

Navigating the Skies of Data A Flight Through Feature Stores

Dave Ruijter



Session
feedback



Dave Ruijter

Solution Architect & Engineer
Blue Rocket IT



dave@blue-rocket.it



@DaveRuijter



linkedin.com/in/DaveRuijter



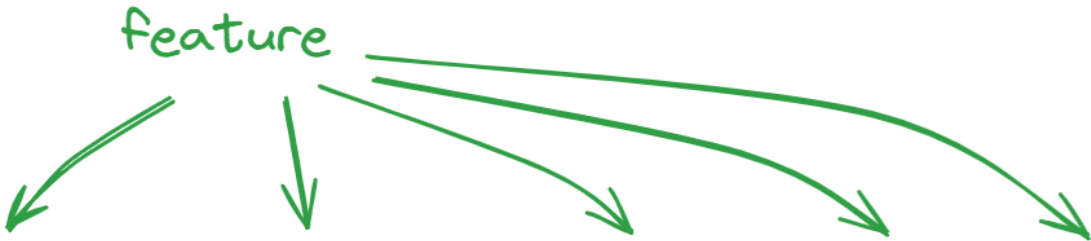
ModernData.ai



What is a Feature Store?


What is a Feature?

feature



wine_id	fixed_acidity	volatile_acidity	residual_sugar	alcohol	ph
1	7.4	0.7	1.9	9.4	3.51
2	7.8	0.88	2.6	9.8	3.2
3	11.2	0.76	1.9	9.8	3.16
4	7.4	0.7	1.8	9.4	3.51

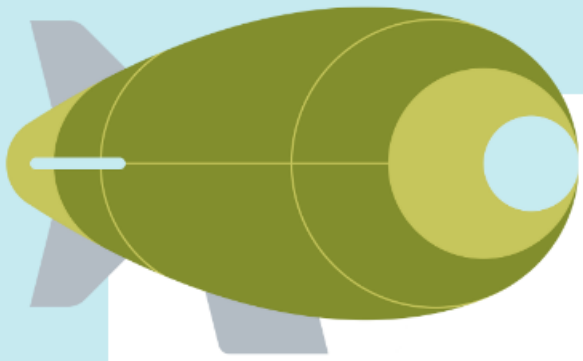
feature value



What is a Feature?

*individual measurable property or characteristic
of a
phenomenon being observed.*

***A machine learning model is only as good
as the features fed into it***



Feature Table vs Fact Table

Feature Table

1. Attributes can be anything
2. Attributes are often normalized or standardized

Fact Table

1. Attributes are (usually) numeric
2. Attributes are 'pure' or an aggregated form of that



Normalization and standardization are preprocessing steps used to make the numerical features in a dataset have a **more uniform scale**. This is important because machine learning algorithms that compute distances or assume normality can perform poorly if features are on wildly different scales.

Here are some examples of normalized or standardized features in a feature table:

1. Min-Max Normalization:

Before Normalization: House Size feature with values ranging from 500 to 5,000 square feet.

After Normalization: Normalized House Size where values are scaled to fall between 0 and 1, using the formula $(\text{value} - \text{min}) / (\text{max} - \text{min})$.

2. Z-Score Standardization:

Before Standardization: Annual Income feature with a mean of \$50,000 and standard deviation of \$20,000.

After Standardization: Standardized Annual Income where each value is transformed using the formula $(\text{value} - \text{mean}) / \text{standard deviation}$, resulting in a feature with a mean of 0 and standard deviation of 1.

3. Log Transformation:

Before Transformation: Website Clicks feature with a highly right-skewed distribution ranging from 1 to 10,000 clicks.

After Transformation: Log of Website Clicks where each value is transformed using the logarithm, reducing the effect of extreme outliers and making the distribution more normal.

4. Binarization:

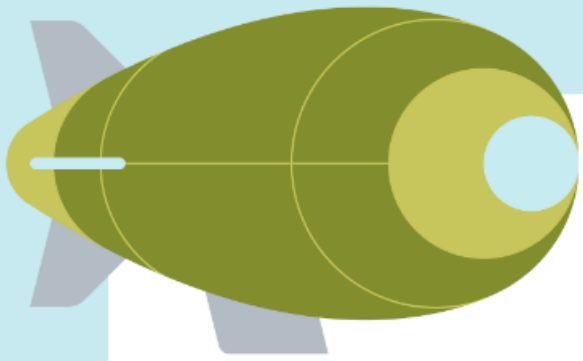
Before Binarization: Age feature with continuous values ranging from 18 to 90 years.

After Binarization: Is Senior binary feature where values are transformed into a binary indicator (0 or 1) depending on whether the Age is above a certain threshold, say 65.

5. Robust Scaler:

Before Scaling: Investment Returns feature with potential extreme outliers due to market volatility.

After Scaling: Robust Scaled Investment Returns where values are scaled using median and interquartile range, making it less sensitive to outliers compared to min-max scaling or z



Difference between feature table and a fact table?

Feature Table

1. Attributes can be anything
2. Attributes are often normalized or standardized
3. Often single dimension
4. Often 'OBT': one big table
5. Often has versions of the same thing over time: snapshots

Fact Table

1. Attributes are (usually) numeric
2. Attributes are 'pure' or an aggregated form of that
3. Often multiple dimensions
4. Often denormalized (~~OBT~~)
5. Often has a single record of the same thing, supported by slowly changing dimensions



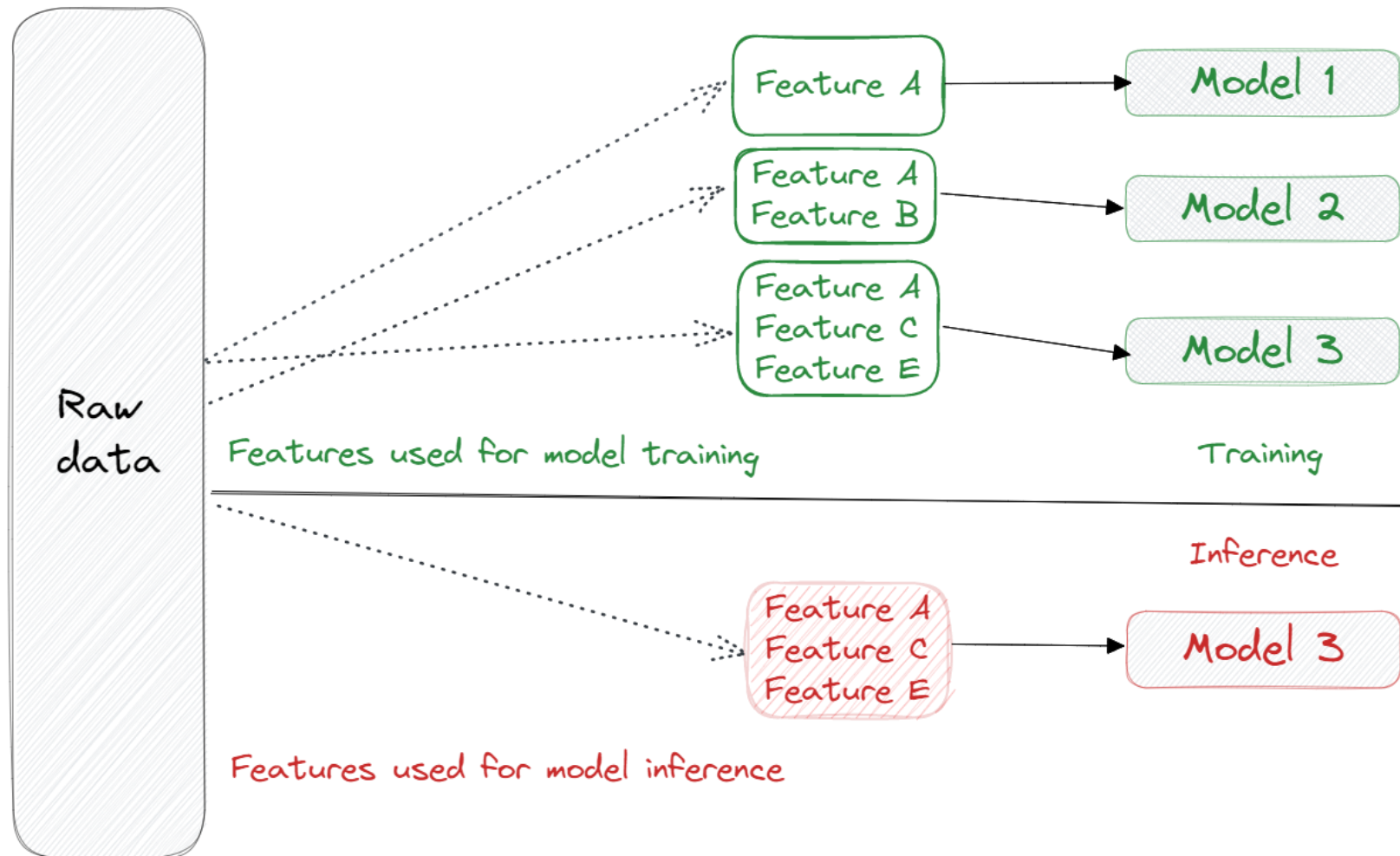


Image source: <https://www.qwak.com/post/what-is-a-feature-store-in-ml>

What is a Feature Store?

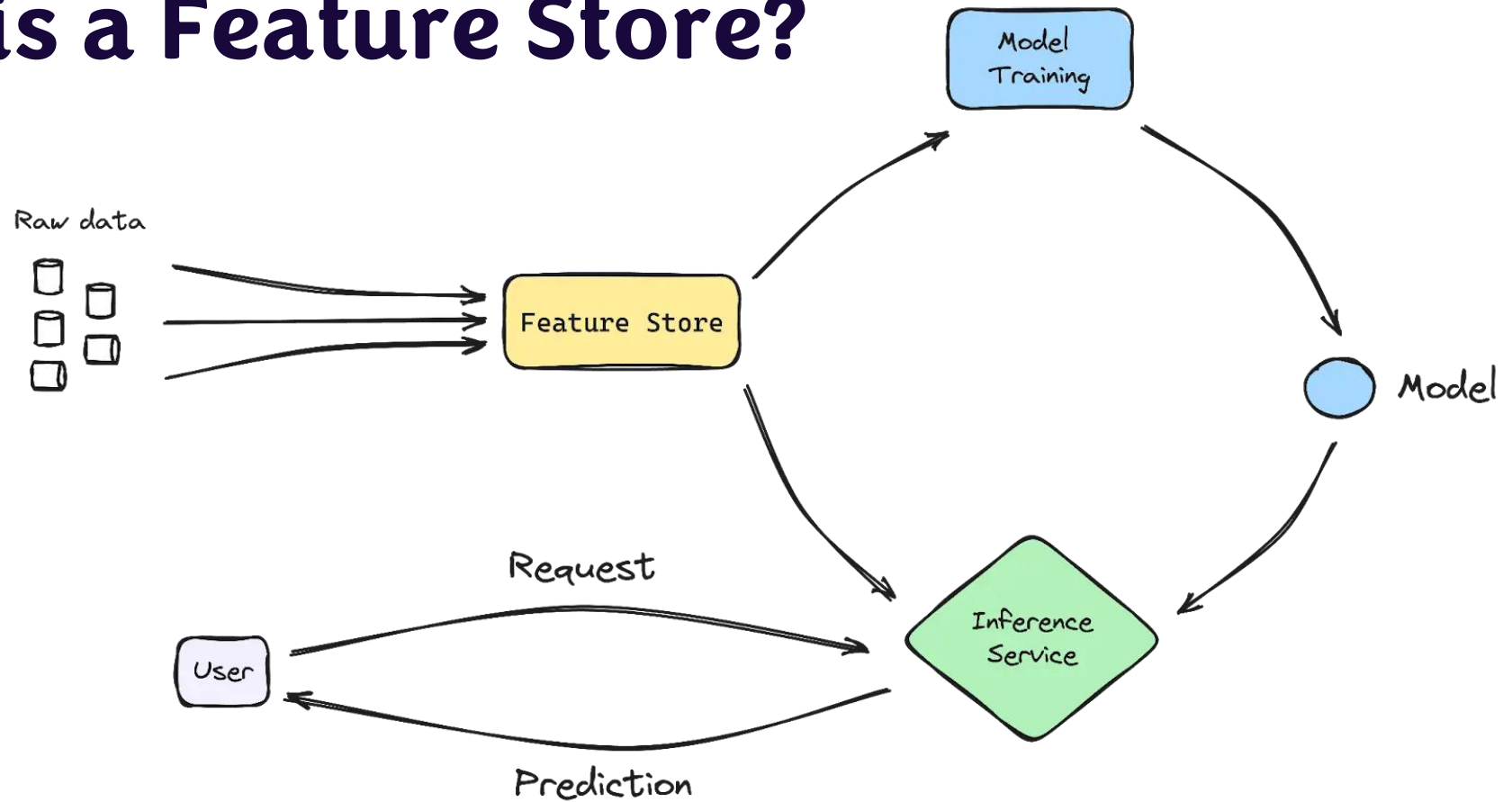


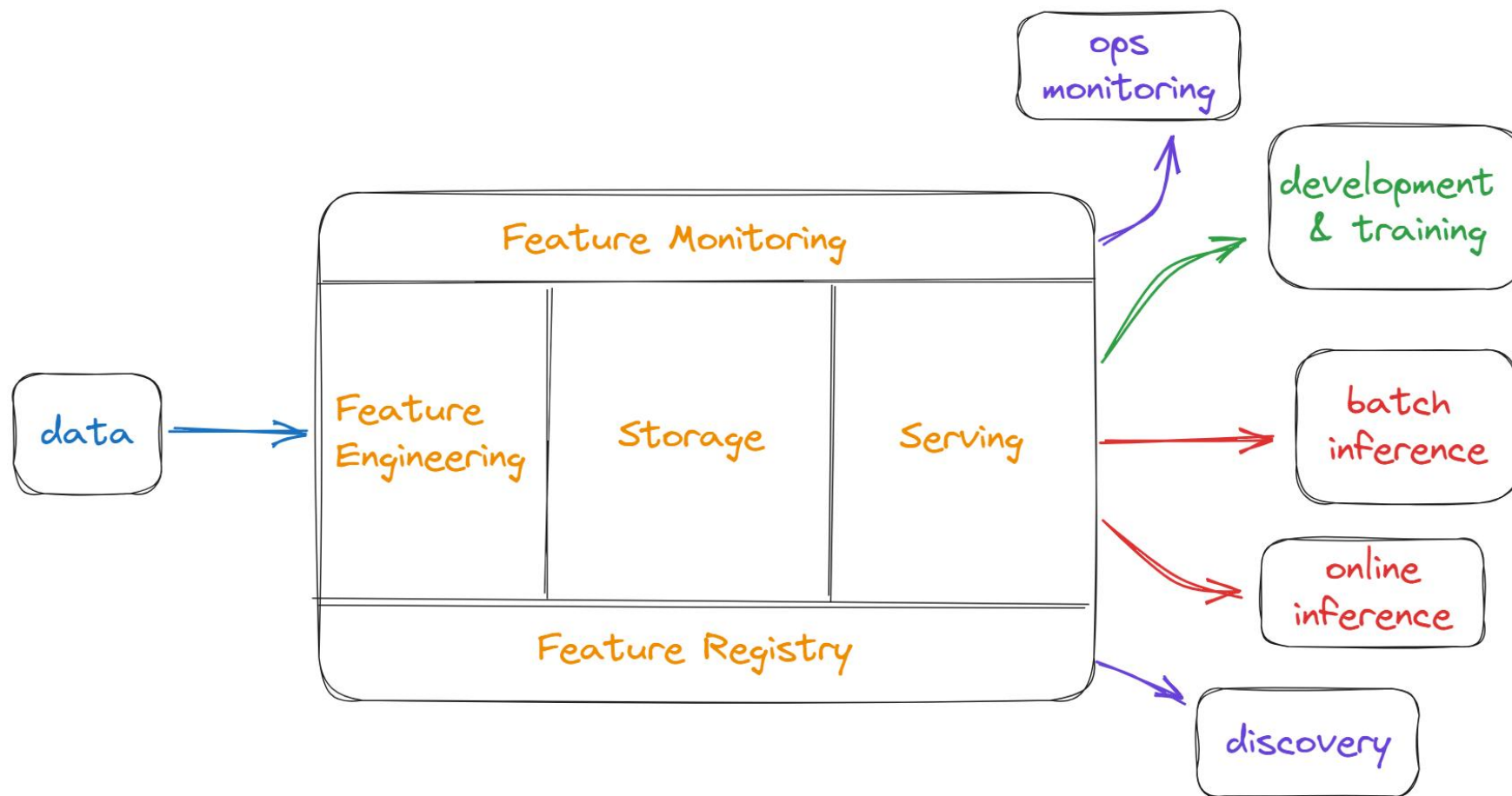
Image source: <https://www.qwak.com/post/what-is-a-feature-store-in-ml>



Core components/features

1. Feature engineering (transformations)
2. Storage
3. Registry
4. Serving
5. Monitoring





Offerings

- Azure ML
- Azure Databricks
- Amazon SageMaker
- Feast
- Tecton
- Iguazio
- Hopsworks
- Kaskada
- Molecula
- RasgoVertex AI





DEMO

Feature Store in Databricks



Benefits of a Feature Store

1. Standardization of features
2. Efficiency and speed
3. Scalability
4. Improved collaboration
5. Quality and compliance





Best Practices in Feature Store Implementation

1. Define clear feature metadata
2. Ensure feature quality
3. Adopt a governance strategy
4. Facilitate collaboration and sharing
5. Monitor and optimize usage and performance



Do I need it?



Q&A on Feature Stores



Session
feedback

