

iPlayground

Swift 也能訓練 Machine Learning 模型？

Create ML 實戰

Machine Learning in Swift

Precondition

Create ML

- Xcode 10
- macOS Mojave
- >= iOS 11

Agenda

Create ML

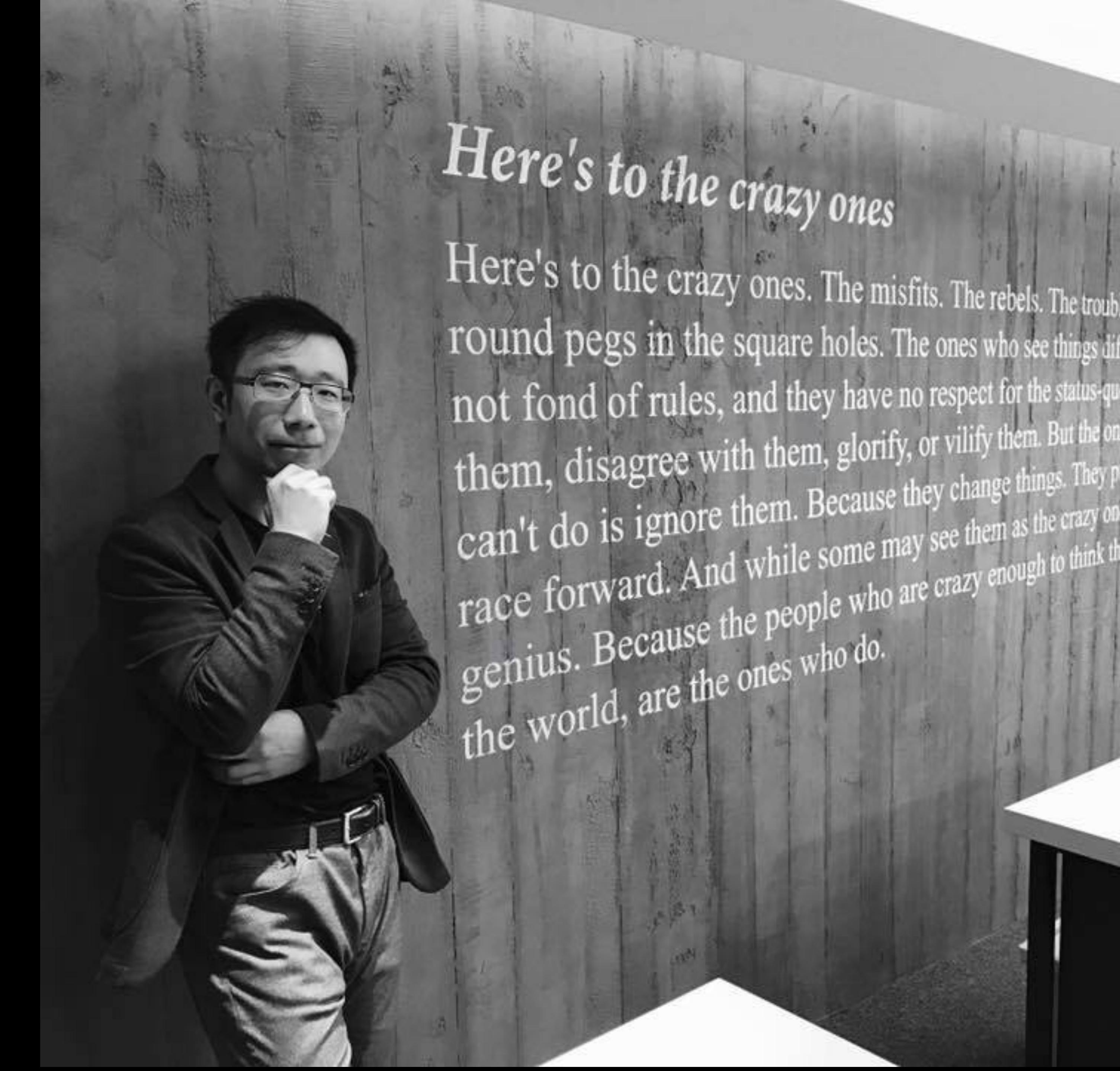
- Machine Learning Training Process
- Create ML Overview
- Image Classification
- Natural Language Processing
- Tabular Data
- Model Accuracy

Introduction

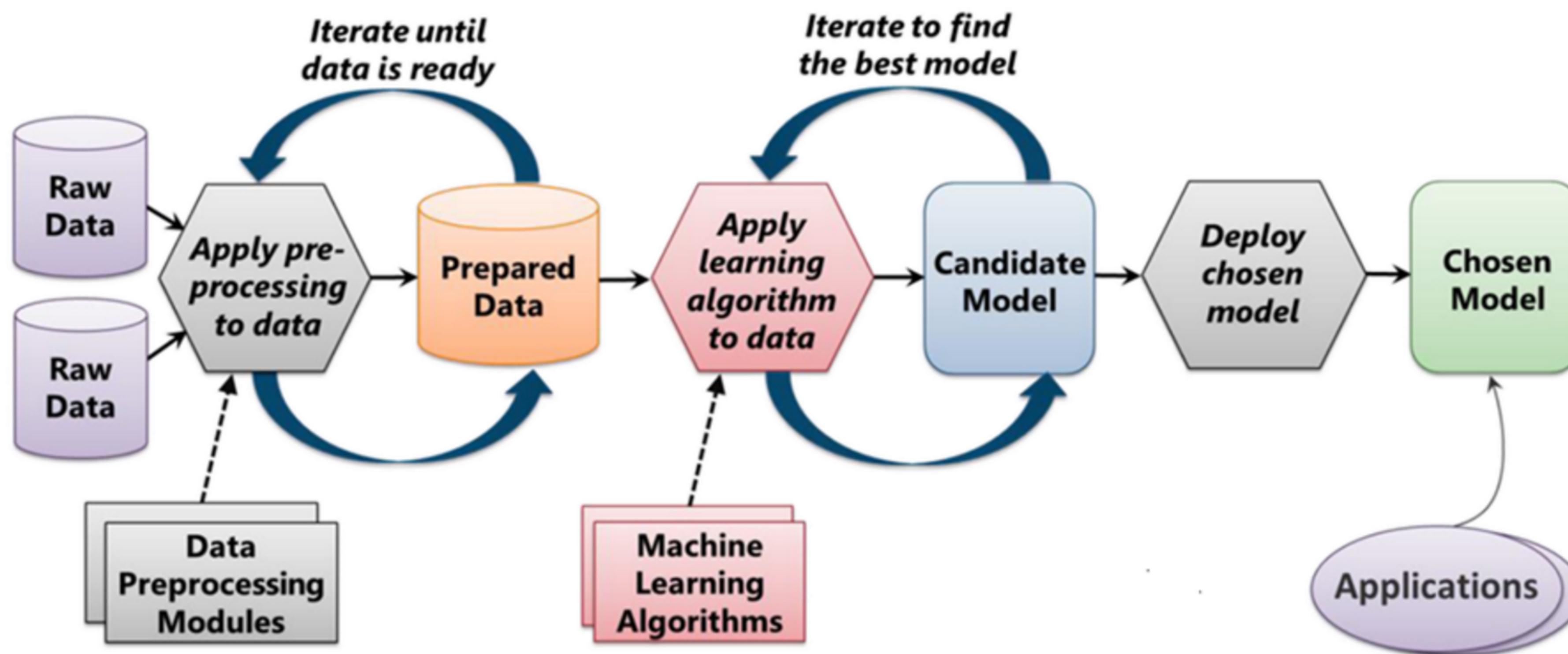
講者簡介

張 景隆

- 台大 HackNTU - iOS 課程講師
- GDG DevFest Taipei 2016 講者
- 東海大學 教學卓越計畫 講師
- 教育部開放式課程-程式開發講師
- MOPCON 2014 講者
- CodeData 作者
- iOS Dev Club 講者 & 核心成員
- 參加蘋果官方主辦 2013年 Tech Talk



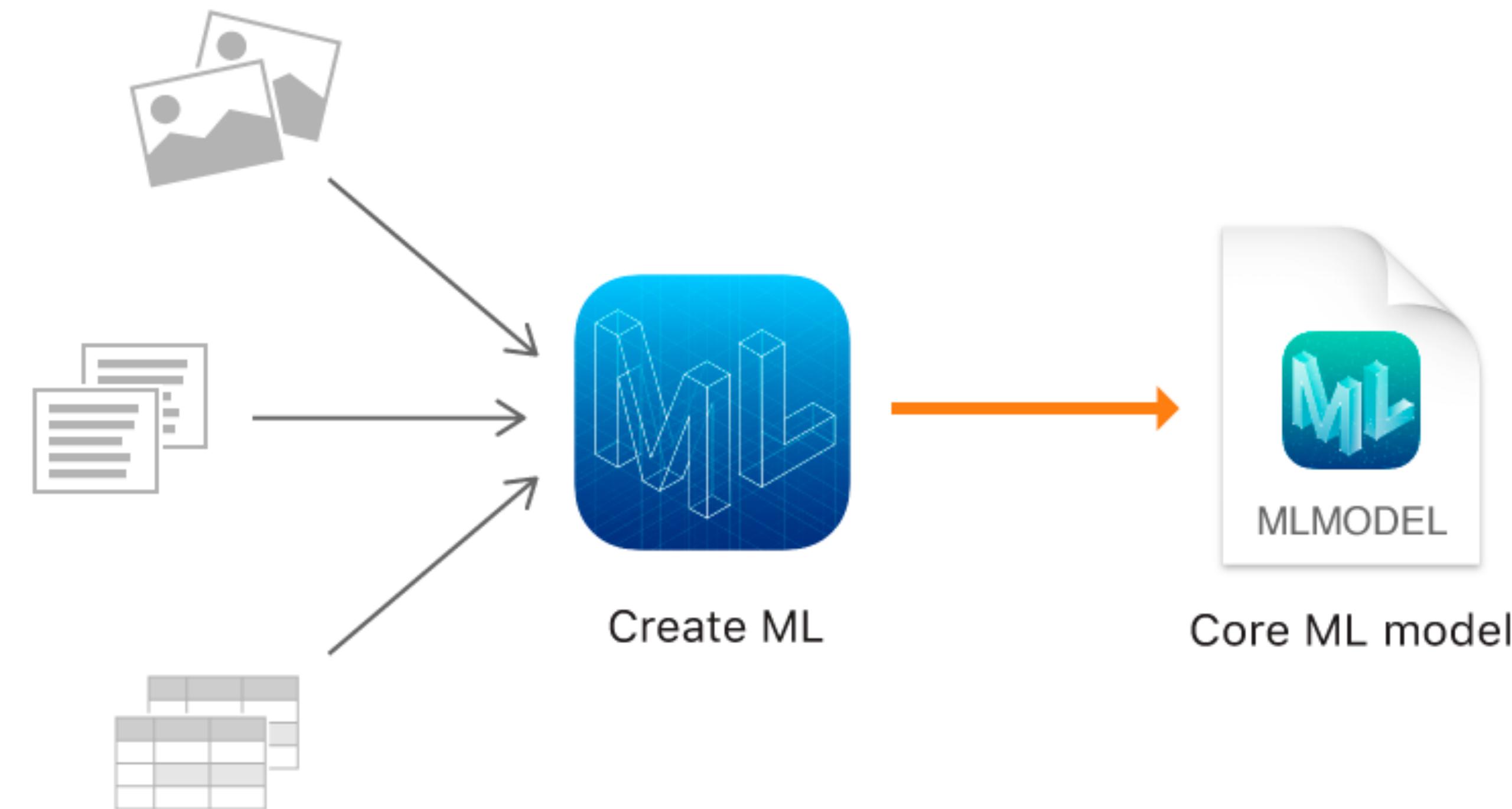
The Machine Learning Process



From "Introduction to Microsoft Azure" by David Chappell

Overview

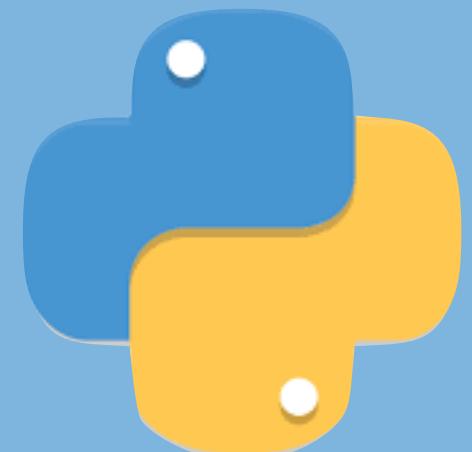
Create ML



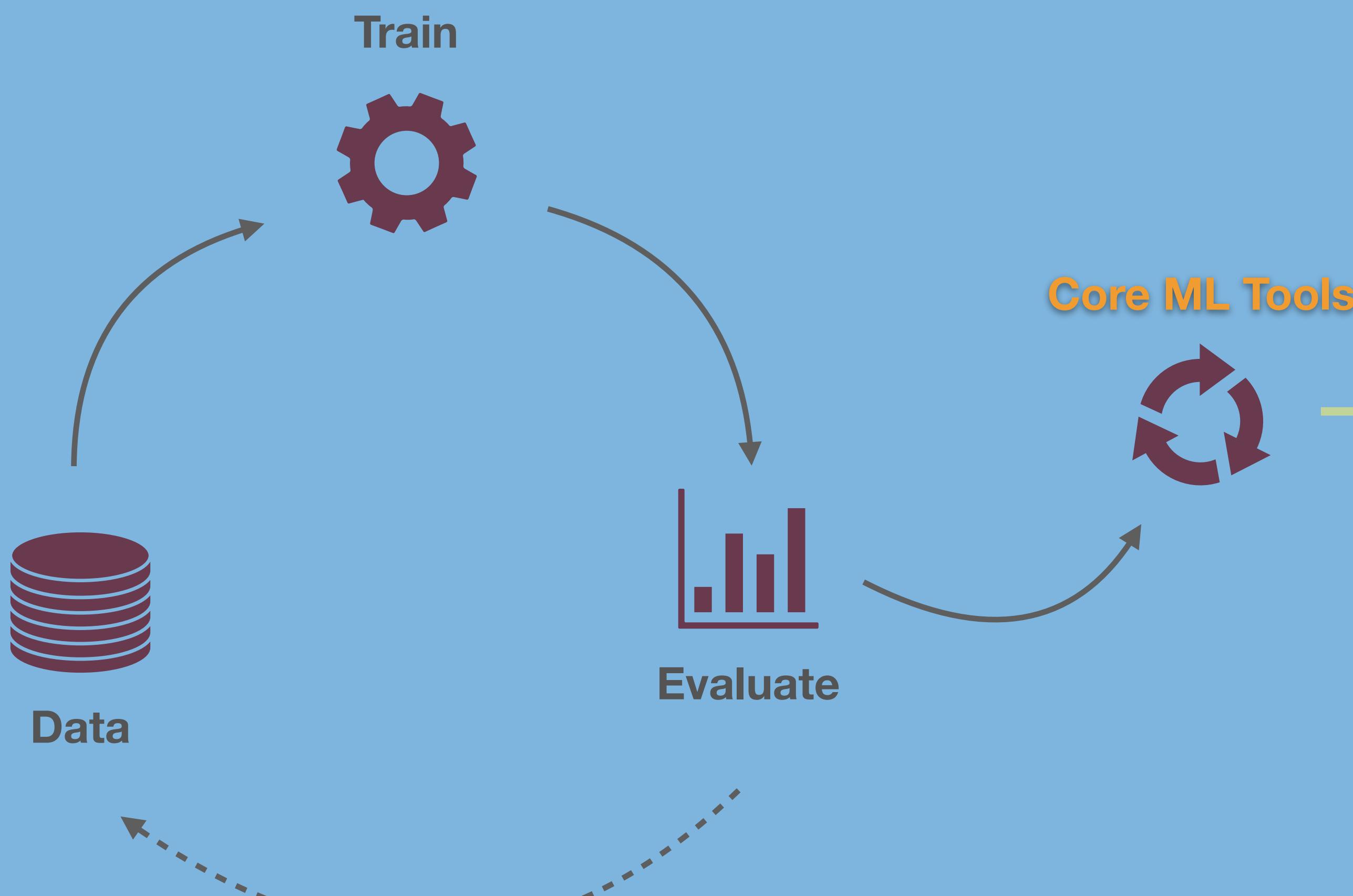
Create ML Workflow

Create ML

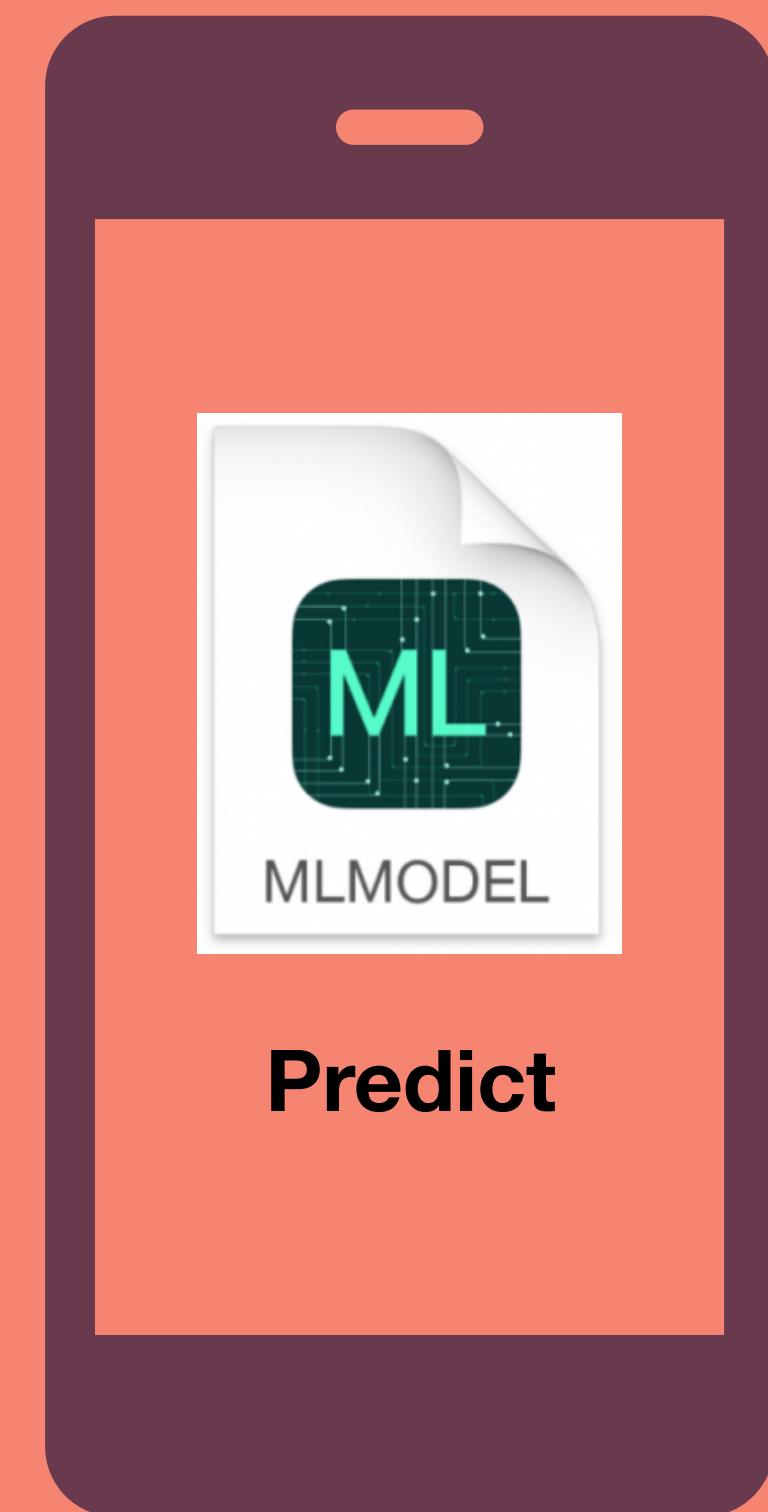




Python



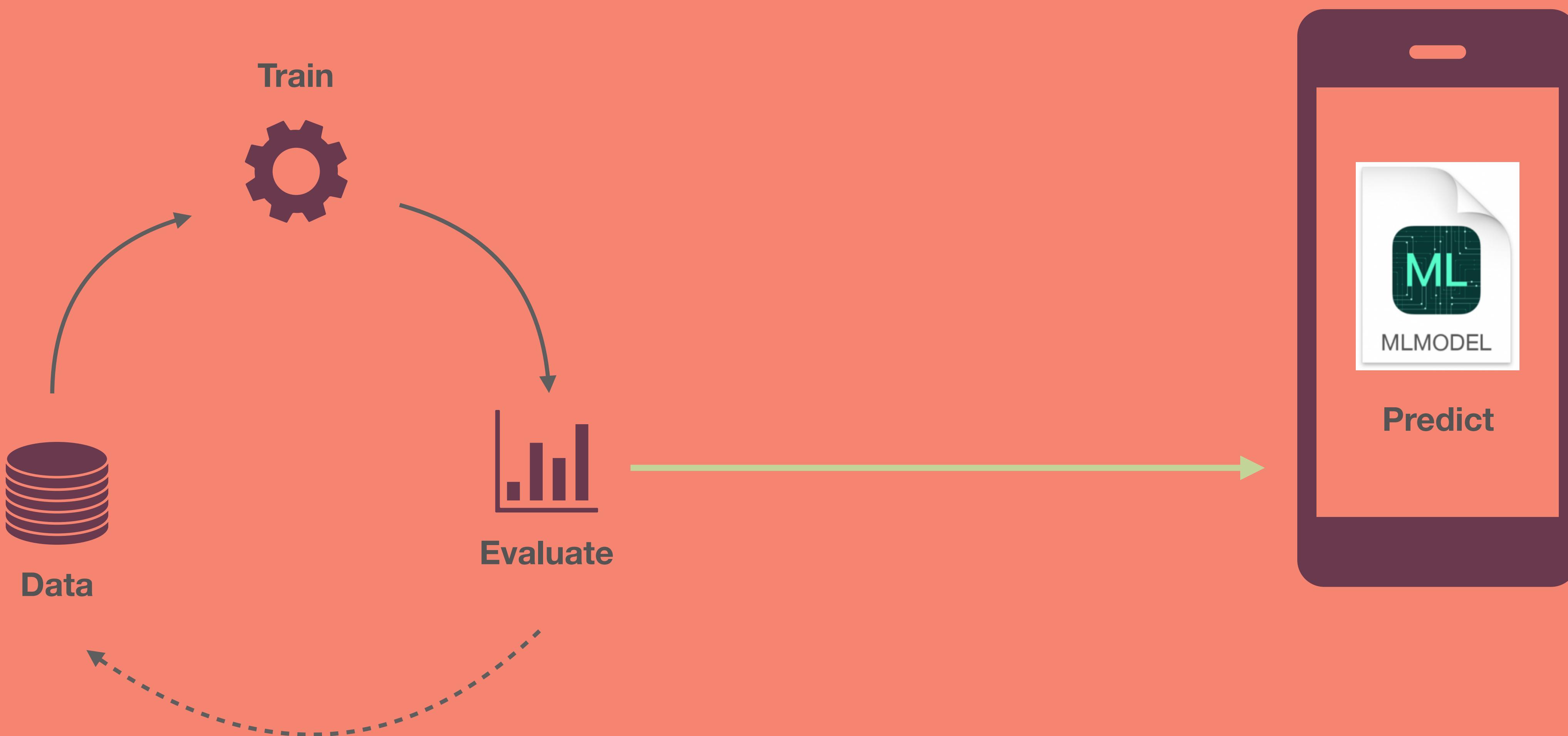
Swift



Predict

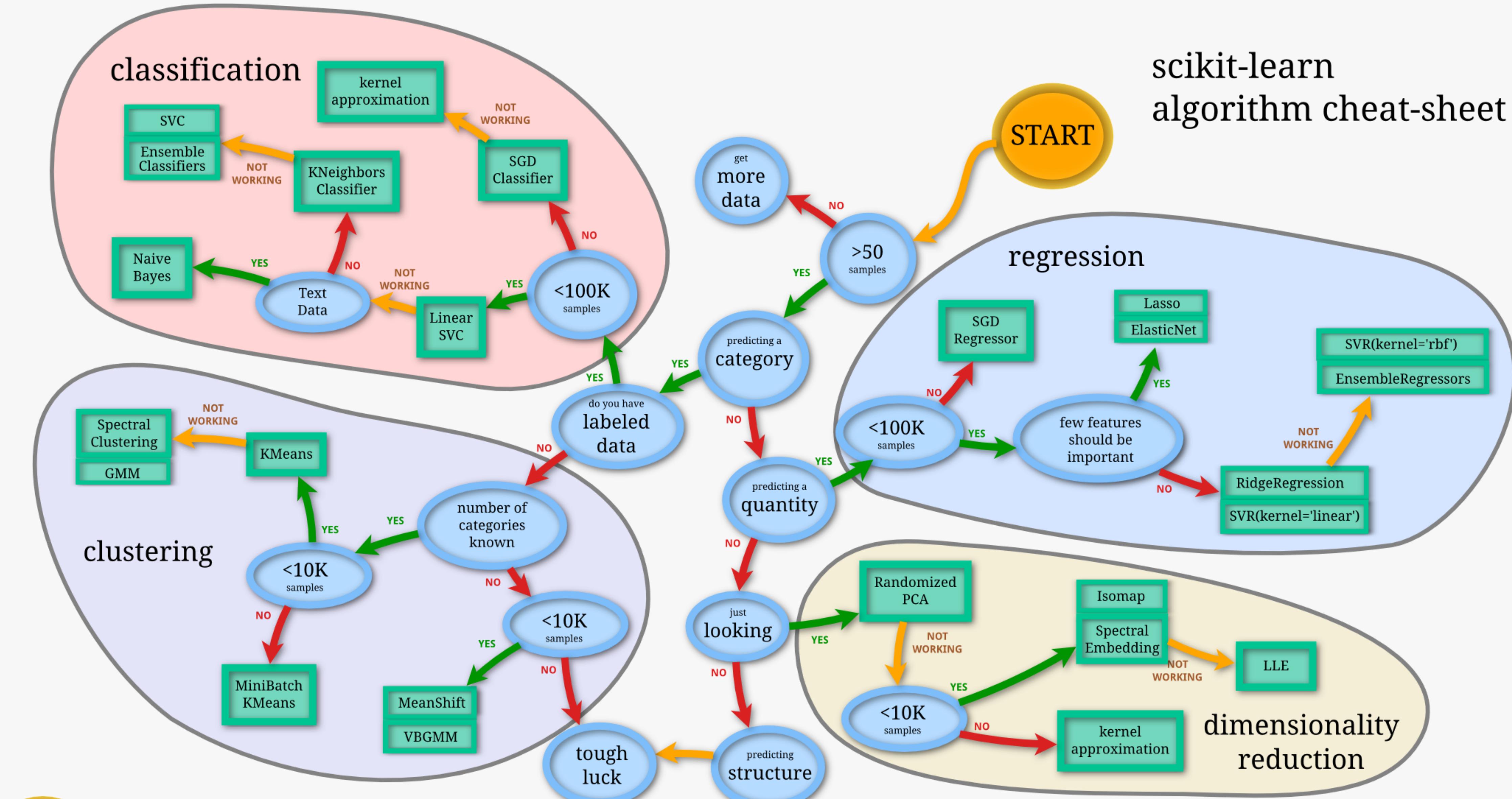


Swift



scikit-learn

algorithm cheat-sheet



Back

scikit
learn



Image Classifier

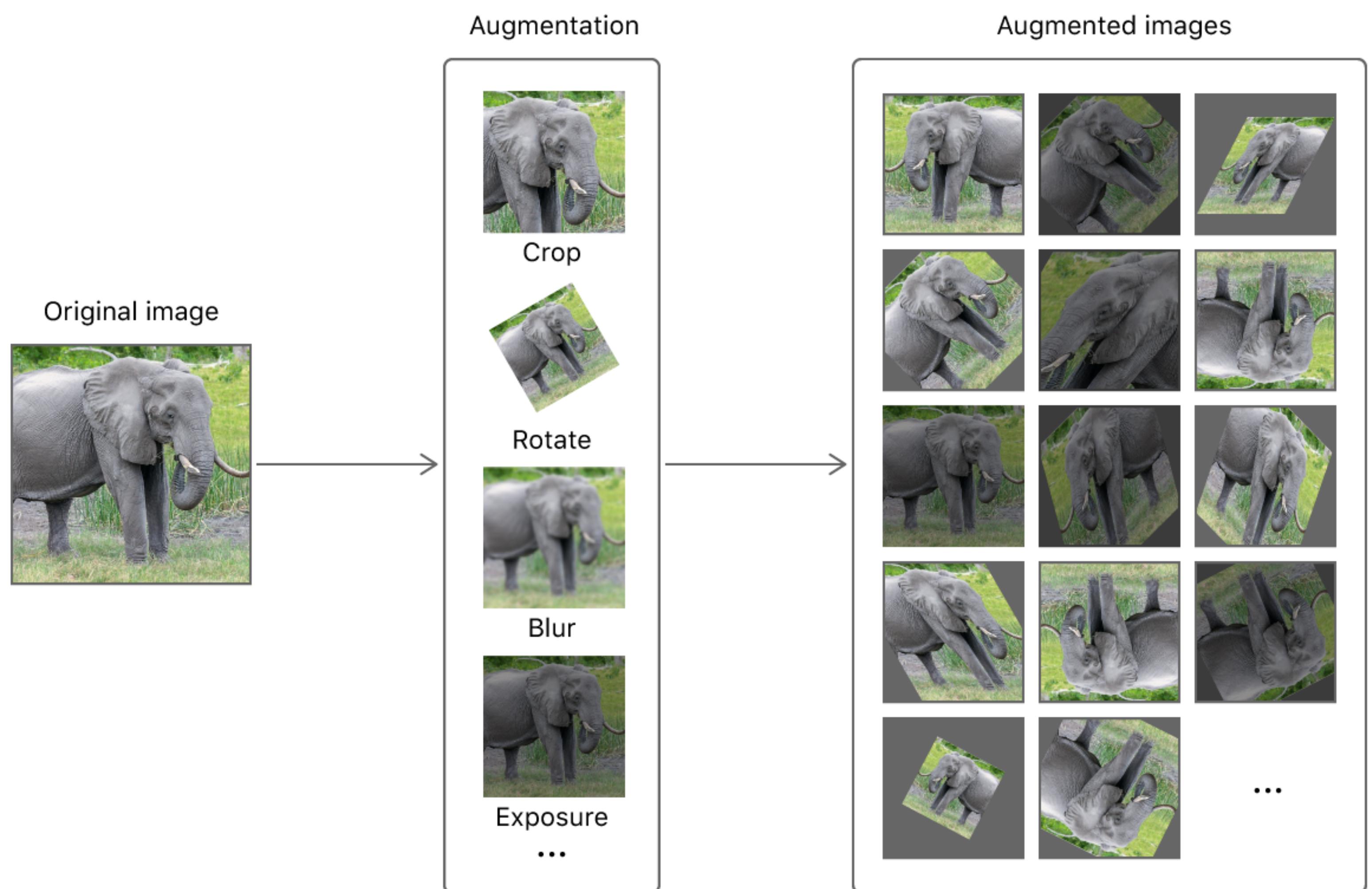
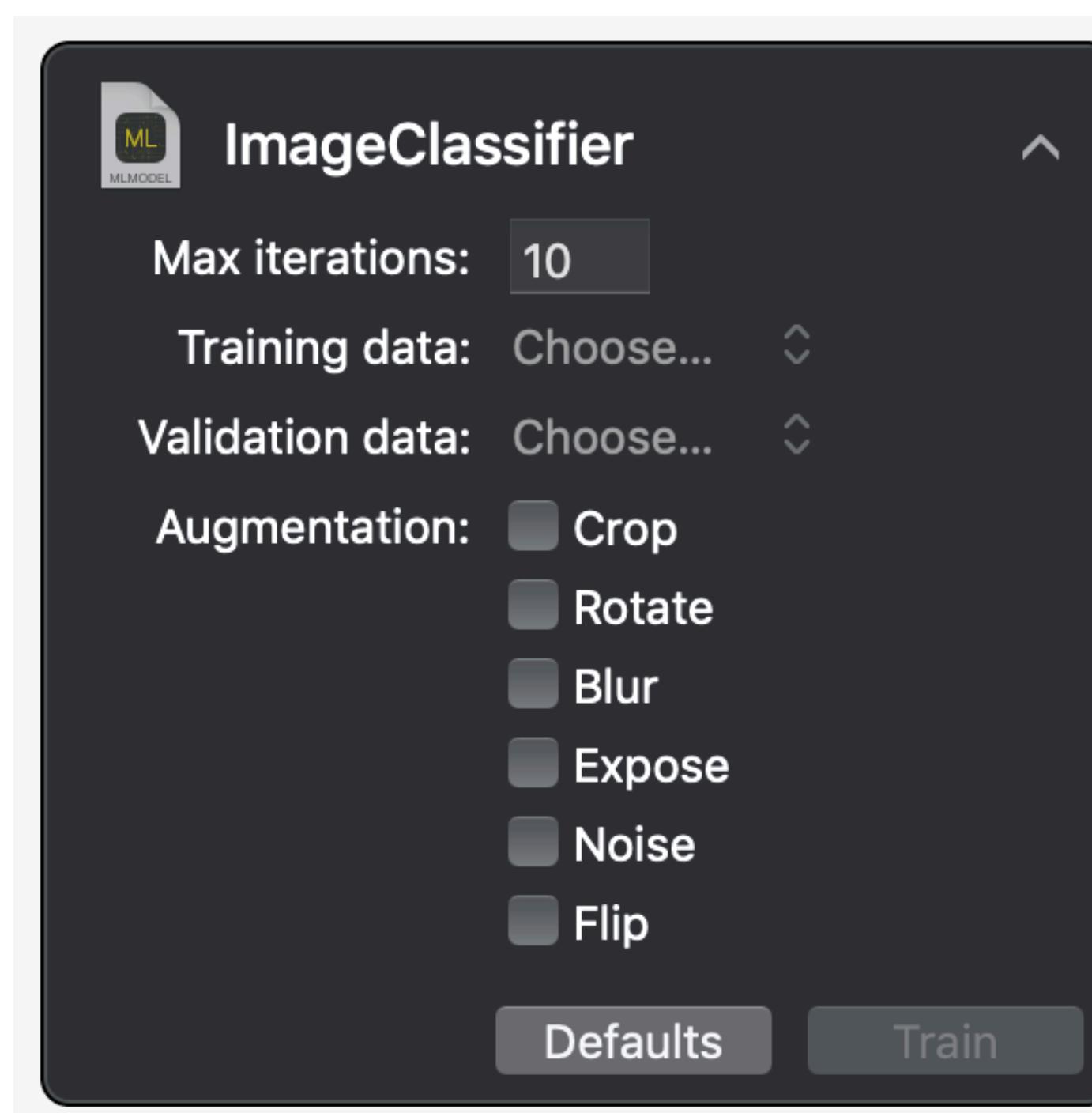
Magic of MLUI

Create ML

```
import CreateMLUI  
  
let builder = MLImageClassifierBuilder()  
builder.showInLiveView()
```

Image Classifier

Create ML





Demo

Hotdog Classifier

Image Classifier Builder in a Playground

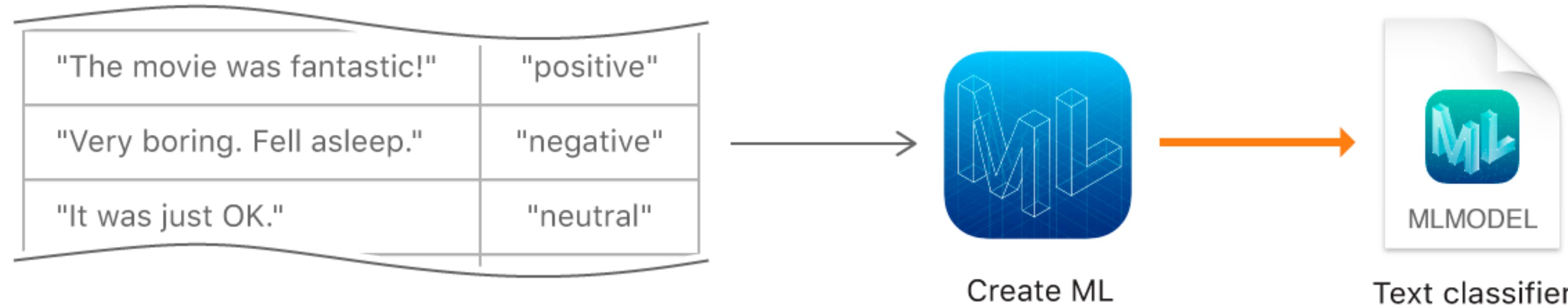
Classifying Images with Vision and Core ML



Natural Language Processing

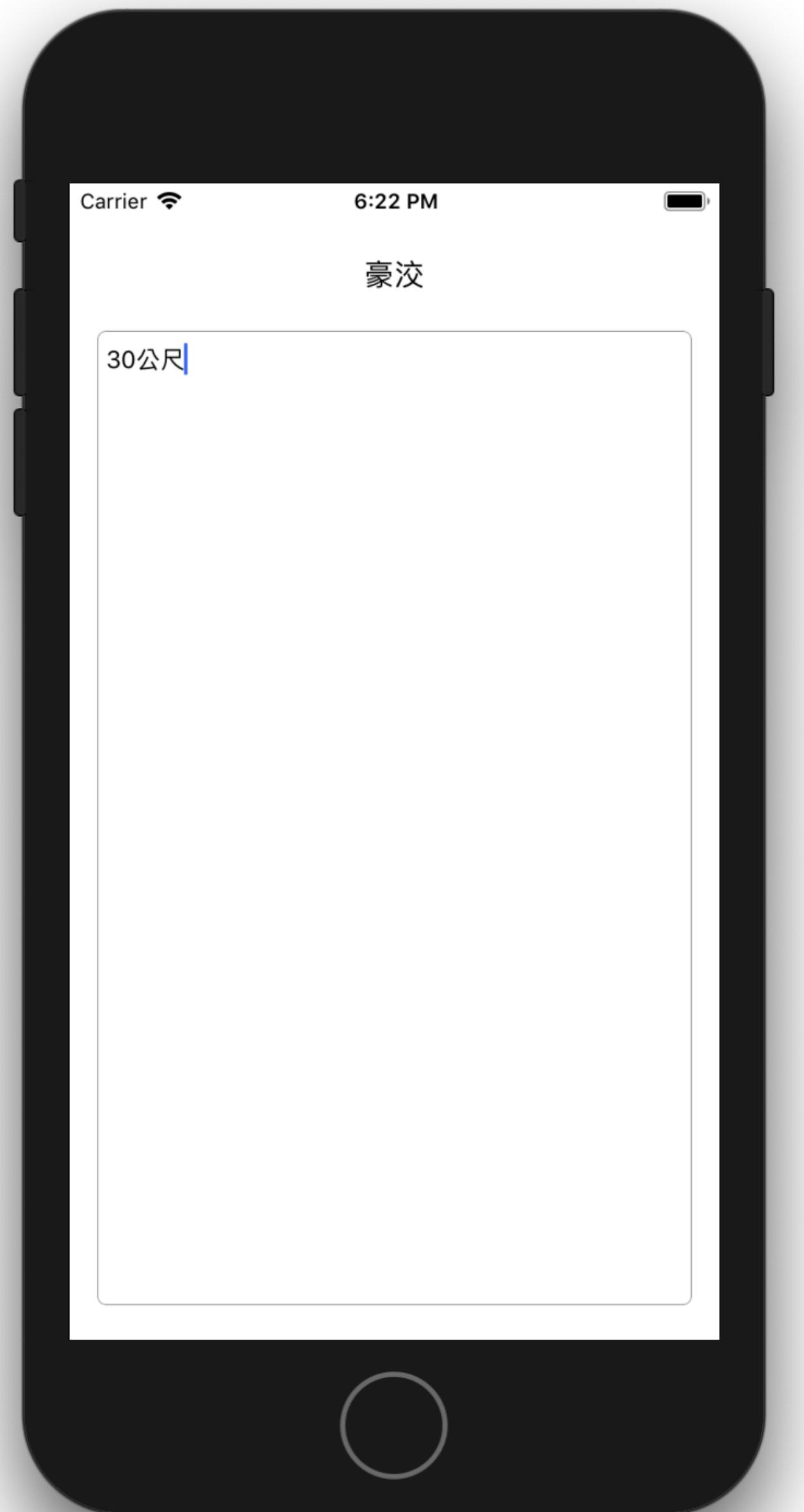
Text Classifier Overview

Create ML



PTT Detector

Create ML



PTT Detector

Create ML

```
[  
  {  
    "label": "豪洨",  
    "text": "在某個交友app，有個被我西裝照吸引的人妻，聊了兩天雙方都很有感，馬上就在遇  
  },  
  {  
    "label": "八卦",  
    "text": "1. 媒體來源:※ 例如蘋果日報、奇摩新聞自由2. 完整新聞標題:※ 標題沒有寫出來  
  },  
  {  
    "label": "八卦",  
    "text": "最近在加油站看到一台KIA的小車車顏色很鮮豔 長ㄉ像機器人發現好像很少看到這'  
  },  
  {  
    "label": "八卦",  
    "text": "陳其邁晚會爆場 空拍照驚人！網友驚呆：誰說高雄五五波？民進黨高雄市長候選人  
  },  
  {  
    "label": "八卦",  
    "text": "原本這是一篇新聞的文基市打造國門廣場 擬拆舊站圓環、陸橋https://www.ptt  
  },
```



Demo

PTT Detector

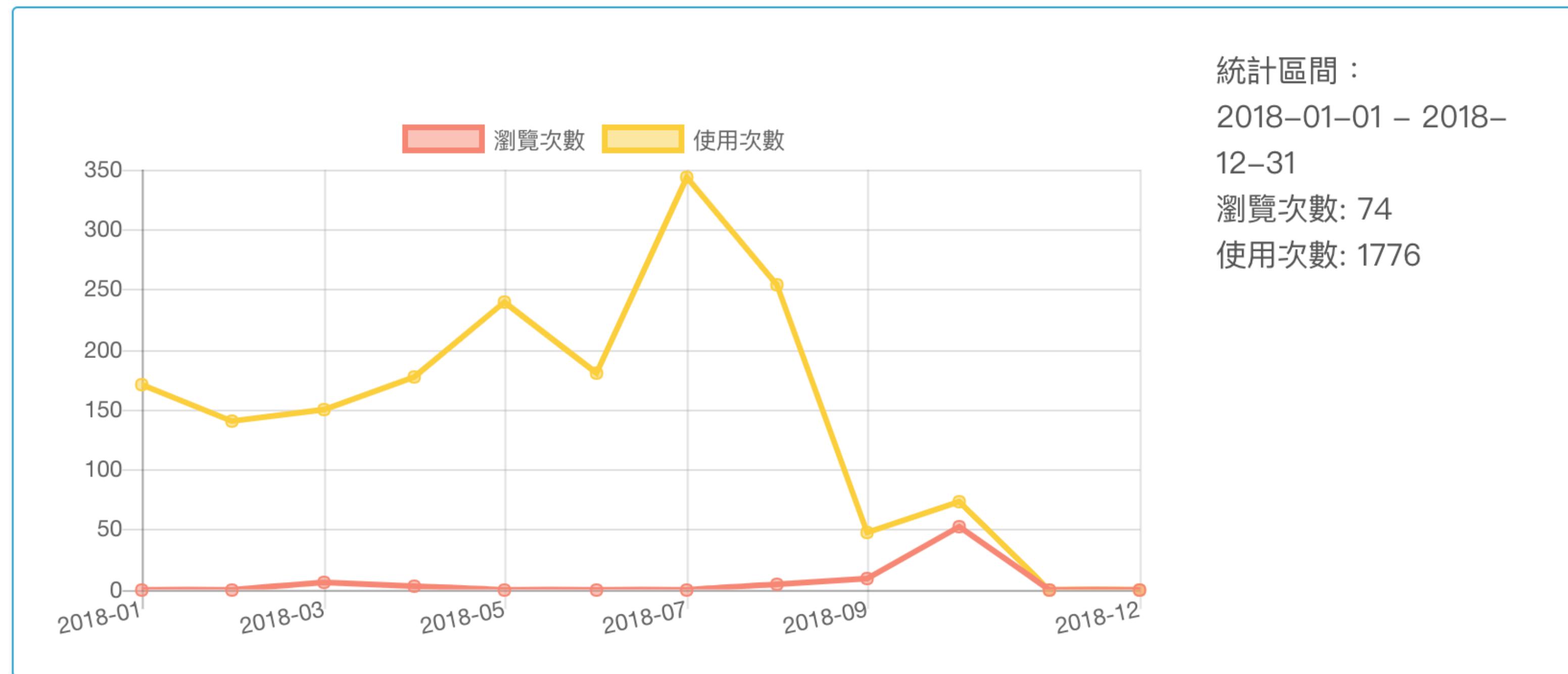


Tabular Data

臺北捷運各站分時進出量統計

Create ML

使用統計



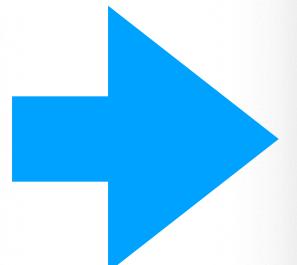
Data Preprocessing

Create ML

Replace \t or \s to ,

MRT_201807.csv

| 日期 | 時段 | 進站 | 出站 | 人次 |
|------------|----|------|------|----|
| 2018-07-01 | 00 | 松山機場 | 松山機場 | 0 |
| 2018-07-01 | 00 | 松山機場 | 中山國中 | 0 |
| 2018-07-01 | 00 | 松山機場 | 南京復興 | 0 |
| 2018-07-01 | 00 | 松山機場 | 忠孝復興 | 0 |
| 2018-07-01 | 00 | 松山機場 | 大安 | 1 |
| 2018-07-01 | 00 | 松山機場 | 科技大樓 | 0 |
| 2018-07-01 | 00 | 松山機場 | 六張犁 | 1 |
| 2018-07-01 | 00 | 松山機場 | 麟光 | 0 |
| 2018-07-01 | 00 | 松山機場 | 辛亥 | 0 |
| 2018-07-01 | 00 | 松山機場 | 萬芳醫院 | 0 |
| 2018-07-01 | 00 | 松山機場 | 萬芳社區 | 1 |



Date,Hour,Inbound,Outbound,Passenger

| | | | |
|---------------|------|------|---|
| 2018-07-01,00 | 松山機場 | 松山機場 | 0 |
| 2018-07-01,00 | 松山機場 | 中山國中 | 0 |
| 2018-07-01,00 | 松山機場 | 南京復興 | 0 |
| 2018-07-01,00 | 松山機場 | 忠孝復興 | 0 |
| 2018-07-01,00 | 松山機場 | 大安 | 1 |
| 2018-07-01,00 | 松山機場 | 科技大樓 | 0 |
| 2018-07-01,00 | 松山機場 | 六張犁 | 1 |
| 2018-07-01,00 | 松山機場 | 麟光 | 0 |
| 2018-07-01,00 | 松山機場 | 辛亥 | 0 |
| 2018-07-01,00 | 松山機場 | 萬芳醫院 | 0 |
| 2018-07-01,00 | 松山機場 | 萬芳社區 | 1 |

Well format CSV



Demo

MRT Passenger Traffic



Predictions and Evaluation

Predictions and Evaluation

Create ML

- **Confusion Matrix**
- **Classification Report**

Confusion Matrix

Create ML

```
[ [ 1   2 ]  
[ 0   4 ] ]
```

Confusion Matrix

Create ML

| | | Actual Value (as confirmed by experiment) | |
|--|-----------|--|--------------------------------|
| | | positives | negatives |
| Predicted Value (predicted by the test) | positives | TP True Positive | FP False Positive |
| | negatives | FN False Negative | TN True Negative |

Classification Report

Create ML

| | precision | recall | f1-score | support |
|--------------------|------------------|---------------|-----------------|----------------|
| 0 | 1.00 | 0.33 | 0.50 | 3 |
| 1 | 0.67 | 1.00 | 0.80 | 4 |
| avg / total | 0.81 | 0.71 | 0.67 | 7 |

某池塘有1400條魚，300隻蝦，300隻蟹。

現在以**捕魚**為目的。

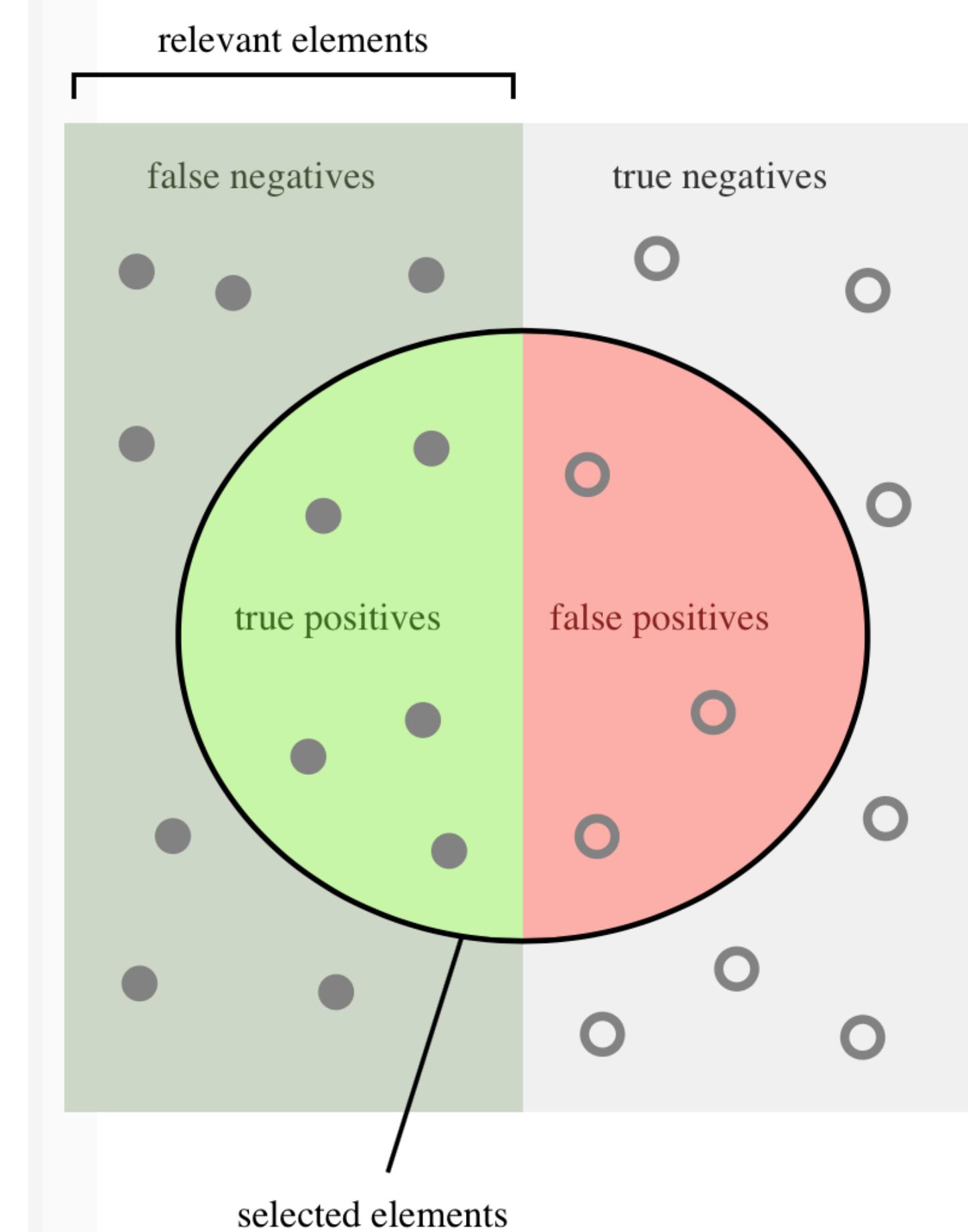
撒一大網，逮著了700條魚，200隻蝦，100隻蟹。

$$\text{準確率} = 700 / (700 + 200 + 100) = 70\%$$

$$\text{召回率} = 700 / 1400 = 50\%$$

$$F\text{值} = 2 * 70\% * 50\% / (70\% + 50\%) = 58.3\%$$

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Recap

Create ML

- New Machine Learning Model Training Flow
- Image Classification
- Natural Language Processing
- Tabular Data
- Model Accuracy

CONTACT ME

Information



張景隆 (Louis Chang)

 @appletone.tw



CoreMLtools

CoreMLtools 2.0

Create ML

```
pip install -U coremltools
```

```
import coremltools
from coremltools.models.neural_network.quantization_utils import *
model = coremltools.models.MLModel('/Users/appleton/Desktop/MRT.mlmodel')
model.predict({'Date':'2018-10-18', 'Hour':8, 'Inbound':u'台北車站', 'Outbound':u'公館'})
```

<https://apple.github.io/coremltools/>