Strathmore
UNIVERSITY

Recap of ML and DL Concepts

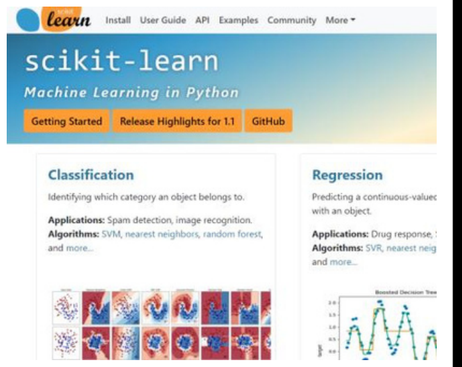
TensorFlow

Keras

A deep learning library

seaborn

matplotlib



pandas

jupyter

PyTorc

End-to-End Machine Learning Workflow

Introduction

- What is ML & DL?
- Types of ML & DL Systems
- Applications and challenges

Data

- Data Collection
- Data Pre-processing
- Feature Engineering

Algorithms

- Supervised Models
- Unsupervised Models
- Model Boosting, Stacking, Ensembling

Training ML Models for Production

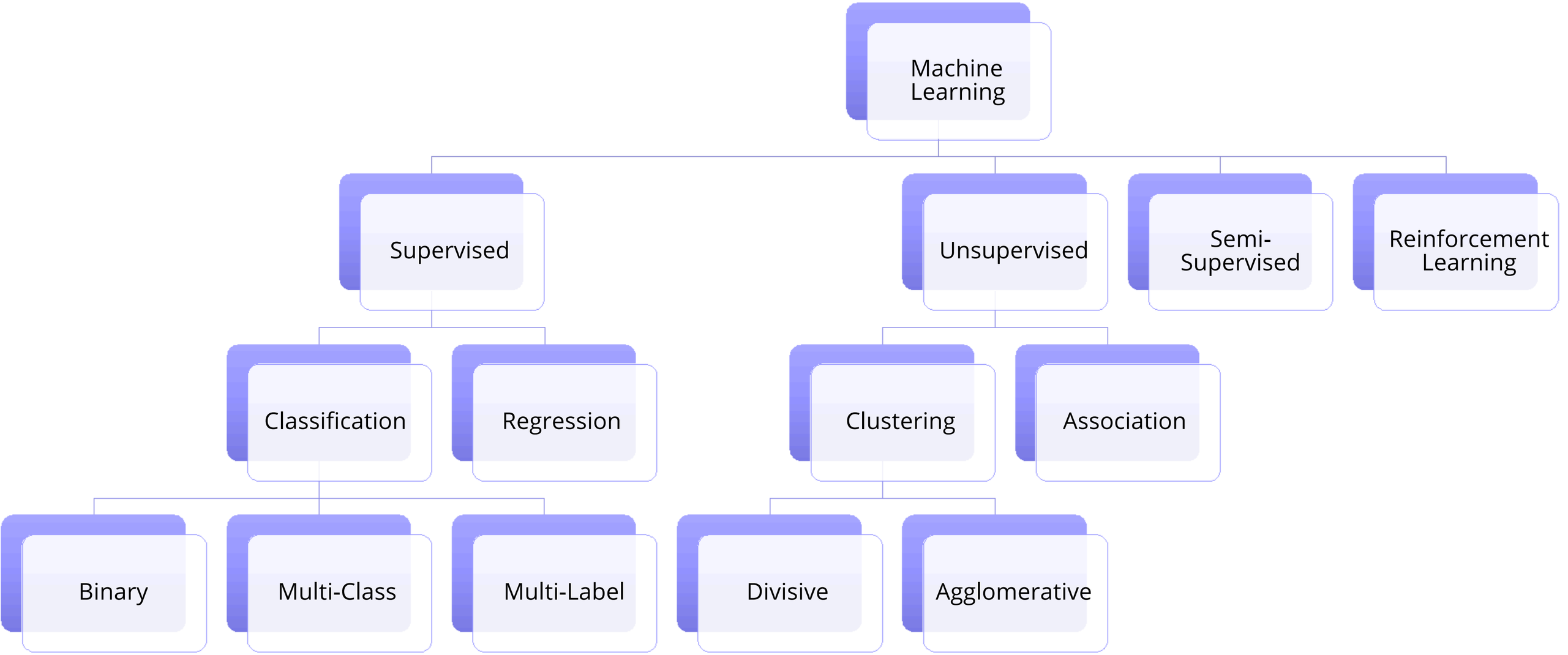
- Problem framing
- Training best practices
- Model validation

Deployment & monitoring

ML is good for:

- Solutions requiring long list of rules
- Solutions requiring extensive fine-tuning
- Complex problems unsolvable by traditional methods e.g. perceptive problems such as image recognition
- Fluctuating environments e.g. data changes, problem changes
- Dealing with large, complex data
- Observable but unstudied phenomenon e.g. computing network logs

Recap: Types of ML Systems

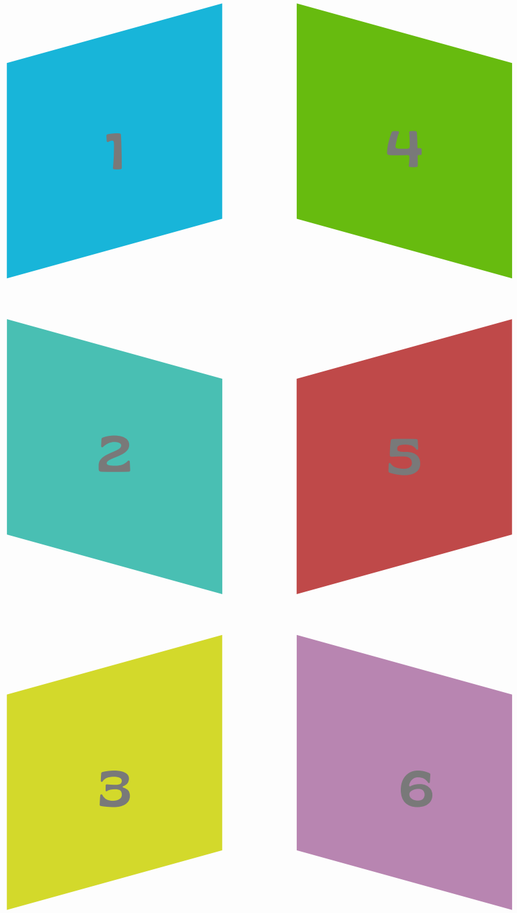


Deep Learning Models

Feedforward Neural Networks(FNNs)

Convolutional Neural Networks (CNNs)

Recurrent Neural Networks (RNNs)

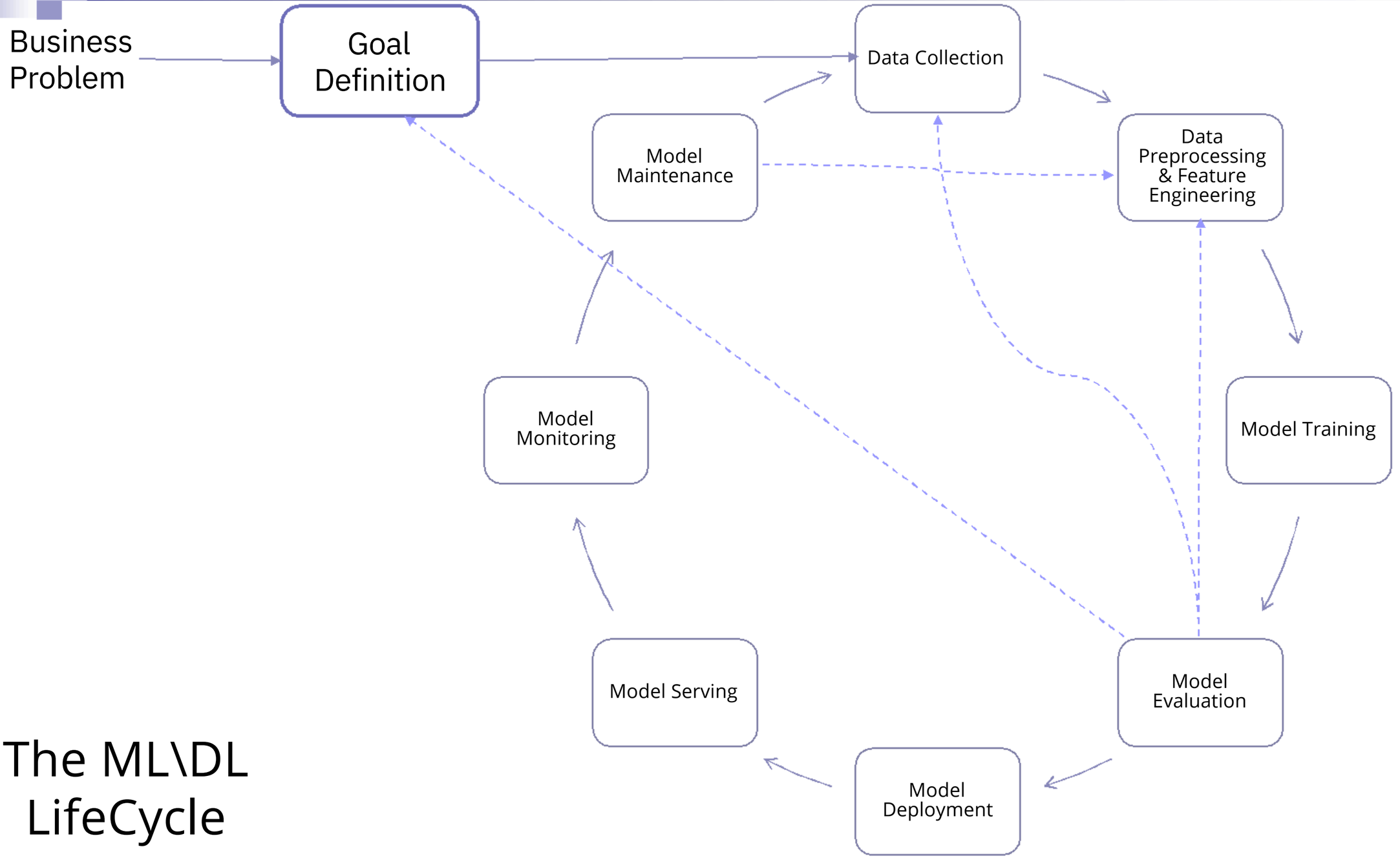


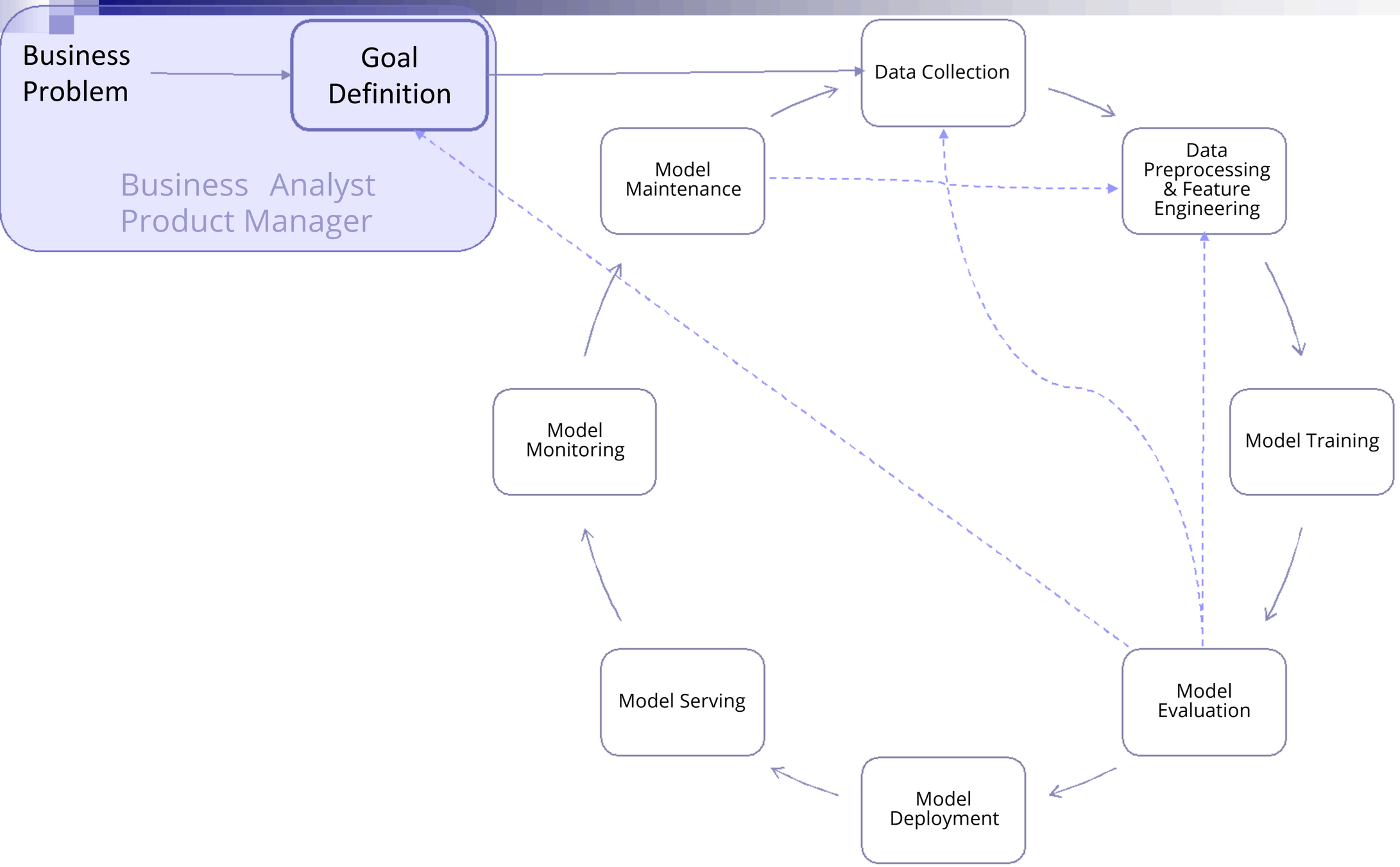
Autoencoders

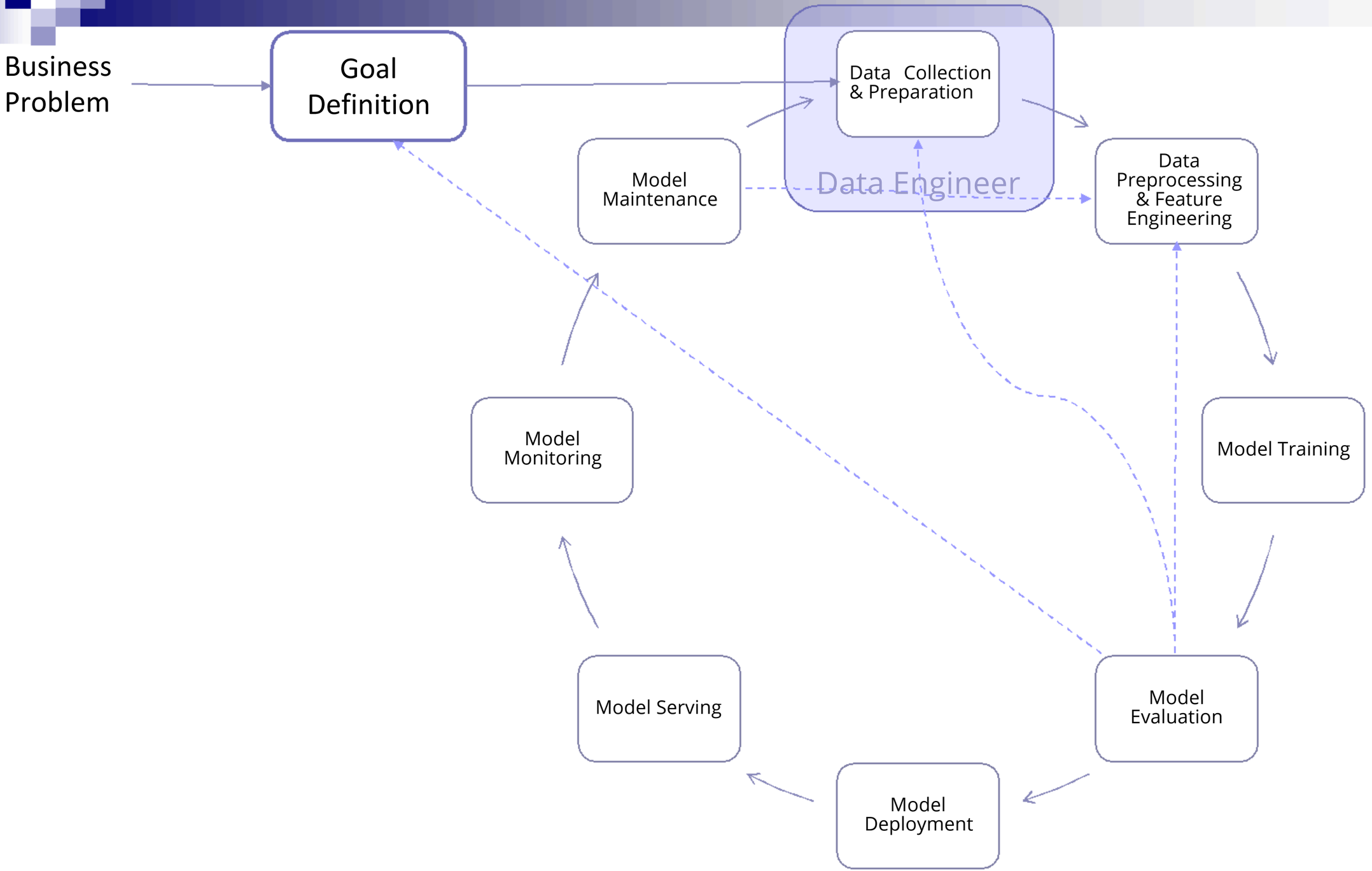
Transformers

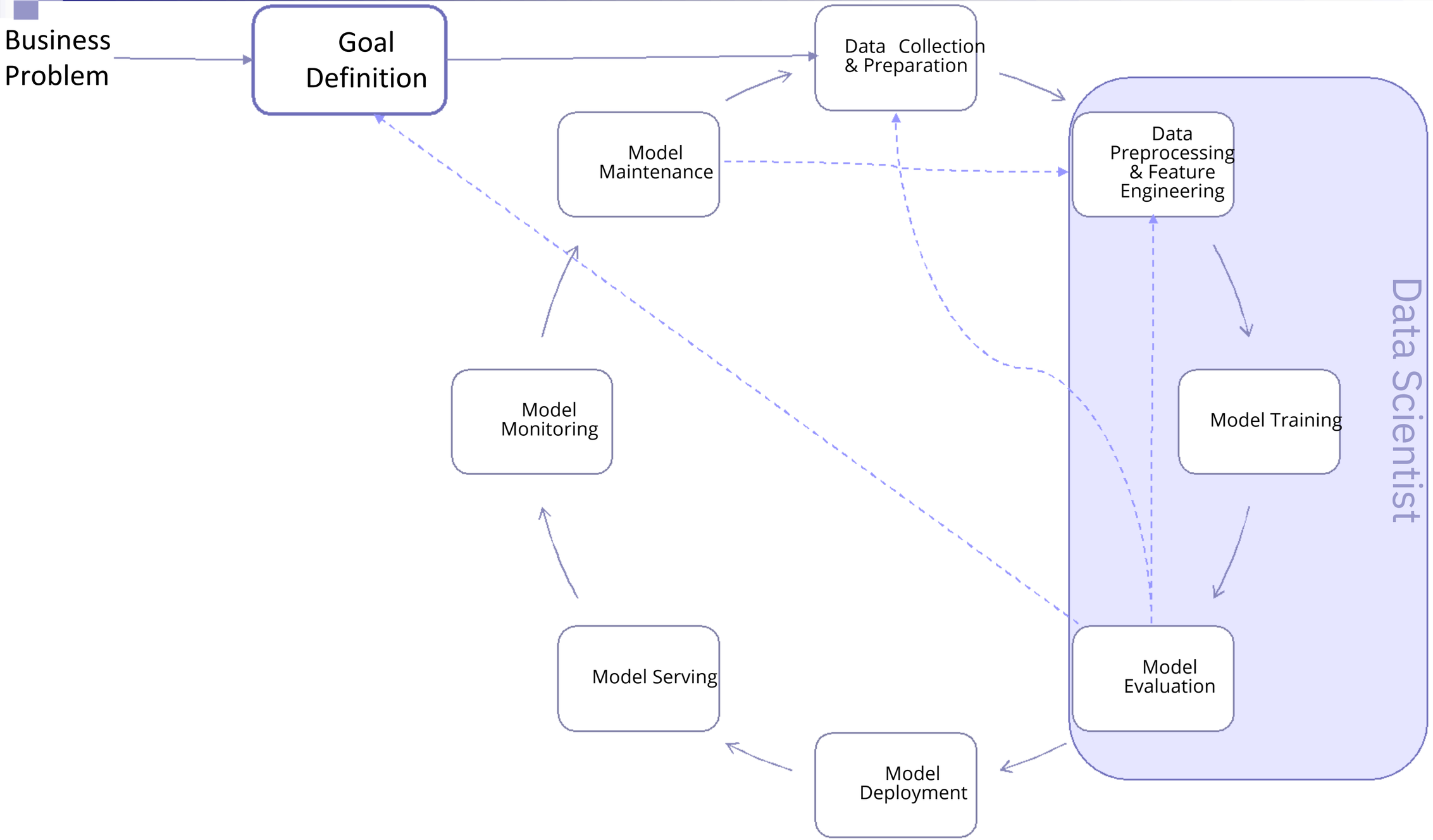
Generative Adversarial Networks (GANs)

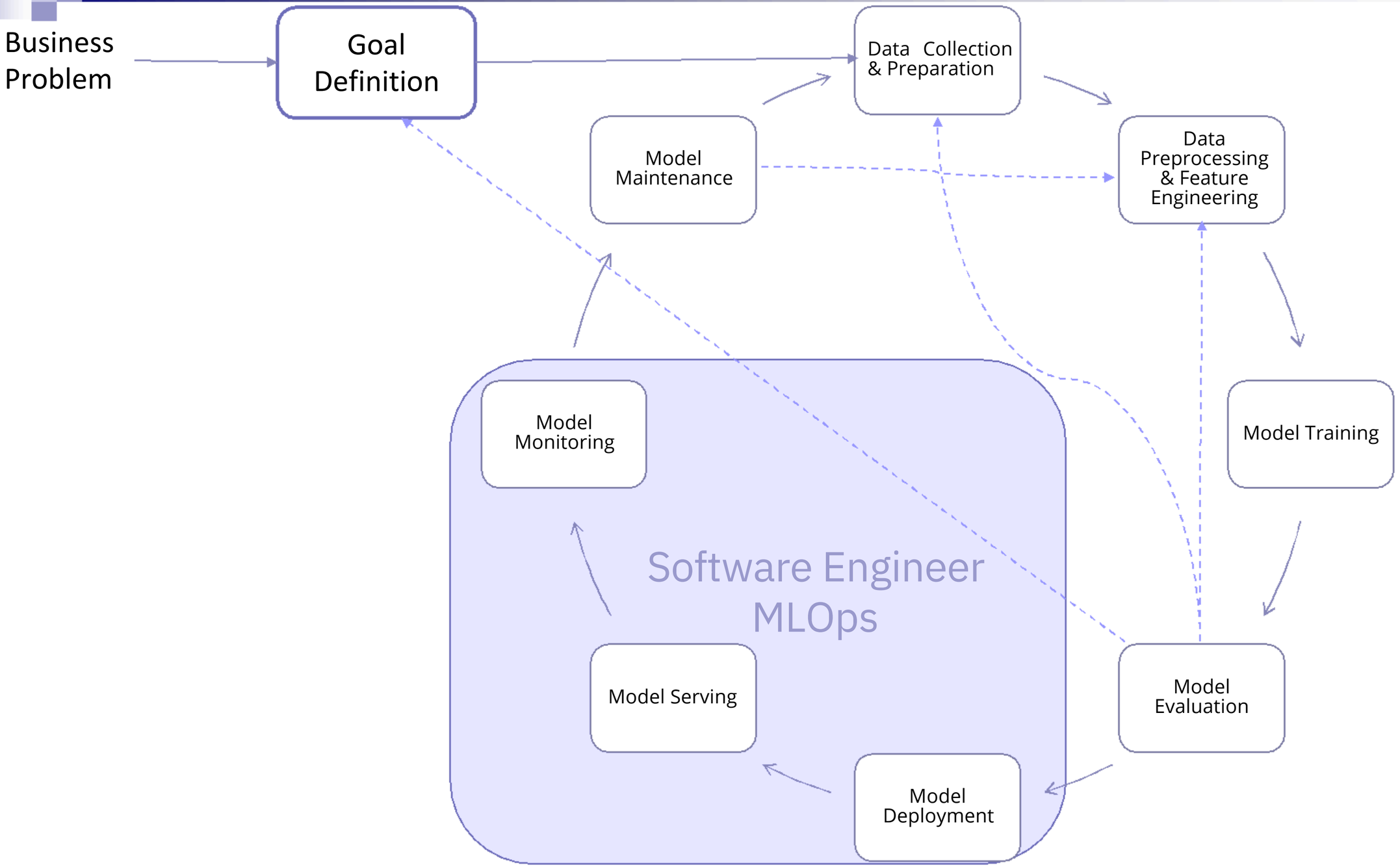
The ML\DL LifeCycle

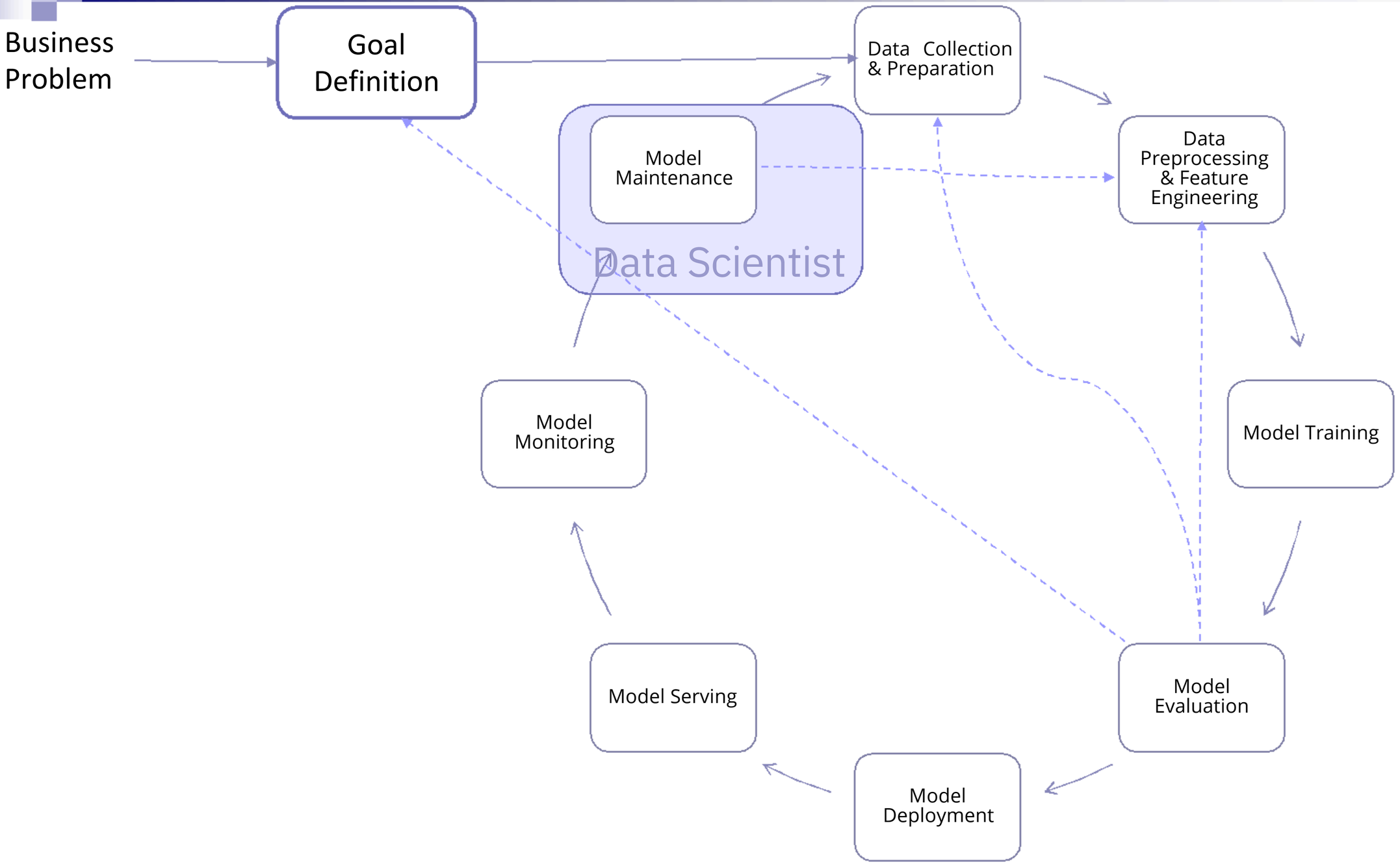



Business Problem
Goal Definition
Business Analyst
Product Manager
Data Collection
Data Preprocessing & Feature Engineering
Model Training
Model Evaluation
Model Deployment
Model Serving
Model Monitoring
Model Maintenance


Business Problem
Goal Definition
Data Collection & Preparation
Data Engineer
Data Preprocessing & Feature Engineering
Model Training
Model Evaluation
Model Deployment
Model Serving
Model Monitoring
Model Maintenance


Business Problem
Goal Definition
Data Collection & Preparation
Model Maintenance
Data Preprocessing & Feature Engineering
Model Monitoring
Model Training
Model Serving
Model Evaluation
Model Deployment
Data Scientist


Business Problem
Goal Definition
Data Collection & Preparation
Data Preprocessing & Feature Engineering
Model Training
Model Evaluation
Model Deployment
Model Serving
Model Monitoring
Model Maintenance
Software Engineer MLOps


Business Problem
Goal Definition
Data Collection & Preparation
Data Preprocessing & Feature Engineering
Model Training
Model Evaluation
Model Deployment
Model Serving
Model Monitoring
Model Maintenance
Data Scientist

EDA –Know your data

*In statistics,**exploratory data analysis**is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.*

EDA Tools/Techniques

- Data set shape and feature types
  - *df.shape, df.dtypes*
- Eye-balling of data
  - Explore oddities by looking at column names etc.

  df.head()
- Univariate analysis

  Understand distributions, outliers, missing values, variance, unique values etc.
  - *df.describe(), box plots, cdf/pdf plots, violin plots*
- Bivariate analysis
  - Understand relationship between 2 variables

  e.g., age vs target

  Box plots, pair plots
- Multivariate analysis

  Understand interactions between multiple variables
  - *Correlation matrix, pair plots, 3D plots etc.*

Clean your data

Missing data

Important: Understand why data is missing

Missing completely at random (MCAR)

Missing data are randomly distributed across the variable and unrelated to other variables.

(no patterns observed, same probability of missing)

Missing at random (MAR)

There might be systematic differences between missing and observed recordsbut these are completely accounted for by other observed variables. (e.g., more data is missing for males vs. females but probability of missing is the same within each group). The term 'random' is a bit of a misnomer

Missing not at random (MNAR)

Missing data systematically differ from the observed values. Related to the variable itself e.g., not stating my preference for brand X because I don't like it.

Handling missing values

- Remove records with missing data
- Leave as-is
- Impute
  - Substitution (Fixed-value impute (mean/mode/median/'unknown')

Row	Age	Weight	Height	Salary
1	18	70		35,000
2	43	65	175	26,900
3	34	87		76,500
4	21	66	187	94,800
5	65	60	169	19,000

$$\hat{Height} \leftarrow \frac{1}{3}(175 + 187 + 169) = 177$$

- Fast, easy but tend to be inaccurate without accounting for other features/correlations or overall data structure; Only suitable for MCAR; May be sensitive to noise e.g., outliers

Feature Engineering

Sidebar:
What is Cardinality?

The number of unique elements in a set:

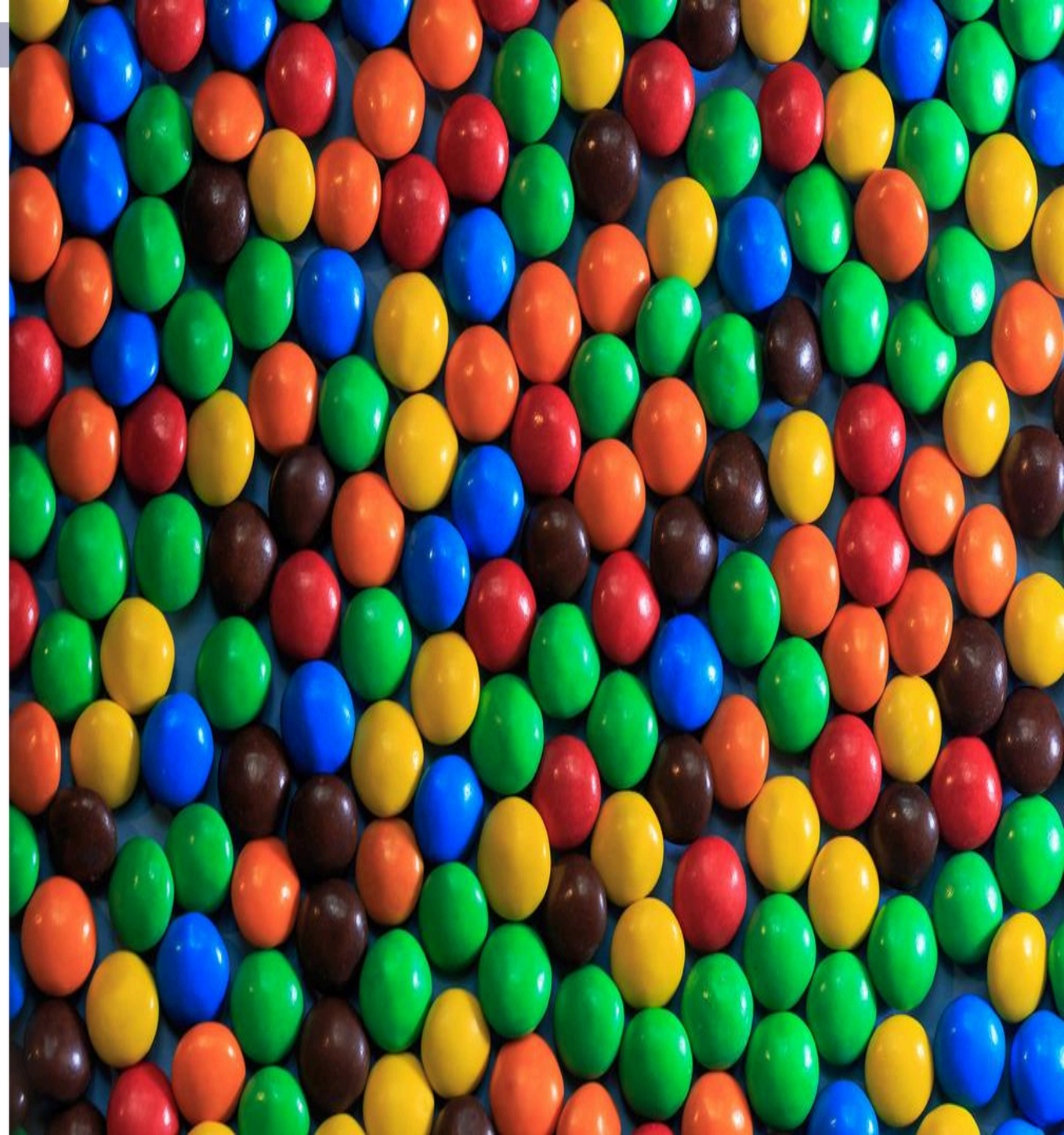
X=$\{4,6,7\}$ Cardinality = 3

X2=$\{9,2,7,3,1\}$ Cardinality =5

Counties dataset:

Cust County
1002
2010
3030
4002
5006
6047

Cardinality = 47



Converting Categorical Features to Numeric

One-Hot Encoding (dummy coding)

	Marital
1	Single
2	Married
3	Divorced
4	Unknown

	Single	Married	Divorced	Unknown
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

-Very simple
-but can create an explosion of features if cardinality is high.
-Is not target-led

(pro-tip: can use aggregation approach to reduce cardinality e.g., ‘Nairobi’, ‘Kiambu’, ‘Nakuru’, ‘Other’)

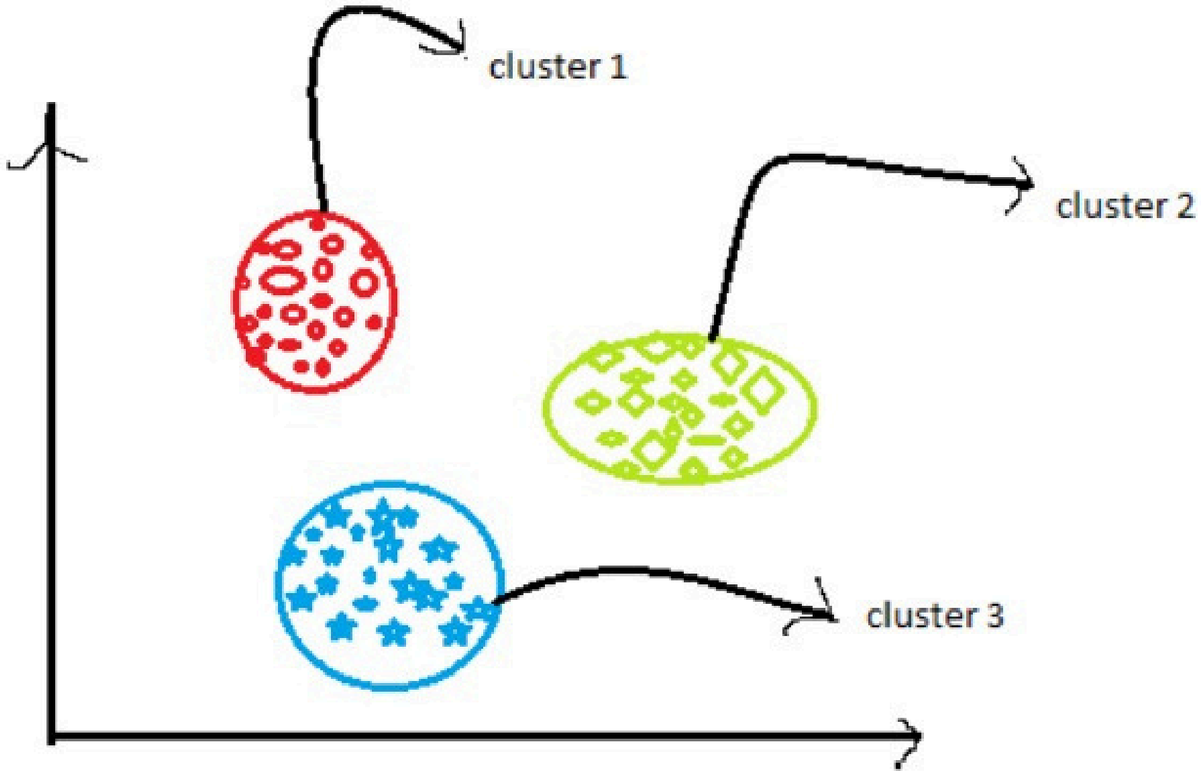
Label Encoding

Mon	1
Tue	2
Wed	3
Thur	4
Fri	5
Sat	6
Sun	7

-Alsosimple
-works better for ordered categories - but may mislead algorithm on scale and distance
–Is not target-led.

Clustering Approach to reduce cardinality

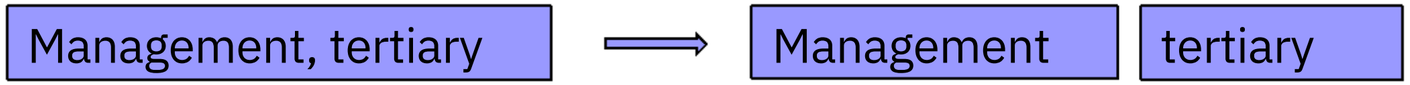
- For high-cardinality features, build similarity clusters and then perform one-hot encoding or proportion representation.



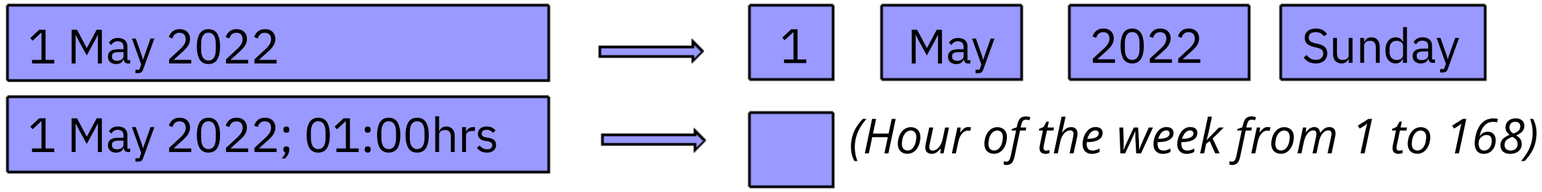
Example, clustering US state codes into fewer categories

Feature Enrichment

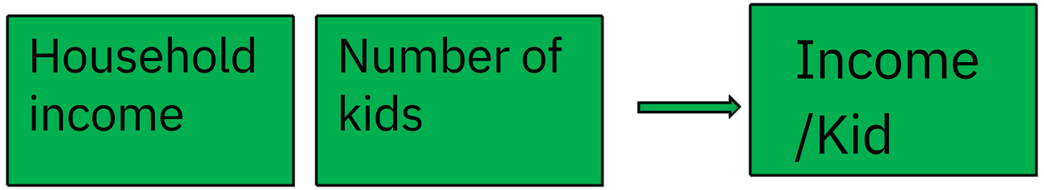
- Feature splitting



- Date extraction



- Combining Features (domain led)



Can apply simple additions, subtractions, polynomials etc. to various features to extract a different dimension

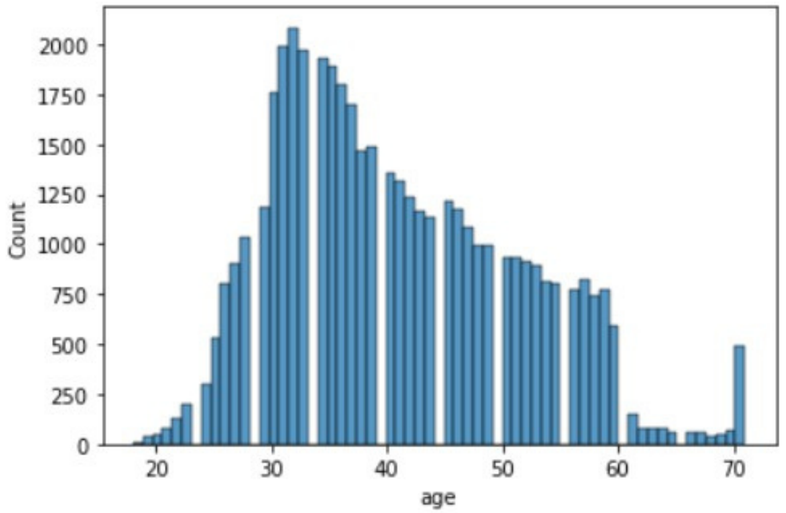
Feature Scaling

Bring your features to the same or similar range of values or distributions

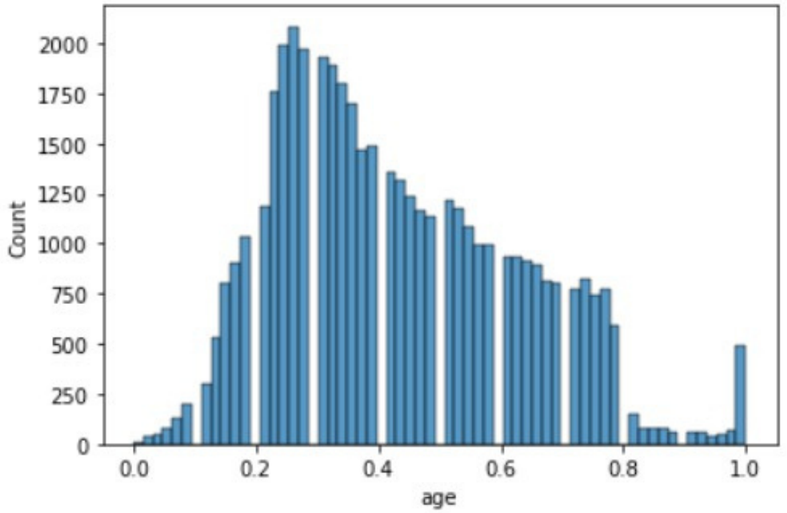
- Normalization (min-max scaling), constrain data into a range (typically [0,1]

$$\bar{x}^{(j)} \leftarrow \frac{x^{(j)} - \min^{(j)}}{\max^{(j)} - \min^{(j)}},$$

Caution: if min and max are outliers, the feature will be squeezed into a very small range. In this case consider robust scaling (*x = (x —median) / inter-quartile range*



from sklearn.preprocessing import MinMaxScaler

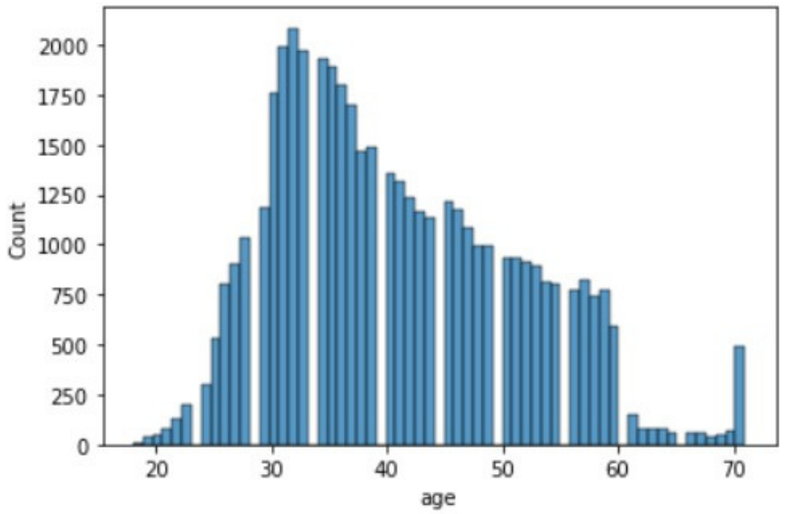


Feature Scaling

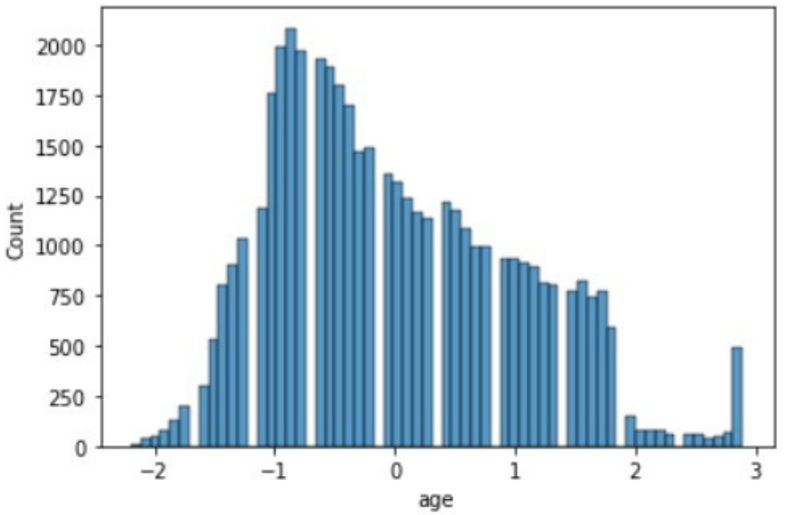
Bring your features to the same or similar range of values or distributions

- Standardization ，rescale data to achieve properties of a standard normal distribution (μ=0, σ=1)

$$\hat{x}^{(j)} \leftarrow \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}},$$



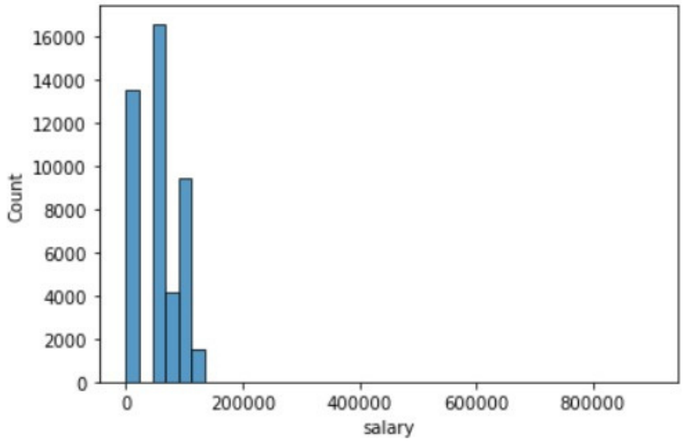
from sklearn.preprocessing import StandardScaler

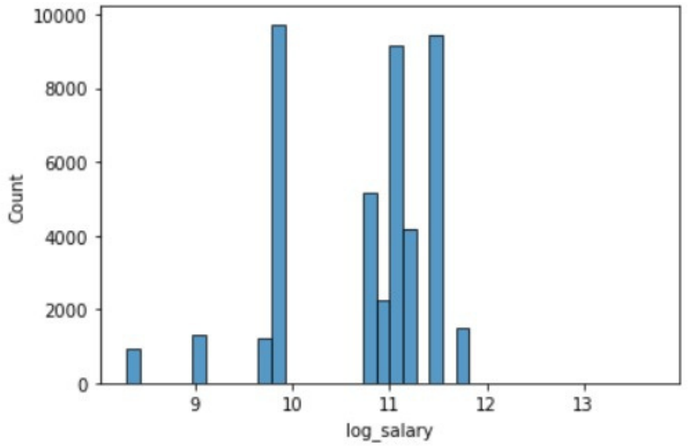


Power Transformation for changing distribution

Techniques used for converting a skewed distribution to a normal distribution/less-skewed distribution.
Many machine learning algorithms prefer or perform better when numerical variables have a Gaussian probability distribution

Log Transform





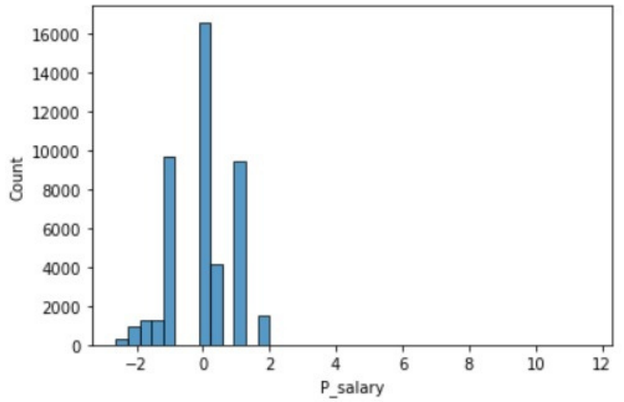
Changes distribution and takes care of extreme values

Box-Cox Transform

λ			
Y'	=	[Y−1] / λ	when λ ≠ 0
Y'	=	ln Y	when λ = 0

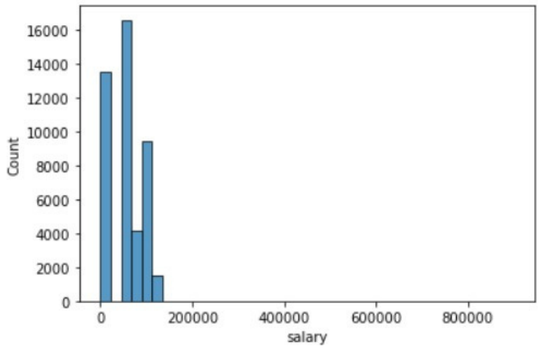
from sklearn.preprocessing import PowerTransformer

PowerTransformer(method = 'box-cox', standardize=True)

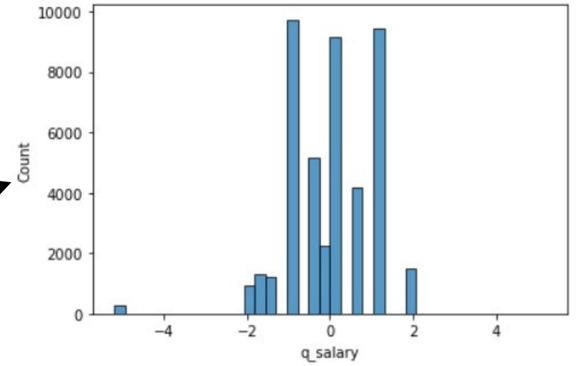


Quantile Transform

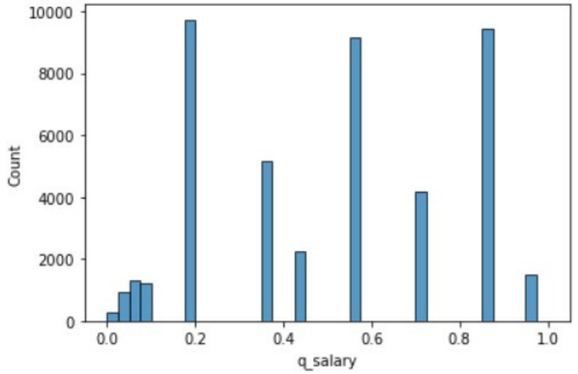
a non-parametric data transformation technique to transform your numerical data distribution to following a certain data distribution (e.g. the**Gaussian Distribution or Uniform Distribution**)



QuantileTransformer (output_distribution='normal')



QuantileTransformer (output_distribution=uniform')



Related technique: Feature Discretization (binning) to replace numerical values with bin numbers

e.g., No of times defaulted on a loan 0,1,2,3,4 can be binnedin to 0 (never) or 1 (have defaulted one + times