Machine للـ Python 🐍 Learning

NumPy • Pandas • Matplotlib • Scikit-learn

معالجة البيانات المفقودة • أمثلة عملية شاملة

📐 NumPy: العمل مع المصفوفات

الاستيراد والإنشاء الأساسي

📝 استيراد NumPy

```
import numpy as np
```

إنشاء المصفوفات

🔧 طرق مختلفة لإنشاء Arrays

```
# 1) من قائمة عادية
a = np.array([1, 2, 3])
# 1 متجهD
b = np.array([[1],[2],[3]])                # متجه عمودي 3×1
c = np.array([[1, 2, 3], [4, 5, 6]])      # مصفوفة 2×3

# 2) np.arange(start, stop, step)
d = np.arange(1, 10, 2)                    # [1 3 5 7 9]
e = np.arange(0, 1, 0.1)                   # [0. 0.1 0.2 ... 0.9]
```

```
# 3) np.linspace(start, stop, num_points)
f = np.linspace(1, 10, 5)                  # [1. 3.25 5.5 7.75
10.]

# 4) أصفار وآحاد
g = np.zeros(5)                            # [0. 0. 0. 0. 0.]
h = np.ones((2, 3))                        # [[1. 1. 1.] [1. 1.
1.]]
i = np.eye(3)                              # مصفوفة الهوية 3×3

# 5) أرقام عشوائية
j = np.random.rand(2, 3)                   # قيم عشوائية 0 .1
k = np.random.randint(1, 10, size=(2, 3)) # أرقام عشوائية صحيحة
```

✅ **مثال 1: خصائص المصفوفة**

الكود:

```
c = np.array([[1, 2, 3], [4, 5, 6]])
print(c.dtype)    # int64
print(c.ndim)     # 2
print(c.shape)    # (2, 3)
print(c.size)     # 6
```

الشرح: شكل المصفوفة 2 سطر و 3 أعمدة، إجمالي 6 عناصر

الوصول للعناصر (Indexing & Slicing)

🔍 الوصول والتقطيع

```
c = np.array([[1, 2, 3], [4, 5, 6]])

# الصفوف والأعمدة
print(c[0])          # [1 2 3] - السطر الأول
print(c[:, 0])       # [1 4] - العمود الأول
print(c[1, 2])       # 6 - العنصر في السطر 1 والعمود 2

# التقطيع
```

```
print(c[:, 1:])      # [[2 3] [5 6]] - كل الأعمدة من 1 فما فوق
print(c[::2])        # [1 2 3] - كل سطر ثاني (خطوة 2)
print(c[:, ::2])     # [[1 3] [4 6]] - كل عمود ثاني
```

✅ مثال 2: عمليات أساسية

الكود:

```
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

print(a + b)         # [5 7 9] - جمع عنصر بعنصر
print(a * b)         # [4 10 18] - ضرب عنصر بعنصر
print(np.dot(a, b))  # 32 - الضرب النقطي
print(a.sum())       # 6
print(a.mean())      # 2.0
print(np.sqrt(a))    # [1. 1.41421356 1.73205081]
```

📊 Pandas: DataFrames

📝 الاستيراد

```
import pandas as pd
```

إنشاء DataFrames

🔧 طرق الإنشاء

```
# 1) من قاموس
data = {
    'الاسم': ['أحمد', 'فاطمة', 'محمد'],
    'العمر': [25, 30, 28],
    'الراتب': [3000, 3500, 3200]
```

```
}
df = pd.DataFrame(data)

# 2) من قائمة قواميس
data2 = [
    {'الاسم': 'علي', 'العمر': 35, 'الراتب': 3800},
    {'الاسم': 'سارة', 'العمر': 29, 'الراتب': 3300}
]
df2 = pd.DataFrame(data2)

# 3) من ملف CSV
df = pd.read_csv('data.csv')

# 4) من ملف Excel
df = pd.read_excel('data.xlsx')
```

✅ مثال 3: استكشاف البيانات

الكود:

```
df = pd.DataFrame({
    'الاسم': ['أحمد', 'فاطمة', 'محمد'],
    'العمر': [25, 30, 28]
})

print(df.head())          # أول 5 سطور (هنا 3)
print(df.tail())          # آخر 5 سطور
print(df.shape)           # (3, 2) - 3 أعمدة، 2 سطور
print(df.info())          # معلومات عن كل عمود
print(df.describe())      # إحصائيات العمر (min, max, mean...)
```

النتيجة: معلومات إحصائية شاملة عن البيانات

التعامل مع الأعمدة

🔧 اختيار وتعديل الأعمدة

```
# اختيار أعمدة
print(df['الاسم'])           # Series واحدة
print(df[['العمر' ,'الاسم']]) # DataFrame
بعمودين

# إضافة عمود
df['الراتب'] = [3200 ,3500 ,3000]
df['الراتب_السنوي'] = df['الراتب'] * 12

# حذف عمود
df.drop('الراتب_السنوي', axis=1, inplace=True)
# أو
del df['الراتب_السنوي']

# تعديل قيم
df['العمر'] = df['العمر'] + 1
df.loc[0, 'الاسم'] = 'محمود'  # تعديل قيمة محددة
```

📈 Matplotlib: الرسوم البيانية

📝 الاستيراد

```
import matplotlib.pyplot as plt
```

رسم بسيط

🔧 رسم منحنى أساسي

```
import numpy as np
import matplotlib.pyplot as plt

# توليد البيانات
x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)
```

```
# الطريقة الأولى (سريعة)
plt.figure(figsize=(10, 6))
plt.plot(x, y, 'b-', linewidth=2)
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.title('دالة الجيب')
plt.grid(True)
plt.show()

# الطريقة الثانية (كائنية)
fig, ax = plt.subplots(figsize=(10, 6))
ax.plot(x, y, 'r-', linewidth=2, label='sin(x)')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title('دالة الجيب')
ax.legend()
ax.grid(True)
plt.show()
```

عدة رسوم في شكل واحد

🔧 Subplots متعددة

```
fig, ((ax1, ax2), (ax3, ax4)) =
plt.subplots(2, 2, figsize=(12, 10))

# الشكل الأول
ax1.plot(x, np.sin(x), 'b-')
ax1.set_title('sin(x)')

# الشكل الثاني
ax2.plot(x, np.cos(x), 'r-')
ax2.set_title('cos(x)')

# الشكل الثالث
ax3.scatter(np.random.rand(50), np.random.rand(50), c='green')
ax3.set_title('نقاط عشوائية')

# الشكل الرابع
ax4.bar(['محمد' ,'فاطمة' ,'أحمد'], [28 ,30 ,25])
```

```
ax4.set_title('الأعمار')

plt.tight_layout()
plt.show()
```

✅ **مثال 4: رسم بيانات من DataFrame**

الكود:

```
df = pd.DataFrame({
    'الشهر': ['يناير', 'فبراير', 'مارس', 'أبريل'],
    'المبيعات': [1000, 1200, 1500, 1800]
})

# رسم المبيعات
plt.figure(figsize=(10, 6))
plt.plot(df['الشهر'], df['المبيعات'], marker='o',
linewidth=2)
plt.ylabel('المبيعات')
plt.title('المبيعات حسب الشهر')
plt.grid(True)
plt.show()
```

⚠️ معالجة البيانات المفقودة

أنواع البيانات المفقودة

🔴 **مشكلة شائعة:** البيانات المفقودة قد تكون NULL, NaN, None, أو قيم فارغة

🔍 اكتشاف البيانات المفقودة

```
import pandas as pd
import numpy as np

# إنشاء DataFrame مع بيانات مفقودة
df = pd.DataFrame({
    'الاسم': ['محمد' ,'فاطمة' ,'أحمد', None],
    'العمر': [25, np.nan, 28, 35],
    'الراتب': [3500 ,3000, None, 3800]
})

print(df)
print("\n--- اكتشاف القيم المفقودة --  ")
print(df.isnull())          # True للقيم المفقودة
print(df.notnull())         # True للقيم الموجودة
print(df.isnull().sum())    # عدد القيم المفقودة لكل عمود
print(df.isnull().sum().sum())  # العدد الكلي
```

طرق التعامل مع البيانات المفقودة

💡 ملخص الخيارات:

- **الحذف:** حذف الصفوف/الأعمدة بقيم مفقودة
- **الملء:** استبدال بمتوسط، وسيط، أو قيمة ثابتة
- **الاستيفاء:** استخدام قيم مجاورة
- **التنبؤ:** استخدام نماذج للتنبؤ بالقيم

1️⃣ الحذف (Dropping)

🗑️ حذف الصفوف/الأعمدة المفقودة

```
# حذف الصفوف بأي قيمة مفقودة
df_dropped = df.dropna()
print(df_dropped)  # يبقى السطر الأول فقط

# حذف صفوف بقيم مفقودة في عمود معين
df_dropped = df.dropna(subset=['العمر'])
```

```
# حذف أعمدة بقيم مفقودة
df_dropped = df.dropna(axis=1)  # يحذف عمود الراتب كله

# حذف إذا كانت جميع القيم مفقودة
df_dropped = df.dropna(how='all')

# 50 حذف إذا كانت نسبة القيم المفقودة أكثر من%
threshold = len(df) * 0.5
df_dropped = df.dropna(thresh=threshold)
```

✅ مثال 5: الحذف العملي

الكود:

```
df_original = df.copy()
df_clean = df.dropna()

print(f"الأصلي: {len(df_original)} سطر")
print(f"بعد الحذف: {len(df_clean)} سطر")

# النتيجة: 3 سطور فقط تبقى (أول سطر يحتوي على كل القيم)
```

2️⃣ الملء (Filling)

✏️ ملء القيم المفقودة

```
# ملء بقيمة ثابتة
df_filled = df.fillna(0)           # ملء بـ 0
df_filled = df.fillna('غير محدد') # ملء نصوص

# ملء بمتوسط العمود
mean_age = df['العمر'].mean()     # 31.0
df_filled = df.fillna({'العمر': mean_age})

# ملء بوسيط (Median)
median_age = df['العمر'].median() # 31.5
df_filled = df.fillna({'العمر': median_age})

# ملء كل عمود بمتوسطه
```

```
df_filled = df.fillna(df.mean())

# الاستيفاء (Forward Fill - تكرار القيمة السابقة)
df_filled = df.fillna(method='ffill')

# الاستيفاء (Backward Fill - تكرار القيمة التالية)
df_filled = df.fillna(method='bfill')

# الاستيفاء الخطي (Linear interpolation)
df_filled = df.interpolate()
```

✅ مثال 6: الملء بالمتوسط

الكود:

```
print("الأصلي:")
print(df)

mean_age = df['العمر'].mean()
print(f"\nمتوسط العمر: {mean_age}")

df_filled = df.fillna({'العمر': mean_age})
print("\nبعد الملء:")
print(df_filled)

# النتيجة: NaN 31.0 العمر يصبح
```

3️⃣ التنبؤ (Prediction)

🤖 استخدام Scikit-learn للتنبؤ

```
from sklearn.impute import
SimpleImputer
from sklearn.impute import KNNImputer

# 1) SimpleImputer (متوسط، وسيط، الأكثر تكرارًا)
imputer_mean = SimpleImputer(strategy='mean')
X_filled = imputer_mean.fit_transform(df[['الراتب' ,'العمر']])
```

```
# 2) KNNImputer (أقرب جيران k استخدام)
imputer_knn = KNNImputer(n_neighbors=3)
X_filled = imputer_knn.fit_transform(df[['الراتب' ,'العمر']])

df_filled = df.copy()
df_filled[['الراتب' ,'العمر']] = X_filled
```

4️⃣ حالات خاصة

🎯 حالات عملية

```
# السيناريو 1: بيانات زمنية (السلاسل الزمنية)
# استخدم الاستيفاء الخطي
df['التاريخ'] = pd.date_range('2024-01-01',
periods=len(df))
df_filled = df.set_index('التاريخ').interpolate(method='linear')

# السيناريو 2: بيانات فئوية (نصوص)
# استخدم الملء بالأكثر تكرارًا
df['الفئة'] = df['الفئة'].fillna(df['الفئة'].mode()[0])

# السيناريو 3: تجاهل الصفوف بأكثر من 30% مفقود
df_clean = df.dropna(thresh=len(df.columns) * 0.7)

# السيناريو 4: ملء ذكي حسب المجموعة
df_filled = df.groupby('الفئة').transform(lambda x:
x.fillna(x.mean()))
```

✅ مثال 7: مقارنة الطرق

الكود:

```
print("الأصلي:")
print(df)

# الطريقة 1: حذف
df1 = df.dropna()
print(f"\nبعد الحذف: {len(df1)} سطر")
```

```
# الطريقة 2: ملء بالمتوسط
df2 = df.fillna(df.mean())
print(f"بعد الملء بالمتوسط: {len(df2)} سطر")
print(df2)

# الطريقة 3: ملء بالوسيط
df3 = df.fillna(df.median())
print(f"بعد الملء بالوسيط: {len(df3)} سطر")
```

Scikit-learn: Machine Learning 🤖

📝 الاستيراد

```
import sklearn
from sklearn.model_selection import
train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score,
recall_score
```

مثال شامل: من البيانات إلى النموذج

🔧 خطوات البناء

```
import pandas as pd
import numpy as np
from sklearn.model_selection import
train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# 1) تحضير البيانات
df = pd.DataFrame({
```

```
    'ساعات_الدراسة': [1, 2, 3, 4, 5, 2, 3, 4],
    'الدرجة': [55, 65, 70, 80, 85, 60, 75, 88]
})

# 2) إزالة البيانات المفقودة
df = df.dropna()

# 3) فصل الـ Features والـ Target
X = df[['ساعات_الدراسة']]
y = df['الدرجة']

# 4) تقسيم البيانات (80% تدريب، 20% اختبار)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# 5) تطبيع البيانات (Scaling)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 6) بناء وتدريب النموذج
model = LinearRegression()
model.fit(X_train_scaled, y_train)

# 7) التنبؤ
y_pred_train = model.predict(X_train_scaled)
y_pred_test = model.predict(X_test_scaled)

# 8) تقييم النموذج
mse = mean_squared_error(y_test, y_pred_test)
r2 = r2_score(y_test, y_pred_test)
print(f"MSE: {mse:.4f}")
print(f"R²: {r2:.4f}")

# 9) التنبؤ بقيمة جديدة
new_hours = np.array([[3.5]])
new_hours_scaled = scaler.transform(new_hours)
prediction = model.predict(new_hours_scaled)
print(f"التنبؤ للطالب الذي يدرس 3.5 ساعة: {prediction[0]:.2f}")
```

تصنيف (Classification)

🎯 نموذج تصنيف

```
from sklearn.ensemble import
RandomForestClassifier
from sklearn.metrics import accuracy_score,
confusion_matrix

# البيانات
df = pd.DataFrame({
    'العمر': [25, 35, 45, 28, 50, 38, 42, 31],
    'الدخل': [2000, 4000, 6000, 3000, 7000, 5000, 5500, 3500],
    'اشترى': [0, 1, 1, 0, 1, 1, 1, 0]  # 0 = لا، 1 = نعم
})

# فصل البيانات
X = df[['العمر', 'الدخل']]
y = df['اشترى']

# تقسيم البيانات
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# تدريب النموذج
clf = RandomForestClassifier(n_estimators=10, random_state=42)
clf.predict(X_train, y_train)

# التنبؤ
y_pred = clf.predict(X_test)

# التقييم
accuracy = accuracy_score(y_test, y_pred)
print(f"دقة النموذج: {accuracy:.2%}")
print(f"مصفوفة الالتباس:\n{confusion_matrix(y_test, y_pred)}")
```

✅ مثال 8: انحدار كامل

الكود:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

# البيانات
X = np.array([[1], [2], [3], [4], [5]])
y = np.array([2, 4, 5, 4, 5])

# النموذج
model = LinearRegression()
model.fit(X, y)

# التنبؤ
y_pred = model.predict(X)
r2 = r2_score(y, y_pred)

print(f"معامل الانحدار: {model.coef_[0]:.4f}")
print(f"الثابت: {model.intercept_:.4f}")
print(f"R²: {r2:.4f}")
```

📋 ملخص سريع - Cheat Sheet

المكتبة	الاستخدام الأساسي	أهم الدوال	مثال بسيط
NumPy	مصفوفات رياضية	array, arange, linspace, dot, sum	`a = np.array([1,2,3])`
Pandas	DataFrames وتحليل البيانات	read_csv, head, describe, fillna, dropna	`df = pd.read_csv('file.csv')`

المكتبة	الاستخدام الأساسي	أهم الدوال	مثال بسيط
Matplotlib	رسم بياني	plot, scatter, bar, hist, show	`plt.plot(x, y); plt.show()`
Sklearn	خوارزميات ML	train_test_split, fit, predict, score	`model.fit(X_train, y_train)`

خطوات ML النموذجية

الخطوة	الكود	الشرح
1️⃣ قراءة البيانات	`df = pd.read_csv('data.csv')`	تحميل البيانات من ملف
2️⃣ استكشاف	`df.head(); df.info(); df.describe()`	فهم البيانات
3️⃣ تنظيف	`df.dropna(); df.fillna(value)`	إزالة/ملء البيانات المفقودة
4️⃣ معالجة	`scaler = StandardScaler(); scaler.fit_transform(X)`	تطبيع البيانات
5️⃣ تقسيم	`train_test_split(X, y, test_size=0.2)`	فصل التدريب والاختبار

الخطوة	الكود	الشرح
6️⃣ تدريب	`model.fit(X_train, y_train)`	بناء النموذج
7️⃣ تنبؤ	`y_pred = model.predict(X_test)`	اختبار النموذج
8️⃣ تقييم	`score = model.score(X_test, y_test)`	قياس الأداء

معالجة البيانات المفقودة - ملخص

🗑️ الحذف
`df.dropna()` - يحذف أي سطر به قيم مفقودة

✏️ الملء بقيمة ثابتة
`df.fillna(0)` - ملء كل القيم المفقودة بـ 0

📊 الملء بمتوسط
`df.fillna(df.mean())` - ملء بمتوسط كل عمود

🤖 الملء بـ Sklearn
`SimpleImputer(strategy='mean')` - ملء ذكي

تم إعداده لطلاب الـ Machine Learning

جميع الأمثلة جاهزة للتطبيق الفوري • اطبع كـ PDF • استخدم كـ Cheat Sheet