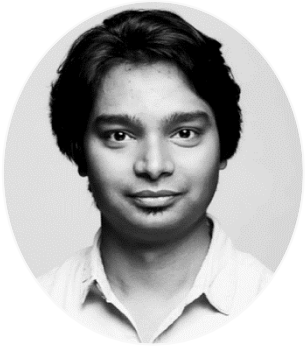
Preparing Data for Machine Learning Model: Part 1



Abhishek Kumar
DATA SCIENTIST | AUTHOR | SPEAKER

@meabhishekkumar

TensorFlow.js Ecosystem

TFJS data

TFJS viz

Overview

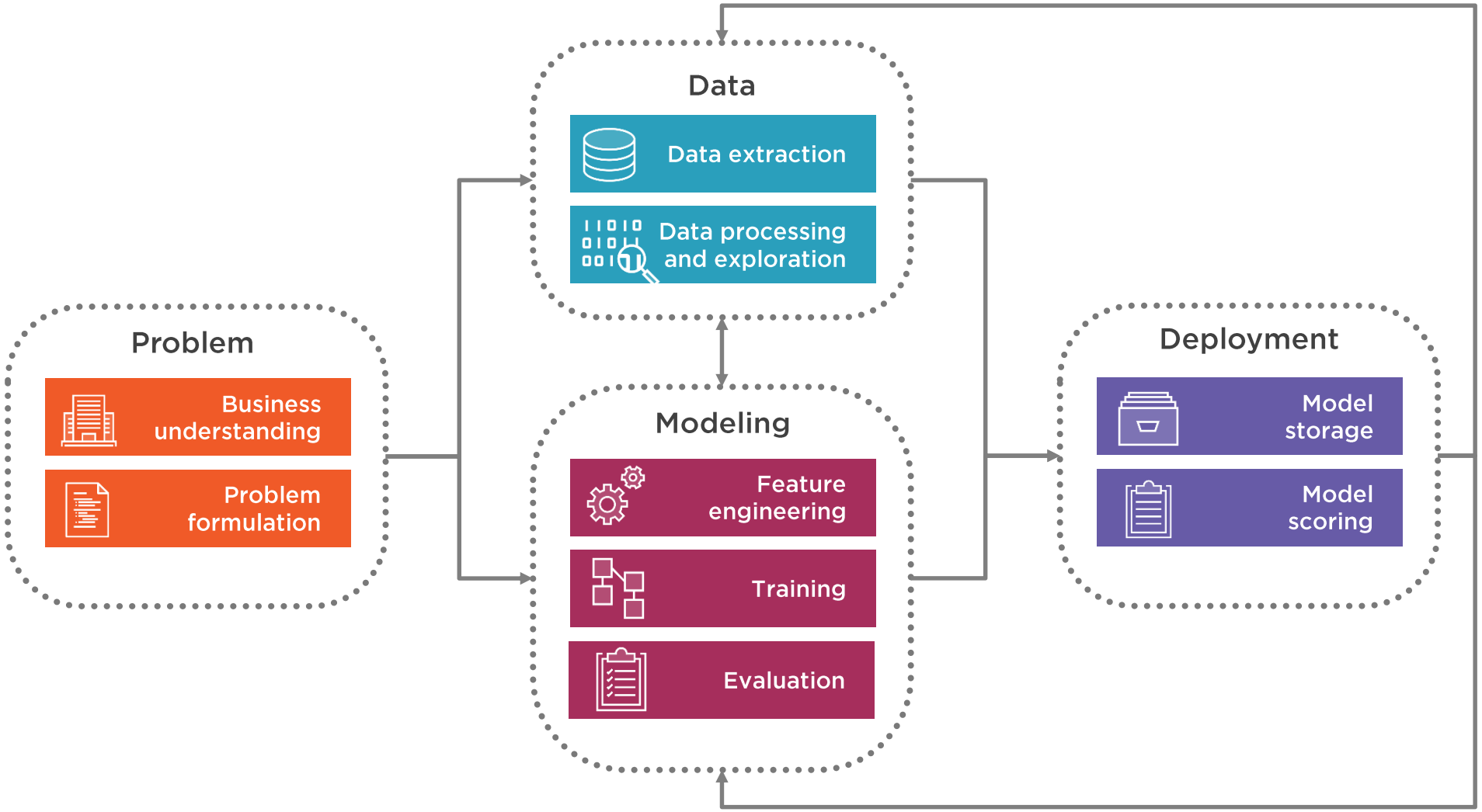
Machine learning workflow

Toxicity detection use case

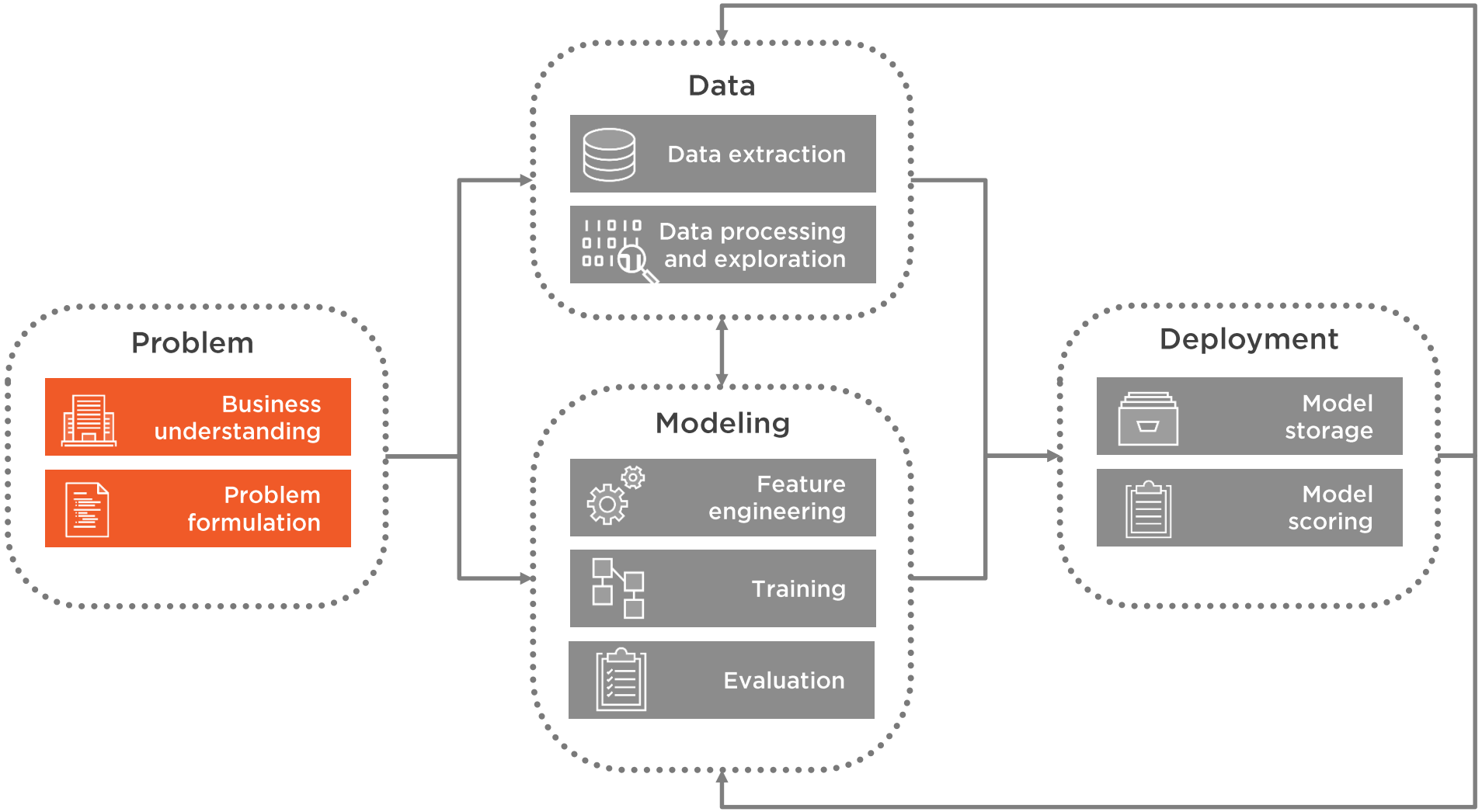
Import dataset

Visualize dataset

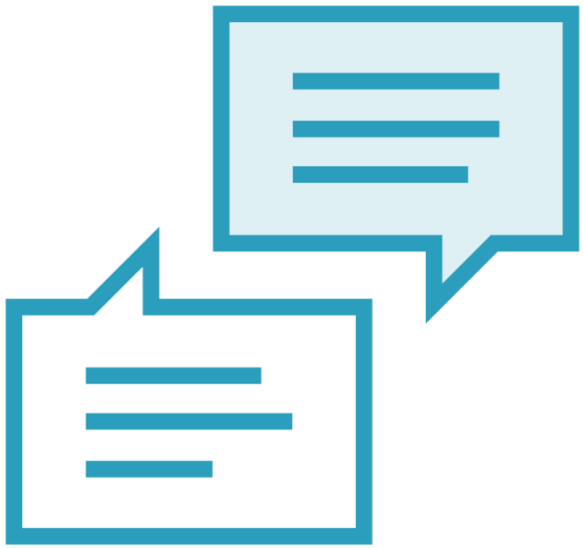
Machine Learning Workflow



Machine Learning Workflow



Toxicity Detection



Detect toxicity in comments

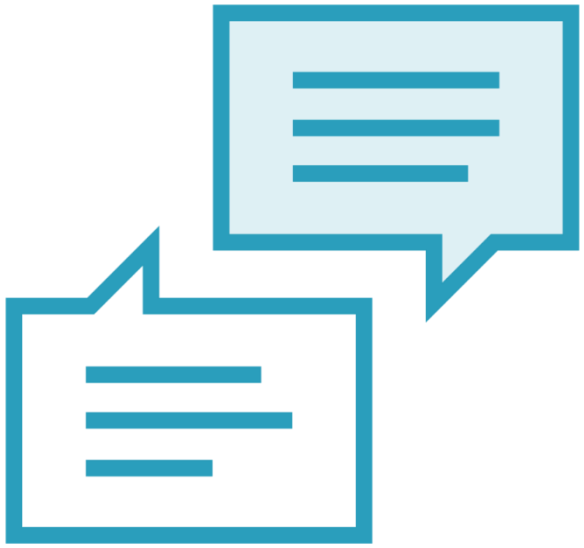
Toxicity dataset

- Conversation AI team (Jigsaw, Google)
- Kaggle challenge
- 159K Wikipedia comments with labels of different toxicity types such as threats, obscenity, insult, identity-based attack

Disclaimer: Dataset contains text that may be considered profane, vulgar, or offensive

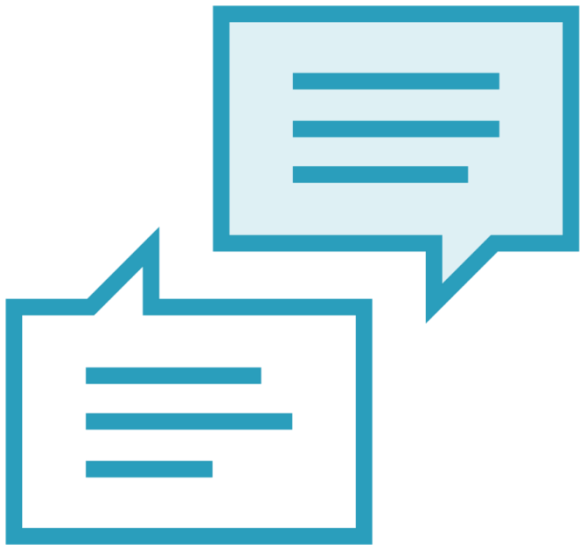
Source : https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

Toxicity Detection Dataset



- ID
- Comment
- Toxic
- Severe toxic
- Obscene
- Threat
- Insult
- Identity hate

Toxicity Detection Dataset



Cleaned version

Toxic column used for binary classification

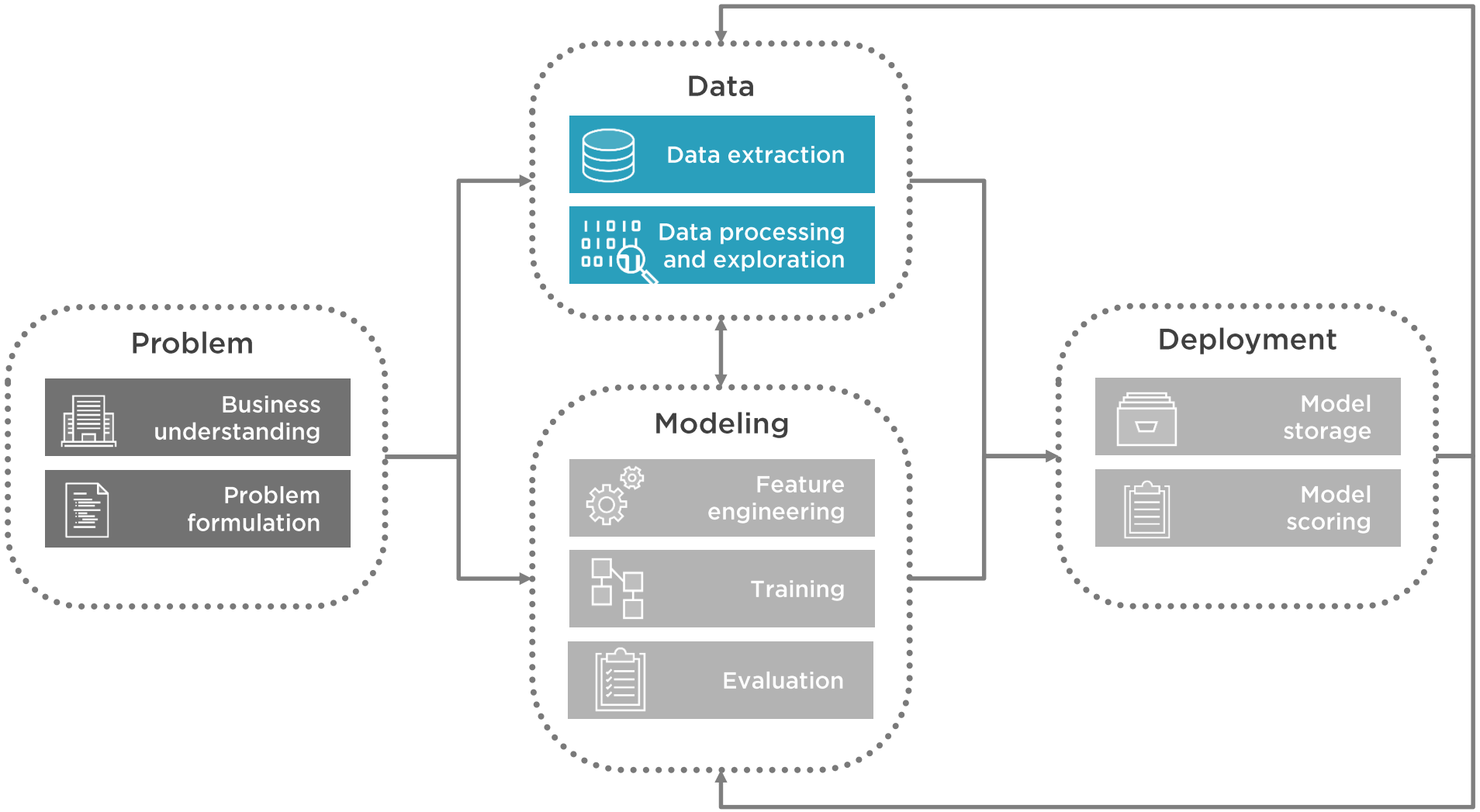
Small sample data

- Faster execution
- Focus on end to end ML workflow
- Learn TensorFlow.js concepts

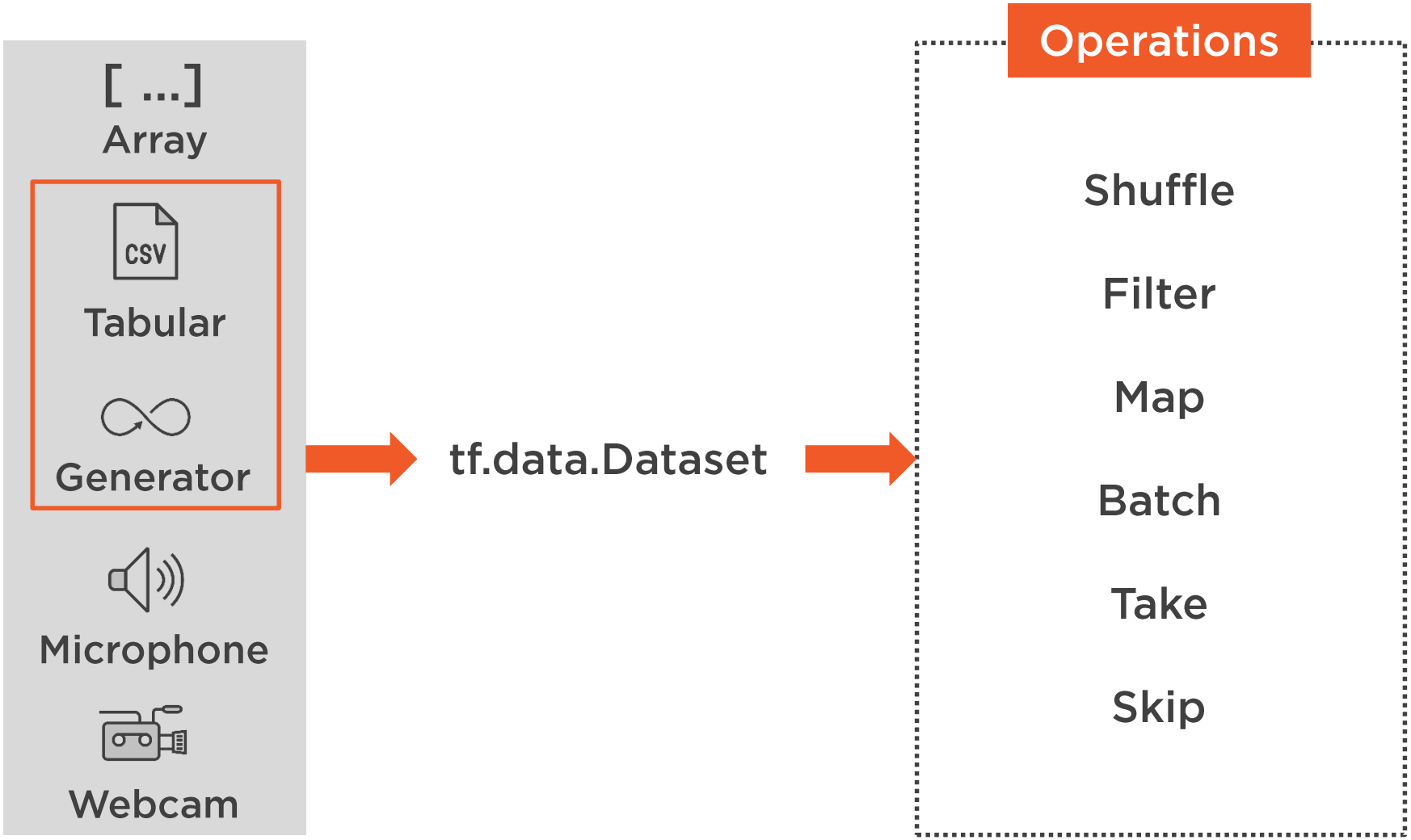
Try on larger dataset

- Node.js
- Preferably with GPU

Machine Learning Workflow



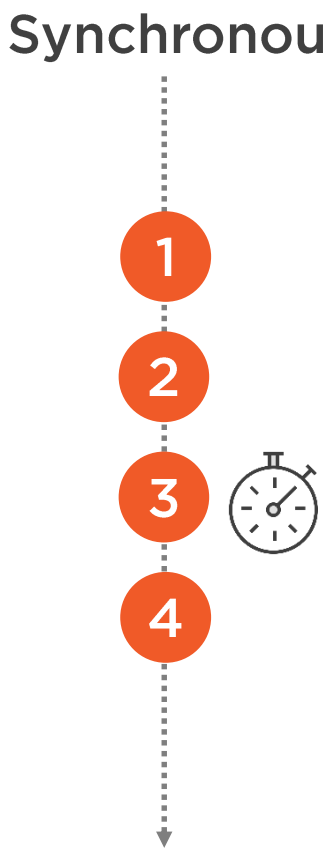
Working with TFJS Data

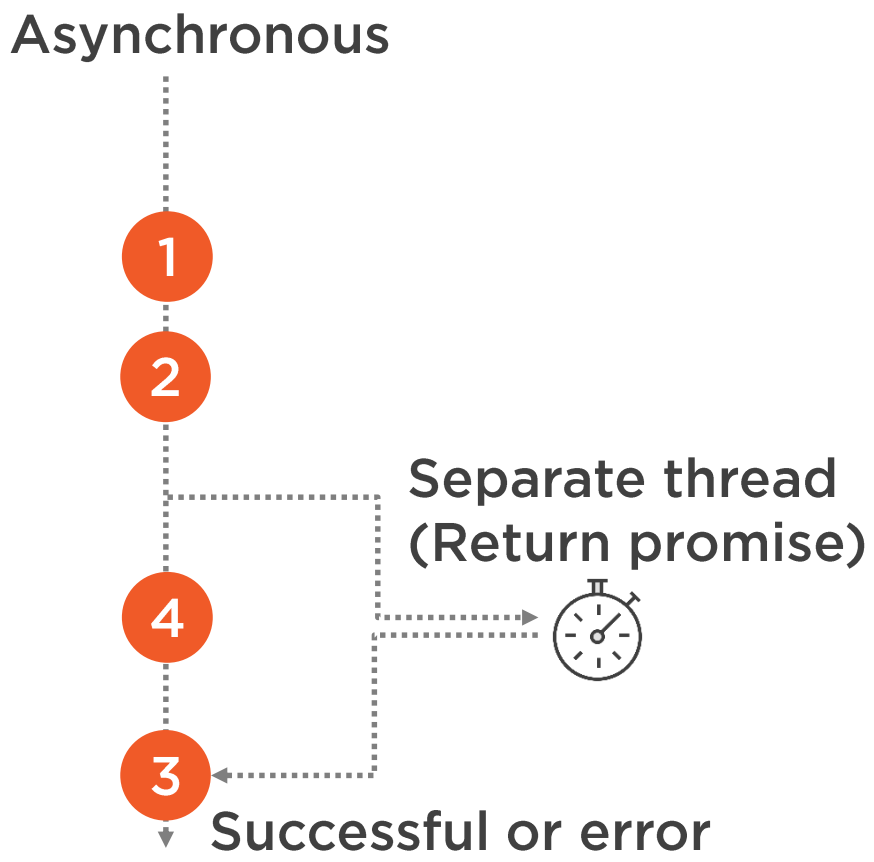


Async Task

JS code

#1
#2
#3 Long running task
#4

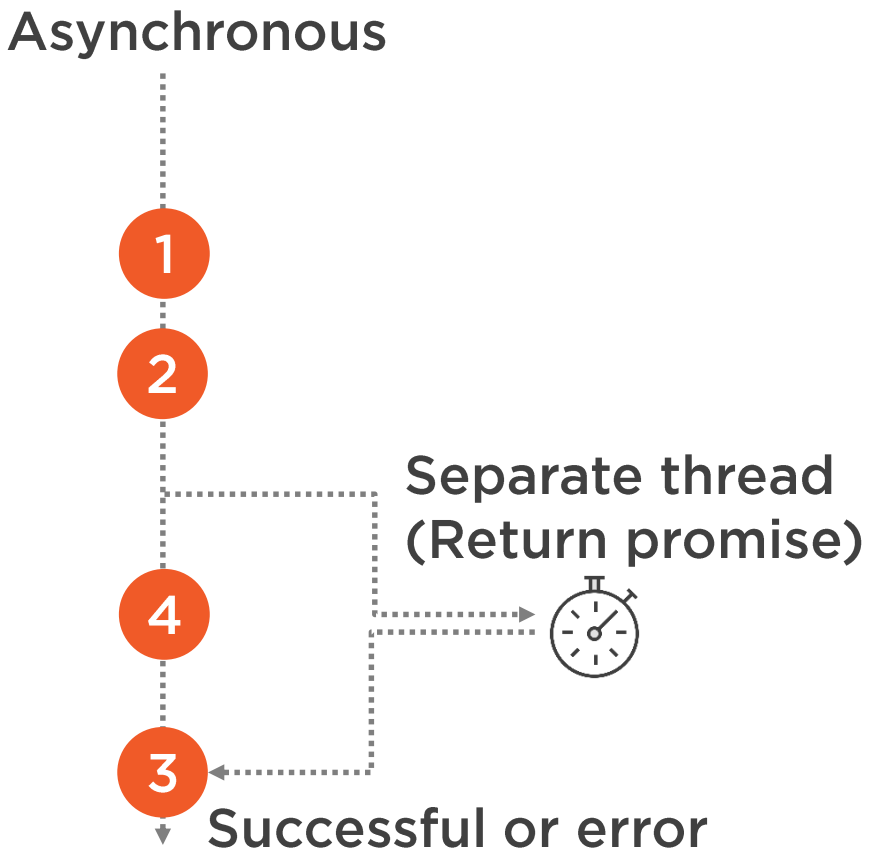




Async Task

JS code

```
console.log("line1");
console.log("line2");
tf.ready().then(function() {
 console.log(tf.getBackend());
});
console.log("line4");
```



Async Await

JS code

```
console.log("line1");
console.log("line2");
tf.ready().then(function() {
 console.log(tf.getBackend());
});
console.log("line4");
```

JS code with Async

```
console.log("line1");
console.log("line2");
async function init(){
    await tf.ready();
    console.log(tf.getBackend());
}
init();
console.log("line4");
```

ES6 Arrow Function

Function keyword

```
console.log("line1");
console.log("line2");
async function init(){
    await tf.ready();
    console.log(tf.getBackend());
}
init();
console.log("line4");
```

Arrow function

```
console.log("line1");
console.log("line2");
const init = async () => {
    await tf.ready();
    console.log(tf.getBackend());
}
init();
console.log("line4");
```

```
const run = async () => {
  const data = await readData(); // first async op
  console.log(data);
  const processed = await doProcessing(data); // pass results to second async op
  console.log(processed);

}

run();
```

Async Await

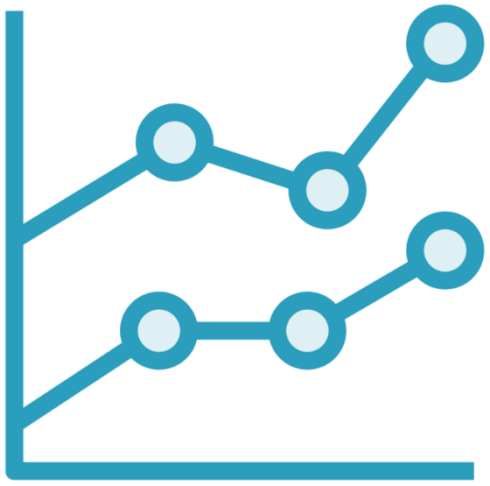
Makes asynchronous programing clean

Demo

Reading data using TFJS data

- Async await

Working with TFVis



Charts

- Bar chart, line chart, scatter plot

Model

- Model inspection
- Training performance
- Model evaluation and metrics

Visor for rendering visualization

Use output with other JS visualization libraries such as Plotly

Demo

Visualizing data using tfvis

Summary

Machine learning workflow

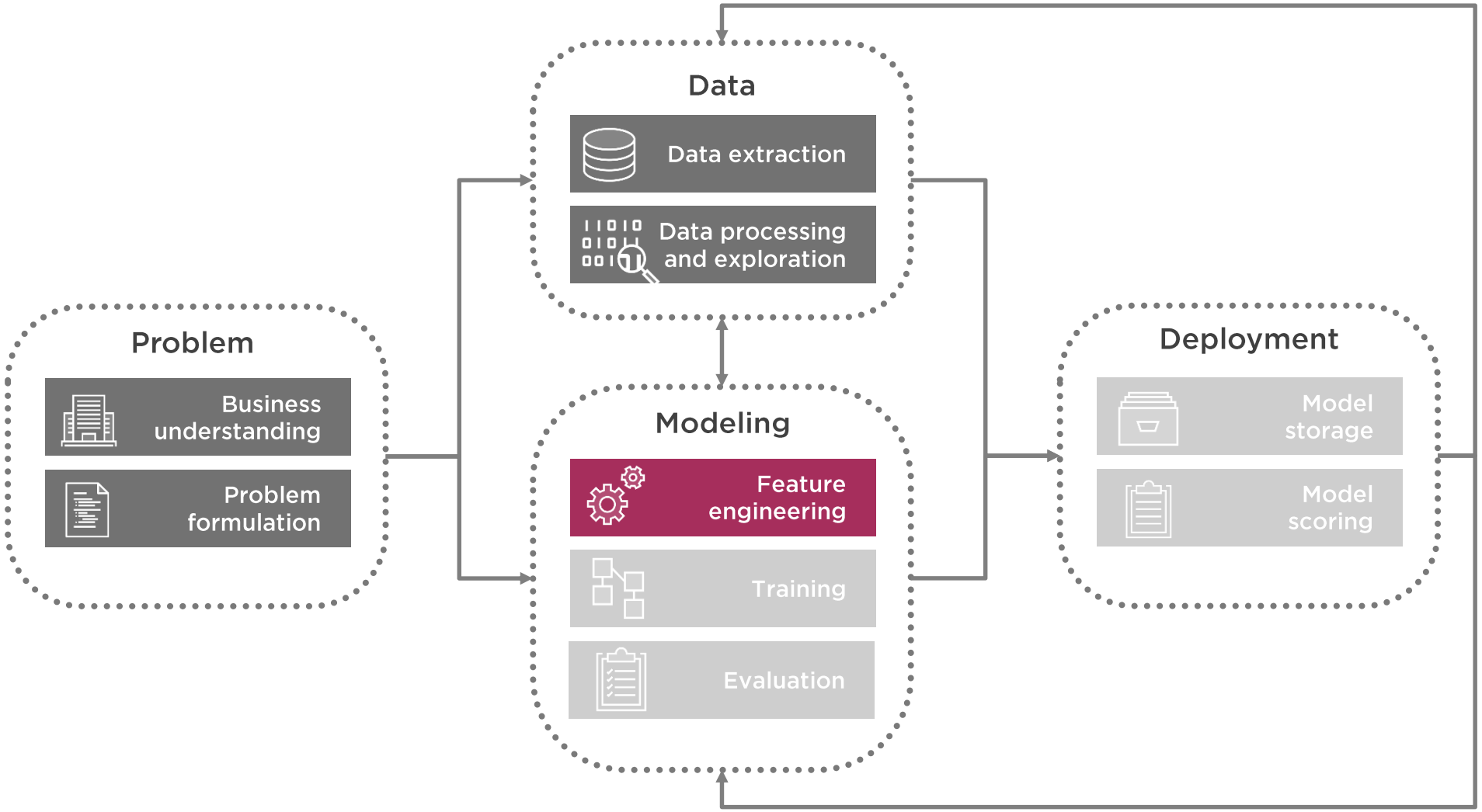
- Iterative
- Problem, data, modeling, deployment

Toxicity detection use case

Explore data

- Read data from CSV file
- Iterate through dataset
- Visualize dataset

Machine Learning Workflow



Up Next: Preparing Data for Machine Learning Model – Part 2