

# Accelerating Data Science through Feature Platform and Generative AI

**FeatureStore Summit, October 2023**

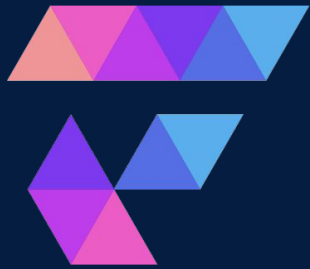
**Xavier Conort**

with the help of Google Bard, ChatGPT and FeatureByte!

# Agenda

- Feature engineering is a complex and challenging task
- Pain points solved by **Feature Platforms**
- The magic of **Transformers**
- How **Generative AI** helped FeatureByte build a **context-aware** automated **feature ideation** solution





# Why feature engineering can be hard

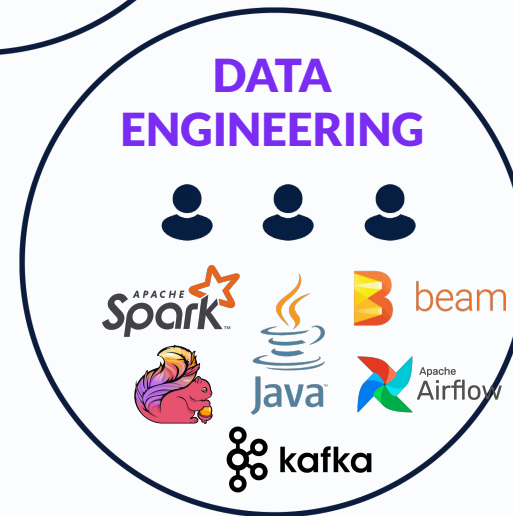
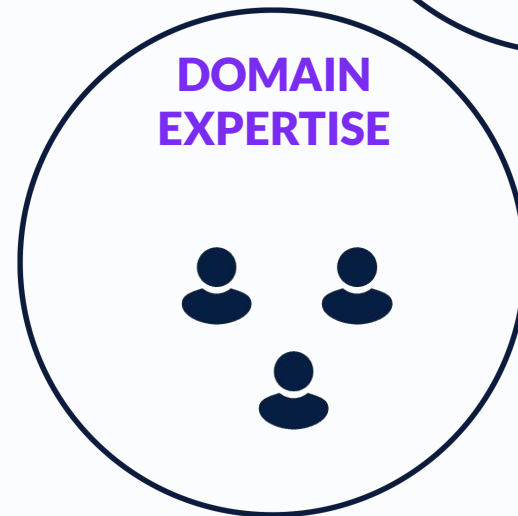
# 3 skills come into play

Data Science skills are necessary to **translate ideas into mathematical forms**



and deal with the **limitless possibilities for features**

**Understanding of the data** is essential to come up with good feature ideas and assess data quality



High risk of time leakage and training-serving inconsistencies: need to be **point-in-time accurate**

while transactional data can be **XXXL**

## ... while Google Bard reminds us of an important principle

- good features are features that are **relevant to the use case, informative and interpretable.**

Feature engineering is the process of transforming raw data into features that are more informative and relevant for machine learning models. It involves selecting, creating, and transforming features to make them more predictive and interpretable.

Features are the inputs that machine learning models use to make predictions. They can be anything from simple categorical variables, such as the color of an object, to complex engineered features, such as the number of times a customer has visited a website in the past month.

The goal of feature engineering is to create features that are:

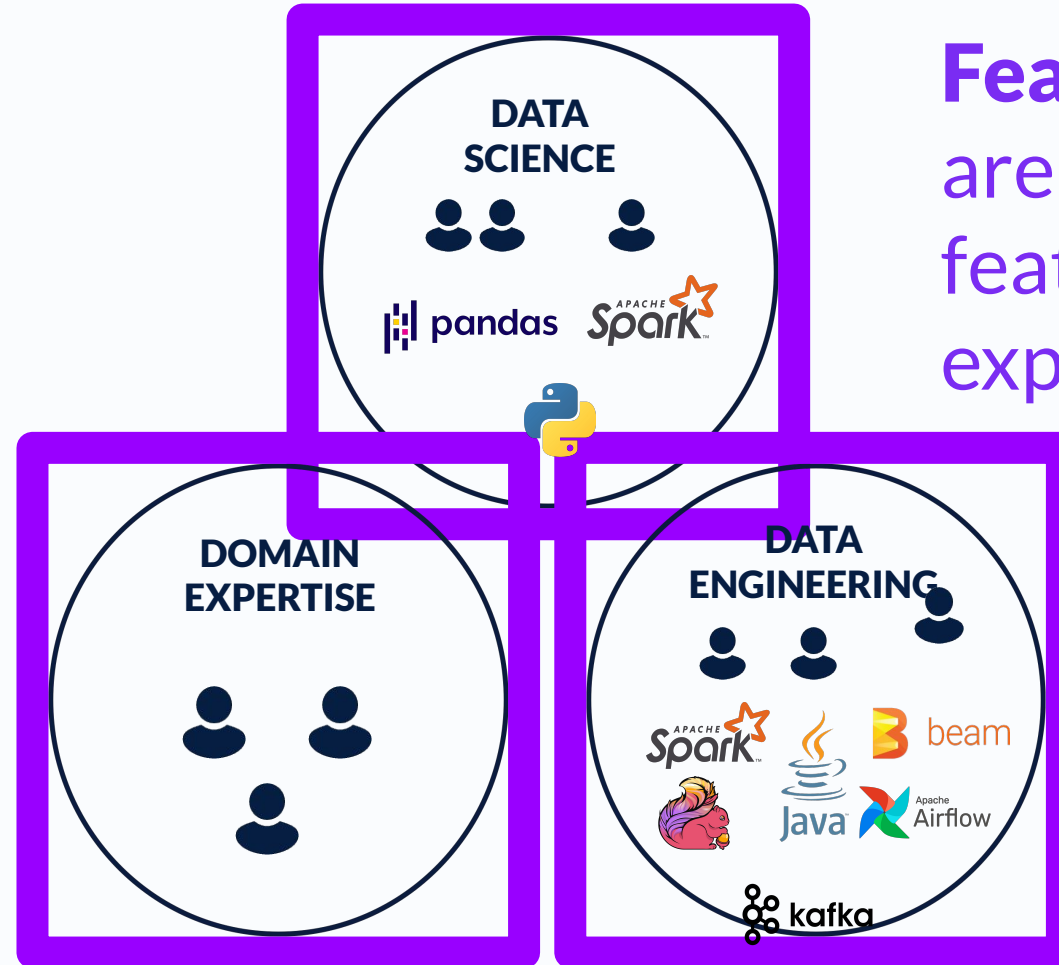
- **Relevant:** The features should be correlated with the target variable that the model is trying to predict.
- **Informative:** The features should provide new information about the data that is not already captured by existing features.
- **Interpretable:** The features should be easy to understand and explain to humans.

Feature engineering can be a complex and challenging task, but it is an essential part of building accurate and interpretable machine learning models.



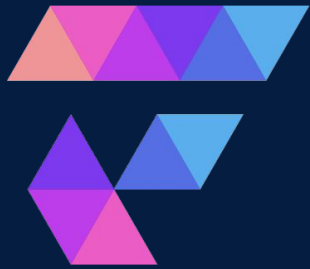
## 2 innovations are poised to radically simplify the effort

**Generative AI** is getting close to emulate a Domain Expert



**Feature Platforms** are facilitating feature experimentation

**And** are streamlining data engineering



# Pain points solved by feature platforms



# Feature platforms are making feature engineering easier in several ways

First, they **reduce the latency** of feature computation (by pre-computing and storing feature values)

Second, they **prevent inconsistencies between training and production**

Third, they **simplify the creation of complex features** (with declarative framework)

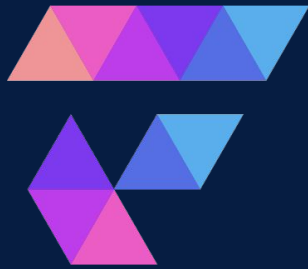
Fourth, they **accelerate experimentation** with new feature ideas (with automated backfilling)

Fifth, they make it **easier to find and reuse** existing features

Finally, they provide a **unified solution from experimentation to deployment**







# The magic of Transformers



# Transformers (LLMs) are also revolutionizing data science

- **As a data scientist:** they are awesome and delivering immediate benefits.
  - Radically simplify a complex task (NLP)
  - In many cases, we don't need any training for great outcomes!
- **As a product manager:** they are opening up new horizons!

But

- **As a Risk manager:** too new, opaque and stochastic
- **As a MLOps engineer:** another transformation to operationalize...



# Do I still need Feature Platforms for Transformers?

Yes! They complement each other. Feature Platforms can improve Transformers operationalization

1. **Point-in-Time correctness**
2. **Low latency** thanks to the pre-computation of features that involve transformers
3. **Caching** (via partial aggregations or other mechanisms) to reduce expensive transformer calls
4. **Transformer Library**



# Can Transformers replace traditional feature engineering?

Maybe in the future, but less magic expected...

1. Deep learning has not delivered its promise with tabular data yet except for:
  - a. regular time series
  - b. sequences of events (successful results for recommendation system)
2. If solutions emerge, it **won't be pre-trained Off the Shelf Transformers like for NLP**
  - a. need a lot of data to train them to get meaningful results.
3. And for many of us, it **may not pass the model validation process:**
  - a. not explainable enough
  - b. and maybe not robust enough if the data is not XXXL



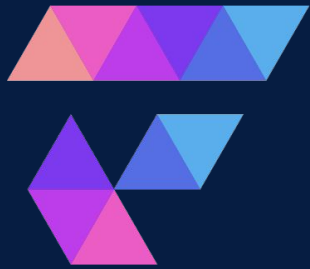
# So why do I think Transformers are revolutionary as a product manager?

"Semantics is to artificial intelligence as physics is to engineering."

John McCarthy, considered to be one of the most important figures in the history of artificial intelligence

"Creativity is just connecting things."

Steve Jobs



**Exciting results from  
Generative AI that can  
already help feature  
engineering**

# Areas where Generative AI is already doing great


## Generative AI:


1. is already **very familiar with data modeling concepts**
2. can **recognize the semantic of the data columns** well beyond numeric and string if meaningful names or good descriptions are provided
3. already **possesses deep domain knowledge** that can **help**
  - a. take some **important decisions** in the feature engineering process
  - b. **assess** how **relevant** a **feature** is to a use case



# Make the difference between different types of numeric columns

is speed column an additive, semi-additive, non-additive, circular or categorical column?

View other drafts 

 A speed column is non-additive.

- Additive columns are columns where the sum of the values in a row is meaningful. For example, the columns "number of apples" and "number of oranges" in a fruit basket are additive columns. The sum of these two columns would give you the total number of fruits in the basket.
- Semi-additive columns are columns where the sum of the values in a row is meaningful, but only up to a certain point. For example, the column "distance traveled" in a car trip is a semi-additive column. The sum of the distances traveled on each leg of the trip is meaningful, but the sum of all the distances traveled would be meaningless, because it would take into account the car's route.
- Non-additive columns are columns where the sum of the values in a row is not meaningful. For example, the column "temperature" in a weather forecast is a non-additive column. The sum of the temperatures for each day of the forecast would not be meaningful, because it would not tell you anything about the overall temperature trend.
- Circular columns are columns where the values are cyclical, such as the hours of the day or the days of the week. The sum of the values in a circular column is not meaningful, because it would simply wrap around to the beginning of the cycle.
- Categorical columns are columns where the values are categories, such as the color of a car or the type of pet. The sum of the values in a categorical column is not meaningful, because it would simply add up the number of items in each category.





# Can recommend strategy to filter data

GX

My use case is to predict fraud with credit card transactions. There are 7 types of transactions: purchase, bank fee, reversal, cash advance, interest, repayment and balance transfer. On which type of transactions should I focus my feature engineering?

## 1. Purchase Transactions:

- This is typically the primary area of concern for many credit card companies. Unauthorized purchases are a common manifestation of credit card fraud.
- Feature considerations: location, frequency, amount, merchant type, time of day, sequence of purchases in a short time frame, any rapid change in purchase behavior, purchases that don't align with past behavior, etc.

## 2. Cash Advance:

- Taking cash out on a credit card can be a red flag, especially if it's unusual for the account or if it's for a large amount.
- Feature considerations: amount, frequency, location (especially international cash advances), time since last cash advance, sequence of consecutive cash advances, etc.

## 3. Reversal Transactions:

- While reversals can be genuine (like returning a purchase), they can also be a sign of fraud or a test by fraudsters to see if the card is active.
- Feature considerations: frequency of reversals, time between purchase and reversal, patterns of purchase-followed-by-reversal, etc.



# Can give Plain English explanation of why a feature is relevant

## Description:

Z-Score of the Txn Amount in relation to the distribution of Txn Amount among all Txns with the same State\_x\_TxnCat as that Txn over a 7d period.

## Relevance:

This feature received a score of 9 out of 10.

This feature is highly relevant for the fraud detection problem. The Z-Score is a statistical measurement that describes a value's relationship to the mean of a group of values. It is measured in terms of standard deviations from the mean. If a Z-Score is 0, it indicates that the data point's score is identical to the mean score. A Z-Score of 1.0 would indicate a value that is one standard deviation from the mean. Z-Scores may be positive or negative, with a positive value indicating the score is above the mean and a negative score indicating it is below the mean. In the context of fraud detection, a transaction amount that is significantly different (either much higher or much lower) from the mean transaction amount for a given state and transaction category over a 7-day period could be

indicative of fraudulent activity. For example, if the average transaction amount for grocery purchases in California is \$100, and a transaction for \$10,000 occurs, this would have a high Z-Score and could be a sign of fraud.

Therefore, this feature is extremely relevant as it can help identify outliers which could potentially be fraudulent transactions. However, it's not a definitive indicator of fraud on its own, as there could be legitimate reasons for the transaction amount to be significantly different from the mean. Hence, the score is 9 and not a perfect 10.



## ... even for complex features

### Description:

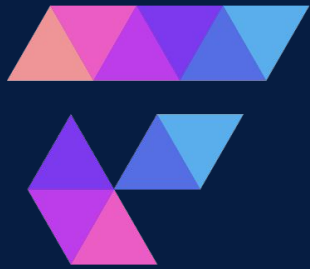
The feature assesses the representation of the Txn\_CardTransactionDescription of a given Txn in that Txn's CUSTOMER's Purchase Txn Amount over a 168d period. A value greater than 1 indicates this Txn\_CardTransactionDescription is over-represented for the particular Customer. The evaluation is done by comparing: - the Percentage of a specific Txn's Customer's total Purchase Txn Amount, that match the CardTransactionDescription of that Txn over a 168d period. - the Percentage of the total Purchase Txn Amount, that match the CardTransactionDescription of a specific Txn over a 168d period.

### Relevance:

This feature received a score of **9** out of 10.

This feature is highly relevant for the fraud detection problem. It captures the behavior of a customer over a significant period (168 days) and compares it with the overall behavior of all customers. If a specific transaction description is over-represented for a particular customer compared to the overall customer base, it could be a sign of fraudulent activity. This feature can help identify unusual patterns in a customer's transactions, which is a key aspect of fraud detection. However, it's not a definitive indicator of fraud on its own, as legitimate customers can also have unique spending patterns. Therefore, it should be used in conjunction with other features to improve the accuracy of the fraud detection model.





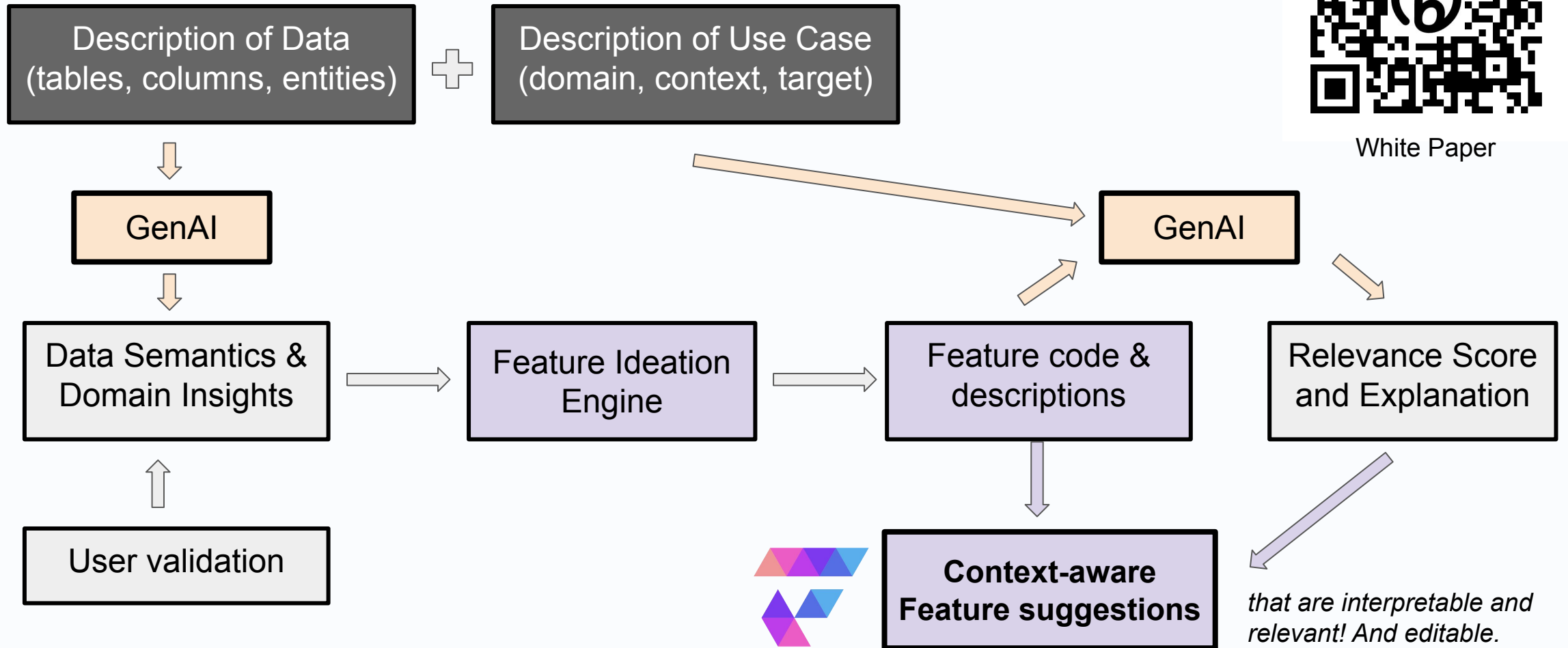
# FeatureByte Copilot



# How FeatureByte's Copilot is currently leveraging GenAI



White Paper



Credit card transaction fraud detection - flagged as fraud within 30 days

Use Case Primary Entity: **TXN** Context: **PURCHASE TRANSACTION** Target: **TRANSACTION FRAUD STATUS 30 DAYS LATER**

- Explore >
- Formulate >
- Create >
- Feature Ideation**
- Experiment >
- Approve >
- Manage >
- Security >
- Admin >

1 Tagging of Incomplete Semantics

2 Feature Ideation

662 Recommended Features (includes 208 catalog features) 0 Other Features All 662 Features

Feature Name	Relevance	Primary Entity	Primary Table	Signal Type	Readiness
<input type="checkbox"/> <b>CARD_Consistency_of_Sum_of_Purchase_Txn_Amount_7d_vs_168d</b> Consistency of the Card measured by the Ratio of the Sum of Purchase Txn Amount for both the 7d and 168d periods	9/10 ?	CARD	TRANSACTIONS	STABILITY	NEW
<input type="checkbox"/> <b>CARD_Consistency_of_Purchase_Txn_Amount_across_Txn_CardTransactionDescriptions_7d_vs_168d</b> Consistency score of the Card measured by the Cosine Similarity between the Distribution representing the cumulative Amount of...	9/10 ?	CARD	TRANSACTIONS	STABILITY	NEW

About SDK Code

Description:

Consistency score of the Card measured by the Cosine Similarity between the Distribution representing the cumulative Amount of Purchase Txn, categorized by their respective Txn's CardTransactionDescription, for both the 7d and 168d periods.

Relevance:

This feature received a score of 9 out of 10.

This feature is highly relevant for the fraud detection problem. Fraudulent transactions often exhibit patterns that deviate significantly from the normal behavior of a cardholder. The consistency of purchase transaction amounts across different transaction descriptions over a period of 7 days compared to 168 days can provide valuable insights into these patterns. For instance, if a cardholder typically makes small purchases at grocery stores and suddenly there's a large purchase at an electronics store, this could be a sign of potential fraud. Similarly, if the cardholder's spending pattern drastically changes in a short period of time (7 days) compared to a longer period (168 days), this could also indicate fraudulent activity. Therefore, this feature, which measures the consistency of such behaviors, is extremely relevant for detecting credit card fraud. However, it's not a perfect 10 because while it's a strong indicator, fraud detection should also consider other factors such as location, time of transaction, and

Displaying 662 of 662 features

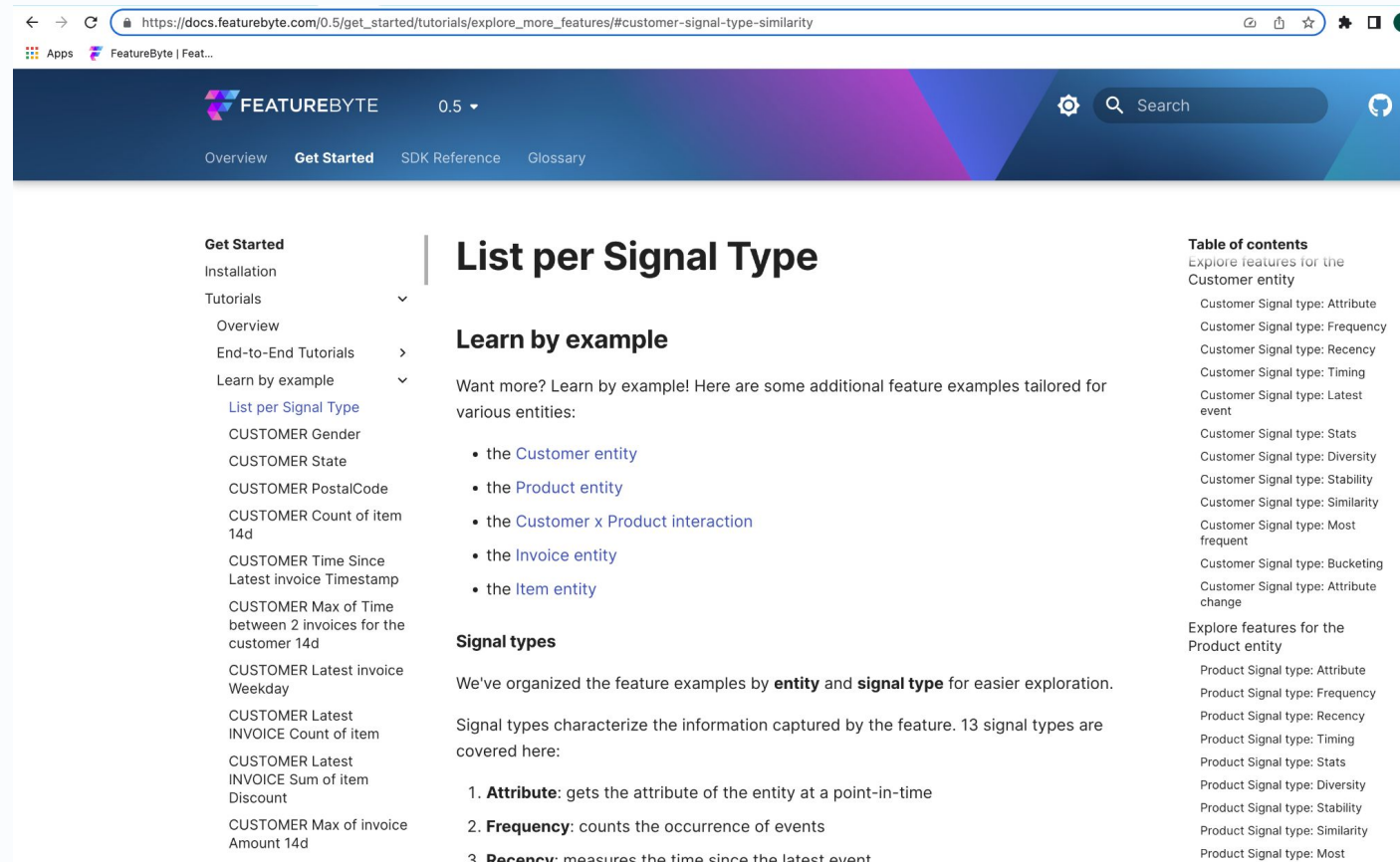
2 Features Selected

Save Features / Feature List

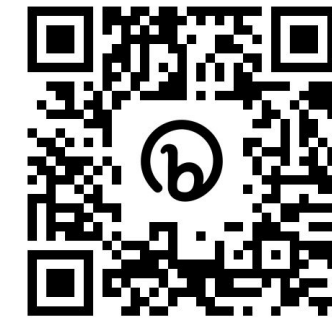
Generate Notebook



# Examples of feature SDK notebooks

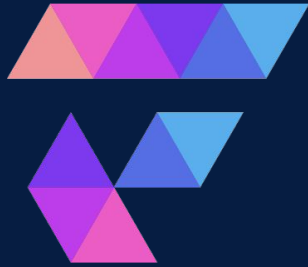


The screenshot shows a web browser window with the URL [https://docs.featurebyte.com/0.5/get\\_started/tutorials/explore\\_more\\_features/#customer-signal-type-similarity](https://docs.featurebyte.com/0.5/get_started/tutorials/explore_more_features/#customer-signal-type-similarity). The page header includes the FeatureByte logo, version 0.5, a search bar, and navigation links for Overview, Get Started, SDK Reference, and Glossary. The main content area is titled "List per Signal Type" and includes a "Learn by example" section with a list of entities: Customer entity, Product entity, Customer x Product interaction, Invoice entity, and Item entity. A "Signal types" section explains that 13 signal types are covered, with three examples: Attribute (gets the attribute of the entity at a point-in-time), Frequency (counts the occurrence of events), and Recency (measures the time since the latest event). A "Table of contents" sidebar on the right lists various signal types for both Customer and Product entities.



Tutorials of FeatureByte's free and source available package





# Conclusion

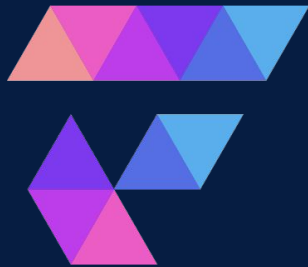




# Feature Platforms and GenAI are accelerating Data Science by

1. **Reducing time-to-value** thanks to:
  - a. quicker experimentation, and smooth transition to deployment
2. **Increasing production accuracy** thanks to:
  - a. improved training / serving consistency
  - b. better NLP with transformers
  - c. opportunity to build context aware feature generation
3. **Increasing transparency and bridging the gap between features and non-technical stakeholders** thanks to:
  - a. Plain English descriptions of features and their relevance to a use case





# Thanks!

**If you are curious to know more about Featurebyte**

Github Repo of the free and source available engine: <https://github.com/featurebyte/featurebyte>

Documentation: <https://docs.featurebyte.com/0.5/>



Video of FeatureByte Enterprise