

# How we build AI Libraries for Front-end Developers and Use AI in Front-end Area

Xiaojian Wang  
Front-end and machine learning engineer  
@Alibaba-inc

1. What AI could do for Front-end

2. DataCook: Javascript Machine Learning Library

3. PipCook: JavaScript CI/CD Pipelines

# 1. What AI could do for Front-end

# What Front-end Could do

UI development

Front-end Infrastructure

Node.js (Server Development)

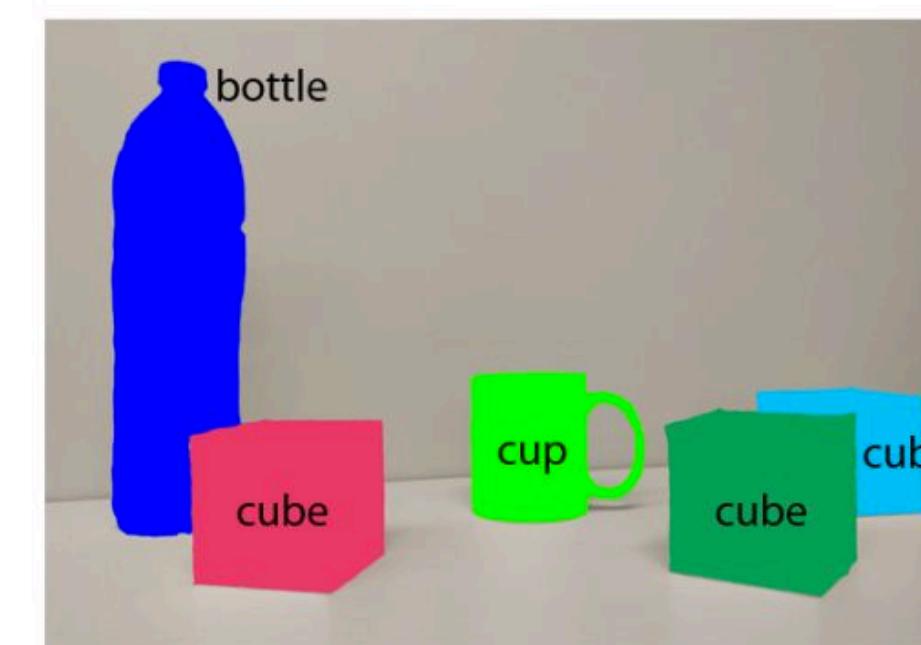
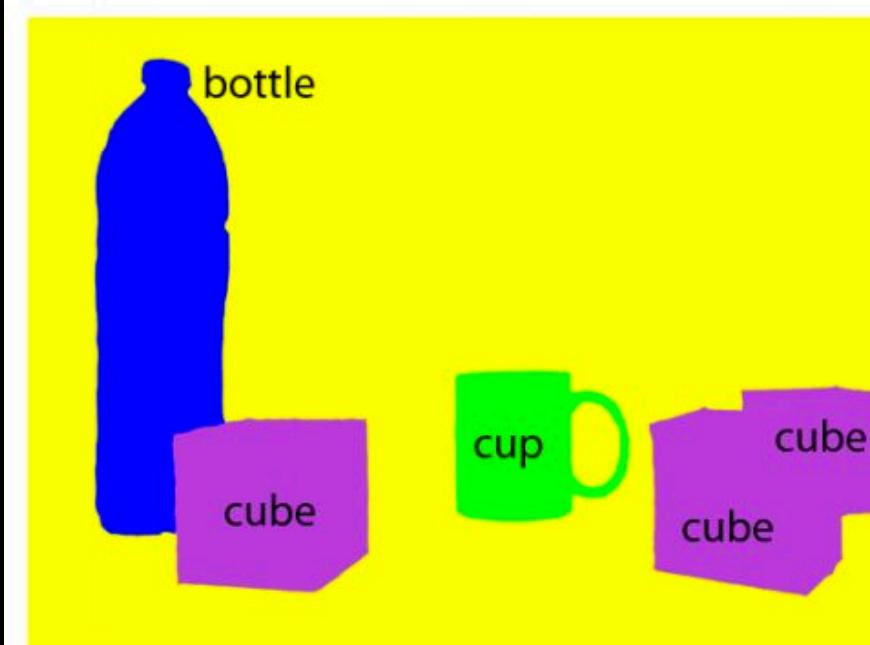
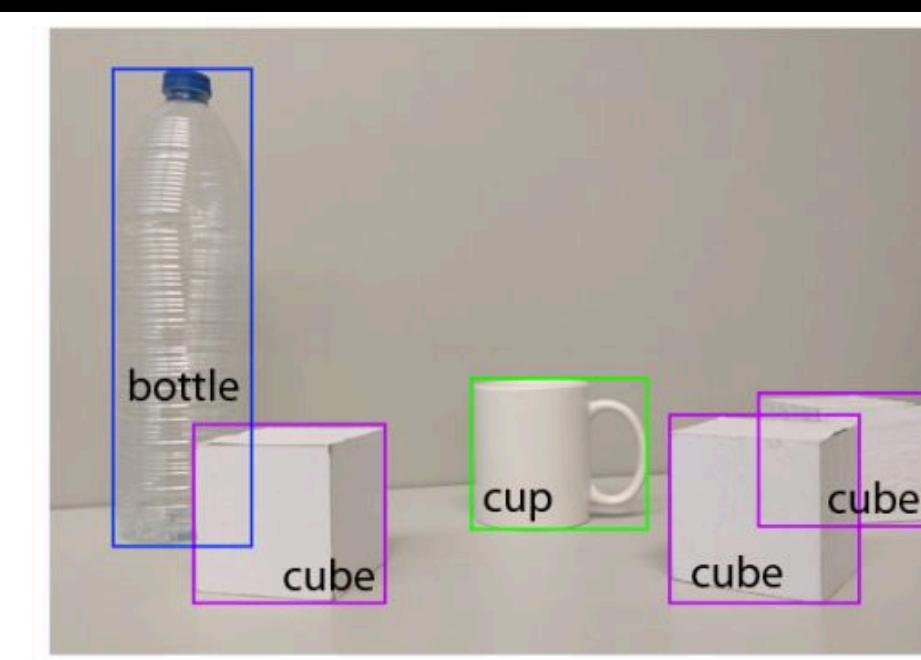
Data Visualization

Games



Human-computer interaction

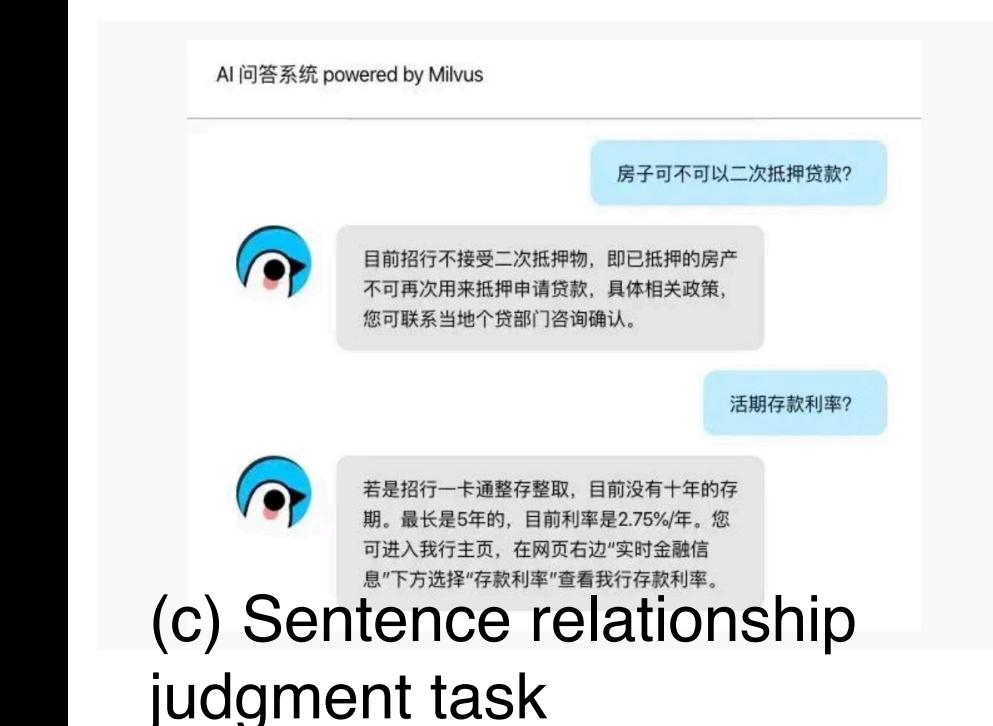
# What can AI do



Computer Vision

General **PER** Baril said that apart from the group of 150,000, **LOC** U.S. and **MISC** British reconnaissance plans had tracked two much smaller groups of refugees -- one of up to 1,000 north of the town of **LOC** Masisi and one of up to 8,000 on the road from **LOC** Bukavu west to **LOC** Kindu. The **MISC** Kisangani office of the medical charity **ORG** Medecins sans Frontieres said on Friday that more than 100,000 refugees were trekking northwest from the **LOC** Goma-Bukavu area and many of them were now in the town of **LOC** Walikale.

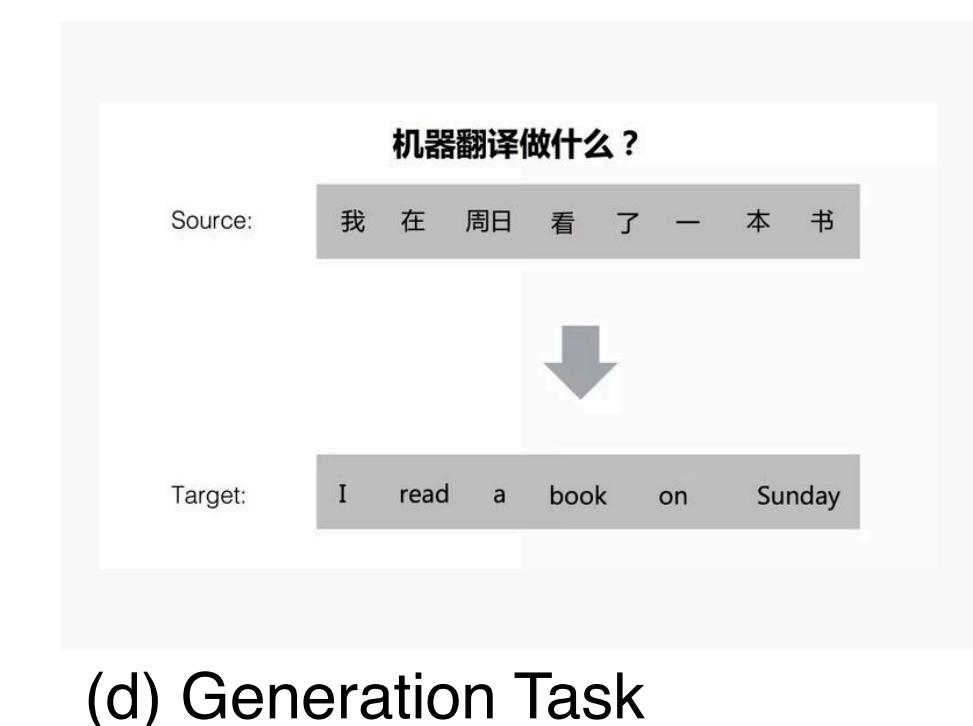
(a) Sequence Tagging



(c) Sentence relationship judgment task



(b) Text Classification



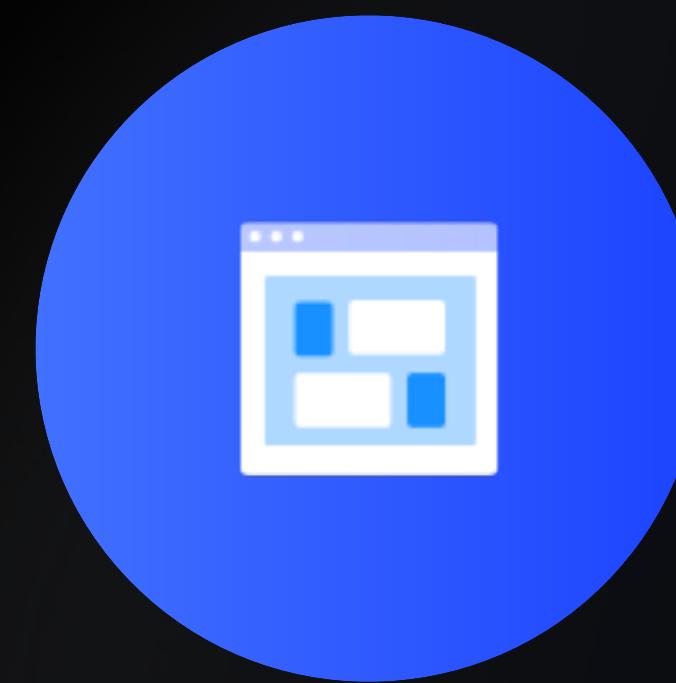
NLP

# What AI could do for Front-end



## Automatic or assisted programming

Improve developer experience and  
increase developer productivity

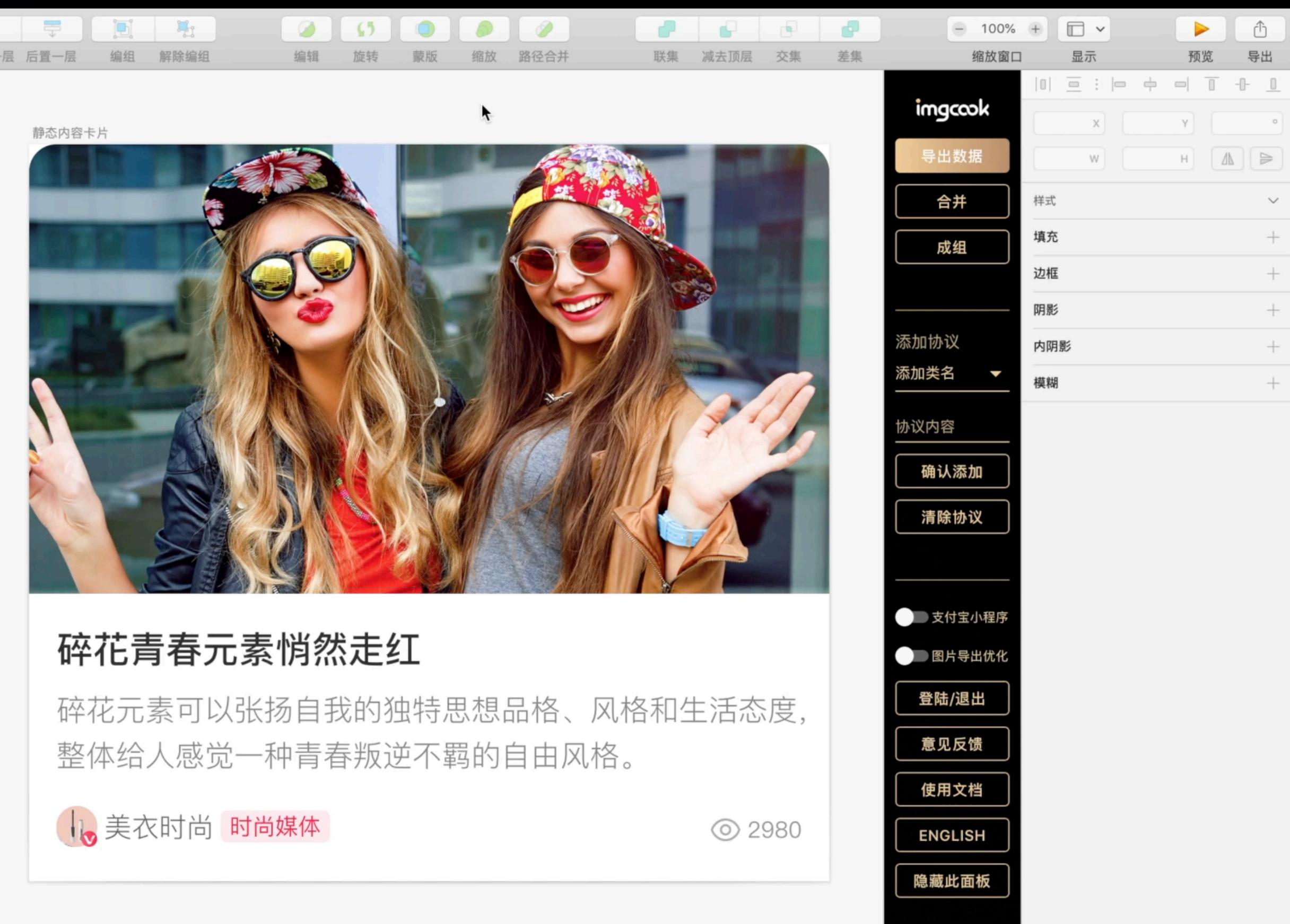


## User experience improvement

Enhance product user experience and  
enhance business value

# Auto Code Generation – ImgCook

Generate Code Automatically from Design Files – [www.imgcook.com](http://www.imgcook.com)



## Layer parsing

Extract structured information from design draft files

## UI information recognition

Recognize semantic information of UI interface images and texts through NLP, image classification, object detection and other models

## Layout Analysis

Infer page layout structure based on layout algorithm

## Code Generation

Generates corresponding Flutter, H5 and other codes

# Auto Code Generation – ImgCook

Use the classification model to identify the semantic type of the icon, which is used to give the classname an easy-to-understand name in the generated code



Like



Comment



Share

```
● ● ●  
<div className="primary-i0">  
  <div className="picture1-wrapper">  
    <img  
      className="picture1"  
      src={'https://img.alicdn.com/imgextra/i4/01CN0-34-32.png'}  
    />  
  </div>  
  <span className="like">Like</span>  
</div>
```

Without Icon  
Recognition

```
● ● ●  
<div className="primary-i0">  
  <div className="icon-good-wrapper">  
    <img  
      className="icon-good"  
      src={'https://img.alicdn.com/imgextra/i4/01CN0-34-32.png'}  
    />  
  </div>  
  <span className="like">Like</span>  
</div>
```

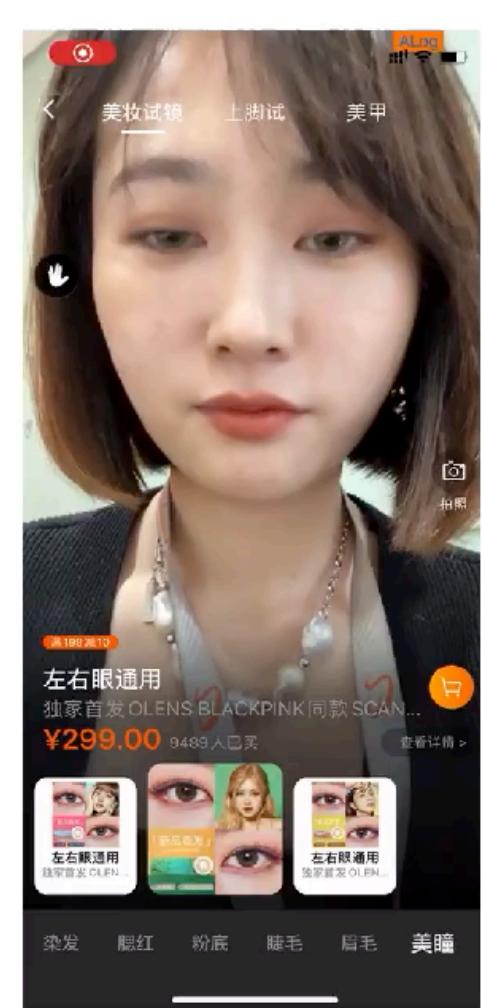
With Icon  
Recognition

# augmented reality

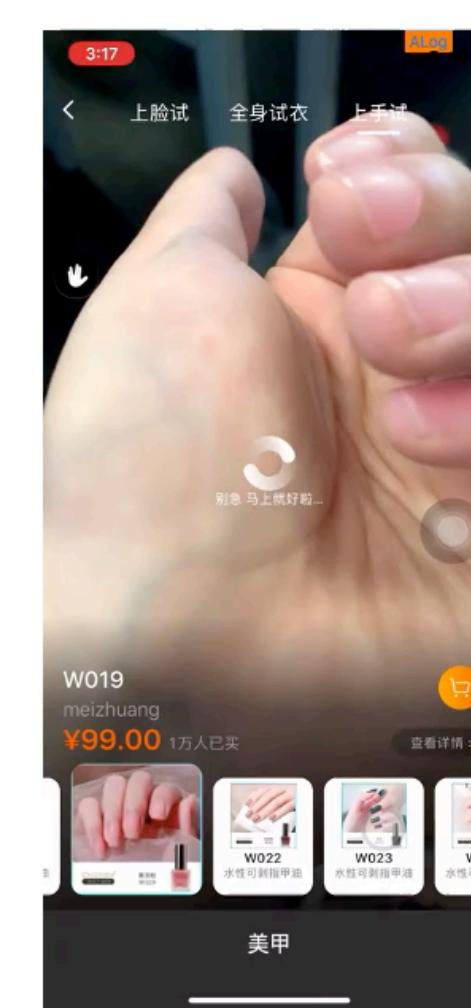
## ▼AR Beauty



AR试妆  
(口红/眼影/睫毛/眼线/眉毛/腮红/粉底)



AR美瞳

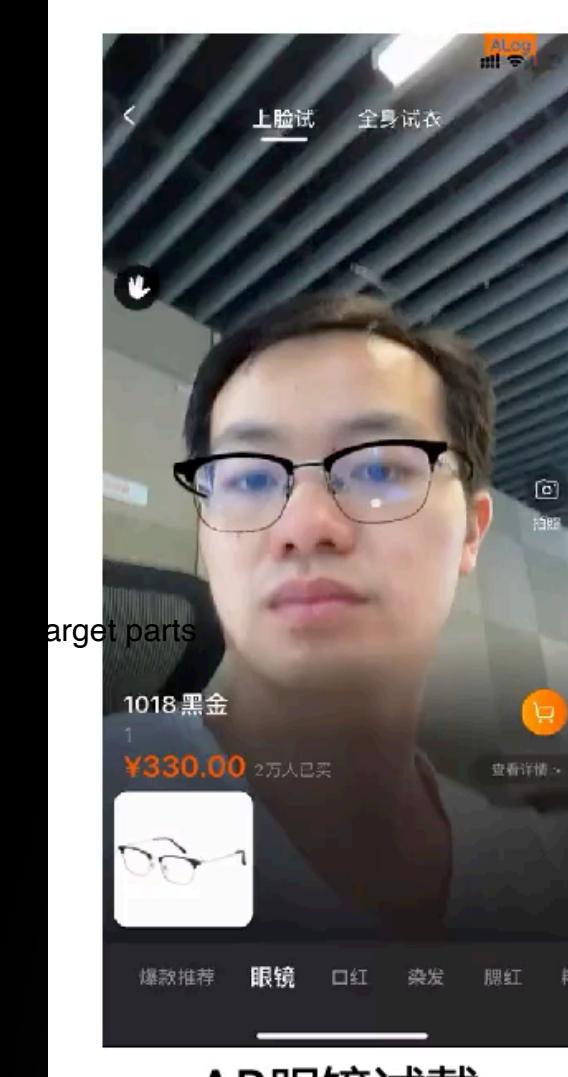


AR美甲



AR美发

## ▼ AR Wear



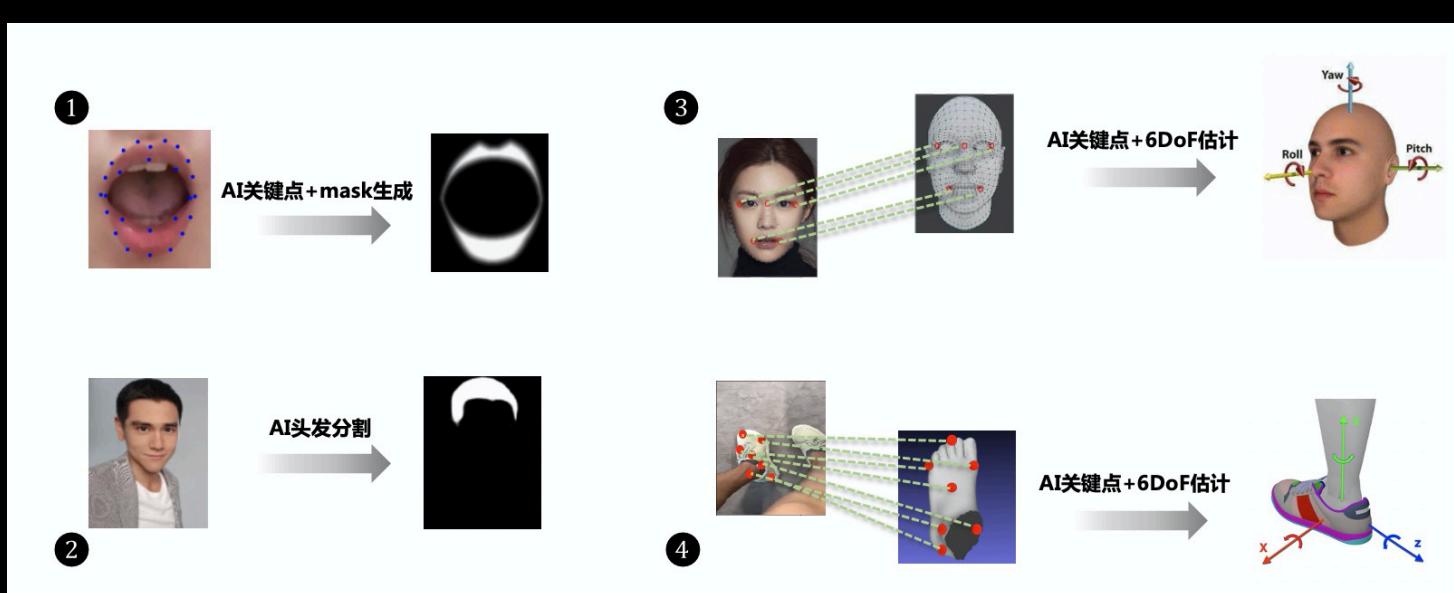
AR眼镜试戴



AR试鞋



AR手表试戴



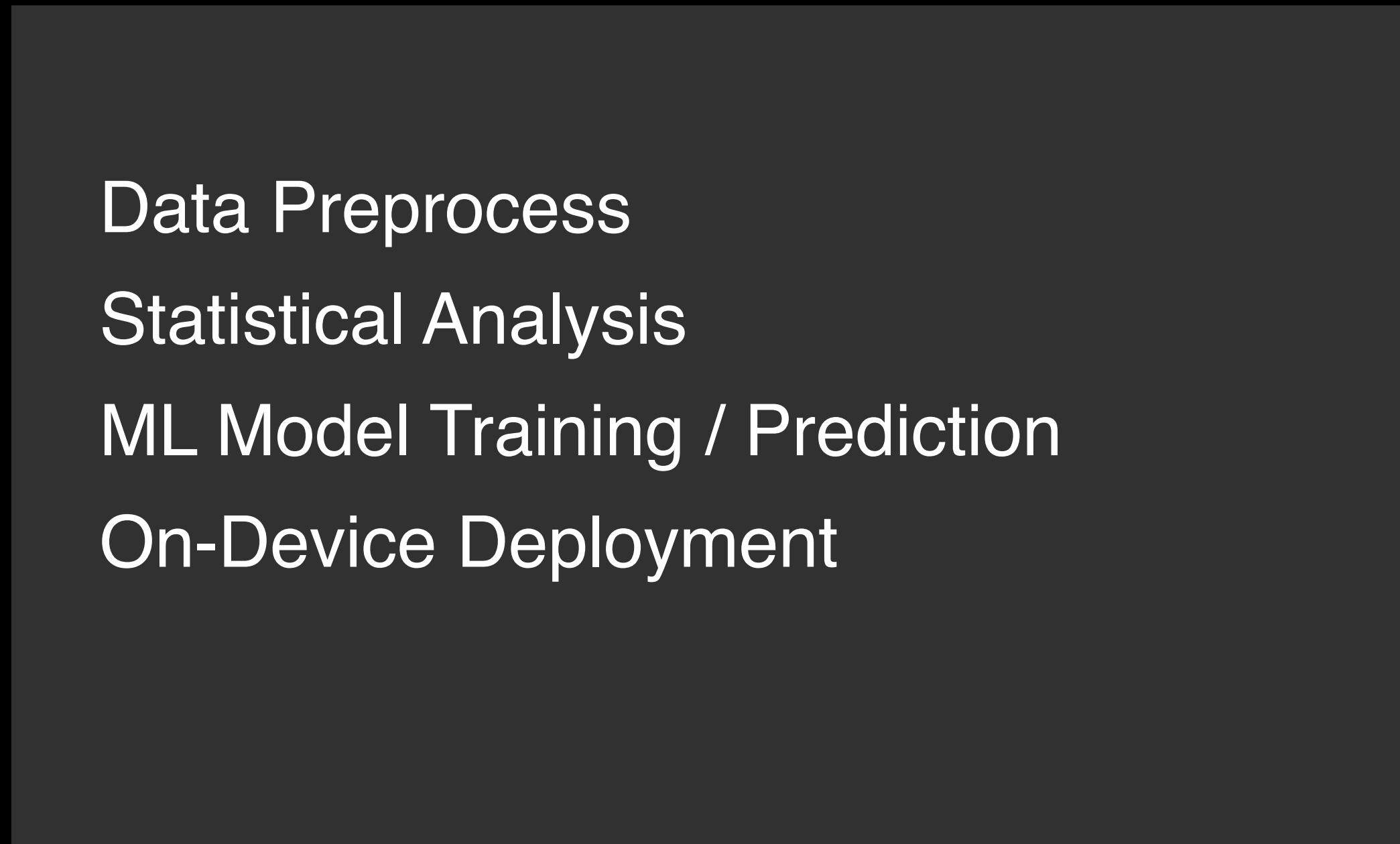
- AI algorithm detects
- Material simulation + coloring/3D model rendering

## 2. DataCook

# Front-end Machine Learning Library

# What is DataCook

Machine Learning and Data Science  
Library for Javascript & Typescript



# Framework of DataCook

## Model

Linear regression / logistic regression / decision tree / PCA / K-Means / factor analysis...

## Preprocess

Data Transformation / encoder / regularization ....

## Metrics

AUC / ROC / precision / recall / accuracy...

## Stat

Random distribution / ANOVA / t-test / correlation...

## Dataset

Data reading / shuffle / sampling...

## Math

Beta / Gamma / t ....

## Linalg

Determinant / inverse / eigen decomposition / SVD...

Basic Computation Framework tensorflow.js

# Installation and Importation

## NPM Installation

```
npm install @pipcook/datacook
```

## Browser Import

```
<script src="https://cdn.jsdelivr.net/npm/@pipcook/datacook/dist/web/index.js"></script>
```

## Import in code

```
import { Model } from '@pipcook/datacook';
const { LinearRegression } = Model;
```

# Basic Usage

```
import { Model } from '@pipcook/datacook';
const { LinearRegression } = Model;
const X = [
  [4, 5],
  [2, 3],
  [1, 4],
  [3, 8],
];
const y = [10, 5.5, 6.5, 12];
// create model
const lm = new LinearRegression();
// train linear model using feature set X and label set y
await lm.fit(X, y);
// get prediction
const yPred = await lm.predict(X);
yPred.print();
// [10, 6, 6, 12]
```

# Example: Naïve Bayes for Spam Detection

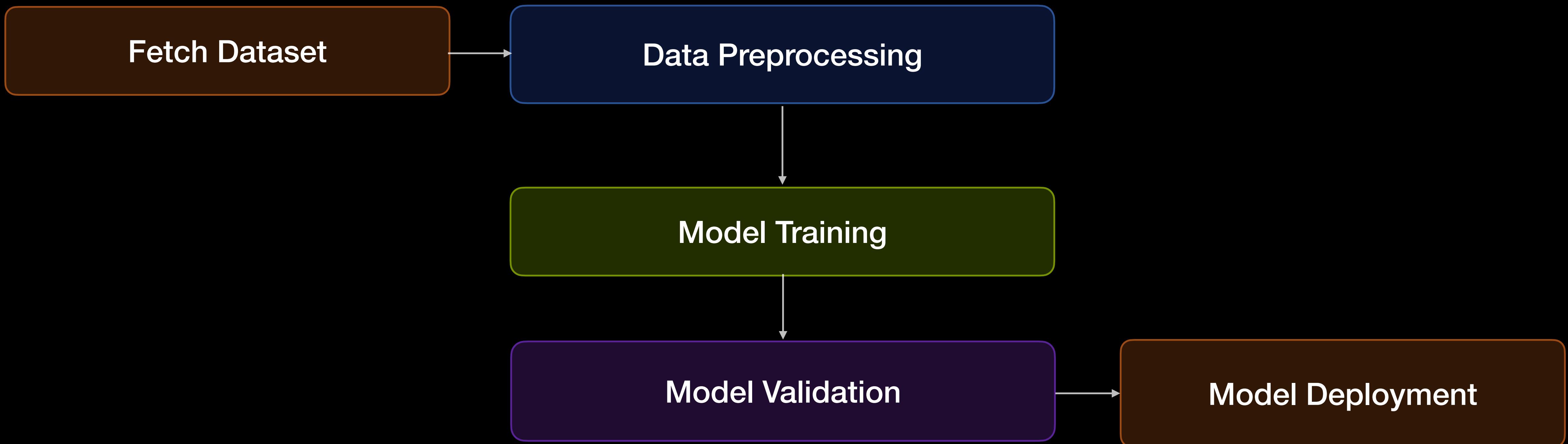
## Dataset

Spam mails dataset at [kaggle.com](https://www.kaggle.com)

#	label	text	label_num
605	ham	Subject: enron methanol ; meter # : 988291 this is a follow up to the note i gave you on monday , 4...	0
2349	ham	Subject: hpl nom for january 9 , 2001 ( see attached file : hplnol 09 . xls ) - hplnol 09 . xls	0
3624	ham	Subject: neon retreat ho ho ho , we ' re around to that most wonderful time of the year - - - neon ...	0
4685	spam	Subject: photoshop , windows , office . cheap . main trending abasements darer prudently fortuitous...	1

# Example: Naïve Bayes for Spam Detection

## Task Process



# Example: Naïve Bayes for Spam Detection

## Fetching data and preprocessing

```
import * as DataCook from '@pipcook/datacook'
import fetch from 'node-fetch';
const { CountVectorizer } = DataCook.Text;
const res = await fetch('http://imgcook.github.io/datacook/assets/dataset/spam.csv');
const text = await res.text();
const data = text.split('\n').map((d) => d.split(','));
const stopwords = await(await fetch('../dataset/stopwords.txt')).text();
const contents = data.map(d => d[1]);
const countVectorizer = new CountVectorizer(contents, stopwords.split('\n'));
const textVec = countVectorizer.transform(contents);
```

Since the input data is of text format, we first use CountVectorizer in DataCook to convert text documents to a matrix of token counts.

# Example: Naïve Bayes for Spam Detection

## Model Training

```
const { MultinomialNB } = DataCook.Model.NaiveBayes;
const mnb = new MultinomialNB();
await mnb.train(textVec, labels);
```

## Model Validation

```
const yPred = await mnb.predict(textVec);
const { accuracyScore } = DataCook.Metrics;
console.log(accuracyScore(yPred, labels)); // 0.9895
```

## Model Save

```
await fs.writeFile('./model.json', mnb.toJson(), null);
await fs.writeFile('./vectorizer.json', countVectorizer.toJson(), null)
```

# Example: Naïve Bayes for Spam Detection

## Model Deployment

### Model Load in Browser

```
fetch('./model.json').then((res) => res.text())
  .then((modelJson) => {
    const { MultinomialNB } = datacook.Model.NaiveBayes;
    const mnb = new MultinomialNB();
    mnb.load(modelJson);
    this.model = mnb;
  });
fetch('./vectorizer.json').then((res) => res.text())
  .then((vectorizerJson) => {
    const { CountVectorizer } = datacook.Text;
    const vectorizer = new CountVectorizer();
    vectorizer.load(vectorizerJson);
    this.vectorizer = vectorizer;
  });
```

### Online Prediction

```
if (this.model && this.vectorizer) {
  const textVector =
    this.vectorizer.transform(this.messageList);
  const predY = await this.model.predict(textVector);
  const predYArray = predY.arraySync();
  this.labels = predYArray;
}
```

# Example: Naïve Bayes for Spam Detection

## Online Demo

<https://imgcook.github.io/datacook/examples/text-classification/index.html>

The screenshot shows a web-based email interface with a sidebar on the left and a main message list on the right.

**Mailboxes:**

- Inbox (499)
- Spam Collection Using Naïve Bayes
- Sent
- Drafts
- Trash
- Archive

**Tags:**

- Work
- Outdoors
- Personal

**Messages / Inbox**

The main area displays several messages, each preceded by a small preview icon:

- "Go until jurong point"
- Ok lar... Joking wif u oni...
- Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
- U dun say so early hor... U c already then say...
- "Nah I don't think he goes to usf"
- "FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send
- Even my brother is not like to speak with me. They treat me like aids patient.
- As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press \*9 to copy your friends Callertune

<https://github.com/imgcook/datacook>

# 3. PipCook: JavaScript CI/CD Pipelines

**Node.js**

**Server-End Developer**

**Front-End Developer**

Node.js

Server-End Developer

Front-End Developer

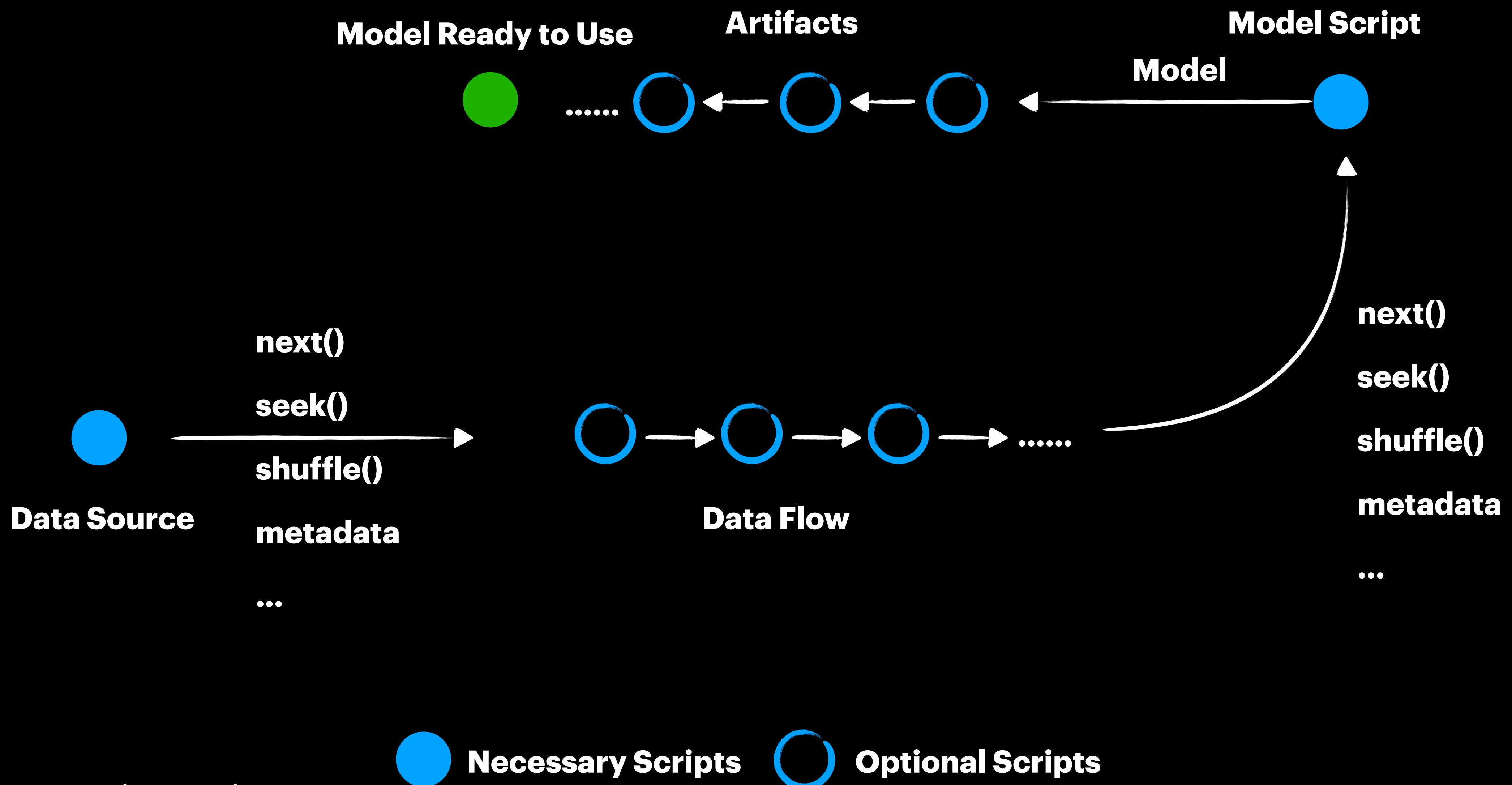
Machine Learning Developer

PipCook



# Machine Learning CI/CD in JavaScript – PipCook

## Pipeline



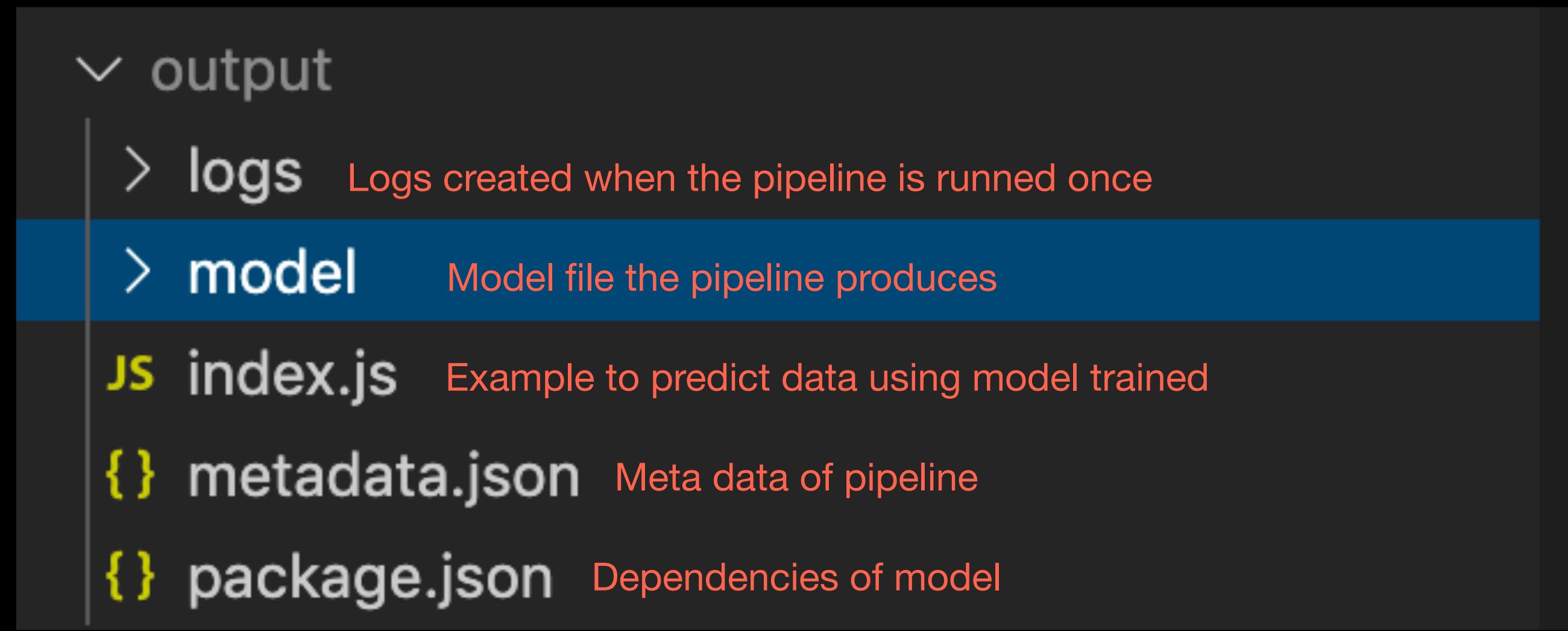
# Machine Learning CI/CD in JavaScript – PipCook

## Pipeline Configuration

```
● ● ●

{
  "specVersion": "2.0",
  "datasource": "https://cdn.jsdelivr.net/gh/imgcook/pipcook-script@c6d5ff7/scripts/image-
classification-mobilenet/build/datasource.js?url=http://pc-github.oss-us-west-
1.aliyuncs.com/dataset/imageclass-test.zip",
  "dataflow": [
    "https://cdn.jsdelivr.net/gh/imgcook/pipcook-script@c6d5ff7/scripts/image-classification-
mobilenet/build/dataflow.js?size=224&size=224"
  ],
  "model": "https://cdn.jsdelivr.net/gh/imgcook/pipcook-script@c6d5ff7/scripts/image-classification-
mobilenet/build/model.js",
  "artifact": [
    {
      "processor": "pipcook-artifact-zip@0.0.2",
      "target": "/tmp/mobilenet-model.zip"
    }
  ],
  "options": {
    "framework": "mobilenet@1.0.0",
    "train": {
      "epochs": 1,
      "validationRequired": true
    }
  }
}
```

# Output of Pipeline



## How to predict the data using the model trained

```
$ pipcook predict ./output/image-classification-mobilenet.json -s ./output/data/validation/blurBackground/71197_223__30.7_36.jpg
```

# Boa.js – Bridge Library Between Python And JavaScript

What if JavaScript Developer would like to use Python Libraries of Machine Learning

<https://github.com/imgcook/boa>

Python

```
● ● ●

import tensorflow as tf

mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test, verbose=2)
```

JavaScript

```
● ● ●

const boa = require('boa');
const tf = boa.import('tensorflow');

const mnist = tf.keras.datasets.mnist
const [[x_train, y_train], [x_test, y_test]] = mnist.load_data()
const [x_train, x_test] = [x_train / 255.0, x_test / 255.0]

const model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten([28, 28]),
    tf.keras.layers.Dense(128),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10)
])
model.compile('adam',
              'sparse_categorical_crossentropy',
              ['accuracy'])
model.fit(x_train, y_train, 5)
model.evaluate(x_test, y_test, 2)
```

# Thanks



<http://imgcook.com/>



<https://github.com/imgcook/datacook>

[imgcook.github.io/datacook](http://imgcook.github.io/datacook)



<https://alibaba.github.io/pipcook/>

<https://github.com/alibaba/pipcook>