Copyright Notice

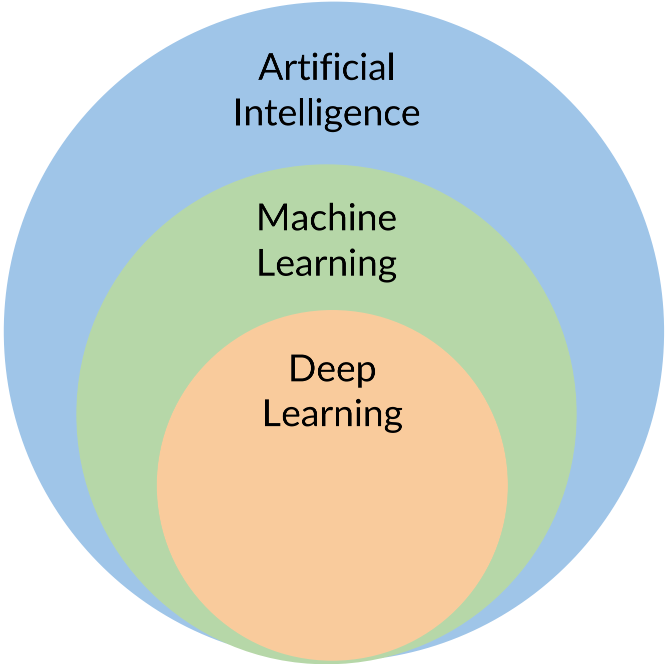
Practical Data Science

Explore the Use Case and Analyze the Dataset

Practical Data Science in the Cloud

Introduction

AI, ML, DL, data science…?

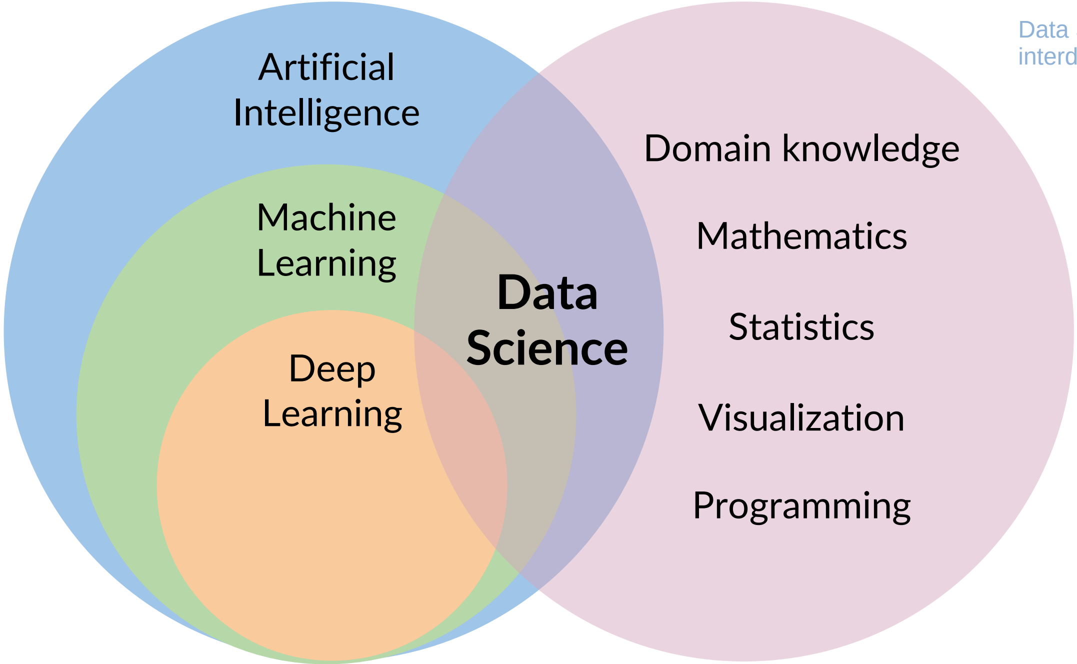


AI is a technique that lets machines mimic Human behaviour.

Machine Learning is a subset of AI that uses statistical methods and algorithms that are able to learn from the data without being explicitly programmed.

Deep Learning is again a subset of ML that uses Artificial Neural Networks to learn from the data.

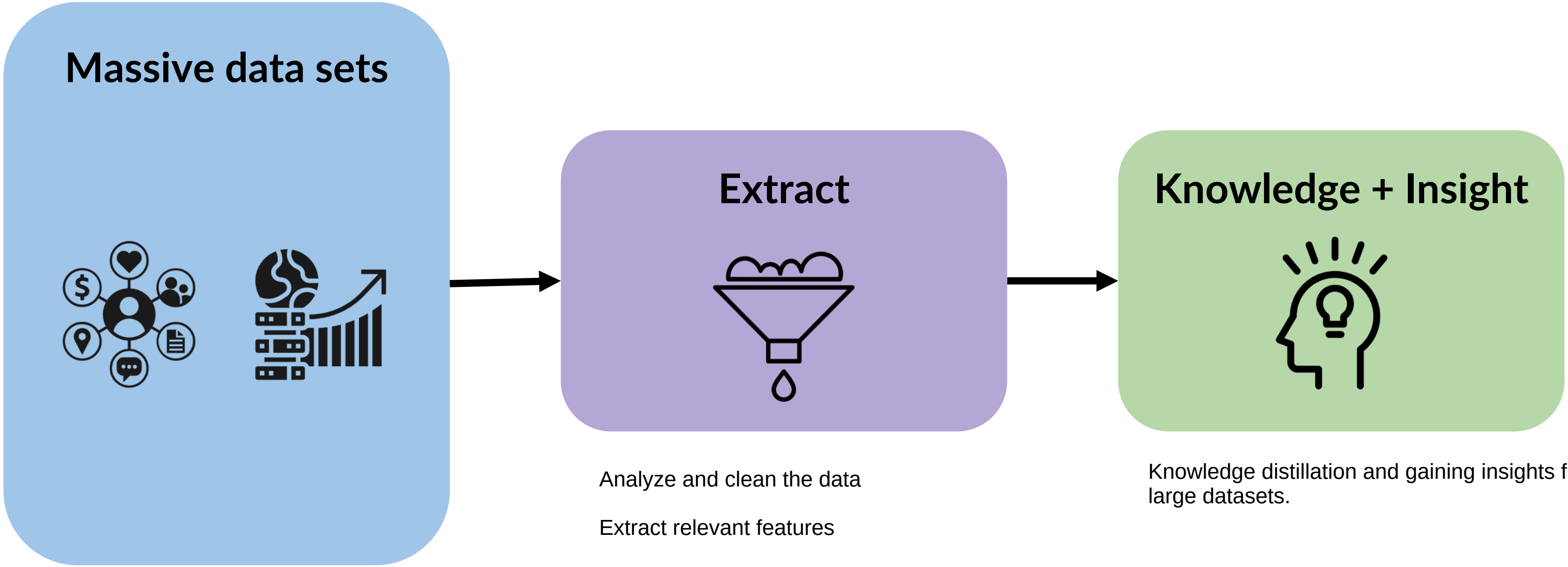
AI, ML, DL, data science…?



Data Science is a discipline that touches all fields. It is an interdisciplinary field.

Practical Data Science?

Practical data science



Massive data sets

Extract

Knowledge + Insight

Can originate from Social Media channels, mobile and web applications, public or company internal data sources, etc

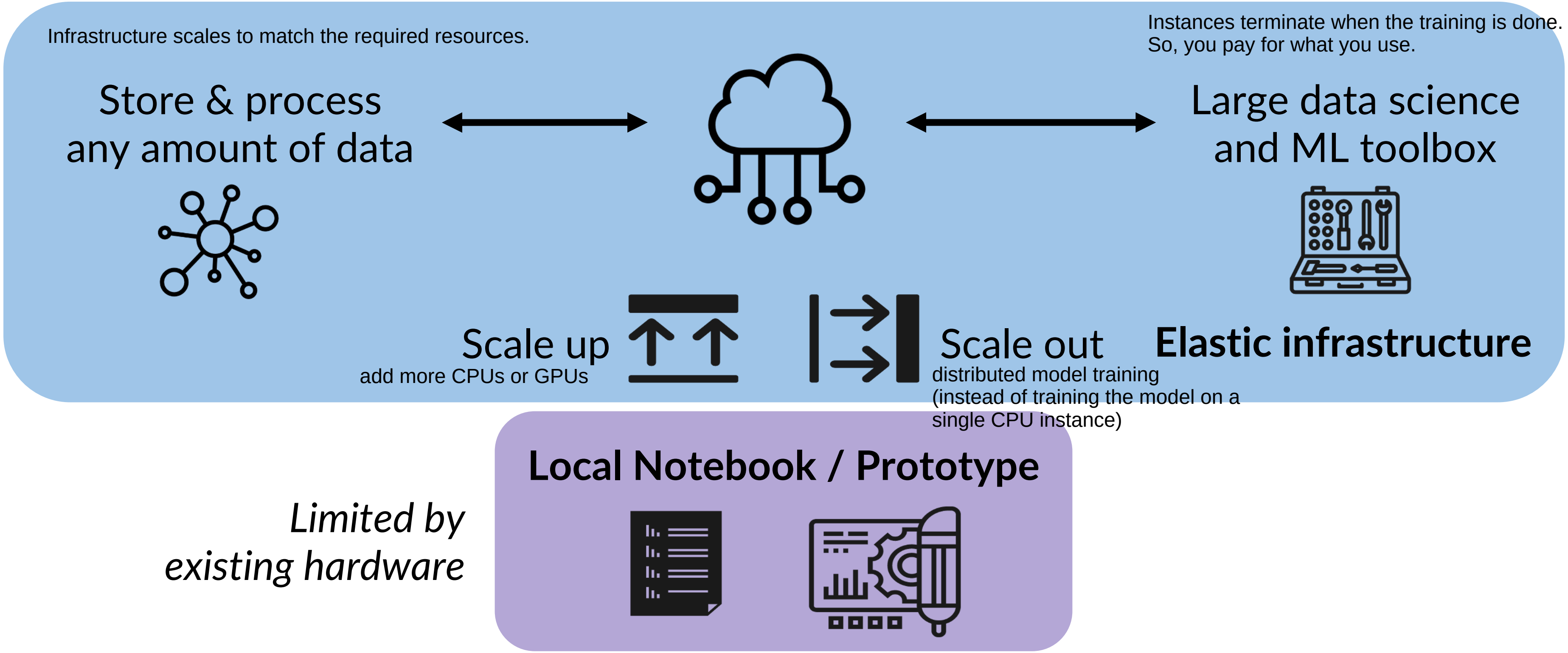
Analyze and clean the data

Extract relevant features

Knowledge distillation and gaining insights from large datasets.

… in the *Cloud*?

Practical data science in the cloud



Infrastructure scales to match the required resources.

Instances terminate when the training is done.
So, you pay for what you use.

Store & process
any amount of data

Large data science
and ML toolbox

Scale up
add more CPUs or GPUs

Scale out
distributed model training
(instead of training the model on a
single CPU instance)

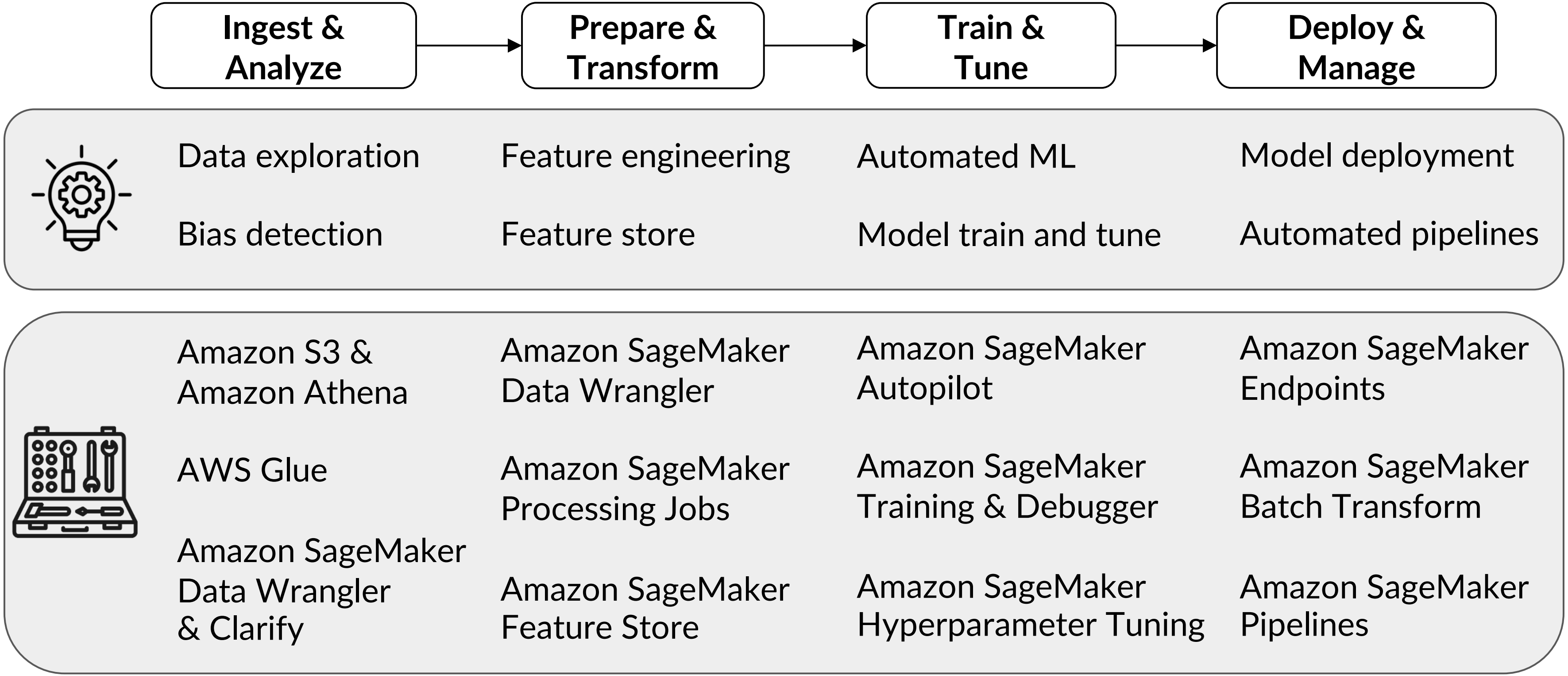
Elastic infrastructure

Local Notebook / Prototype

*Limited by
existing hardware*

Data science and ML toolbox

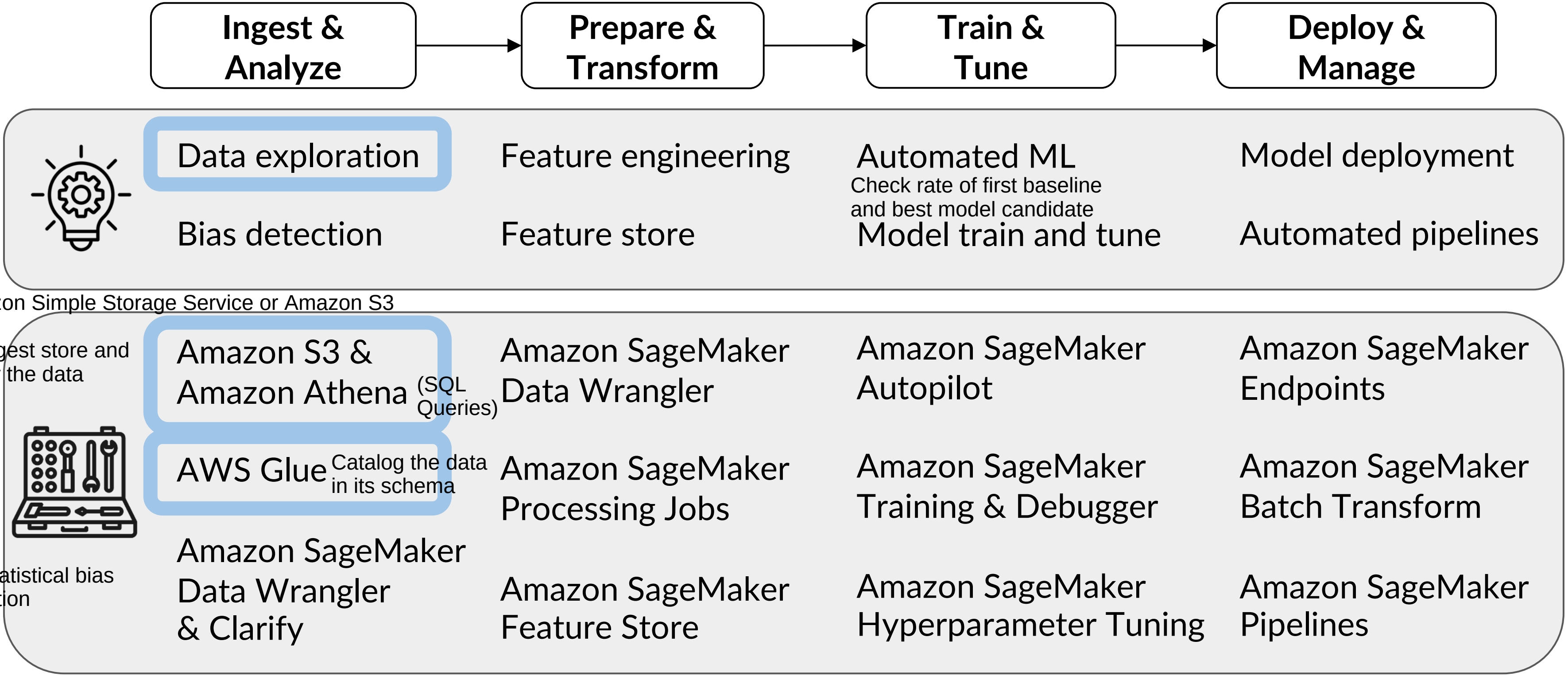
Machine Learning Workflow



Ingest & Analyze	Prepare & Transform	Train & Tune	Deploy & Manage
Data exploration	Feature engineering	Automated ML	Model deployment
Bias detection	Feature store	Model train and tune	Automated pipelines
Amazon S3 & Amazon Athena	Amazon SageMaker Data Wrangler	Amazon SageMaker Autopilot	Amazon SageMaker Endpoints
AWS Glue	Amazon SageMaker Processing Jobs	Amazon SageMaker Training & Debugger	Amazon SageMaker Batch Transform
Amazon SageMaker Data Wrangler & Clarify	Amazon SageMaker Feature Store	Amazon SageMaker Hyperparameter Tuning	Amazon SageMaker Pipelines

Machine Learning Workflow

Learn different model deployment and strategies and how to orchestrate the model development as an automated pipeline.



Ingest & Analyze → Prepare & Transform → Train & Tune → Deploy & Manage

Ingest & Analyze	Prepare & Transform	Train & Tune	Deploy & Manage
Data exploration	Feature engineering	Automated ML Check rate of first baseline and best model candidate	Model deployment
Bias detection	Feature store	Model train and tune	Automated pipelines
Amazon S3 & Amazon Athena (SQL Queries)	Amazon SageMaker Data Wrangler	Amazon SageMaker Autopilot	Amazon SageMaker Endpoints
AWS Glue Catalog the data in its schema	Amazon SageMaker Processing Jobs	Amazon SageMaker Training & Debugger	Amazon SageMaker Batch Transform
Amazon SageMaker Data Wrangler & Clarify	Amazon SageMaker Feature Store	Amazon SageMaker Hyperparameter Tuning	Amazon SageMaker Pipelines

Amazon Simple Storage Service or Amazon S3

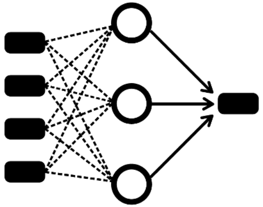
To ingest store and query the data

For statistical bias detection

Use Case and Dataset

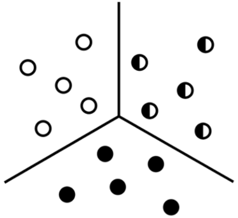
Introduction

Popular ML tasks and learning paradigms



Classification & Regression

Supervised



Clustering

Unsupervised

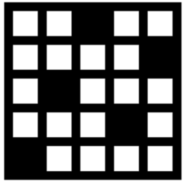
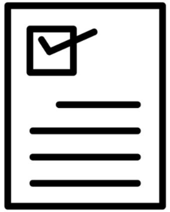


Image Processing

Computer Vision

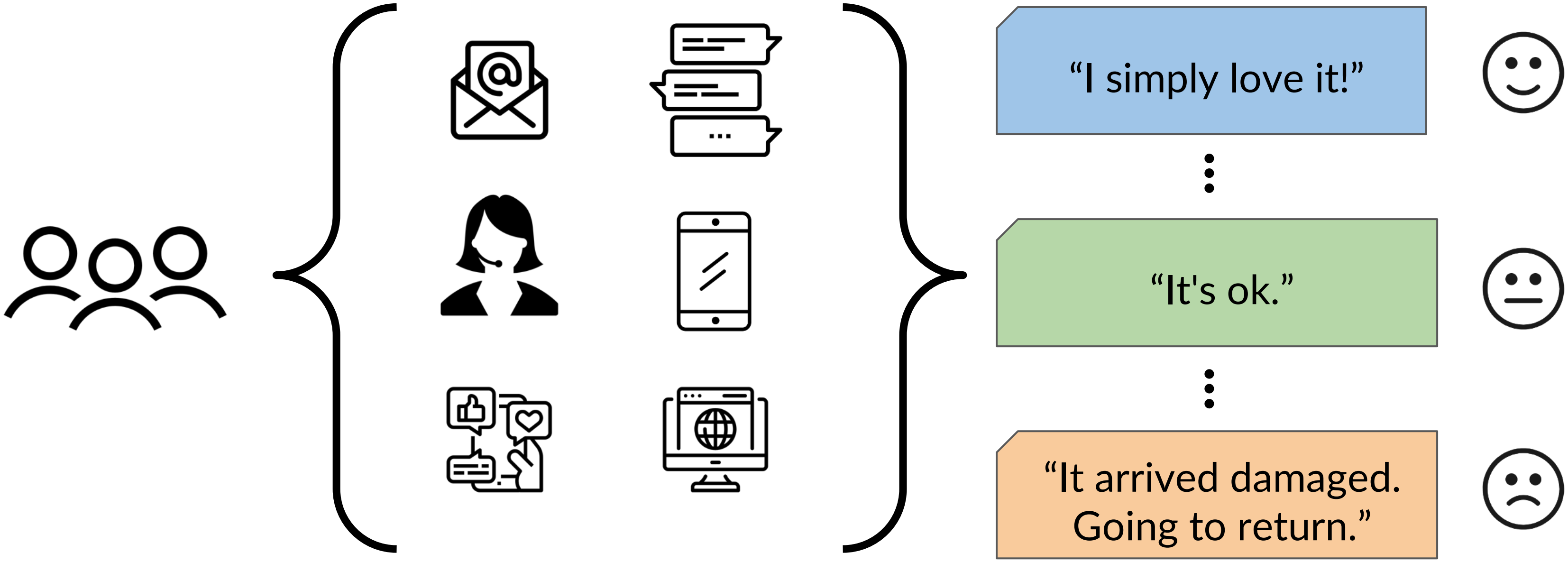


Text Analysis

NLP / NLU

Natural Language Processing
Natural Language Understanding

Multi-class classification for sentiment analysis of product reviews



“I simply love it!”

“It's ok.”

“It arrived damaged. Going to return.”

- Create an NLP model that will take product reviews as inputs.
- Then use the model to classify the sentiment of the reviews into the three classes of positive, neutral, and negative.

Working with product reviews data

Input feature for model training	Label for model training
Review Text	**Sentiment**
I simply love it!	1 (positive)
It's ok.	0 (neutral)
It arrived damaged, going to return	-1 (negative)

Data Ingestion & Exploration

- SCALABILITY is a great advantage of working on the cloud.

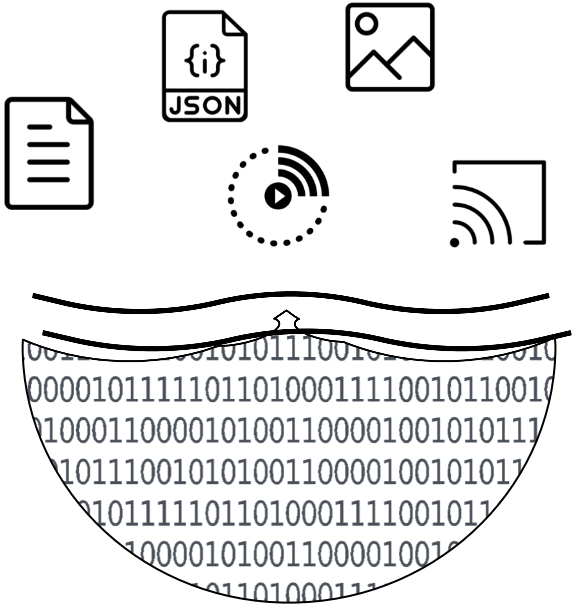
The infrastructure scales elastically with the size of your data.
Imagine your company is collecting all customer feedback across all online channels.

You need to capture customer feedback streaming from social media channels, feedback captured and transcribed through support center calls, incoming emails, mobile apps, and website data, and much more.

Deal with structured data (CSV files) and unstructured data, such as, support center call audio files

Elastically scale the storage capacities as new data arrives. - Cloud based data lakes address this problem

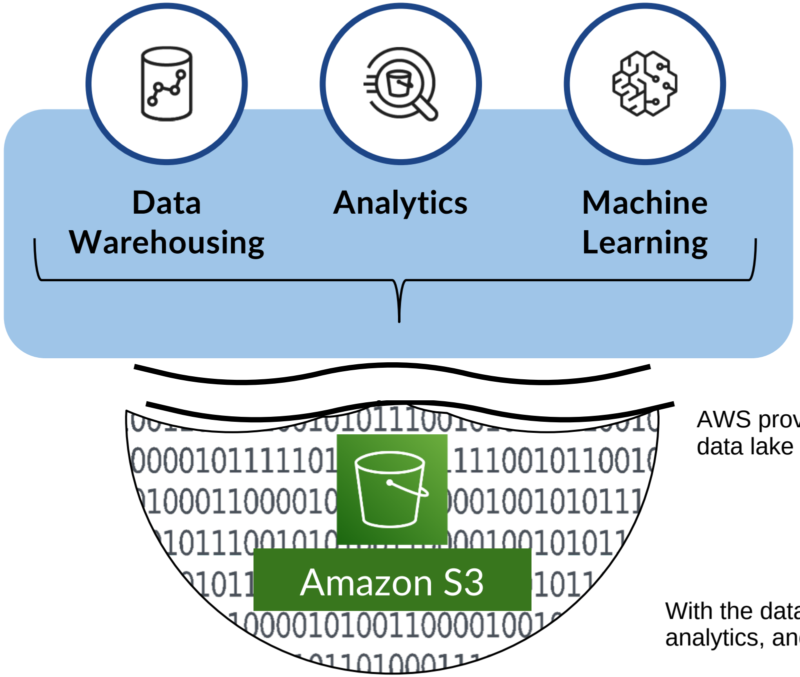
Ingest data into data lakes



- Centralized and secure repository
- Store, discover and share data at any scale
  - structured relational data such as CSV or TSV files
  - semi-structured data JSON OR XML files
  - unstructured data images, audio, and media files
  - streaming data an application delivering continuous feed of log files, or feeds from social media channels, into your data lake.
- Governance The data needs to be governed.

With data arriving at any time you need to implement ways:
 - discover
- catalog the new data.
- the data needs to comply with the political data security, privacy, and governance regulations.

Data lakes on Amazon S3



Data lakes are often built on object storage, such as Amazon S3.

File Storage: It stores and manages the files as individual files organized in hierarchical file folder structures.

In contrast, Block Storage stores and manages data as individual chunks called the blocks. Each block receives a unique identifier, but no additional metadata is stored with that block.

With Object Storage, data is stored and managed as objects, which consists of the data itself, any relevant metadata, such as when the object was last modified, and a unique identifier.

- Amazon Simple Storage Service (Amazon S3)

Object storage is super helpful when storing and retrieving growing amounts of data of any type. Hence, it is a great foundation for data lakes.

- Object storage

Amazon S3 gives you access to durable and high- available object storage in the cloud.

- Durable, available, exabyte scale

AWS provides additional tools and services to assist you in building a secure, compliant, and auditable data lake on top of S3.

- Secure, compliant, auditable

With the data lake in place, you can now use this centralized data repository to enable data warehousing, analytics, and Machine Learning.

AWS Data Wrangler is an open source Python library,

AWS Data Wrangler

- Open source Python library
- Connects pandas DataFrames and AWS data services
- Load/unload data from
  - data lakes
  - data warehouses
  - databases

```
!pip install awswrangler

import awswrangler as wr
import pandas as pd

# Retrieving the data directly from Amazon
S3
df = wr.s3.read_csv(
         path='s3://bucket/prefix/')
```

A Data Catalog is a collection of metadata, combined with data management and search tools, that helps analysts and other data users to find the data that they need, serves as an inventory of available data, and provides information to evaluate fitness data for intended uses.

Register data with AWS Glue Data Catalog

AWS Glue Data Catalog: This data catalog service is used to register or catalog the data stored in S3data lake, or bucket, as an individual container for object is called.

Using the Data Catalog Service, you create a reference to data "S3-to-table" mapping.

AWS Glue Data Catalog

The AWS Glue table, which is created inside an AWS Glue database, only contains the metadata information such as the data schema.

Name	reviews
Database	dsoaws_deep_learning
Classification	csv
Location	s3://<bucket>/<prefix>

- Creates reference to data ("S3-to-table" mapping)
- Just metadata / schema stored in tables
- No data is moved
- *AWS Glue Crawlers* can be set up to automatically
  - infer data schema
  - update data catalog

It's important to note that no data is moved. All the data remains in your S3 location.

You catalog where to find the data and which schema should be used, to query the data.

Instead of manually registering the data, you can also use AWS Glue Crawler.

A Crawler can be used and set up to run on a schedule (ETL) or to automatically find new data, which includes inferring the data schema and also to update the data catalog.

Register data with AWS Glue Data Catalog

AWS Glue
Data Catalog

Name	reviews
Database	dsoaws_deep_learning
Classification	csv
Location	s3://<bucket>/<prefix>

```
import awswrangler as wr

# Create a database in the
# AWS Glue Data Catalog
wr.catalog.create_database(
         name=...)

# Create CSV table (metadata only) in the
# AWS Glue Data Catalog
wr.catalog.create_csv_table(
         table=...,
         column_types=...,
     ...)
```

Query data with Amazon Athena

Athena is an interactive queries service that lets you run standard SQL queries to explore your data. Athena is serverless, which means you don't need to set up any infrastructure to run those queries, and, no matter how large the data is that you want to query, you can simply type your SQL query, referencing the dataset schema you provided in the AWS Glue Data Catalog. No data is loaded or moved.

Amazon Athena

- Query data in S3
- Using SQL
- No infrastructure to set up
- Schema lookup in AWS Glue Data Catalog
- No data to load

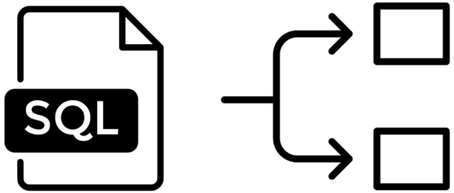
Python

```python
import awswrangler as wr

# Create Amazon Athena S3 bucket
wr.athena.create_athena_bucket()

# Execute SQL query on Amazon Athena
df = wr.athena.read_sql_query(
    sql=...,
    database=...)
```

SQL

```sql
'SELECT product_category FROM reviews'
```

Query data with Amazon Athena



- Complex analytical queries
- Gigabytes > Terabytes > Petabytes
- Scales automatically
- Runs queries in parallel
- Based on Presto

  Presto: an open source distributed SQL engine, developed for this exact use case, running interactive queries against data sources of all sizes.
- No infrastructure setup / no data movement required

Data Visualization

Popular Python data analysis & visualization tools

pandas

```
pip install pandas
```

NumPy

```
pip install numpy
```

matplotlib

```
pip install matplotlib
```

seaborn

```
pip install seaborn
```

How many reviews are in each *sentiment class*?

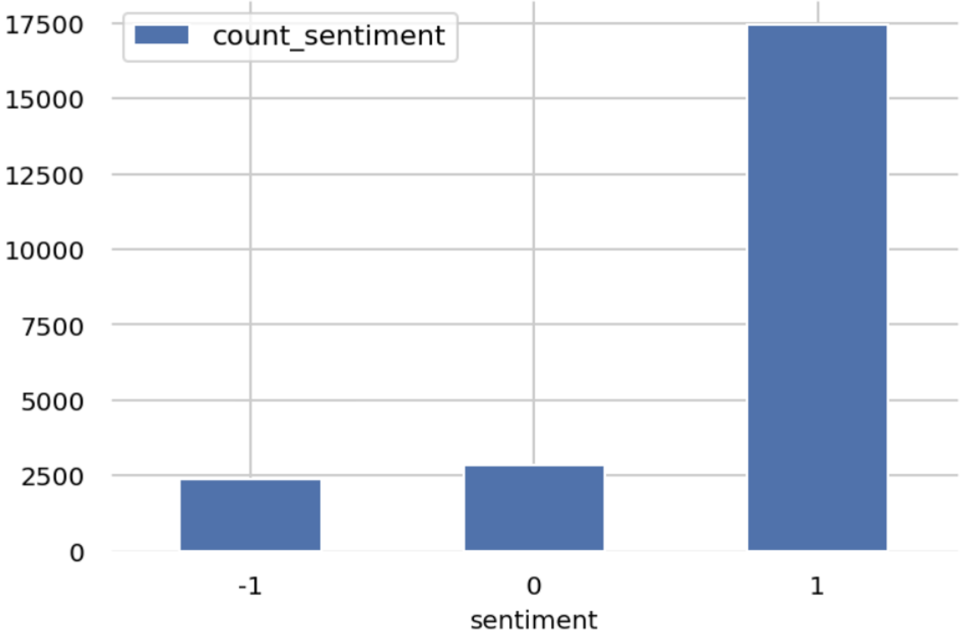
SQL Query

```sql
SELECT sentiment, COUNT(*) AS count_sentiment
FROM dsoaws_deep_learning.reviews
GROUP BY sentiment
ORDER BY sentiment DESC, count_sentiment
```

Python visualization code

```python
import matplotlib.pyplot as plt
chart = df.plot.bar(
         x="sentiment",
     y="count_sentiment")

plt.xlabel("sentiment")
plt.show(chart)
```

How many reviews are in each *sentiment class*?



What is the distribution of review lengths? *(number of words)*

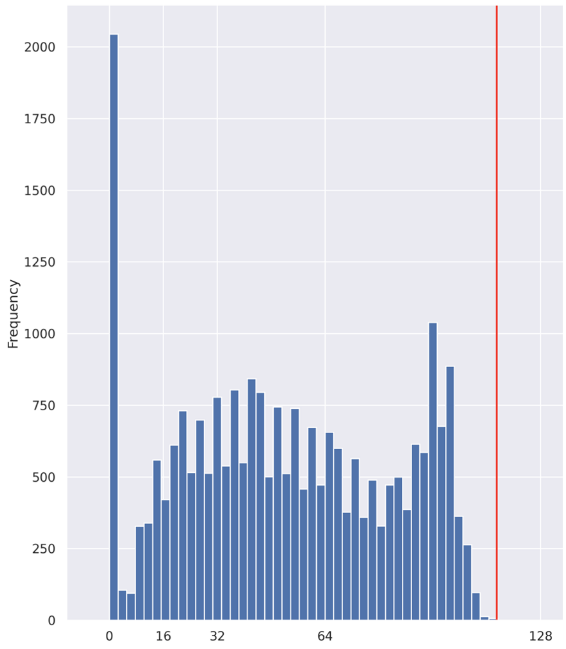
SQL Query

```
SELECT CARDINALITY(SPLIT(review_body, ' ')) as num_words
FROM dsoaws_deep_learning.reviews
```

Python visualization code

```
summary = df["num_words"].describe(
     percentiles=[0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00])

df["num_words"].plot.hist(
     xticks=[0, 16, 32, 64, 128, 256], bins=100,
     range=[0, 256]).axvline(x=summary["100%"], c="red")
```

What is the distribution of review lengths? *(number of words)*



```
mean         52.51
std          31.38
min           1.00
10%          10.00
20%          22.00
30%          32.00
40%          41.00
50%          51.00
60%          61.00
70%          73.00
80%          88.00
90%          97.00
100%        115.00
```