



Using Machine Learning Models

数据科学 – 机器学习模型入门

July 2020

Microsoft Reactor | Ryan Chung

```
led by player" to  
s.load_image("kg.png")  
  
(self):  
    initialize Dog object and create Text o  
g, self).__init__(image = Dog.image  
x = games.mouse.x  
bottom = games.scr  
  
re = games.Text(value = 0, size = 24  
top = 5, right = gam  
  
reen.add(self.score)  
1 = games.Text(value = 0, size = 24  
top = 5, left = gam
```



Ryan Chung

Instructor / DevelopIntelligence
Founder / MobileDev.TW

@ryanchung403 on WeChat
Ryan@MobileDev.TW





Reactor



developer.microsoft.com/reactor/
@MSFTReactor on Twitter

DS On-line Workshop agenda 数据科学在线研讨会议程

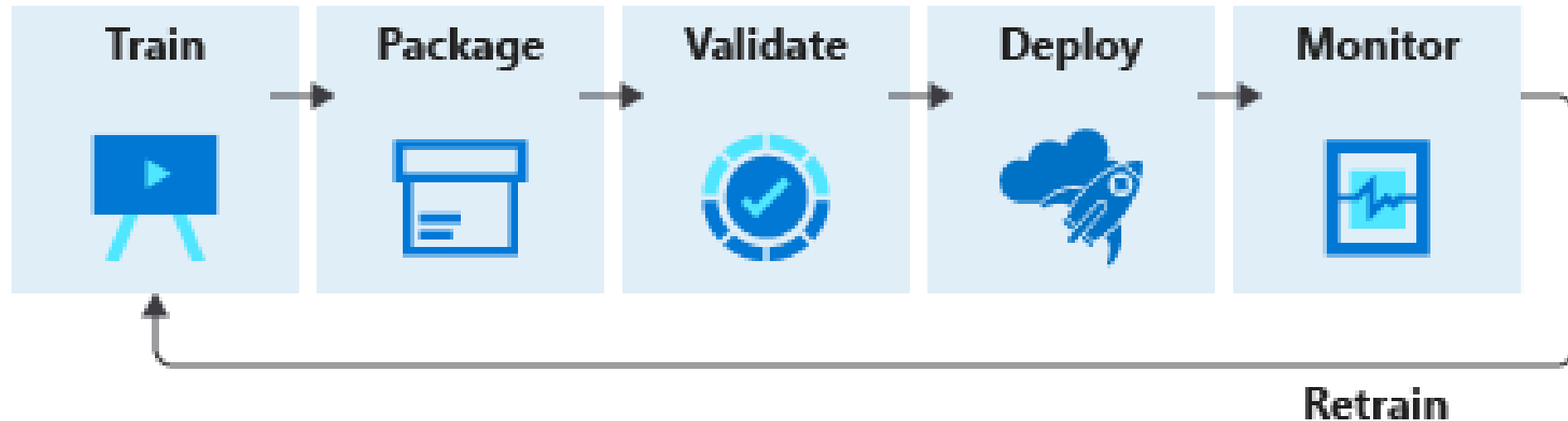
19:30	Welcome 开场
19:35	Overview of ML 机器学习介绍
20:00	How to choose Machine Learning Algorithm 如何选择算法
20:20	5 -minute lab break 中场休息 / 实作练习
20:25	The Workflow and Evaluation for ML 机器学习流程与评估指针
20:45	Intro to Azure Machine Learning Solutions 微软机器学习解决方案
21:00	Event end 研讨会结束

Azure Machine Learning 微软Azure机器学习

- 云端环境
- 可以进行模型的训练/部署/自动化/管理/追踪
- 适用于
 - 传统机器学习 / 深度学习 / 监督式学习 / 非监督式学习
- 使用弹性
 - 可自行撰写Python/R 或 使用Azure ML 图形化界面



Azure Machine Learning Model Workflow



机器学习

定义

- 计算机算法可以透过经验来自动学习(Tom Mitchell)

种类

- 监督式学习 (分类、回归)
- 非监督式学习 (分群、关联)
- 强化学习 (实时、脱机)

监督式学习

分类

- 是什么(已知标签)
- 是或不是(二元判断)

Analyze image:

輸入一個圖片網址，然後按下 分析圖片 按鈕。

Image to analyze: <https://www.petmd.com/site>

Response:

```
{
  "categories": [
    {
      "name": "动物_猫",
      "score": 0.99609375
    }
  ],
  "color": {
    "dominantColorForeground": "Black",
    "dominantColorBackground": "Brown",
    "dominantColors": [
      "Black",
      "Brown",
      "White"
    ],
    "accentColor": "BD760E",
    "isBwImg": false,
    "isBWImg": false
  },
  "description": {
    "tags": [
      "猫",
      "室内",
      "白色",
      "看着",

```

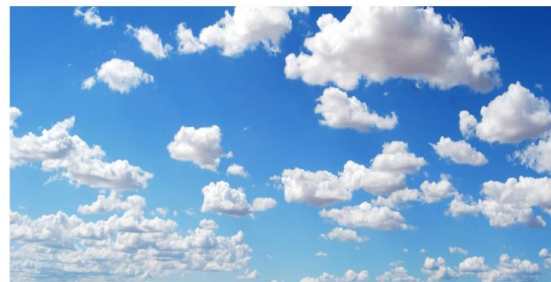
Source image:



白色的猫

修改范例出现预测结果摘要

- Predictions Array
- probability > 0.8
- tagName == "Sky"



應該是天空! (信心:1)



不是天空吧

监督式学习

Regression 数值

- 数值预测(房价、温度、销售量)

engine-size	fuel-system	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price
130	mpfi	3.47	2.68	9	111	5000	21	27	13495
130	mpfi	3.47	2.68	9	111	5000	21	27	16500
152	mpfi	2.68	3.47	9	154	5000	19	26	16500
109	mpfi	3.19	3.4	10	102	5500	24	30	13950
136	mpfi	3.19	3.4	8	115	5500	18	22	17450
136	mpfi	3.19	3.4	8.5	110	5500	19	25	15250
136	mpfi	3.19	3.4	8.5	110	5500	19	25	17710

监督式学习

常见使用案例

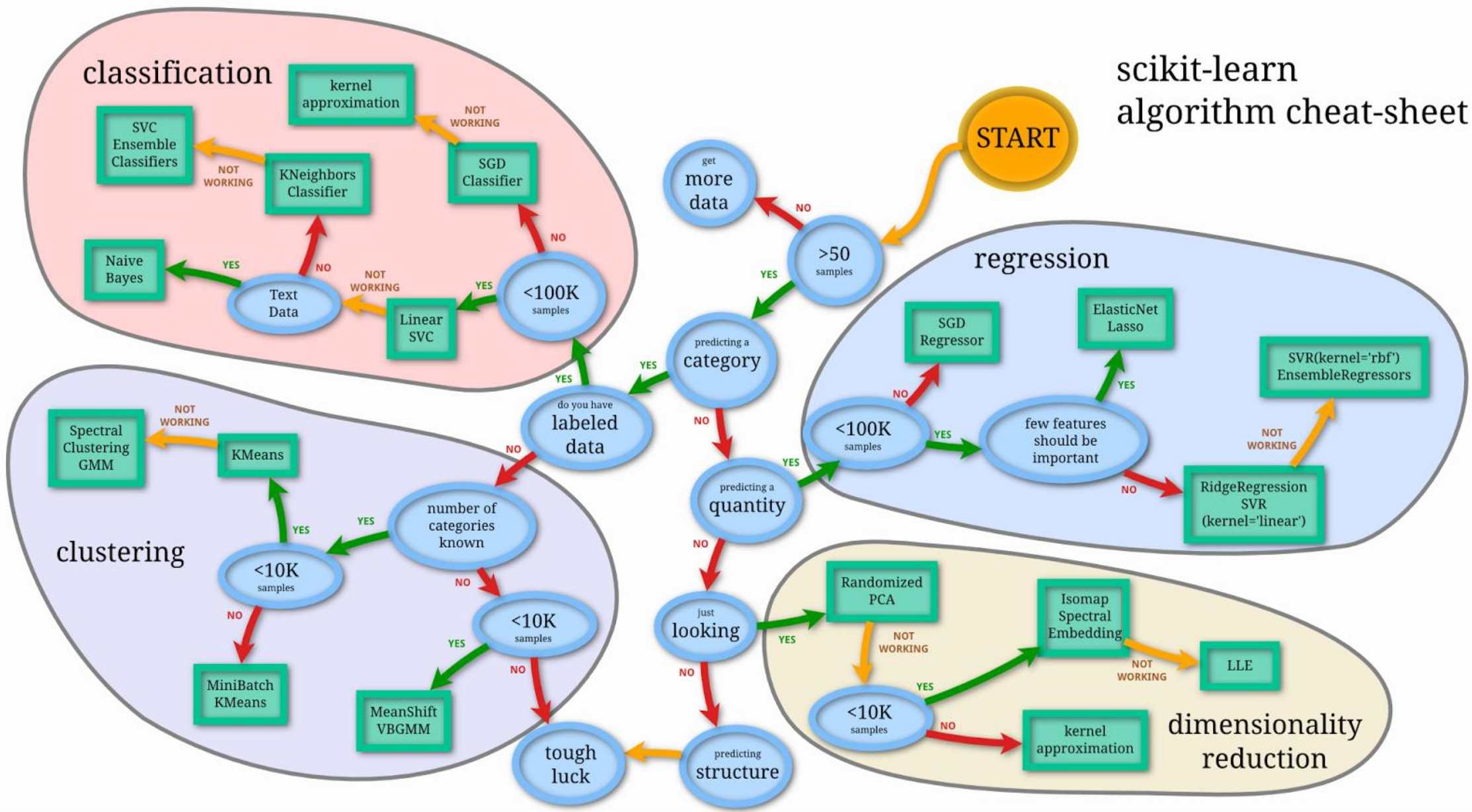
- 图片分类
- 光学文字分类(OCR)
- 脸部辨识
- 情绪分析(sentiment)
- 自然语言处理
- 机器翻译
- 字幕产生
- 事件侦测

题目分类

监督式/非监督式 VS. 数据连续/可数

数据类型 Data Type	监督式学习 Supervised	非监督式学习 Unsupervised
Discrete 离散的	分类 Classification	丛集 Clustering
Continuous 连续的	回归 Regression	降维 Dimensionality Reduction

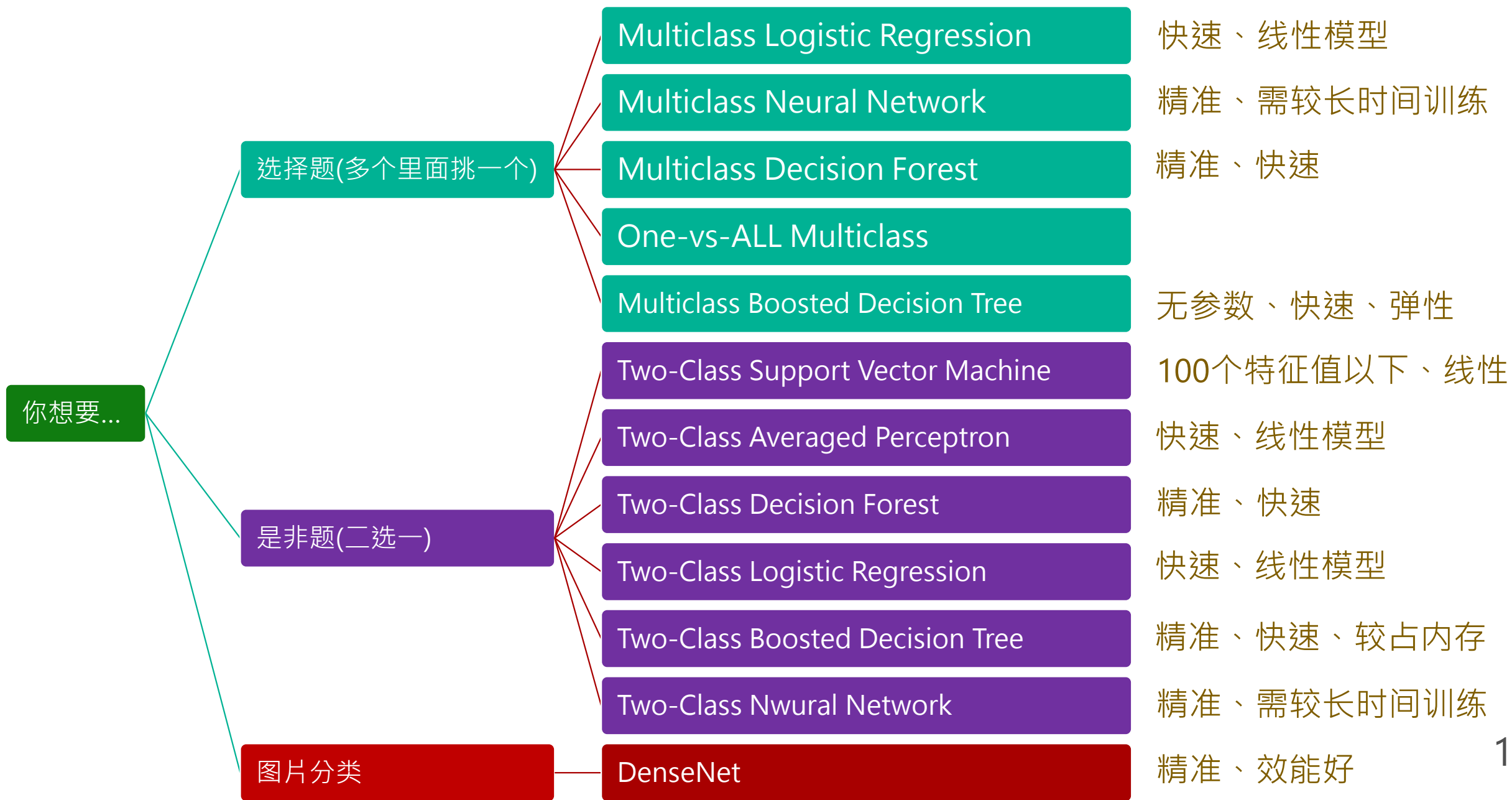
题目分类



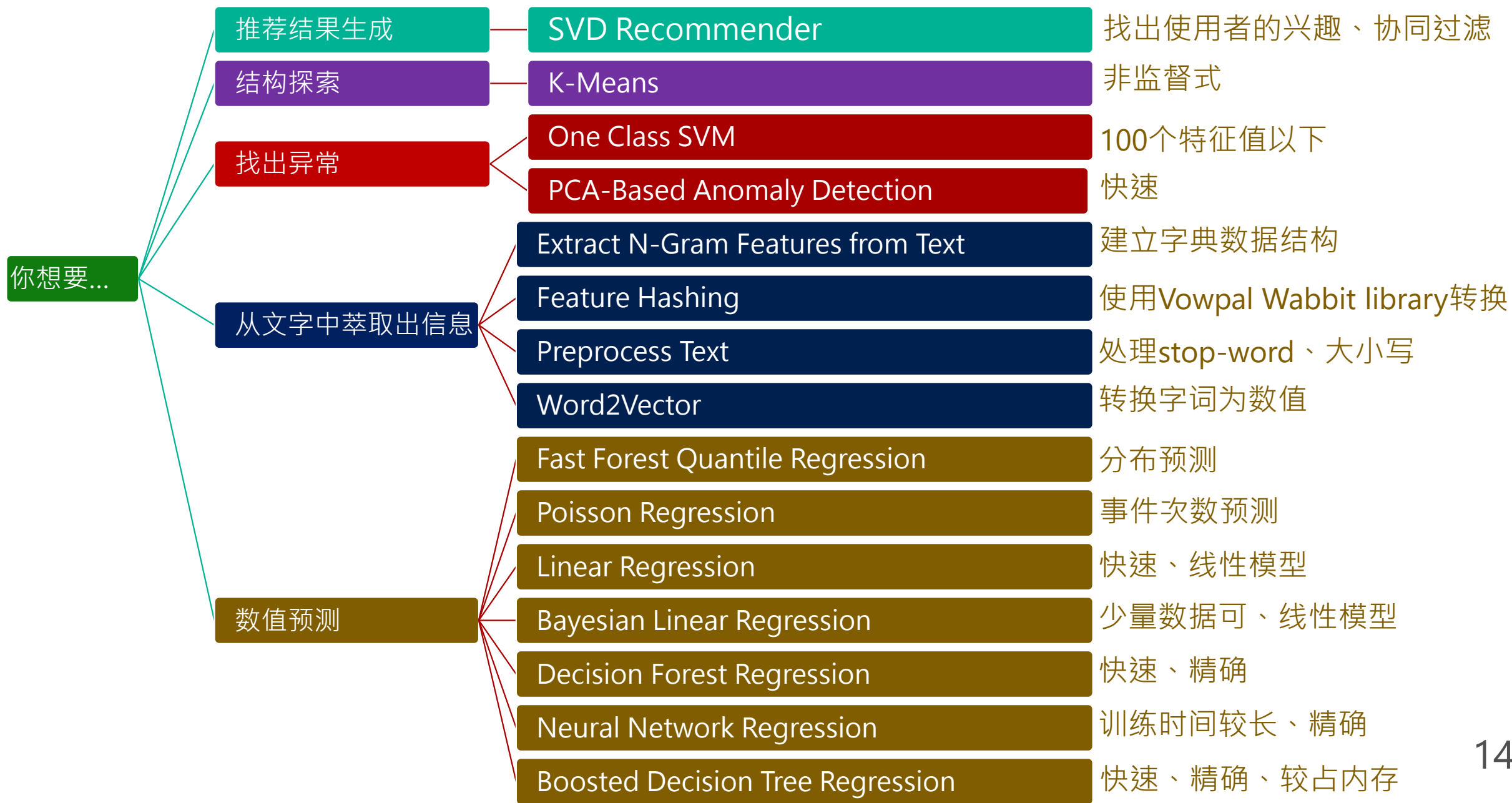
如何选择？

- 你的目标是什么？
 - 参考后面两页的问题分类
- 其他需求
 - Accuracy 精确度
 - 训练时间
 - 线性关系
 - 参数
 - 特征值种类

算法选择 – 从目标出发



算法选择 – 从目标出发





机器学习模型简介

- 预测式算法
 - 从现在与过去的数据来进行预测，例如天气、潜在客户
- 分类算法
 - 给予数据进行训练后，产生一个能够辨别类别的系统
- 时间序列预测算法
 - 概念上与第一类相近，但使用方法不同

机器学习运作流程



练习一：房价预测	练习二：铁达尼号生存预测
	
Linear Regression	Logistic Regression

练习：房价预测

取得资料

- pandas
- read_csv
- 资料观察

资料清理

- 遗漏值处理
- 格式转换

资料切割

- 训练 70%
- 测试 30%

模型选择与使用

- sklearn

结果分析与验证

- metrics

```
#import modules
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#%matplotlib inline
import seaborn as sns
```

```
#import dataset
```

```
df = pd.read_csv("data/Housing_Dataset_Sample.csv")
```

```
#observing dataset
```

```
df.head()
```

```
df.describe().T
```

```
sns.distplot(df['Price'])
```

```
sns.jointplot(df['Avg. Area Income'],df['Price'])
```



练习：房价预测



```
#prepare to train model
```

```
#X是所有可能的影响变因
```

```
#取得所有的列的0,1,2,3,4字段
```

```
X = df.iloc[:, :5]
```

```
#y是目标值
```

```
y = df['Price']
```

```
#split to training data & testing data
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=54)
```


练习：房价预测



```
#using linear regression model
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(X_train, y_train)
```

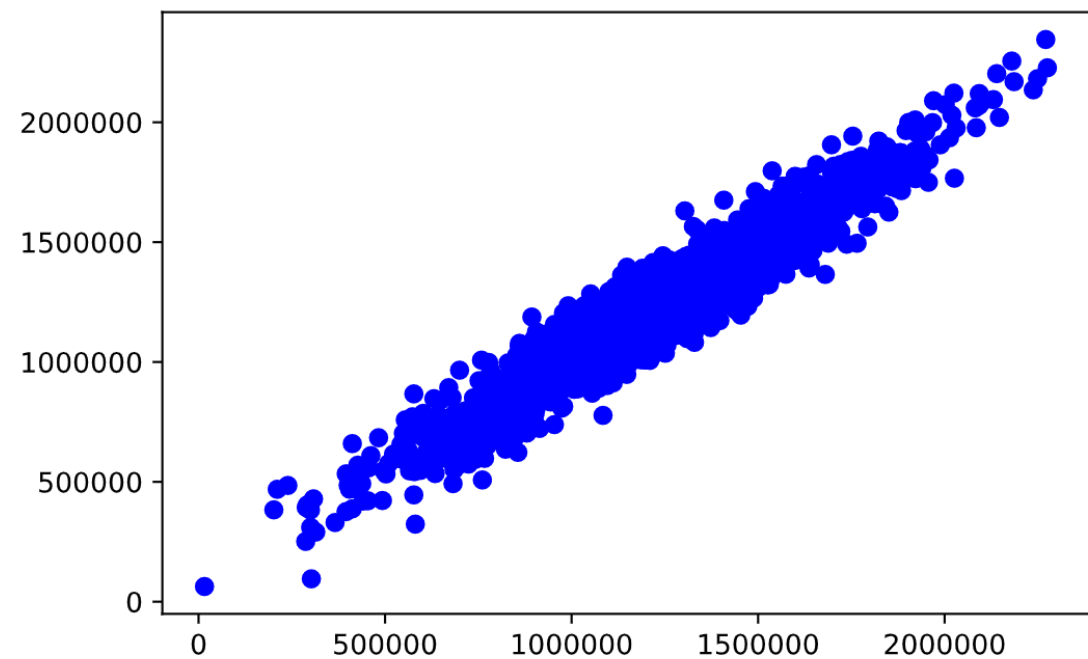
```
#get the result
predictions = reg.predict(X_test)
predictions
```

练习：房价预测



```
from sklearn.metrics import r2_score  
r2_score(y_test, predictions)  
plt.scatter(y_test, predictions, color='blue')
```

0.9216604865707106



练习：铁达尼号生存预测



```
#import modules
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#%matplotlib inline
import seaborn as sns

#import dataset
df = pd.read_csv("data/train_data_titanic.csv")
df.head()
df.info()
```

域名	说明
PassengerId	乘客编号
Survived	是否存活(0 : 否、1 : 是)
Pclass	船票等级(1等、2等、3等)
Name	乘客姓名
Sex	性别
Age	年龄
Sibsp	有多少兄弟姊妹/配偶在船上
Parch	有多少父母/小孩在船上
Ticket	船票编号
Fare	票价
Cabin	舱房编号
Embarked	登船港口 C 瑟堡 Q 皇后镇 S修咸顿

练习：铁达尼号生存预测



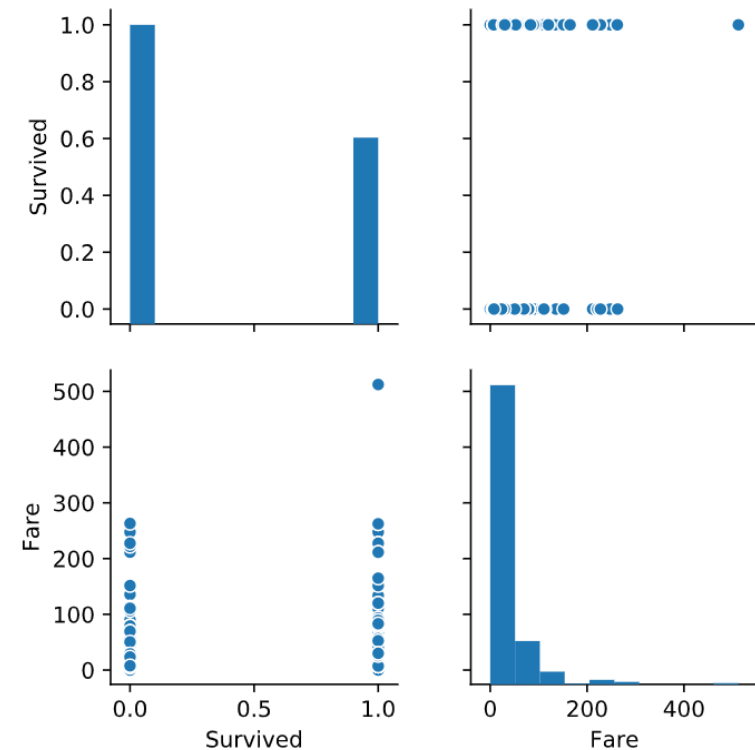
#Remove the columns model will not use

```
df.drop(['Name', 'Ticket'], axis=1, inplace=True)
df.head()
```

```
sns.pairplot(df[['Survived', 'Fare']], dropna=True)
```

#data observing

```
df.groupby('Survived').mean()
df.head()
df['SibSp'].value_counts()
df['Parch'].value_counts()
df['Sex'].value_counts()
```



练习：铁达尼号生存预测



#Handle missing values

```
df.isnull().sum()>(len(df)/2)
```

#Cabin has too many missing values

```
df.drop('Cabin',axis=1,inplace=True)
```

```
df.head()
```

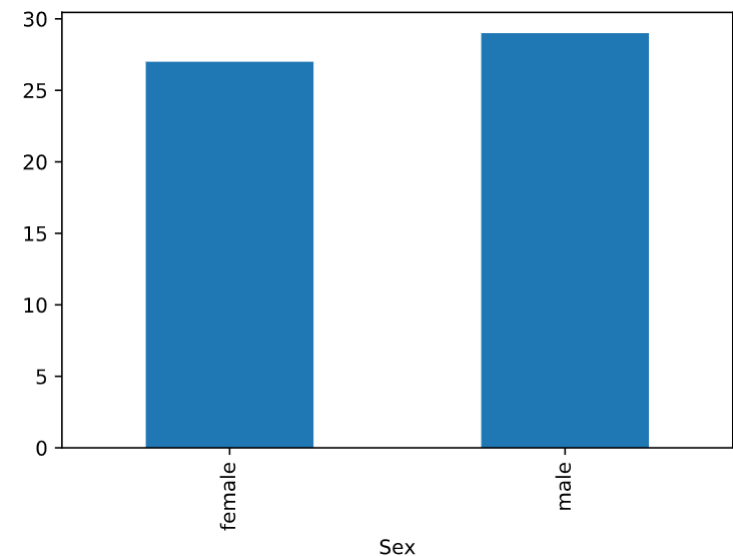
```
df['Age'].isnull().value_counts()
```

#Age is also have some missing values

```
df.groupby('Sex')['Age'].median().plot(kind='bar')
```

#缺失值男生就用男生的平均、女生就用女生的平均值来填补

```
df['Age'] = df.groupby('Sex')['Age'].apply(lambda x: x.fillna(x.median()))
```



练习：铁达尼号生存预测



#发现还有Embarked还有缺2个

```
df['Embarked'].value_counts()
```

#找出第一个次数最多的，发现是S

```
df['Embarked'].value_counts().idxmax()
```

```
df['Embarked'].fillna(df['Embarked'].value_counts().idxmax(), inplace=True)
```

```
df['Embarked'].value_counts()
```

#所有缺失值搞定！

```
df.isnull().sum()
```

```
df['Embarked'].value_counts()
```

```
S      644
```

```
C      168
```

```
Q       77
```

```
Name: Embarked, dtype: int64
```

练习：铁达尼号生存预测



#将Sex, Embarked进行转换

#Sex转换成是否为男生、是否为女生，Embarked转换为是否为S、是否为C、是否为Q

```
df = pd.get_dummies(data=df, columns=['Sex', 'Embarked'])
```

```
df.head()
```

#是否为男生与是否为女生只要留一个就好，留下是否为男生

```
df.drop(['Sex_female'], axis=1, inplace=True)
```

```
df.head()
```

练习：铁达尼号生存预测



```
df.corr()
```

```
#Prepare training data
```

```
#把Survived, Pclass丢掉
```

```
X = df.drop(['Survived', 'Pclass'], axis=1)
```

```
y = df['Survived']
```

```
#split to training data & testing data
```

```
from sklearn.model_selection import train_test_split
```

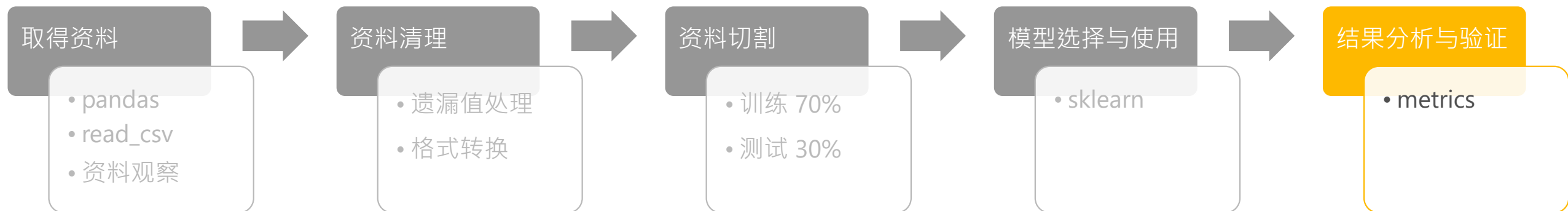
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=67)
```

练习：铁达尼号生存预测



```
#using Logistic regression model
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train, y_train)
predictions = lr.predict(X_test)
```

练习：铁达尼号生存预测



#Evaluate

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

#support是个别tag的真实个数

```
print(classification_report(y_test, predictions))
```

```
print(accuracy_score(y_test, predictions))
```

```
print(confusion_matrix(y_test, predictions))
```

```
pd.DataFrame(confusion_matrix(y_test, predictions), columns=['True Survived', 'True not Survived'], index=['Predict Survived', 'Predict not Survived'])
```

	True Survived	True not Survived
Predict Survived	146	16
Predict not Survived	29	77

常见评量方式

- 回归

- mean_squared_error
- mean_absolute_error
- explained_variance_score
- r2_score

- 分类

- Precision
- Recall
- F1 Score
- Accuracy

常见评量方式

n = 100	预测为No		预测为Yes	
实际上是 No	TN	35	FP	15 (Type I Error)
实际上是 Yes	FN	5 (Type II Error)	TP	45

Precision 准确率 = $\frac{\text{模型预测为Yes且实际上为Yes}}{\text{模型预测为Yes的个数}}$

Recall 召回率 = $\frac{\text{实际上为Yes而模型也预测为Yes}}{\text{实际上为Yes的所有个数}}$

F1 Score = $2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$

Accuracy 精准率 = $\frac{\text{模型预测为Yes且实际上为Yes} + \text{模型预测为No且实际上为No}}{\text{所有预测的个数}}$

使用时间

机率为Yes或No比例相当时，大多数可用Accuracy

- 因为当Yes或No明显比例偏高时，就全部猜那一边Accuracy会大幅提升

怕Type I Error的，要用Precision

- Type I Error 就是预测为Yes但实际为No
- 例如门禁系统把陌生人当成自家人

怕Type II Error的，要用Recall

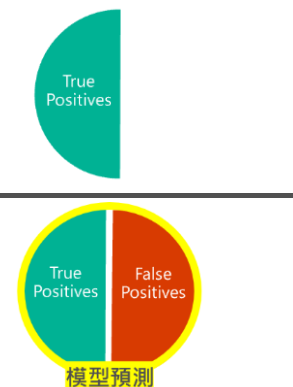
- Type II Error 就是预测为No但实际为Yes
- 例如广告投放判断不是潜在客户但结果却是潜在客户

F1 Score 可以避免Precision & Recall的极端误差

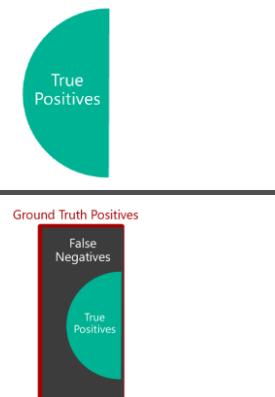
Precision & Recall

- Precision – 准确率(你的模型判断是对的, 有多少真的是对的)
- Recall – 召回率(真的是对的的项目中, 你的模型找到几个)
- 准确率是从模型的角度出发、召回率是用真实的状况来看

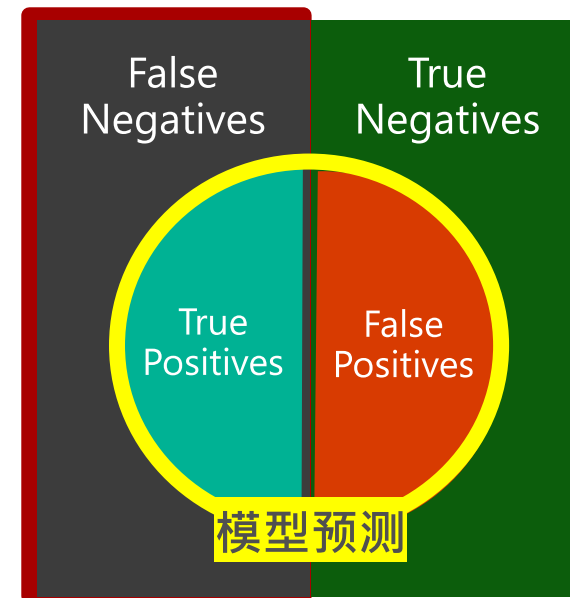
$$\text{Precision 准确率} = \frac{\text{模型预测为Yes且实际上为Yes}}{\text{模型预测为Yes的个数}}$$





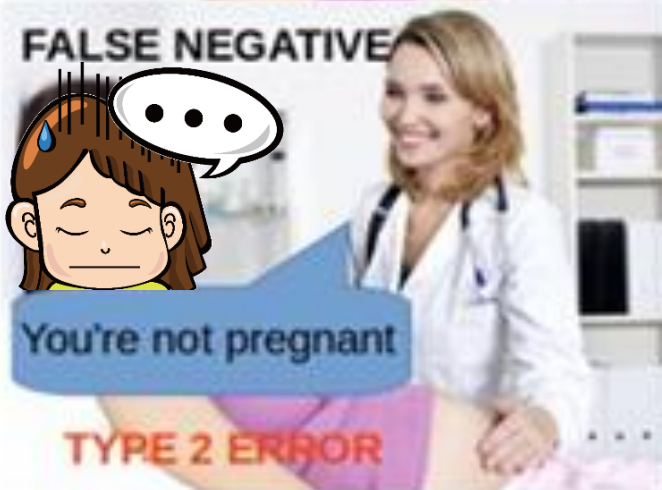

$$\text{Recall 召回率} = \frac{\text{实际上为Yes而模型也预测为Yes}}{\text{实际上为Yes的所有个数}}$$



Ground Truth Positives



范例：是否怀孕的判断

		Actual Values	
		1	0
Predicted Values	1	<p>TRUE POSITIVE</p> 	<p>FALSE POSITIVE</p> 
	0	<p>FALSE NEGATIVE</p> 	<p>TRUE NEGATIVE</p> 

Recall & Precision练习

模型预测结果



Cat



Dog



Cat

Precision 准确率 = $\frac{\text{模型预测为Yes且实际上为Yes}}{\text{模型预测为Yes的个数}}$

Precision for Cat = _____

Precision for Dog = _____

Precision for Mouse = _____

Precision for Whole Model = $\frac{\text{每一种类别的精确率加总}}{\text{类别数}}$

= _____

Recall & Precision练习

模型预测结果



Cat



Dog



Cat

Recall 召回率 = $\frac{\text{实际上为Yes而模型也预测为Yes}}{\text{实际上为Yes的所有个数}}$

Recall for Cat = _____

Recall for Dog = _____

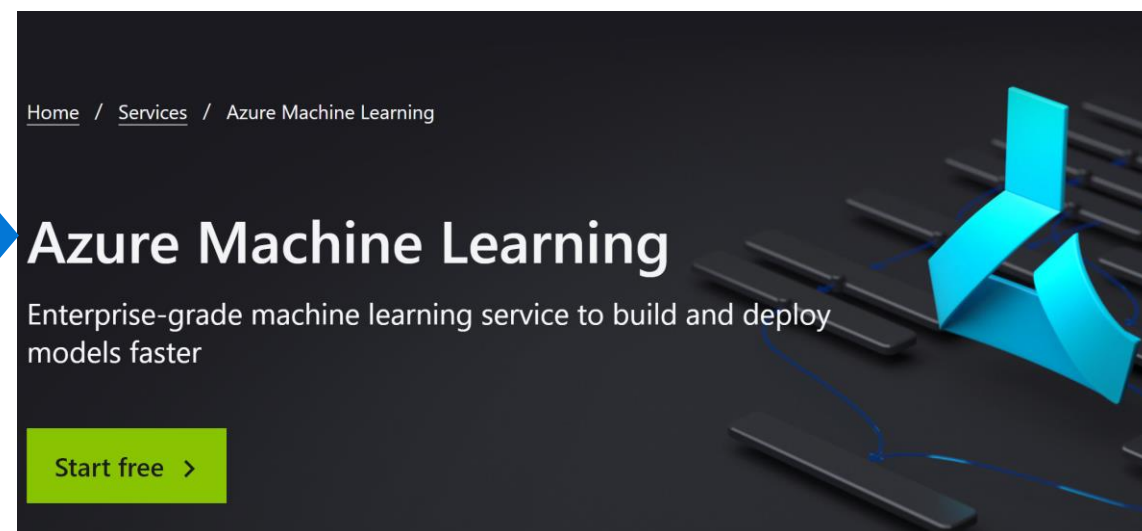
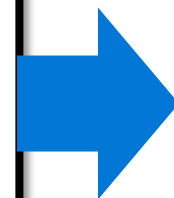
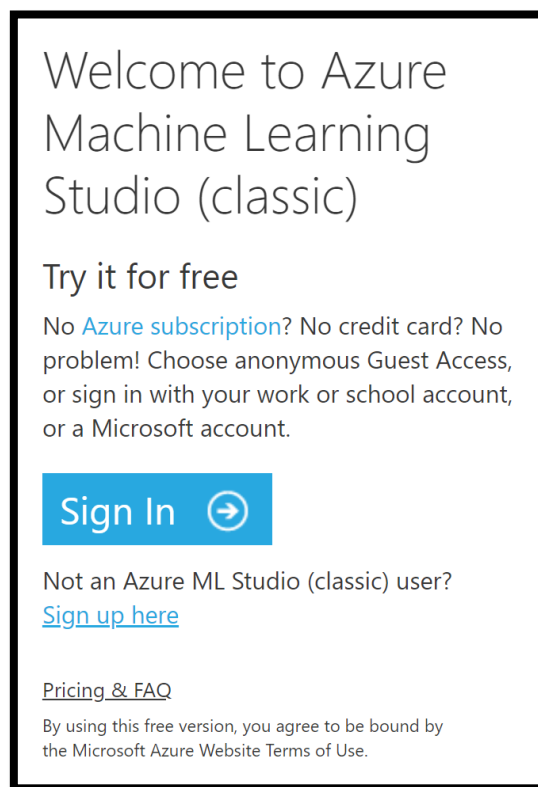
Recall for Mouse = _____

Recall for Whole Model = $\frac{\text{每一种类别的召回率加总}}{\text{类别数}}$

= _____

Azure 机器学习平台服务

- Azure Machine Learning Designer 微软Azure机器学习设计师
 - 最新推出，以拖拉方式建立机器学习流程
- Jupyter Notebook
 - 可以在云端上使用，上面有许多范例也可以自己从头建立
- VS Code Extension 插件
 - 可结合本地端运行
- CLI Extension
 - 以指令列方式使用
- Reinforcement learning
 - 实验中，使用Ray RLlib



Azure Machine Learning Designer 拖拉式界面

The screenshot displays the Azure Machine Learning Designer interface, which is a drag-and-drop environment for building machine learning pipelines. The interface is divided into several sections:

- Left Sidebar:** Contains navigation options such as "New", "Home", "Notebooks", "Automated ML", "Designer" (highlighted), "Datasets", "Experiments", "Pipelines", "Models", and "Endpoints".
- Search Bar:** A search box with the text "Search" is located at the top of the left sidebar.
- Dataset List:** A list of datasets is shown, including "Flight Delays Data".
- Transformation List:** A list of data transformation operations is displayed, including "Add Columns", "Add Rows", "Apply Math Operation", "Apply SQL Transformation", "Clean Missing Data", "Clip Values", "Convert to CSV", "Convert to Dataset", and "Edit Metadata".
- Canvas:** The main workspace where the pipeline is built. It shows a workflow starting with the "Flight Delays Data" dataset, followed by a "Normalize Data" transformation. The "Normalize Data" transformation has a slider control.
- Top Bar:** Includes a title "Flight Delays" and a settings icon. Below the title is a toolbar with icons for "Autosave on" (checked), "Save", "Undo", "Redo", "Copy", "Paste", "Delete", "Zoom In", "Zoom Out", "Fit", and "Refresh".

Azure Machine Learning – 自动化ML(UI)

[首頁](#) > [新增](#) >

機器學習 

Microsoft



機器學習



已儲存

Microsoft

建立

概觀

方案

Azure Machine Learning 可為開發人員及資料科學家提供各種具有生產力的機器學習模型建置、訓練與部署體驗。建立 Azure Machine Learning 工作區即可訓練、管理及部署機器學習實驗及 Web 服務。

實用的連結

[文件](#)

[定價詳細資料](#)

[Azure AI 資源庫](#)

Azure Machine Learning – 自动化ML(UI)

- 省略标签步骤，进行检阅+建立

[首頁](#) > [新增](#) > [機器學習](#) >

機器學習

建立機器學習工作區

基本 標籤 檢閱 + 建立

專案詳細資料

選取用以管理部署資源及成本的訂用帳戶。使用像資料夾這樣的資源群組來安排及管理您的所有資源。

工作區詳細資料

指定工作區的名稱、區域和版本。

工作區名稱 * ⓘ

HelloAutoML



區域 * ⓘ

東亞



工作區版本 * ⓘ

企業



Automated ML

- 新增自动化ML回合

Microsoft Azure Machine Learning

HelloML > Automated ML (preview)

Automated ML (preview)

無需撰寫任何程式碼，即可讓自動化 ML 依據您的資料訓練及尋找最佳模型。 [深入了解自動化 ML](#)

+ 新增自动化 ML 回合

沒有任何最近的自动化 ML 回合可顯示。
按一下 [新增自动化 ML 回合] 建立您的第一個回合
[深入了解如何建立自动化 ML 回合](#)

文件

[概念: 什麼是自动化 ML?](#)

选取数据集

- 建立数据集 -> 来自本机档案

建立新的自动化 ML 回合

- 选取资料集
- 设定执行
- 工作类型与设定

选取资料集

從下面的清單中選取資料集，或建立新的資料集。自動化 ML 目前只支援用表格式資料撰寫執行。

+

建立資料集

▼

來自本機檔案

來自資料存放區

來自 Web 檔案

來自開放資料集

僅顯示支援的資料集

搜尋以篩選項目...

資料集類型	建立日期	已修改
<div><div></div><div>沒有任何可顯示的 資料集</div></div>		

上一步

下一個

取消

填写数据集名称 BankMarketing

從本機檔案建立資料集

☒ 基本資訊

☐ 資料存放區和檔案選取

☐ 設定與預覽

☐ 結構描述

☐ 確認詳細資料

基本資訊

名稱 *



資料集版本

BankMarketing

1

資料集類型 * ①

表格式



描述

資料集描述

上一步

下一個

取消

默认数据存放区 -> 设定上传路径 -> 下一步

從本機檔案建立資料集

- 基本資訊
- 資料存放區和檔案選取
- 設定與預覽
- 結構描述
- 確認詳細資料

資料存放區和檔案選取

選取或建立資料存放區 *

- 目前選取的資料存放區: workspaceblobstore (Azure Blob 儲存體) (預設)
- 先前建立的資料存放區
- 建立新的資料存放區

為您的資料集選取檔案 *

建立資料集之後，這些檔案將會上傳至您的預設 Blob 儲存體，並可在工作區中使用。支援的檔案類型包含: 分隔檔 (也就是 csv、tsv)、Parquet、JSON Lines 與純文字。

瀏覽 已選取 1 個檔案。大小總計為 3.958 MiB。已上傳 0/1 個檔案

檔案名稱	大小 (MiB)	上傳 %	狀態
bankmarketing_train.csv	3.958		

上傳路徑

dataset 檔案將會上傳至 '\$(上傳路徑)07-08-2020_035156_UTC'

☐ 跳過資料驗證 ①

设定与预览

- 确认侦测是否正确：
 - 文件格式-分隔符
 - 分隔符-逗号
 - 编码-UTF-8
 - 所有档案都有相同标头
 - 略过资料列-无

設定與預覽

這些設定是自動偵測而來。請驗證選取項目正確，否則請予以更新

檔案格式

分隔符號

逗號

範例

Field1,Field2,Field3

編碼

UTF-8

資料行標頭

所有檔案都有相同的標頭

略過資料列

無

123	識..	123	age	Abc	job	Abc	marital	Abc	education	Abc	defa
1		57			technician		married		high.school		no
2		55			unknown		married		unknown		unknov
3		33			blue-collar		married		basic.9y		no
4		36			admin.		married		high.school		no

选择要实验的字段

- 将 day_of_week 关闭

Create dataset from local files

☒ Basic info

☒ Settings and preview

☒ **Schema**

☐ Confirm details

Include	Column name	Properties	Type
<input type="checkbox"/>	Path	Not applicable to selecte... ▾	String
<input checked="" type="checkbox"/>	age	Not applicable to selecte... ▾	Integer
<input checked="" type="checkbox"/>	job	Not applicable to selecte... ▾	String
<input checked="" type="checkbox"/>	marital	Not applicable to selecte... ▾	String
<input checked="" type="checkbox"/>	education	Not applicable to selecte... ▾	String
<input checked="" type="checkbox"/>	default	Not applicable to selecte... ▾	String
<input checked="" type="checkbox"/>	housing	Not applicable to selecte... ▾	String
<input checked="" type="checkbox"/>	loan	Not applicable to selecte... ▾	String
<input checked="" type="checkbox"/>	contact	Not applicable to selecte... ▾	String
<input checked="" type="checkbox"/>	month	Not applicable to selecte... ▾	String

Back

Next

Cancel

确认建立

從本機檔案建立資料集

- ✓ 基本資訊
- ✓ 資料存放區和檔案選取
- ✓ 設定與預覽
- ✓ 結構描述
- 確認詳細資料

確認詳細資料

資料集類型
表格式

檔案設定

檔案格式
分隔符號
分隔符號
逗號
編碼
UTF-8
資料行標頭
所有檔案都有相同的標頭
略過資料列
無

☐ 在建立後分析此資料集

上一步

建立

选取数据集完成

- 勾选BankMarketing -> 下一个

[HelloML](#) > [Automated ML \(preview\)](#) > 啟動回合

✓ 成功: 已成功建立 BankMarketing 資料集

建立新的自動化 ML 回合

● 選取資料集

○ 設定執行

○ 工作類型與設定

選取資料集

從下面的清單中選取資料集，或建立新的資料集。自動化 ML 目前只支援用表格式資料撰寫執行。

+ 建立資料集 ▾

☒ 僅顯示支援的資料集

🔍 搜尋以篩選項目...

資料集名稱

☒ BankMarketing

上一步

下一個

设定执行

- 输入实验名称(HelloAutoML)、目标数据行(y)、选取计算丛集(建立新的计算)

建立新的自动化 ML 回合

- ✓ 选取资料集
- 设定执行
- 工作类型与设定

设定执行

设定实验。从现有实验中选取名称或定义新名称，然后选取要使用的目标资料行及训练计算。 [深入了解如何设定实验](#)

资料集

BankMarketing (检视资料集)

实验名称 *

☐ 选取现有的 ☒ 新建

新增实验名称

HelloAutoML

目标资料行 * i

y

选取计算丛集 * i

HelloML

[建立新的计算](#) [重新整理计算](#)

上一步

下一个

建立新的计算

- 计算名称
- 虚拟机类型
- 虚拟机大小
- 节点数目下限
- 节点数目上限

新增计算叢集 ⓘ

計算名稱 * ⓘ

HelloAutoML

區域 * ⓘ

westcentralus

虛擬機器類型 *

CPU (中央處理器)

虛擬機器優先順序 * ⓘ

專用

低優先順序

虛擬機器大小 * ⓘ

Standard_DS12_v2

4 核心, 28 GB (RAM), 56 GB (磁碟)

節點數目下限 * ⓘ

0

節點數目上限 * ⓘ

1

相應減少之前的閒置秒數 * ⓘ

120

下載自動化的範本

建立

取消

设定执行

- 确认计算丛集后 -> 下一个

建立新的自动化 ML 回合

- ☒ 选取资料集
- ☒ 设定执行
- ☐ 工作类型与设定

设定执行

设定实验。从现有实验中选取名称或定义新名称，然后选取要使用的目标资料行及训练计算。[深入了解如何设定实验](#)

资料集

BankMarketing ([检视资料集](#))

实验名称 *

☐ 选取现有的 ☒ 新建

新增实验名称

HelloAutoML

目标资料行 * [i](#)

y

选取计算丛集 * [i](#)

HelloAutoML

[建立新的计算](#) [重新整理计算](#)

上一步

下一个

取消

选取工作类型

- 分类

建立新的自动化 ML 回合

- ✓ 选取资料集
- ✓ 设定执行
- 工作类型与设定

选取工作类型

为实验选取机器学习服务工作类型。如有需要，还可使用其他设定来微调实验。



分类



预测目标资料行中数种类别的其中一种。是/否、蓝、红、绿。



启用深度学习 [i](#)



迴歸

预测连续数值



時間序列预测

根据时间预测值

[⚙️ 检视其他组态设定](#) [🔍 检视特徵化设定](#)

上一步

完成

取消

选取工作类型

• 分类

建立新的自动化 ML 回合

- ✓ 选取资料集
- ✓ 设定执行
- 工作类型与设定

选取工作类型

为实验选取机器学习服务工作类型。如有需要，还可



分类

预测目标资料行中数种类别的其中之一。是



启用深度学习 ⓘ



迴歸

预测连续数值



時間序列預測

根据时间预测值



檢視其他組態設定 ⓘ 檢視特徵化設定

其他組態



主要計量 ⓘ

精確度



✓ 解釋最佳模型 ⓘ

封鎖的演算法 ⓘ

自动化 ML 在訓練期間不會使用的演算法清單。

✓ 結束準則

訓練作業時間 (小時) ⓘ

1

計量分數閾值 ⓘ

計量分數閾值

✓ 驗證

驗證類型 ⓘ

自動

✓ 並行

並行反覆次數上限 ⓘ

1

上一步

完成

取消

完成 -> 等待自动化ML进行

- 自动依算法精确度进行排序

持續時間
45 分 9.645 秒

最佳模型摘要


演算法名稱

VotingEnsemble

精確度

0.91927  檢視所有其他計量

取樣

100% 

演算法名稱

VotingEnsemble

SparseNormalizer, XGBoostClassifier

StackEnsemble

SparseNormalizer, XGBoostClassifier

MaxAbsScaler, LightGBM

解釋

[檢視說明](#)

精確度 ↓

0.91927

0.91411

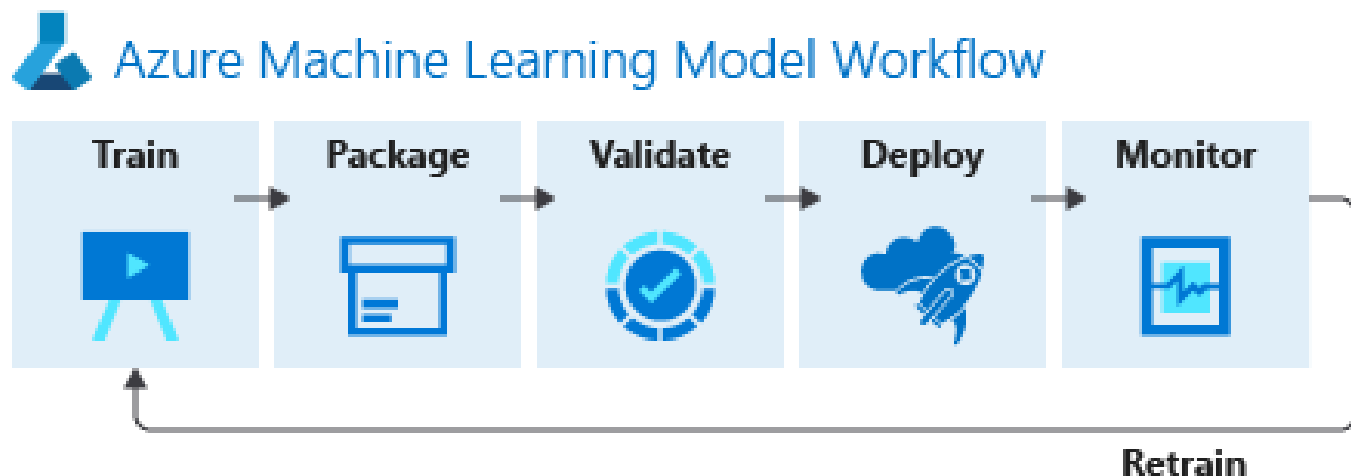
0.91290

0.91259

0.91199

Azure Machine Learning 微软Azure机器学习

- 云端环境
- 可以进行模型的训练/部署/自动化/管理/追踪
- 适用于
 - 传统机器学习 / 深度学习 / 监督式学习 / 非监督式学习
- 使用弹性
 - 可自行撰写Python/R 或 使用Azure ML 图形化界面



Azure Machine Learning Designer

- 微软Azure机器学习设计师
- 最新推出，以拖拉方式建立机器学习流程

Jupyter Notebook

- 可以在云端上使用，上面有许多范例也可以自己从头建立

VS Code Extension 插件

- 可结合本地端运行

CLI Extension

- 以指令列方式使用

Reinforcement learning

- 实验中，使用Ray RLlib



Reactor



developer.microsoft.com/reactor/
@MSFTReactor on Twitter

议程结束 感谢聆听



请记得填写课程回馈问卷
<https://aka.ms/ReactorFeedback>

© 2019 Microsoft Corporation. All rights reserved. The text in this document is available under the Creative Commons Attribution 3.0 License, additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are not included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.