مكتبات البايثون لتحليل البيانات

مكتبة نمباي NumPy Library

يرمز اسم مكتبة **نمباي** (NumPy) إلى **البايثون العددي** (Numerical Python)، وهي مكتبة قياسية للعمل مع البيانات العددية في البايثون، يمكن استخدامها لإجراء مجموعة متنوعة من العمليات الرياضية على المصفوفات.

الجدول 3.3: وظائف مكتبة نيمباي

وظيفة	المعنى
add(arr1, arr2,...)	إضافة المصفوفات.
multiply(arr1,arr2,...)	ضرب المصفوفات.
absolute(arr)	تُرجع القيمة المطلقة لكل عنصر في المصفوفة المدخلة.
maximum(arr1,arr2,...)	تُرجع القيمة القصوى في المصفوفة المدخلة.

وظيفة (Method):

هي دالة مرتبطة **بكائن** (Object) ويتم تعريفها داخل الفئة (Class). على سبيل المثال: np.add(arr1, arr2).

ابدأ بإنشاء قائمة بسيطة في مفكرة جوبيتر الخاصة بك. هذه قائمتك:

مصفوفة (Array):

هي نوع من البيانات يمكنه الاحتفاظ بعدد ثابت من القيم التي لها نفس نوع البيانات.

```
myList = [-3,-2,-1,0,1,2,3,4,5,5,5,6,7,8]

print(type(myList))
print(myList)
```

```
<class 'list'>
[-3, -2, -1, 0, 1, 2, 3, 4, 5, 5, 5, 6, 7, 8]
```

الشكل 3.10: وضع قائمة في مفكرة جوبيتر

استخدم مكتبة نمباي، وفي هذا المقطع البرمجي ستستخدم وظيفة **القيمة المطلقة (absolute())** لطباعة القيم المطلقة للقائمة.

عند استخدام مكتبة، يمكنك أن تعطيها اسمًا لاستخدام وظائفها في مقطعك البرمجي.

عند استخدام وظيفة من المكتبة، اكتب اسم المكتبة ثم نقطة ثم اسم الدالة.

```
import numpy as np

a = np.absolute(myList)
print(a)
```

```
[3 2 1 0 1 2 3 4 5 5 5 6 7 8]
```

الشكل 3.11: استخدام مكتبة نمباي

مكتبة بانداس Pandas Library

تأخذ مكتبة بانداس البيانات وتنشئ كائن البايثون، وهناك نوعان رئيسيان من الكائنات:

> **المتسلسلة** (Series): عبارة عن مصفوفة أحادية البعد قادرة على حمل أي نوع من البيانات (**الأعداد الصحيحة** (Integers)، و**السلاسل النصية** (Strings)، و**الأرقام العشرية** (Floats)، وكائنات البايثون وغيرها).

> إطار البيانات (DataFrame): هو هيكل بيانات ثنائي الأبعاد يبدو مشابهًا جدًا لجدول في ورقة عمل إكسل.

لكل كائن أساليبه وسماته الخاصة. يمكنك إنشاء متسلسلة أو إطار بيانات من الصفر (من القوائم والقواميس وما إلى ذلك) كما يمكن استيراد البيانات من مصادر البيانات، مثل إكسل و CSV، و SQL، و JSON، والمزيد.

الجدول 3.4: الاختلافات بين مكتبات بانداس ونيمباي

	بانداس	نيمباي
أنواع البيانات	يعمل مع البيانات المجدولة.	يعمل مع البيانات العددية.
أنواع الكائنات	متسلسلة (Series)، إطار البيانات (DataFrame).	مصفوفات.
الأداء	يتعامل مع مئات الآلاف من البيانات.	يعالج خمسين ألف صف أو أقل.
استخدام الذاكرة	يستهلك المزيد من الذاكرة.	يستهلك ذاكرة أقل.
الاستخدام	تحليل البيانات وتصويرها.	إجراء الحسابات.

كائن المتسلسلة Series Object

الآن، ستقوم بتحويل هذه القائمة إلى كائن المتسلسلة. للقيام بذلك، عليك تضمين مكتبة باندا‌س في مفكرتك. ولاستخدام مكتبة في البايثون، يمكنك إضافة كلمة **استيراد** (Import) واسم المكتبة في بداية مقطعك البرمجي.

```
import pandas as pd
```

```
s = pd.Series(myList, name='Numbers')

print(s)
```

```
0     -3
1     -2
2     -1
3      0
4      1
5      2
6      3
7      4
8      5
9      5
10     5
11     6
12     7
13     8
Name: Numbers, dtype: int64
```

في مفكرة جوبيتر، عليك استيراد المكتبة مرة واحدة فقط ثم يمكنك استخدامها في المفكرة بأكملها.

سمات كائن المتسلسلة Attributes of Series Object

في الجدول 3.5 يتم تقديم بعض السمات الأكثر شيوعًا التي يمكنك استخدامها لكائن المتسلسلة.

الجدول 3.5: سمات كائن المتسلسلة

السمة	المعنى
name	تُرجع اسم المتسلسلة.
size	تُرجع حجم المتسلسلة.
is_unique	تُرجع صواب (True) إذا كانت قيم كائن المتسلسلة فريدة، وإلا فإنها تُرجع خطأ (False).
hasnans	تُرجع صواب (True) إذا كان كائن المتسلسلة المعطى لديه قيم مفقودة، وإلا فإنها تُرجع خطأ (False).

السمة (Attribute):

قيمة مرتبطة بالكائن الذي يشار إليه بالاسم باستخدام تعبيرات منقطة. على سبيل المثال، إذا كان الكائن **طالب** (student) وكانت السمة **درجة** (grade)، فسيتم الإشارة إليها student.grade.

في الحوسبة، NaN ترمز إلى ليس رقمًا (Not a Number).

طبّق بعض هذه السمات في كائن المتسلسلة.

```
# What is the name of the Series?
print("The name of the series is:", s.name)
```

```
The name of the series is: Numbers
```

```
# Print Series size
print("Size of the series is:", s.size)
```

```
Size of the series is: 14
```

```
print("Are the elements of this series unique?", s.is_unique)
```

```
Are the elements of this series unique? False
```

```
# Check if there are empty rows in the Series (nan = Not A Number)
print("Are there empty values in the series?", s.hasnans)
```

```
Are there empty values in the series? False
```

كائن إطار البيانات DataFrame Object

الأداة التحليلية الأكثر شيوعًا واستخدامًا هي إكسل. يمكنك العمل مع ملفات إكسل في مفكرة جوبيتر باستخدام مكتبة بانداس. لفتح ملف إكسل في مفكرة جوبيتر، تحتاج إلى أن تكون هذه الملفات (ملف الإكسل والمفكرة) في نفس المجلد.

مكتبة نظام التشغيل OS Library

للتحقق من ملف العمل الخاص بك، يمكنك استخدام مكتبة **نظام التشغيل** (OS)، حيث أنها توفر في بايثون وظائف لإنشاء وإزالة دليل (مجلد)، وجلب محتوياته، وتغيير أو تحديد المجلد الحالي، إلى أخره.

```
import os
os.getcwd()
```

```
'C:\\Users\\Documents\\Jupyter examples'
```

getcwd يرمز إلى

احصل على مجلد العمل الحالي

.(get current working directory)

الشكل 3.14: مكتبة نظام التشغيل

الآن، ستقوم بتحويل ملف الإكسل التالي إلى إطار البيانات لمعالجة بياناته.

```
data = pd.read_excel('saudischools.xlsx')
```

```
data
```

سمات كائن إطار البيانات Attributes of a DataFrame Object

في الجدول التالي، يتم تقديم بعض السمات الأكثر شيوعًا، والتي يمكنك توظيفها في الحصول على معلومات حول إطار البيانات.

الجدول 3.6: سمات كائن إطار البيانات

السمة	المعنى
shape	تُرجع أبعاد إطار البيانات.
size	تُرجع العدد الإجمالي للعناصر في إطار بيانات (n x m)
dtypes	تُرجع نوع القيمة لكل عمود.
columns	تُرجع أسماء أعمدة إطار البيانات.
axes	تُرجع عدد الصفوف وأسماء الأعمدة.

```
# Printing the table dimensions
data.shape
```

```
(5597, 6)
```

```
# Return the total number of elements in the array (n x m)
data.size
```

```
33582
```

```
# Return the type of the value of each column
data.dtypes
```

يمكنك إضافة تعليقات إلى مقطعك البرمجي باستخدام (#) في بداية العبارة.

في مكتبة بانداس، عادة ما يكون نوع بيانات الكائن سلسلة نصية string.data.

الجدول 3.7: أنواع بيانات بانداس

نوع بيانات بانداس	نوع بيانات البايثون
object	str or mixed
int64	int
float64	float
bool	bool
datetime64	NA
timedelta[ns]	NA
category	NA

الفهرسة Indexing

الفهرس (Index) هو قائمة بالأعداد الصحيحة أو التسميات التي تستخدمها لتحديد الصفوف أو الأعمدة بشكل فريد. في باندا، تتضمن الفهرسة بشكل أساسي اختيار صفوف وأعمدة محددة من البيانات من إطار البيانات، حيث يمكن اختيار جميع الصفوف وبعض الأعمدة، أو اختيار بعض الصفوف وجميع الأعمدة، أو بعض من كل صف وعمود. اختيار **المجموعة الفرعية** (Subset Selection) هو مصطلح آخر للفهرسة.

لتشاهد بعض الأمثلة على الوظائف التي يمكنك استخدامها للفهرسة.

الجدول 3.8: وظائف الفهرسة

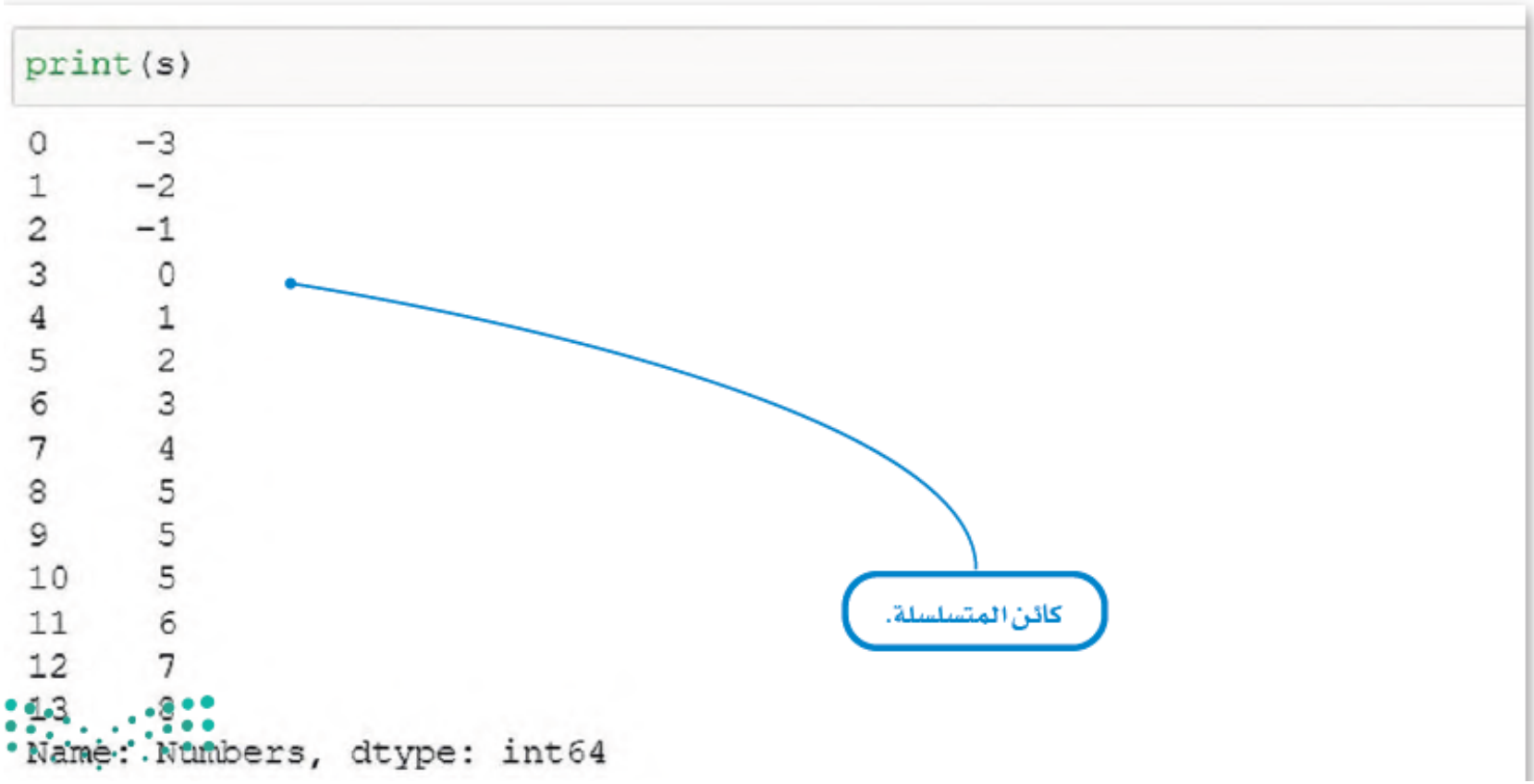
الوظيفة	المعنى
head()	تُرجع العناصر الأولى من الكائن.
tail()	تُرجع العناصر الأخيرة من الكائن.
value_counts()	تُرجع القيم الفريدة للكائن.
idxmax()	تُرجع قيمة فهرس العنصر الأقصى.
idxmin()	تُرجع قيمة فهرس العنصر الأدنى.

استخدام الفهرسة في كائن المتسلسلة Using Indexing in a Series Object

طبّق وظائف الفهرسة هذه على كائن المتسلسلة الذي قمت بإنشائه. اطبع كائن المتسلسلة أولًا، لتذكر محتوياته.

```
print(s)
```

```
0    -3
1    -2
2    -1
3     0
4     1
5     2
6     3
7     4
8     5
9     5
10    5
11    6
12    7
13    8
Name: Numbers, dtype: int64
```



```
x=4
print("the value of the index",x, "is",s[x])
```

```
the value of the index 4 is 1
```

```
# Return the first 2 rows of the series
s.head(2)
```

```
0    -3
1    -2
Name: Numbers, dtype: int64
```

```
# Return the last rows of the series
s.tail()
```

```
9      5
10     5
11     6
12     7
13     8
Name: Numbers, dtype: int64
```

```
# Return a count of the unique values of the series
s.value_counts()
```

```
 5     3
-3     1
-2     1
-1     1
 0     1
 1     1
 2     1
 3     1
 4     1
 6     1
 7     1
 8     1
Name: Numbers, dtype: int64
```

القيمة الافتراضية لعدد الصفوف للوظيفتين ()head و()tail هي 5 لكل من المتسلسلة وإطار البيانات.

Using Indexing in DataFrame Object استخدام الفهرسة في كائن إطار البيانات

```
# Printing the first 10 rows of the table
data.head(10)
```

```
data.tail()
```

```
# Acessing the dataframe attribute 'columns' to print the names of
# the table's columns
for col in data.columns:
    print(col)
```

```
data.describe()
```

تستخدم وظيفة describe() لعرض بعض التفاصيل الإحصائية الأساسية.

	مجموع الطلبة	مجموع المعلمين	مجموع الإداريين
count	5597.000000	5597.000000	5597.000000
mean	1110.124352	89.510988	19.459175
std	2950.990275	192.359535	66.800341
min	0.000000	0.000000	0.000000
25%	31.000000	4.000000	0.000000
50%	136.000000	17.000000	1.000000
75%	808.000000	82.000000	10.000000
max	36416.000000	2090.000000	1152.000000

الشكل 3.19: استخدام الفهرسة في كائن إطار البيانات

تصفية البيانات أو اختيار مجموعة بيانات جزئية
Filtering Data or Subset Selection

في بعض الأحيان لا تحتاج إلى مجموعة البيانات بأكملها. تحتاج إلى عزل بعض البيانات المحددة. للقيام بذلك، تحتاج إلى إضافة بعض المرشحات. هناك العديد من الأساليب لاختيار مجموعة جزئية من إطار البيانات أو متسلسة. الأسلوب الأسهل هو استخدام الفهرسة المنطقية (Boolean Indexing)، ولكن الأسلوب الأكثر قوة هو باستخدام طرق loc و iloc. أولاً ستتعلم الفهرسة المنطقية ثم أسلوب loc و iloc.

تصفية البيانات:

تصفية البيانات هو عملية اختيار جزء أصغر من مجموعة البيانات الخاصة بك واستخدام تلك المجموعة الجزئية للعرض أو التحليل.

الفهرسة المنطقية Boolean Indexing

هي نوع من الفهرسة التي تستخدم القيم الفعلية لمجموعة البيانات، وفيها تحتاج إلى استخدام **المُعامِلات المنطقية** (Boolean Operator)، وتُكتب المُعامِلات المنطقية بشكل مختلف في مفكرة جوبيتر عن بايثون.

الجدول 3.9: المُعامِلات المنطقية في مفكرة جوبيتر

جوبيتر	بايثون
&	and
|	or
~	not

لتشاهد بعض الأمثلة على كائن المتسلسلة.

```
# Return the elements of the series that satisfy the expression s>0
s[s > 0]
```

```
4     1
5     2
6     3
7     4
8     5
9     5
10    5
11    6
12    7
13    8
Name: Numbers, dtype: int64
```

```
s[(s < -1) | (s > 6)]
```

```
0    -3
1    -2
12    7
13    8
Name: Numbers, dtype: int64
```

```
# Printing not(s<0) => (s>=0)
s[~(s < 0)]
```

```
3     0
4     1
5     2
6     3
7     4
8     5
9     5
10    5
11    6
12    7
13    8
Name: Numbers, dtype: int64
```

الشكل 3.20: تصفية البيانات في الكائنات المتسلسلة

الفهرسة مع أسلوب Loc و Iloc Indexing with Loc and Iloc Methods

تُعد طريقتي iloc و loc ضمن الطرق الأكثر شيوعًا للفهرسة في مكتبة بانداس.

- **loc**: يختار الصفوف والأعمدة مع مسميات محددة (أسماء الأعمدة).
- **iloc**: يختار الصفوف والأعمدة في مواضع الأعداد الصحيحة المحددة (أرقام الصفوف والأعمدة).

وإليك أدناه بعض الأمثلة على أستخدام كائن إطار البيانات بأسلوب ()loc.

في هذا المثال، ستستخدم طريقة ()loc لطباعة الصفوف الخمسة الأولى من عمودين محددين.

```
# Choosing the first 5 rows of the columns 'المنطقة الإدارية' and 'المرحلة'
data.loc[:4,['المرحلة','المنطقة الإدارية']]
```

في هذا المثال، ستنشئ إطار بيانات جديدًا يسمى studentsReg. وسيحتوي إطار البيانات هذا على عمودين: عمود واحد للمنطقة وآخر لعدد الطلبة.

```
# Create a dataframe called studentsReg with two columns Region and Number of Students
studentsReg = data.loc[:,['المنطقة الإدارية','مجموع الطلبة']]
studentsReg
```

المنطقة الإدارية	مجموع الطلبة

والآن، سوف تستخدم طريقة iloc() لتحديد جميع عناصر الصف الأول من إطار البيانات.

تذكر، الفهرسة في بايثون تبدأ من 0.

```
# Print all the elements from the [row] of the table
data.iloc[0]
```

```
المنطقة الإدارية                الرياض
المرحلة                      التعليم المستمر
نوع المدرسة                    تعليم الكبار
مجموع الطلبة                          826
مجموع المعلمين                         0
مجموع الإداريين                        0
Name: 0, dtype: object
```

الشكل 3.24: طباعة عناصر الصف الأول من إطار البيانات

وفي هذا المثال، سوف تستخدم حلقة **for** لطباعة الصفوف العشرة الأولى من العمود الأول من إطار بيانات studentsReg.

```
for i in range (10):
    print(studentsReg.iloc[i][1])
```

```
826
1040
190
34668
285
71
183
16018
548
63
```

الشكل 3.26: العناصر المطبوعة لإطار البيانات

المجموعات والتجميع Grouping and Aggregating

تسمى عملية وضع عناصر مجموعة البيانات في مجموعات بناءً على بعض المعايير وتطبيق الوظائف على هذه المجموعات بالتجميع. في مكتبة بانداس؛ يتم تنفيذ هذا الإجراء باستخدام وظيفة (df.groupby()).

فعلى سبيل المثال، تخيل أن لديك مجموعة بيانات لأفضل هدّافي كرة السلة في كل العصور. إذا كنت ترغب في معرفة عدد اللاعبين في مجموعة البيانات هذه لفريق معين، فيمكنك تجميع هذه البيانات حسب عمود "الفريق" وتطبيق دالة المجموع (sum()) على البيانات المجمّعة.

دالة التجميع:

دالة تقوم بحسابات رياضية مع قيم صفوف متعددة والتي يتم تجميعها معًا، ونتيجة لذلك ترجع قيمة موجزة واحدة.

دوال التجميع الأكثر شيوعًا هي sum، count، max، min and mean.

وظيفة Groupby
Groupby Method

باستخدام وظيفة groupby() يمكنك تقسيم بياناتك إلى مجموعات مختلفة، ويمكن أن يساعدك هذا في إجراء حسابات لتحليل البيانات بشكل أفضل.

الجدول 3.10: الدوال التجميعية

الدالة	المعنى
sum	تُرجع مجموع قائمة الأرقام.
max	تُرجع العدد الأقصى لقائمة الأرقام.
min	تُرجع العدد الأدنى لقائمة الأرقام.
mean	تُرجع متوسط قائمة الأرقام.

تنظيف البيانات Data Cleaning

من المهم جدًا أن تكون البيانات التي ستحللها صحيحة ، قبل البدء بتحليلها ، وهذا يعني أنه يجب إزالة البيانات المكررة أو المشوّشة أو غير الدقيقة من مجموعة البيانات الخاصة بك، وإذا بقيت هذه البيانات كما هي، فلن تكون نتائج تحليلها صحيحة.

تنظيف البيانات:

تنظيف البيانات هو عملية إصلاح أو إزالة البيانات غير الصحيحة أو المشوّشة أو المنسقة بشكل غير صحيح أو المكررة أو غير المكتملة من مجموعة البيانات.

الشكل 3.30: عملية تنظيف البيانات

الجدول 3.11: وظائف تنظيف البيانات

الوظيفة	المعنى
duplicated()	تُرجع قيمة منطقية لكل صف يحتوي على بيانات مكررة.
value_counts()	تُرجع القيم الفريدة في مجموعة البيانات.
isnull()	تُرجع قيمة منطقية لكل خلية فارغة من مجموعة البيانات.
dropna()	يحذف الصفوف الفارغة.

البيانات المكررة Duplicated Data

للتحقق مما إذا كانت مجموعة البيانات الخاصة بك تحتوي على بيانات مكررة، فيمكنك أن تستخدم الوظيفة df.duplicated(). وتعطي هذه الوظيفة قيمة منطقية لكل صف حسب احتواءه على بيانات مكررة.

> صواب (True) للبيانات المكررة.

> خطأ (False) للبيانات غير المكررة.

سترى كيفية التعامل مع الصفوف المكررة في مجموعة البيانات.

```
dup = data.duplicated()

# To see how many duplicated rows there are in the table
dup.value_counts()

False    5426
True      171
dtype: int64
```

عدد النسخ المكررة

الشكل 3.31: استخدام وظيفة df.duplicated()

يوجد في مجموعة البيانات الخاصة بك 171 صفًا مكررًا.

لحذف هذه الصفوف تستخدم وظيفة drop_duplicates()، حيث تحذف هذه الطريقة الصفوف المكررة.

بعد حذف الصفوف المكررة، عليك تحديث مجموعة البيانات الخاصة بك للتحقق من إزالة الصفوف المكررة.

```
# Now remove duplicated rows from the table
data = data.drop_duplicates()

dup = data.duplicated()

dup.value_counts()

False    5426
dtype: int64
```

لا يوجد
صفوف مكررة.

الشكل 3.32: استخدام وظيفة drop_duplicates()

الخلايا الفارغة Empty Cells

للتحقق مما إذا كانت مجموعة البيانات الخاصة بك بها قيم مفقودة، يمكنك استخدام وظيفة data.isnull()، حيث ترجع قيمة منطقية لكل خلية من مجموعة البيانات:

> صواب (True) للخلايا الفارغة

> خطأ (False) للخلايا الممتلئة

سترى كيف يمكنك عد الخلايا الفارغة في مجموعة البيانات.

في هذا المثال تحسب الخلايا الفارغة لكل عمود.

```
# get the number of empty cells per column
missing_values_count = data.isnull().sum()
missing_values_count
```

```
المنطقة الإدارية    5
المرحلة             6
نوع المدرسة         5
مجموع الطلبة        4
مجموع المعلمين      4
مجموع الإداريين     4
dtype: int64
```

عدد الخلايا الفارغة في كل عمود.

الشكل 3.33: عد الخلايا الفارغة لكل عمود

يمكنك رؤية عدد الخلايا الفارغة في كل عمود.

لحذف هذه الصفوف، تستخدم وظيفة dropna()، وستقوم بحذف الصفوف الفارغة.

بعد حذف الصفوف الفارغة، عليك تحديث مجموعة البيانات الخاصة بك للتحقق من إزالة هذه الصفوف.

```
# Drop the missing values
data = data.dropna()
```

```
missing_values_count = data.isnull().sum()
missing_values_count
```

```
المنطقة الإدارية    0
المرحلة             0
نوع المدرسة         0
مجموع الطلبة        0
مجموع المعلمين      0
مجموع الإداريين     0
dtype: int64
```

لا يوجد خلايا فارغة

الشكل 3.34: حذف الصفوف الفارغة

البيانات الخاطئة Wrong Data

في بعض الأحيان قد تحتوي مجموعة البيانات الخاصة بك على بيانات خاطئة. فعلى سبيل المثال، في مجموعة البيانات الخاصة بك لا يمكنك الحصول على أرقام سالبة في عدد عمود الطلبة، وللتحقق مما إذا كانت مجموعة البيانات الخاصة بك تحتوي على بيانات خاطئة، عليك كتابة مقطع برمجي مخصص على حسب مجموعة البيانات الخاصة بك.

في هذا المثال سوف تتحقق من الأرقام السالبة في أعمدة مجموعة البيانات.

يعتمد نوع البيانات التي يمكن اعتبارها خاطئة على مجموعة البيانات. عليك أن تقرر ماذا تفعل بهذه البيانات الخاطئة، فقد ترغب في حذفها أو استبدالها بقيم أخرى.

```
# Check if there are negative elements in the columns that have numbers
data[data['مجموع الطلبة'] <0].nunique()
```

```
0    المنطقة الإدارية
0              المرحلة
0          نوع المدرسة
3         مجموع الطلبة
3        مجموع المعلمين
3      مجموع الإداريين
dtype: int64
```

```
data[data['مجموع المعلمين'] <0].nunique()
```

```
0    المنطقة الإدارية
0              المرحلة
0          نوع المدرسة
3         مجموع الطلبة
3        مجموع المعلمين
3      مجموع الإداريين
dtype: int64
```

```
data[data['مجموع الإداريين'] <0].nunique()
```

```
0    المنطقة الإدارية
0              المرحلة
0          نوع المدرسة
3         مجموع الطلبة
3        مجموع المعلمين
3      مجموع الإداريين
dtype: int64
```

تصوير البيانات

كما ذكر سابقًا، فإن تصوير البيانات هو التمثيل البياني للمعلومات والبيانات. إن تصوير البيانات يجعلها أيسر فهمًا وتحليلًا. باستخدام العناصر المرئية مثل المخططات والرسوم البيانية والخرائط، فإنك تجعل البيانات أكثر سهولة وفهمًا وقابلية للاستخدام.

في هذا الدرس، ستستخدم مفكرة جوبيتر لتصوير بياناتك. ويدعم جوبيتر تصوير البيانات بالاستعانة بمكتبات البايثون.

أنواع تصوير البيانات
Types of Data Visualization

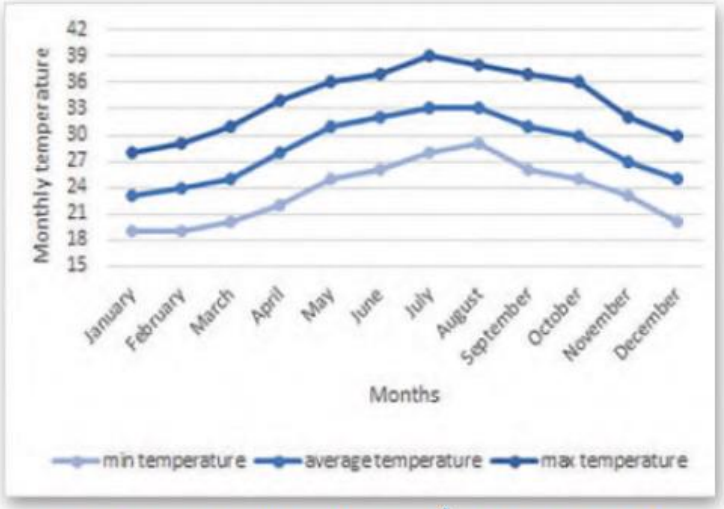
أكثر أنواع تصوير البيانات شيوعًا هي:

- المخططات (الخطية، الشريطية، الدائرية)
- الرسوم البيانية
- المخطط النقطي
- المخطط المدرّج التكراري
- الجداول
- الخرائط

يتم تمثيل البيانات بشكل مختلف باستخدام الأنواع المختلفة لتصوير البيانات. يجب عليك اختيار نوع الرسم البياني حسب ما تريد تحقيقه من تقريرك.

المخطط الخطّي Line Chart

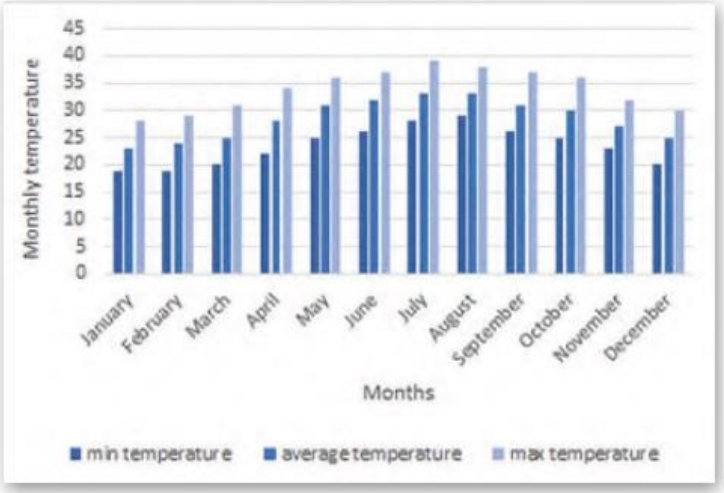
المخطط الخطّي هو تقنية تصوير بيانات، بحيث يتم رسم كل قيمة لمتغير مستقل على مدى فترة زمنية وتتصل هذه القيم بخطوط مستقيمة. عادة ما يكون المحور الأفقي متغيرًا مستمرًا مثل الوقت، والمحور الرأسي هو قيم المتغير المستقل. وتكمن بعض المزايا في بساطته في تمثيل تغيير المتغير بمرور الوقت والذي يمكن أن يساعد في اكتشاف التوجهات والأنماط. ويمكنك رسم خطوط متعددة على نفس الرسم البياني ومقارنة تقدم أكثر من متغير مستقل واحد في نفس الفترة الزمنية.



الشكل 3.37: مخطط خطّي يوضح المتوسط السنوي لدرجات الحرارة المنخفضة والمتوسطة المسجلة في أبها

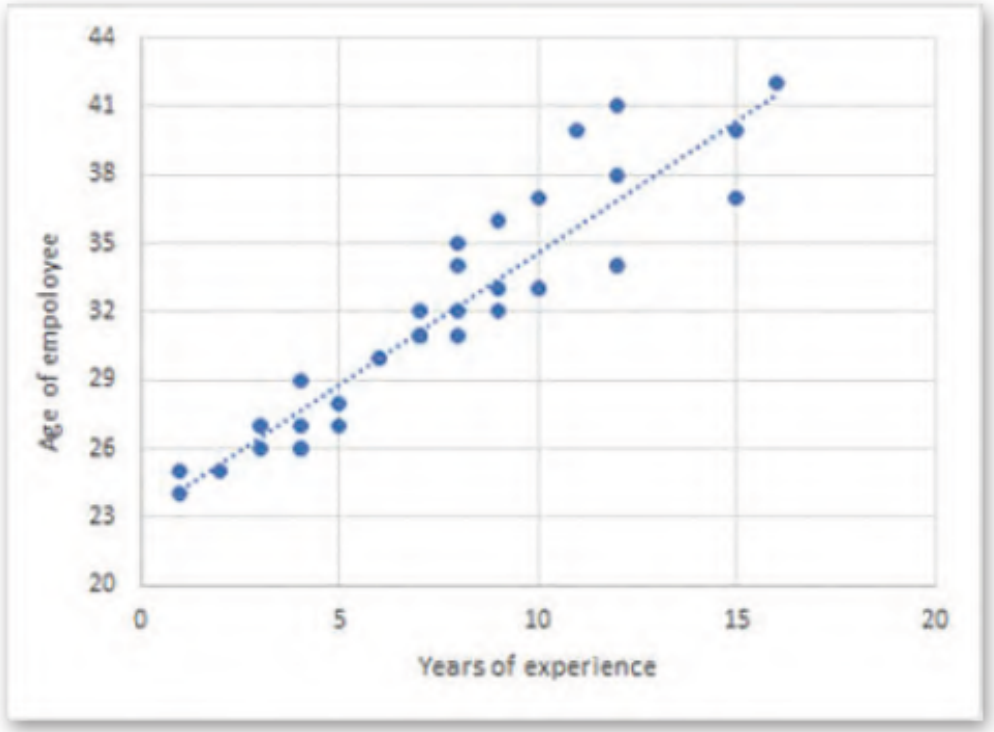
المخطط الشريطي Bar Chart

المخطط الشريطي يمثّل عناصر متغير فئوي على المحور الأفقي (س)، بينما توضح الأعمدة قيم تلك العناصر من خلال ارتفاعها نسبة إلى قيم المحور الرأسي (ص). يمكن أن تكون المخططات الشريطية عمودية أو أفقية، وعادة ما تسمى المخططات الشريطية العمودية مخططات الأعمدة. وهناك العديد من أنواع المخططات الشريطية مثل المخططات الشريطية المجمّعة، والمخططات الشريطية المكدّسة، والمخططات الشريطية مع أشرطة الخطأ، وغيرها المزيد.



المخطط النقطي Scatter Plot

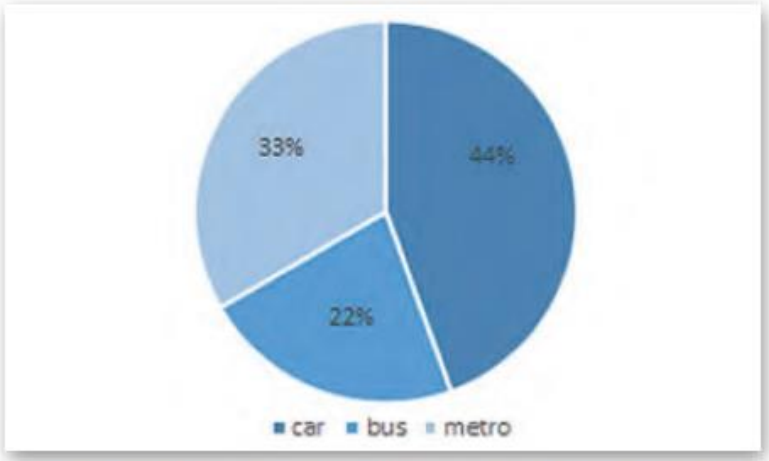
المخطط النقطي هو طريقة لتصوير البيانات باستخدام النقاط لتمثيل قيم المتغيرات المختلفة، وتكون هذه النقاط مبعثرة على الشكل، ومن هنا جاء الاسم. موقع هذه النقط على محوري (س) و (ص) يمثل قيمها، ويمكنك استخدام ألوان مختلفة لرسم النقاط، حيث يمثل كل لون متغير معين. وعندما تكون قيم المتغيرات التي تمت دراستها بيانات متقطعة، فإن المخطط النقطي يكون أكثر ملاءمة من المخطط الخطّي، حيث أنه أكثر قابلية للتطبيق لتمثيل المتغيرات ذات القيم المستمرة (الحقيقية). وهناك أنواع مختلفة من المخطط النقطي بناءً على الارتباط بين المتغيرات (ايجابي، سلبي، لاغي).



الشكل 3.39: مخطط نقطي يبين وجود ارتباط إيجابي بين سنوات الخبرة وعمر الموظف

المخطط الدائري Pie Chart

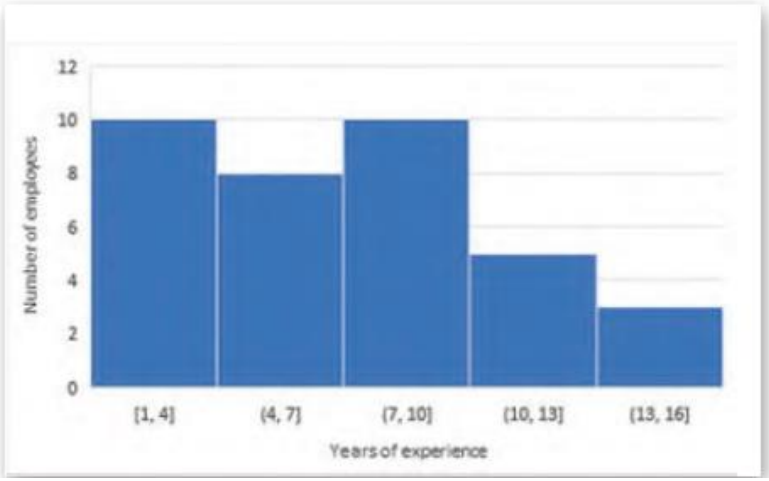
المخطط الدائري هو مخطط يشبه الفطائر، مقسّم إلى شرائح تمثل القيم النسبية لبعض المتغيرات في فئة معينة. تمثل كل شريحة من المخطط فئة مختلفة. هناك العديد من أنواع المخططات الدائرية، مثل **المخططات الدائرية المجوّفة** (Doughnut Charts) و**المخططات نصف المجوّفة** (Half-Doughnut Pie Charts) و**المخططات الدائرية متعددة الطبقات** (Multilayered Pie Charts).



الشكل 3.40:مخطط دائري يبين النسبة المئوية لوسيلة النقل المفضلة

المخطط المدرّج التكراري Histogram

يعد المخطط المدرّج التكراري (الهيستوقرام) أحد أقدم تقنيات تصوير البيانات، حيث يشبه المخططات الشريطية ولكنه يختلف عنها في أنه يظهر تواتر البيانات العددية، بينما المخططات الشريطية تُعد طريقة لمقارنة فئات البيانات. وعندما تريد إنشاء مخطط المدرّج التكراري، فعليك بتجميع البيانات في نطاقات يتم رسمها بعد ذلك على شكل أعمدة متصلة ببعضها البعض، ويُظهر ارتفاع الأعمدة عدد البيانات الموجودة في كل نطاق.



البيانات الفئوية هي متغيرات متقطعة، ويمكن أن يكون لها عدد معين من القيم، فعلى سبيل المثال عدد الطلبة في كل منطقة من المملكة العربية السعودية. ويمكن أن يكون للبيانات المستمرة أي قيمة بين الحد الأدنى والقيمة القصوى، على سبيل المثال، الوقت أو درجة الحرارة.

مكتبة مات بلوت ليب Matplotlib Library

من أجل تصوير بياناتك، تحتاج إلى استيراد مكتبة جديدة، وهي التي تسمى مات بلوت ليب. وتحتوي هذه المكتبة على بعض الأساليب الجاهزة التي يمكنك استخدامها لجعل المخطط الخاص بك أكثر قابلية للفهم، ويمكنك الاطلاع على هذه الأساليب في الجدول 3.12. وباستخدام هذه المكتبة، يمكنك تقديم بياناتك في أي مخطط تريده. في هذا الدرس، ستستخدم هذه الأساليب لإنشاء مخططات بناءً على إطار البيانات الخاص بك.

لدعم النص العربي داخل المخططات التي أنشأتها مكتبة مات بلوت ليب، تحتاج إلى تحويل النص العربي إلى تنسيق يمكن عرضه بشكل صحيح. ستستخدم مكتبتي البايثون:

arabic_reshaper <

bidi.algorithm <

من خلال تشغيل المقطع البرمجي التالي في مفكرة جوبيتر الخاصة بك، يتم تنزيل هاتين المكتبتين وتثبيتهما تلقائيًا.

```
!pip install arabic-reshaper
!pip install python-bidi
```

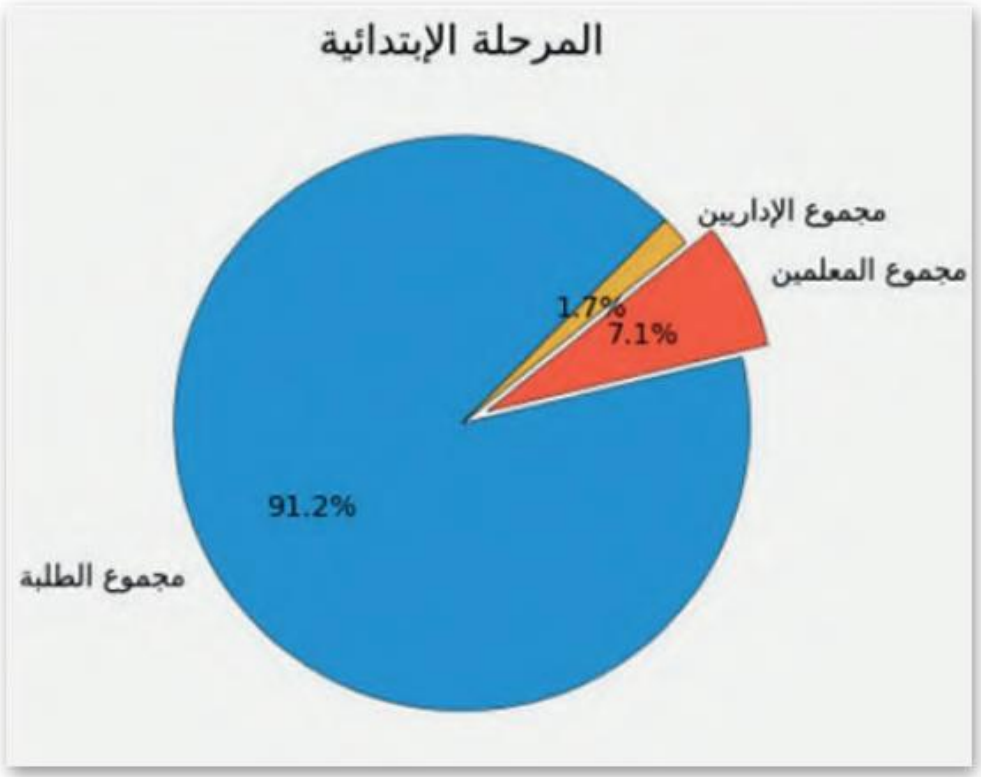
الجدول 3.12: طرق مكتبة مات بلوت ليب (Matplotlib)

الطريقة	المعنى
bar()	ينشئ مخطط شريطي.
pie()	ينشئ مخطط دائري.
set_title()	يحدد عنوان المخطط.
set_ylabel()	يحدد تسمية محور y.
set_xlabel()	يحدد تسمية محور x.
show()	ينشئ المخطط.

المخطط الدائري Pie Chart

سترى خطوات إنشاء المخطط الدائري في مفكرة جوبيتر.

ستنشئ إطار بيانات جديد يسمى groupsP من مجموعة البيانات التي استخدمتها في الدرس السابق. قم بتجميع بياناتك حسب المرحلة واحصل على **المتوسط** (mean()) للطلبة والمعلمين والإداريين، ثم قم بفرز إطار البيانات هذا بمتوسط عدد المسؤولين.



الشكل 3.50:مخطط دائري

نمذجة البيانات التنبؤية

تلجأ المؤسسات والشركات لاستخدام النمذجة التنبؤية لتحليل الأحداث المستقبلية المتعلقة بنشاطها التجاري، وذلك بهدَف اتخاذ أفضل القرارات. ويُمكِن استخدام نماذج التّنبؤ لفهم ومعرفة شرائح وفئات المستهلكين، ولتقدير المبيعات المُحتملة، أو لفهم ومعرفة القضايا الأمنية للحسابات.

ما هي النمذجة التنبؤية؟ What is Predictive Modeling?

تُعتبر التحليلات التنبؤية فرعًا من فروع علم تحليل البيانات المتقدم، وتَستعين هذه التحليلات بالبيانات السابقة، إلى جانب طُرق أخرى كالنمذجة الإحصائية، وتنقيب البيانات، وتعلُّم الآلة، وذلك لتقديم التنبؤات حول النتائج المُستقبلية لقرارات أو لعمليات مُعينة تقوم بها الشركات أو المؤسسات. وتستخدم الشركات والمؤسسات التحليلات التنبؤية للتعرُّف على أنماط مُعينة في هذه البيانات يُمكِن من خلالها تحديد الفُرص والمخاطر. فعلى سبيل المثال، تَجمعُ خدمة الأرصاد الجوية البيانات بشكلٍ يومي عن المتغيرات المختلفة المتعلقة بحالة الطقس مثل درجات الحرارة والرطوبة وغيرها، مما يُمكّنُها من التنبؤ بحالة الطقس في الأيام القادمة.

تُستَخدم التحليلات التنبؤية على نطاقٍ واسع في مجال الرعاية الصحية وذلك بِهدف تحسين طُرق تشخيص وعلاج المرضى المصابين بالأمراض المُزمِنة، وتستخدم إدارات الموارد البشرية والشركات نماذج التنبؤ في تحسين عمليات اختيار وتعيين الموظفين، وأما البنوك فتستخدمها بشكلٍ واسع للكشف عن عمليات الاحتيال.

النمذجة التنبؤية (Predictive Modeling):
هي أسلوبٌ إحصائي تُستَخدمُ فيه النتائج والبيانات السابقة للتنبؤ بالأحداث أو النتائج المستقبلية.

فئات النمذجة التنبؤية Predictive Modeling Categories

تتمثل مُهمة المُتعلِّم في النمذجة التنبؤية بالوصول إلى الدالة أو العلاقة الوظيفية التي تربط متغيرات الإدخال بالمخرجات (التنبؤات) في **بيانات التدريب** (Training Data)، وذلك بصرف النظر عن طبيعة ومُعاملات تلك الدالة.

بمجرد الوصول إلى هذه العلاقة الوظيفية، يُمكن استخدامها للتنبؤ بقيَم المُخرجات بناءً على متغيرات الإدخال المختلفة. وتصنف النماذج التنبؤية إلى فئتين: فئة تحتوي على عدد محدد من المُعاملات وتسمى **بالنموذج المُعامَلي** (Parametric Model)، وفئة لا تحتوي على عدد محدد من المُعامِلات، ويطلق عليها تسمية **النموذج غير المُعامِلي** (Non-Parametric Model).

1. النماذج المُعاملية Parametric Models

تعتبر الافتراضات جزءًا أساسيًا من أي نموذج من نماذج البيانات، فهي تُحسِّن التنبؤات وتجعل النموذج أسهل للفهم. يَضعُ النموذج المُعامَلي افتراضات محددة حول شكل الدالة التي سيتم تعيينها، ويفترض مجموعة محددة مُسبقًا من المُعامِلات، وذلك بشكل مستقل عن تلك الموجودة في أمثلة التدريب، وهكذا فإن النموذج المُعامِلي يقوم بتلخيص بيانات التدريب من خلال هذه المجموعة من المُعامِلات.

2. النماذج غير المُعاملية Non-Parametric Models

إن نماذج تَعلُّم الآلة غير المُعامِلية ليست مَعنيَّة بتكوين الافتراضات حول **دالة التعيين** (Mapping Function)، فيمكن لمثل هذه النماذج مثلًا تقدير طبيعة العلاقة الوظيفية من خلال بيانات التدريب. وتُعدُّ هذه النماذج خيارًا ممتازًا لتحليل الكميات الكبيرة من البيانات بدون أي معرفة سابقة عنها.

المُعامِل (Parameter):
يمكن وصف المُعامِل بأنه متغير جوهري وأساسي في تكوين النموذج.

يعتمد المتخصصون في عمل تحليلات النماذج التنبؤية على البيانات من المصادر التالية:

بيانات عملياتية (Transactional Data).
بيانات العملاء (Customer Data).
البيانات الطبية (Medical Data).
البيانات المالية (Financial Data).
المعلومات الديموغرافية (Demographic Data).
البيانات الجغرافية (Geographic Data).
بيانات التسويق الرقمي (Digital Marketing Data).
إحصائيات الويب (Web Traffic Statistics).

إحصائيات الويب (Web Traffic Statistics).

الجدول 4.1: مقارنة بين النماذج المُعامِلية وغير المُعامِلية

المعيار	النماذج المُعامِلية	النماذج غير المُعامِلية
بيانات التدريب	تتطلب بيانات تدريب أقل من النماذج غير المُعامِلية.	تتطلب بيانات أكثر بكثير من النماذج المُعامِلية لتقدير العلاقة أو دالة التعيين.
سرعة التدريب	أسرع إنجازًا من الناحية الحسابية، ويُمكن تدريبها بشكل أسرع لوجود مُعامِلات محدودة للتدريب.	تستغرق وقتًا أطول للتدريب، حيث تتضمن تحليل علاقات أكثر تعقيدًا يتم تقديرها أثناء عملية التدريب.
الملاءمة	قد لا تُقدّم هذه النماذج أفضل ملاءمة للبيانات، ومن المستبعد أن تتطابق تمامًا مع دالة التعيين.	تُوفّر هذه النماذج تنبؤات أكثر دقة من النماذج المُعامِلية من حيث ملاءمة البيانات، ولكن الخوارزميات في هذه النماذج تكون أكثر عرضة لمشكلة فرط التخصيص (Overfitting).
التعقيد	تتميز إجراءاتُها بسهولة فهمها وتفسيرها.	إجراءاتُها أكثر تعقيدًا وصعوبة سواء من ناحية إمكانية التفسير أو الفهم.

مهام النمذجة التنبؤية Predictive Modeling Tasks

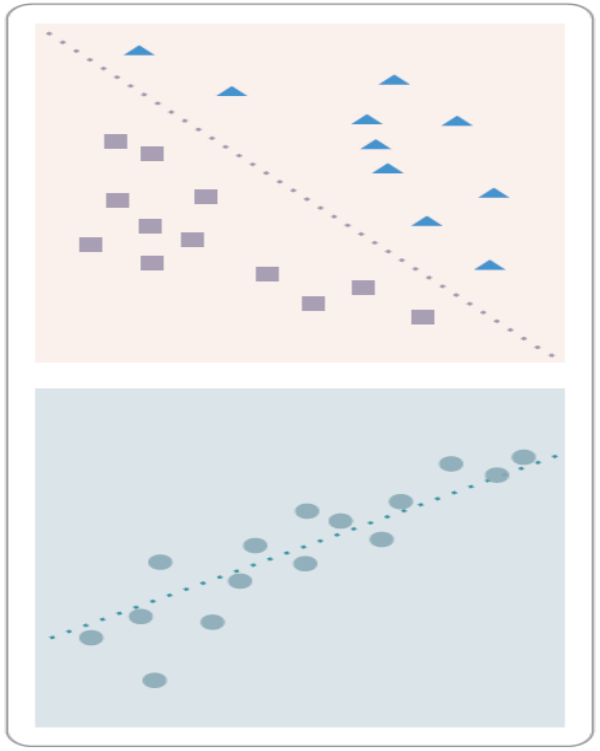
تُعدُّ نماذج **التصنيف** (Classification) و**الانحدار** (Regression) من أهم وأكثر النماذج استخدامًا في مهام النمذجة التنبؤية.

1. التصنيف Classification

يعتمد نموذج التصنيف على عملية تقييم المتغيرات المُدخلة ثم تصنيفها ضمن مجموعات لتكوين بيانات المُخرجات، وبذلك فإن المتغير الذي سيتم توقُّعه ستكون له قيمًا متقطّعة (Discrete)، وقد تكون هذه القيم ببساطة مجرد إجابة لسؤال معين بـ "نعم" أو "لا". ويُستخدم نموذج التصنيف في تقييم عمليات التمويل والبيع بالتجزئة، حيث بمقدوره جمع المعلومات بسرعة وتصنيفها في مجموعات لتقديم الإجابات عن الأسئلة المتعلقة بتلك العمليات.

2. الانحدار Regression

يعتمد نموذج الانحدار على مبدأ إيجاد علاقات رياضية تربط بين متغيرين، بحيث يُمكن تنبؤ أحدهما من خلال معرفة المتغير الآخر، ويُطلق على المتغير المُدخل اسم **المتغير المستقل** (Independent Variable)، بينما يُطلق على المتغير المُخرج اسم **المتغير التابع** (Dependent Variable)، ويتنبأ هذا النموذج بالقيم المحتملة للمتغيرات التابعة من خلال معالجة قِيم المتغيرات المستقلة. يتم تمثيل هذا النموذج بيانيًا في صورة خط مستقيم (انحدار خطي) يتقارب مع جميع نقاط البيانات المستقلة. ويمكن لنموذج الانحدار على سبيل المثال التنبؤ بمدة بقاء شخص إبان دخول المستشفى، ويمثل عدد الأيام في المستشفى المتغير التابع، أما معدل النبض لذلك الشخص مثلًا فيمثل المتغير المستقل.



شكل 4.2: يوضح الفرق بين التصنيف (الشكل العلوي) والانحدار (الشكل السفلي)، حيث يمثّل التصنيف الخط المُنقط وهو الحد الخطي الفاصل بين فئتين مختلفتين، بينما يُمثل الخط المُنقط في الانحدار العلاقة الخطية بين متغيرين.

الجدول 4.2: مقارنة بين التصنيف والانحدار

التصنيف	الانحدار
التصنيف هو التنبؤ بالمُخرجات لفئة متقطعة بمعنى أن المتغير المُخرج يجب أن يكون عددًا صحيحًا.	الانحدار هو التنبؤ بناتج كمي مستمر بمعنى أن المتغير المُخرج يجب أن يكون قيمة مستمرة أو عددًا حقيقيًا.
تُستخدم خوارزمية التصنيف لتعيين قيمة المُدخل (x) مع المتغير المُخرج ذو القيم المتقطّعة (y).	تُستخدم خوارزمية الانحدار لتعيين قيمة المُدخل (x) مع المتغير المُخرج ذو القيم المستمرة (y).

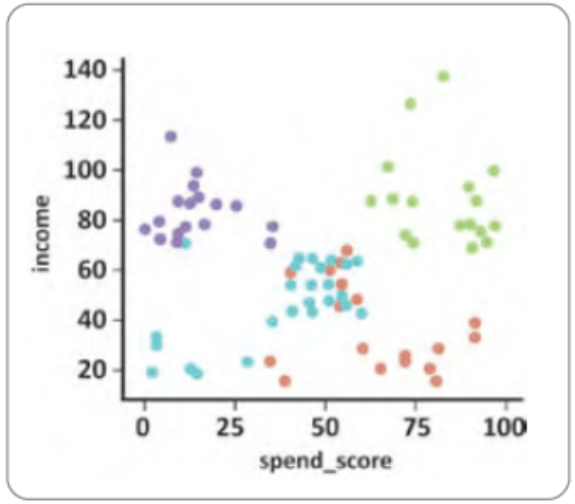
من المهام الشائعة الأخرى للنمذجة التنبؤية:

3. التوقُّع Forecasting

وهو إجراء وتقديم تقديرات رقمية معينة بناءً على تحليل البيانات السابقة والتي يطلق عليها البيانات التاريخية. وتَستخدم شركات الاستثمار التوقُّعات للتنبؤ بأسعار الأسهم في التداولات اليومية أو طويلة الأجل، ويعتبر نموذج التوقُّع من أكثر نماذج التنبؤ شيوعًا حيث يتميز بامكانيات استخدام كثيرة في العديد من المجالات.

4. التجميع Clustering

يُصنِّف نموذج التجميع البيانات إلى مجموعات بناءً على الخصائص المتشابهة بينها، ثم يستخدم بيانات كل **مجموعة** (Cluster) لتحديد النتائج على نطاق واسع لكل مجموعة. وهناك نوعان من طرق التجميع يتم استخدامهما في هذا النموذج: **التجميع الصلب** (Hard Clustering) يعتمد على تصنيف البيانات إلى مجموعات متميزة، حيث يمكن أن تنتمي كل نقطة بيانات إلى مجموعة واحدة فقط، و**التجميع الناعم** (Soft Clustering) يعتمد على تعيين احتمالات لكل نقطة بيانات، حيث يمكن أن تنتمي نقاط البيانات إلى أكثر من مجموعة واحدة. ويُمكن للشركات استخدام نموذج التجميع لتحديد استراتيجيات التسويق لفئات معينة من المستهلكين.



شكل 4.3: مثال على التجميع لأربع مجموعات بناء على قيمتى الدخل ومعدل الإنفاق

6. السلاسل الزمنية Time Series

تَستخدِم نماذج السلاسل الزمنية قِيمَ البيانات المتوفرة سابقًا ضمن تسلسل زمني مُحدد كعوامل الإدخال في مجموعة البيانات؛ وذلك من أجل التنبؤ بقيم جديدة أو أحداث مستقبلية، ويمكن لهذه النماذج تقديم التوقُّعات المستقبلية لاتجاهات أو أحداث فريدة أو متعددة. يمكن لنماذج السَّلاسل الزمنية أيضًا تحليل تأثير العوامل الخارجية كتلك الموسمية والعارضة (غير المتوقّعة) التي قد تحدث على القيم والاتجاهات المستقبلية، على سبيل المثال يمكن لشركة صناعات إلكترونية استخدام نموذج السلاسل الزمنية لتحليل الوقت المطلوب لمعالجة الطلبيات على مدار العام الماضي، وبالتالي يمكن للنموذج التنبؤ بمتوسط وقت المعالجة الشهري.

تُستخدم طرق أخرى للنمذجة التنبؤية في المسائل الأكثر تعقيدًا.

مِن طُرق النمذجة التنبؤية:

أشجار القرار (Decision Trees).
التعزيز الاشتقاقي (Gradient Boosting).
النماذج الخطية العامة (General linear Models).
الشبكات العصبية (Neural Networks).
نماذج بروفيت (Prophet Models).

عملية النمذجة التنبؤية The Predictive Modeling Process

يمكن تعريف النمذجة التنبؤية ببساطة على أنها عملية تنفيذ خوارزميات على مجموعات من البيانات لإنشاء التنبؤات، ويتم في هذه العملية إنشاء نموذج وتدريبه، ثم التحقق من صِحته وإدخال التحسينات عليه عند الحاجة، للحصول على المعلومات المناسبة التي تُلبي احتياجات المؤسسة. وتتكون الخطوات الأساسية لإجراء النمذجة التنبؤية بشكل نموذجي من:

1. جمع البيانات وتنظيفها Data collection and cleaning

إن من المهم القيام بجمع البيانات من جميع المصادر المتوفرة بهدف استخراج المعلومات اللازمة لعملية النمذجة، وبعد ذلك تتم عملية تنظيفها من الشوائب والقيم الشاذة للحصول على تقديرات دقيقة. وتُطبَّق هذه الخطوة على: البيانات المختلفة مثل عمليات البيع والشراء والاستبانات الخاصة بالعملاء، والبيانات الإحصائية الخاصة بالاقتصاد والمسح السكاني، والبيانات التي يتم جمعها بشكلٍ آلي عبر الويب ومن خلال الأجهزة المختلفة وغير ذلك.

2. تحويل البيانات Data transformation

تتم عملية تحويل البيانات بتوحيد بُنية وصياغة البيانات باستخدام عمليات معالجة دقيقة للحصول على البيانات في صورتها النهائية، وتشمل هذه العملية تحديد نطاقات معينة لقيم البيانات وإزالة القيم الغريبة والبيانات الشاذة من خلال **تحليل الارتباط** (Correlation Analysis).

3. صياغة النموذج التنبؤي Formulation of the Predictive Model

تتضمن عملية صياغة النموذج التنبؤي القيام بتحديد طرق التنبؤ المناسبة حسب الحاجة، فيمكن مثلًا استخدام شجرة القرار في عملية التصنيف، بينما يجب استخدام نموذج التعزيز الاشتقاقي حين تكون المهمة تتعلق بالانحدار. ويتم أثناء هذه العملية تحديد بيانات التدريب والاختبار في النموذج، حيث يتم تدريب خوارزمية الإجراء المحدد باستخدام بيانات التدريب المتاحة، ثم يتم تطبيق النموذج الناتج على البيانات لاختبارها وتحديد أداء النموذج.

4. الاستنتاجات أو الاستدلالات Inferences or conclusions

في النهاية يتم استخراج الاستدلالات واستخلاص الاستنتاجات من النموذج، والتي تُساعد في الإجابة على أسئلة الأعمال.

شكل 4.4: مخطط عملية النمذجة التنبؤية

Benefits and Limitations of Predictive Modeling ميزات وتحديات النمذجة التنبؤية

ميزات النمذجة التنبؤية :

تحسين إستراتيجيات التسويق والمبيعات وخدمة العملاء.
تحسين التنافسية المبنية على المعرفة وتوظيف الإستراتيجيات لاكتساب ميزة المنافسة.
تعزيز جودة المنتجات والخدمات.
التحليل الدقيق لمتطلبات المستهلك.
توفير التوقُّعات للعوامل الخارجية التي تؤثر على الإنتاجية أو سير العمل.
المساهمة في إدارة المخاطر المالية والاستثمارية.
توفير التنبؤ بالموارد أو بالمخزون من المواد المختلفة.
التنبؤ بالتوجهات المستقبلية للأعمال.
دعم عملية إدارة القوى العاملة وتحليل المشاكل المتعلقة بها.

تحديات النمذجة التنبؤية :

أمن وخصوصية البيانات.
التعامل مع حجم كبير من البيانات.
تحديات إدارة البيانات.
الحاجة المستمرة لتكييف النماذج مع القضايا والمشاكل المستجدة.

أدوات النمذجة التنبؤية
Predictive Modeling tools

توجد أدوات النمذجة التنبؤية الحديثة على صورة مَنصات متكاملة تدعم تطوير الخوارزميات وتحليل البيانات وتقديم النتائج الموثوقة، ويَتمُّ استخدام هذه الأدوات من قِبَلِ الشركات والمؤسسات البحثية لإخراج استنتاجات دقيقة وشاملة يمكنها المساهمة في اتخاذ القرارات الفعالة.

الأدوات المتاحة:

منصة H2O للذكاء الاصطناعي (H2O Driverless AI).
منصة IBM واتسون ستوديو (IBM Watson Studio).
منصة رابيد ماينر ستوديو (RapidMiner Studio).
منصة ساب للتحليلات السحابية (SAP Analytics Cloud).
منصة ساس (SAS).
منصة IBM الحزمة الإحصائية للعلوم الاجتماعية (IBM SPSS).
منصة أوراكل لعلم البيانات (Oracle Data Science).

جدول 4.3: تطبيقات النمذجة التنبؤية

التطبيق	الوصف
المبيعات	يمكن أن يساهم التحليل التنبؤي في تحديد مكانة الشركة المالية من حيث المبيعات والأرباح، فمن خلال الكشف عن الحالات الشاذة والتباين في البيانات المالية السابقة للأقسام المختلفة في الشركة، يمكن للنمذجة تحديد الأقسام ذات الأداء المنخفض مثل قسم المبيعات، وهذا يؤدي إلى تحسين أداء الشركة وإدخال التحسينات على الأقسام أو العمليات بما يتناسب مع إستراتيجيات النمو والأداء المتميز.
التسويق	يمكن للشركات استهداف فئات معينة من العملاء بالحملات الترويجية لمنتجات أو خدمات معينة، وذلك من خلال التحليل والتنبؤ استنادًا إلى البيانات السابقة، كما يمكن لها أيضًا توقُّع استجابات ومتطلبات هؤلاء العملاء، وهنا يكمُن أحد الأسباب الرئيسة في قيام الشركات بجمع البيانات السابقة. تُعدُّ معرفة رغبات العملاء والتنبؤ بالمنتجات والخدمات التي يرغبون بالحصول عليها في المستقبل من أهم إستراتيجيات التسويق الحديثة.
وسائل التواصل الاجتماعي	تُعدُّ وسائل التواصل الاجتماعي مصدرًا أساسيًا للبيانات الضخمة غير المنظَّمة وغير المتجانسة، والتي تتكوَّن من مشاركة ملايين الأشخاص يوميًا في الحديث عن القضايا والمواضيع المختلفة، ويُعدُّ تحليل بياناتها من أكثر التطبيقات استخدامًا للنمذجة التنبؤية، حيث يَسمحُ للمؤسسات والشركات باستكشاف اهتمامات العملاء وبالتالي تطوير خططها المستقبلية وفقًا لذلك.