Microsoft

Using Machine Learning Models

数据科学 – 机器学习模型入门

Jan 2021

Microsoft Reactor | Ryan Chung

Microsoft

Ryan Chung

Instructor / DevelopIntelligence
Founder / MobileDev.TW

@ryanchung403 on WeChat
Ryan@MobileDev.TW

Microsoft

Reactor

developer.microsoft.com/reactor/

@MSFTReactor on Twitter

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

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

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

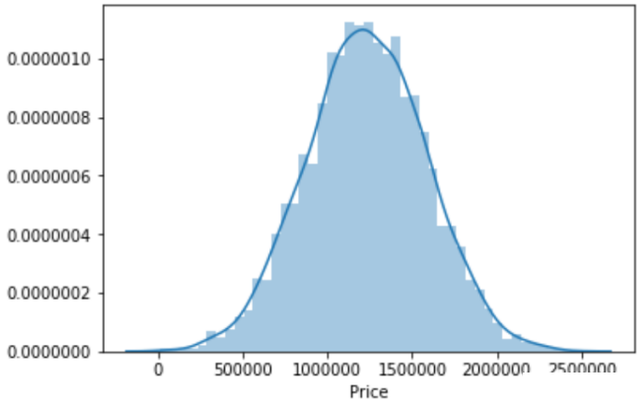
结果分析与验证
- metrics

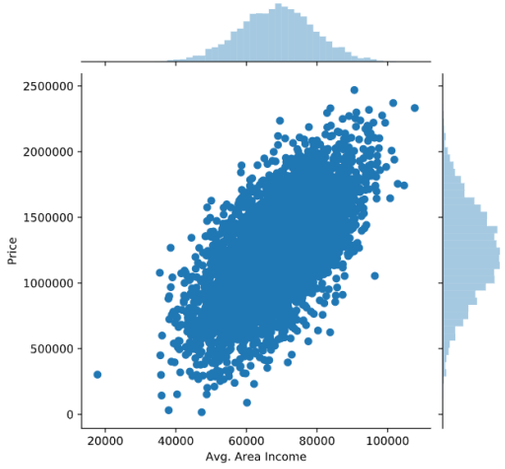
```
#import modules
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#%matplotlib inline
import seaborn as sns

#import dataset

df = pd.read_csv("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

取得资料
- pandas
- read_csv
- 资料观察

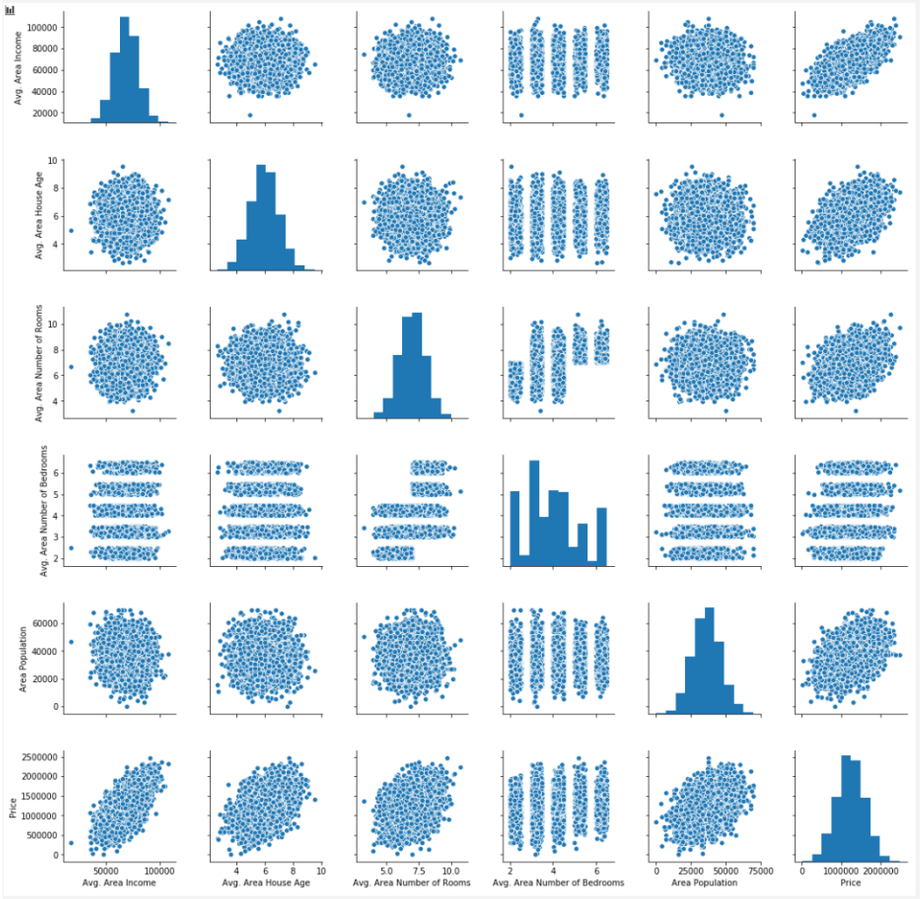
资料清理
- 遗漏值处理
- 格式转换

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

```
sns.pairplot(df)
```



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

```
#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

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

```
#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
```

```
[24] predictions

array([ 614607.96220755, 1849444.80372635, 1118945.08884266, ...,
        834789.03428584, 1787928.10906905, 1455422.23696488])
```

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%

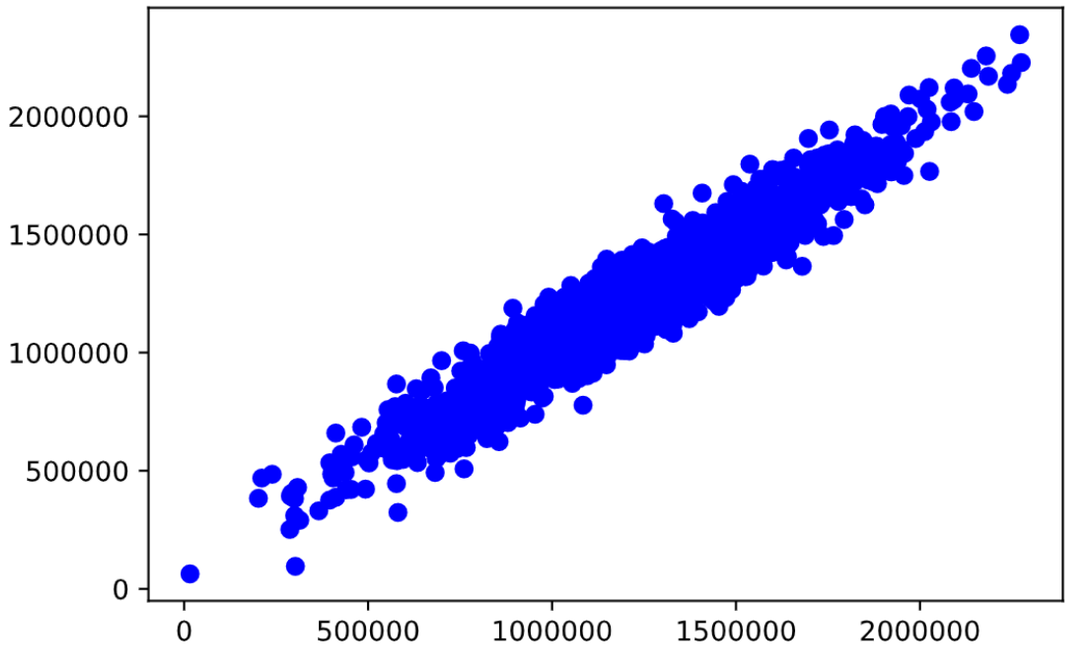
模型选择与使用
- sklearn

结果分析与验证
- metrics

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

0.9216604865707106

如何让点的分布更视觉化?



练习：铁达尼号生存预测

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("train_data_titanic.csv")
```

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

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

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

练习：铁达尼号生存预测

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

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

练习：铁达尼号生存预测

取得资料
- pandas
- read_csv
- 资料观察

资料清理
- 遗漏值处理
- 格式转换

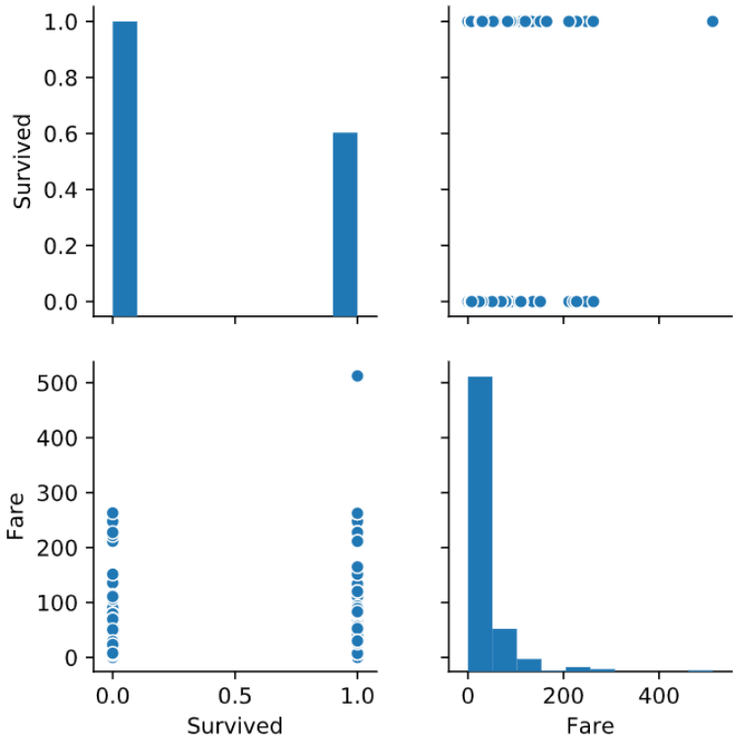
资料切割
- 训练 70%
- 测试 30%

模型选择与使用
- sklearn

结果分析与验证
- metrics

```
#Remove the columns model will not use
df.drop(['Name','Ticket'],axis=1,inplace=True)
df.head()

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



练习：请尝试使用其他栏位观察与Survived之间的关联

练习：铁达尼号生存预测

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

```
#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.groupby('Survived').mean()
```

Survived	PassengerId	Pclass	Age	SibSp	Parch	Fare
0	447.016393	2.531876	30.626179	0.553734	0.329690	22.117887
1	444.368421	1.950292	28.343690	0.473684	0.464912	48.395408

存活者
平均年龄稍低一些!
票价平均较高一些!

练习：铁达尼号生存预测

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

```
#data observing
df['SibSp'].value_counts()
df['Parch'].value_counts()
df['Sex'].value_counts()
```

```
male      577
female    314
Name: Sex, dtype: int64
```

```
0    608
1    209
2     28
4     18
3     16
8      7
5      5
Name: SibSp, dtype: int64
```

```
0    678
1    118
2     80
5      5
3      5
4      4
6      1
Name: Parch, dtype: int64
```

练习：铁达尼号生存预测

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

```
#Handle missing values
df.isnull().sum()
len(df)
len(df)/2
df.isnull().sum()>(len(df)/2)
```

```
[23] df.isnull().sum()

PassengerId       0
Survived          0
Pclass            0
Sex               0
Age             177
SibSp             0
Parch             0
Fare              0
Cabin           687
Embarked          2
dtype: int64
```

```
[24] len(df)

891
```

```
[25] len(df)/2

445.5
```

```
[26] df.isnull().sum()>len(df)/2

PassengerId    False
Survived       False
Pclass         False
Sex            False
Age            False
SibSp          False
Parch          False
Fare           False
Cabin           True
Embarked       False
dtype: bool
```

练习：铁达尼号生存预测

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

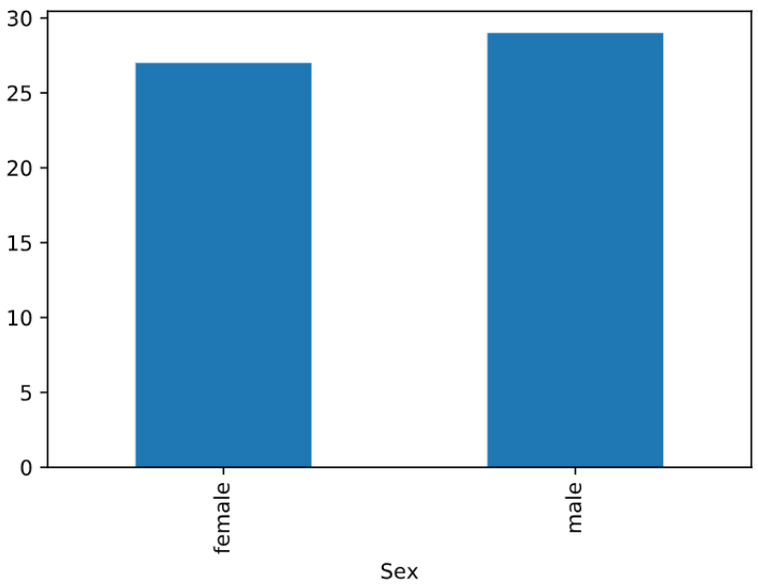
```
#Cabin has too many missing values
df.drop('Cabin',axis=1,inplace=True)
#Age is also having some missing values
df['Age'].isnull().value_counts()
df.groupby('Sex')['Age'].median().plot(kind='bar')
```

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

False    714
True     177
Name: Age, dtype: int64
```

```
[40] df.groupby('Sex')['Age'].median()

Sex
female    27.0
male      29.0
Name: Age, dtype: float64
```



练习：铁达尼号生存预测

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

```
#缺失值男生就用男生的平均(29)、女生就用女生的平均值(27)来填补
df['Age'] = df.groupby('Sex')['Age'].apply(lambda x: x.fillna(x.median()))
```

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

29.00    144
27.00     71
24.00     30
22.00     27
18.00     26
         ...
55.50      1
66.00      1
70.50      1
23.50      1
0.42       1
Name: Age, Length: 88, dtype: int64
```

练习：铁达尼号生存预测

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

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

```
[45] df.isnull().sum()

PassengerId    0
Survived       0
Pclass         0
Sex            0
Age            0
SibSp          0
Parch          0
Fare           0
Embarked       2
dtype: int64
```

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

S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

加起来889笔 (少2笔)

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

S    646
C    168
Q     77
Name: Embarked, dtype: int64
```

填补后

练习：铁达尼号生存预测

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

→

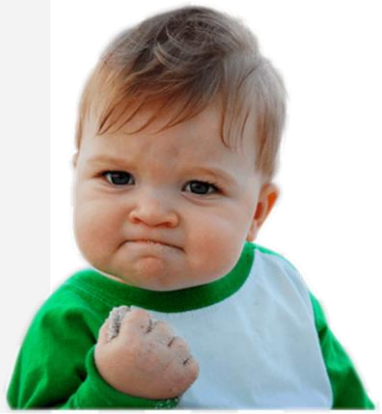
结果分析与验证
- metrics

#所有缺失值搞定！

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

```
[51] df.isnull().sum()

PassengerId    0
Survived       0
Pclass         0
Sex            0
Age            0
SibSp          0
Parch          0
Fare           0
Embarked       0
dtype: int64
```



练习：铁达尼号生存预测

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

```
#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()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	1	0	3	22.0	1	0	7.2500	1	0	0	1
1	2	1	1	38.0	1	0	71.2833	0	1	0	0
2	3	1	3	26.0	0	0	7.9250	0	0	0	1
3	4	1	1	35.0	1	0	53.1000	0	0	0	1
4	5	0	3	35.0	0	0	8.0500	1	0	0	1

练习：铁达尼号生存预测

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

```
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

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

```
#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

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

```
#Evaluate
from sklearn.metrics import confusion_matrix, accuracy_score, recall_score, precision_score
accuracy_score(y_test, predictions)
recall_score(y_test, predictions)
precision_score(y_test, predictions)
```

```
0.832089552238806
0.7264150943396226
0.8279569892473119
```

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

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

练习：铁达尼号生存预测

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

```
#Try Decision Tree
from sklearn import tree
tr = tree.DecisionTreeClassifier(random_state=0, max_depth=5)
tr.fit(X_train, y_train)
tr_predictions = tr.predict(X_test)
```

练习：铁达尼号生存预测

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

```
#Evaluate
accuracy_score(y_test, tr_predictions)  0.8470149253731343
recall_score(y_test, tr_predictions)    0.7924528301886793
precision_score(y_test, tr_predictions) 0.8155339805825242

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

	Predict not Survived	Predict Survived
True not Survived	143	19
True Survived	22	84

练习：铁达尼号生存预测

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

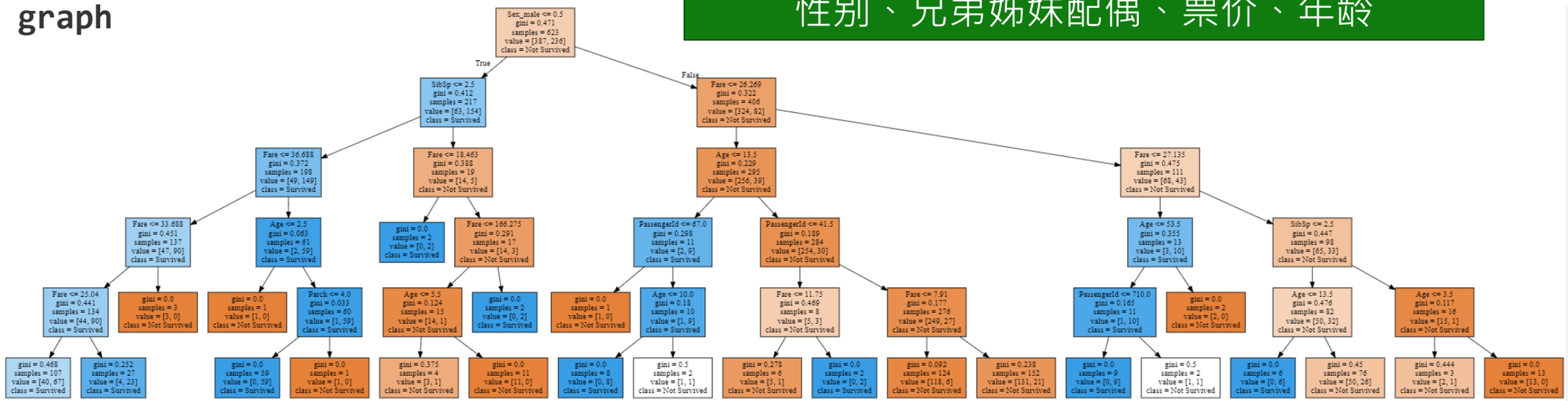
结果分析与验证
- metrics

```
import graphviz

dot_file = tree.export_graphviz(tr, out_file=None, feature_names=X.columns,
class_names=['Not Survived','Survived'], filled=True, rounded=False)

graph = graphviz.Source(dot_file)
graph
```

关键特征
性别、兄弟姊妹配偶、票价、年龄



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 $= \dfrac{\text{模型预测为}\boldsymbol{Yes}\text{且实际上为}\boldsymbol{Yes}}{\text{模型预测为}\boldsymbol{Yes}\text{的个数}}$

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

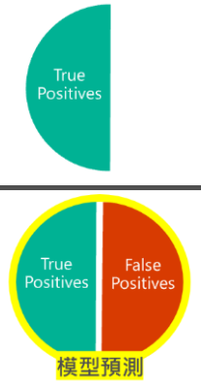
F1 Score $= 2 * \dfrac{\boldsymbol{Precision} * \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}}$

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

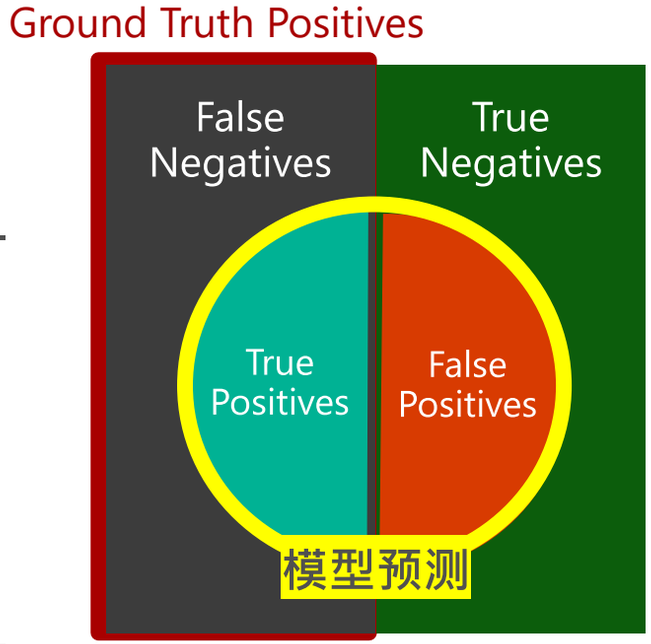
Precision & Recall

- Precision – 准确率(你的模型判断是对的中，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

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

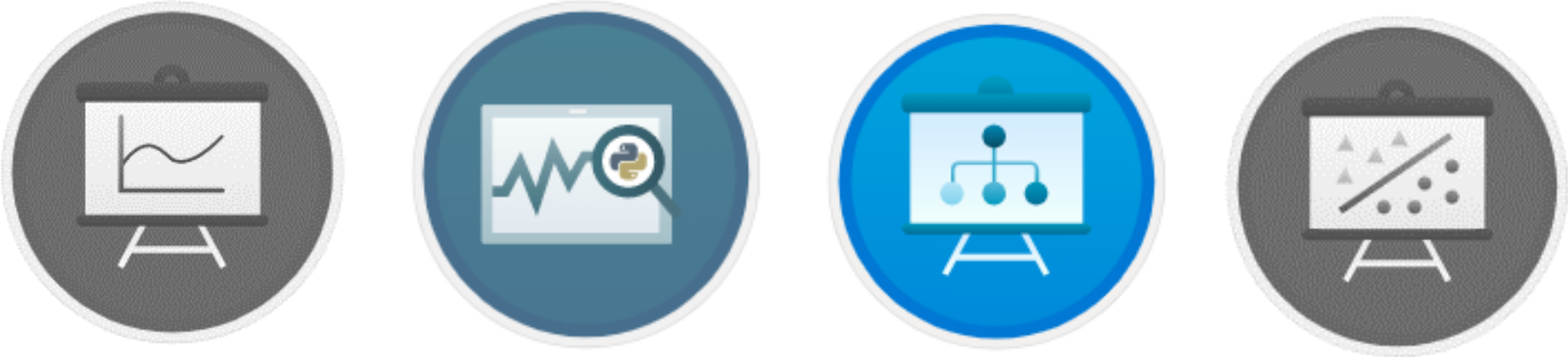


Recall 召回率 $= \dfrac{\text{实际上为}\boldsymbol{Yes}\text{而模型也预测为}\boldsymbol{Yes}}{\text{实际上为}\boldsymbol{Yes}\text{的所有个数}} =$



小结

- 面对问题首先厘清是分类/数值预测/时间序列相关，再决定方法
- 了解各特征意义，确认遗漏值状况，选择合适方式填补或直接舍去
- 选择算法、选择评估方式，如需要可搭配可视化方式显示



Microsoft

Reactor

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

Microsoft

议程结束
感谢聆听

请记得填写课程回馈问卷 (Event ID : XXXXX)
https://aka.ms/Reactor/Survey