الطبعة الأولى — 2026

AI IN TRADING

الذكاء الاصطناعي في التداول

الدليل الشامل من المبتدئ إلى المحترف — التعلم الآلي، التعلم العميق، معالجة
لغة، والتعلم المعزز مطبّقة على الأسواق المالية ببيانات حقيقية وكود Python
كامل

Machine Learning | Deep Learning | & NLP LLMs | Reinforcement Learning

33 فصلًا · مسرد مصطلحات · ملاحق مرجعية

حقوق النشر

الذكاء الاصطناعي في التداول (AI IN TRADING)
الدليل الشامل لمفاهيم الأموال الذكية، والتحليل الأساسي، والتحليل الفني

الطبعة الأولى — 2026

جميع الحقوق محفوظة © 2026 لـ Ahmed Abdelhafez وAhmed Abu Omran.

لا يجوز إعادة إنتاج أي جزء من هذا الكتاب أو توزيعه أو نقله بأي شكل من الأشكال أو بأي وسيلة — سواء بالتصوير، أو التسجيل، أو أي وسيلة إلكترونية أو ميكانيكية أخرى، أو عبر أي نظام لتخزين المعلومات واسترجاعها — دون إذن كتابي مسبق من المؤلفَين، باستثناء الاقتباسات القصيرة المستخدمة في المراجعات النقدية وبعض الاستخدامات غير التجارية الأخرى التي يسمح بها قانون حقوق النشر.

يُعد كتاب "الذكاء الاصطناعي في التداول"، وجميع ما يتضمنه من رسوم توضيحية وأشكال ومخططات، عملًا أصليًا أنتجه المؤلفان، ويخضع لحماية قوانين الملكية الفكرية وحقوق النشر المعمول بها. أي إعادة إنتاج أو توزيع غير مصرح به لهذا العمل، أو أي جزء منه، قد يُعرّض صاحبه لعقوبات مدنية وجنائية، وستُتخذ الإجراءات القانونية اللازمة إلى أقصى حد يسمح به القانون.

إخلاء المسؤولية

هذا الكتاب مُعَدّ لأغراض تعليمية وتثقيفية فقط. ينطوي التداول والاستثمار في الأسواق المالية على مخاطر جوهرية لخسارة رأس المال، وقد لا يكون مناسبًا لكل شخص. لا ينبغي اعتبار أي محتوى وارد في هذا الكتاب استشارة مالية أو استثمارية أو قانونية أو ضريبية، ولا توصية أو دعوة لشراء أو بيع أي أداة مالية.

محتوى هذا الكتاب، وجميع الأطر والمفاهيم والقواعد الواردة فيه، مستمدّ من تجربة تداول حقيقية وشخصية خاضها المؤلفان على مدار سنوات، بما فيها من نجاحات وإخفاقات شكّلت فهمهما العملي للأسواق. أما المخططات والأشكال التوضيحية المرسومة داخل هذا الكتاب تحديدًا، فهي رسوم توضيحية أُعدّت خصيصًا لتجسيد وتبسيط تلك المفاهيم بأوضح صورة ممكنة للقارئ، وليست نسخة حرفية عن صفقة أو حساب أو أداة مالية بعينها؛ فأداء الأسواق الفعلي، ماضيًا أو مستقبلًا، لا يتكرر بالشكل نفسه ولا يمكن ضمانه.

لا يقدّم المؤلفان أي إقرار أو ضمان بخصوص دقة أو اكتمال محتوى هذا الكتاب، ويُخليان مسؤوليتهما صراحةً عن أي ضمانات ضمنية. تقع مسؤولية أي قرار تداولي بالكامل على عاتق القارئ. لن يتحمّل المؤلفان أي مسؤولية عن أي خسارة أو ضرر، مالي أو غير مالي، ينشأ بشكل مباشر أو غير مباشر عن استخدام أو تطبيق أي معلومة واردة في هذا الكتاب.

يُنصح القراء بإجراء أبحاثهم الخاصة واستشارة مختص مالي مرخّص قبل اتخاذ أي قرار استثماري أو تداولي.

معلومات الإصدار

عنوان الكتاب: الذكاء الاصطناعي في التداول
المؤلفان: Ahmed Abdelhafez، Ahmed Abu Omran
الطبعة: الأولى، 2026
اللغة: العربية (مع بعض المصطلحات الإنجليزية المتخصصة)

المؤلفان

Ahmed Abdelhafez

مؤلف مشارك

Ahmed Abu Omran

مؤلف مشارك

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

رسالة من المؤلفَين

وُلد هذا الكتاب بعد رحلة طويلة — سنوات قضيناها بين المخططات، وبين صفقاتنا الخاسرة بقدر ما بين رابحاتها، وفي ذلك المسار البطيء، والمُحبِط أحيانًا، الذي حوّل خبرة متناثرة إلى شيء يمكن تعليمه فعلًا لغيرنا.

لم نكتب هذا الكتاب بحثًا عن اختصارات سهلة. ما بين يديك أقرب إلى سجلٍّ عملي: الأطر التي عدنا إليها مرارًا، والأخطاء التي كلّفتنا قبل أن تُعلّمنا شيئًا، والتنقيح البطيء لمنهج استقرّ أخيرًا تحت ضغط ظروف السوق الحقيقية. إن درست هذا الكتاب بجدّية وطبّقته بصبر، فسيمنحك أساسًا حقيقيًا — أساسًا بُني ليدوم طويلًا، لا ليخدم صفقتك القادمة فقط.

لكننا نريد أن نكون صادقَين معك قبل أن تقلب الصفحة التالية: لا شيء مما ستقرأه — لا هيكل السوق، ولا مفاهيم الأموال الذكية، ولا المؤشرات — هو الدرس الحقيقي. الدرس الحقيقي أهدأ من كل ذلك بكثير. إنه الهدوء. إنه الانضباط. إنه القدرة على الجلوس مع صفقة خاسرة دون أن تدعها تُملي عليك قرارك التالي.

ستمر بفترات تشعر فيها أن استراتيجيتك لا تُقهر، حين يبدو كل إعداد صحيحًا وكل حدس مصيبًا. لا تخلط بين ذلك وبين الاحتراف الحقيقي. فبدون إدارة مخاطر وانضباط نفسي خلف أي ميزة تداولية، لن تصمد هذه الميزة أمام عدد كافٍ من الصفقات. رأينا ذلك يحدث لغيرنا، ورأيناه يحدث لنا في بدايات مشوارنا أيضًا.

لذا خذ الأطر الواردة في هذا الكتاب على محمل الجد — فهي تعمل فعلًا، وقد بنيناها لتدوم. لكن خذ الانضباط على محمل جدٍّ أكبر. فالاحتراف الحقيقي في هذا المجال لا يبدأ بالسيطرة على السوق، بل يبدأ بالسيطرة على النفس.

— *Ahmed Abdelhafez* و*Ahmed Abu Omran*

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

المحتويات

الجزء الأول: أساسيات الذكاء الاصطناعي للتداول 11

الفصل 1: مقدمة إلى الذكاء الاصطناعي في الأسواق المالية 11

1.1 ما هو الذكاء الاصطناعي؟ 11

1.2 لماذا الذكاء الاصطناعي الآن؟ 11

1.3 المجالات الأربعة الرئيسية 12

1.4 التحليل البشري مقابل التحليل بالذكاء الاصطناعي 12

1.5 المفاهيم الأساسية الأربعة 13

1.6 أول كود في الكتاب: الفكرة كاملة في 15 سطرًا 14

1.7 المصطلحات والرموز الأساسية 15

ملخص الفصل 15

نقاط رئيسية 15

تمرين عملي 15

دراسة حالة 16

الفصل 2: إعداد البيئة التقنية (Python) 17

2.1 لماذا Python؟ 17

2.2 التثبيت خطوة بخطوة 17

2.3 خريطة المكتبات 18

2.4 مصادر البيانات عبر الواجهات البرمجية (APIs) 18

2.5 أول تحليل حقيقي كامل 19

ملخص الفصل 21

نقاط رئيسية 21

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

تمرين عملي 21
دراسة حالة 21

الجزء الثاني: جمع البيانات ومعالجتها 22

الفصل 3: مصادر البيانات المالية 22

3.1 البيانات التاريخية (OHLCV): العمود الفقري 22
3.2 خريطة أنواع البيانات الثلاثة 23
3.3 جودة البيانات: القيم الناقصة 24
3.4 جودة البيانات: القيم الشاذة 24
ملخص الفصل 25
نقاط رئيسية 25
تمرين عملي 25
دراسة حالة 25
أمثلة تطبيقية من بيانات السوق الحقيقية 26

الفصل 4: معالجة البيانات وتحضيرها للنماذج 27

4.1 لماذا المعالجة نصف المشروع؟ 27
4.2 تحويل المقاييس (Normalization / Standardization) 27
4.3 هندسة المميزات (Feature Engineering) 28
4.4 التقسيم الثلاثي: تدريب / تحقق / اختبار 29
4.5 التحيز المستقبلي (Look-ahead Bias): العدو الأول 29
ملخص الفصل 30
نقاط رئيسية 30
تمرين عملي 31
دراسة حالة 31

©2026 Ahmed Abdelhafez و Ahmed Abu Omran - جميع الحقوق محفوظة

أمثلة تطبيقية من بيانات السوق الحقيقية 31

الجزء الثالث: التعلم الآلي الكلاسيكي في التداول 32

الفصل 5: التعلم الخاضع للإشراف — التصنيف 32

5.1 صياغة التداول كمشكلة تصنيف 32

5.2 الخوارزميات الأربع الأساسية 32

5.3 النتائج الحقيقية — وقراءتها الصادقة 32

5.4 ما بعد الدقة: مصفوفة الالتباس 33

5.5 من الاحتمال إلى قرار تداول 34

ملخص الفصل 35

نقاط رئيسية 35

تمرين عملي 35

دراسة حالة 35

أمثلة تطبيقية من بيانات السوق الحقيقية 36

الفصل 6: التعلم الخاضع للإشراف — الانحدار 37

6.1 من الفئة إلى القيمة 37

6.2 الخوارزميات الأساسية 37

6.3 مقاييس التقييم 37

6.4 النتائج الحقيقية — ولماذا R^2 سالب؟ 37

6.5 فخ «توقع السعر» الشهير 39

ملخص الفصل 39

نقاط رئيسية 39

تمرين عملي 39

دراسة حالة 39

©2026 Ahmed Abdelhafez و Ahmed Abu Omran - جميع الحقوق محفوظة

أمثلة تطبيقية من بيانات السوق الحقيقية 40

الفصل 7: التعلم غير الخاضع للإشراف 41

7.1 التعلم بلا معلّم 41

7.2 التجميع: أنظمة السوق بـ K-Means 41

7.3 تقليل الأبعاد: PCA 42

7.4 اكتشاف الشذوذ: Isolation Forest 43

ملخص الفصل 44

نقاط رئيسية 44

تمرين عملي 44

دراسة حالة 45

أمثلة تطبيقية من بيانات السوق الحقيقية 45

الفصل 8: التطبيق العملي — بناء نموذج تنبؤي متكامل 46

8.1 المشروع: XGBoost على EUR/USD من الألف إلى الياء 46

8.2 الخطوات الخمس 46

8.3 النتائج الفعلية — والاعتراف قبل التفسير 47

8.4 من الدقة إلى المحفظة: الاختبار الخلفي 47

8.5 تشريح الأخطاء الخمسة الشائعة (وكيف نجونا منها) 48

ملخص الفصل 48

نقاط رئيسية 49

تمرين عملي 49

دراسة حالة 49

أمثلة تطبيقية من بيانات السوق الحقيقية 49

الجزء الرابع: التعلم العميق في التداول 50

©2026 Ahmed Abdelhafez و Ahmed Abu Omran - جميع الحقوق محفوظة

الفصل 9: الشبكات العصبية الاصطناعية 50

9.1 لماذا «العميق»؟ 50

9.2 كيف تتعلم الشبكة: ثلاث خطوات تتكرر 50

9.3 المعركة الحقيقية: الإفراط — بالأرقام الفعلية 52

9.4 قواعد عملية للبيانات المالية 52

ملخص الفصل 53

نقاط رئيسية 53

تمرين عملي 53

دراسة حالة 53

أمثلة تطبيقية من بيانات السوق الحقيقية 54

الفصل 10: الشبكات المتكررة وLSTM 55

10.1 السوق سلسلة، لا جدول 55

10.2 لماذا تفشل RNN البسيطة: قياس فعلي للتلاشي 55

10.3 الحل: خلية LSTM 56

10.4 بناء LSTM للتنبؤ السعري — الكود الكامل 56

ملخص الفصل 57

نقاط رئيسية 57

تمرين عملي 58

دراسة حالة 58

أمثلة تطبيقية من بيانات السوق الحقيقية 58

الفصل 11: نماذج المحولات (Transformers) 59

11.1 الفكرة الثورية: الانتباه بدل الذاكرة 59

11.2 الانتباه الذاتي بالأرقام — على بيانات حقيقية 59

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

11.3 محوّل عملي للتنبؤ السعري 61
11.4 محوّل أم LSTM؟ الجواب الصادق 61
ملخص الفصل 62
نقاط رئيسية 62
تمرين عملي 62
دراسة حالة 62
أمثلة تطبيقية من بيانات السوق الحقيقية 63

الفصل 12: التطبيق العملي — التنبؤ بالنوافذ العميقة على EUR/USD 64

12.1 المشروع: هل يهزم العمقُ البسيطَ فعلًا؟ 64
12.2 النتائج الفعلية — الجدول الذي يتجنبه المسوّقون 64
12.3 فماذا كان LSTM/المحوّل سيغيران؟ 66
12.4 إعادة الصياغة: حيث يستحق العمق ثمنه 66
12.5 قائمة الفحص النهائية لأي مشروع تعلم عميق مالي 66
ملخص الفصل 67
نقاط رئيسية 67
تمرين عملي 67
دراسة حالة 67
أمثلة تطبيقية من بيانات السوق الحقيقية 68

©2026 Ahmed Abdelhafez و Ahmed Abu Omran - جميع الحقوق محفوظة

الجزء الأول: أساسيات الذكاء الاصطناعي للتداول

الفصل 1: مقدمة إلى الذكاء الاصطناعي في الأسواق المالية

1.1 ما هو الذكاء الاصطناعي؟

الذكاء الاصطناعي (Artificial Intelligence) هو قدرة البرامج على أداء مهام كانت تتطلب تقليديًا ذكاءً بشريًا: التعرف على الأنماط، التعلم من التجربة، واتخاذ القرارات. الفرع الأهم لنا كمتداولين هو **التعلم الآلي (Machine Learning)**: بدلًا من كتابة قواعد يدوية («إذا كسر السعر المتوسط فاشترِ»)، نعطي الخوارزمية **بيانات تاريخية** وتستخرج هي القواعد والأنماط منها بنفسها.

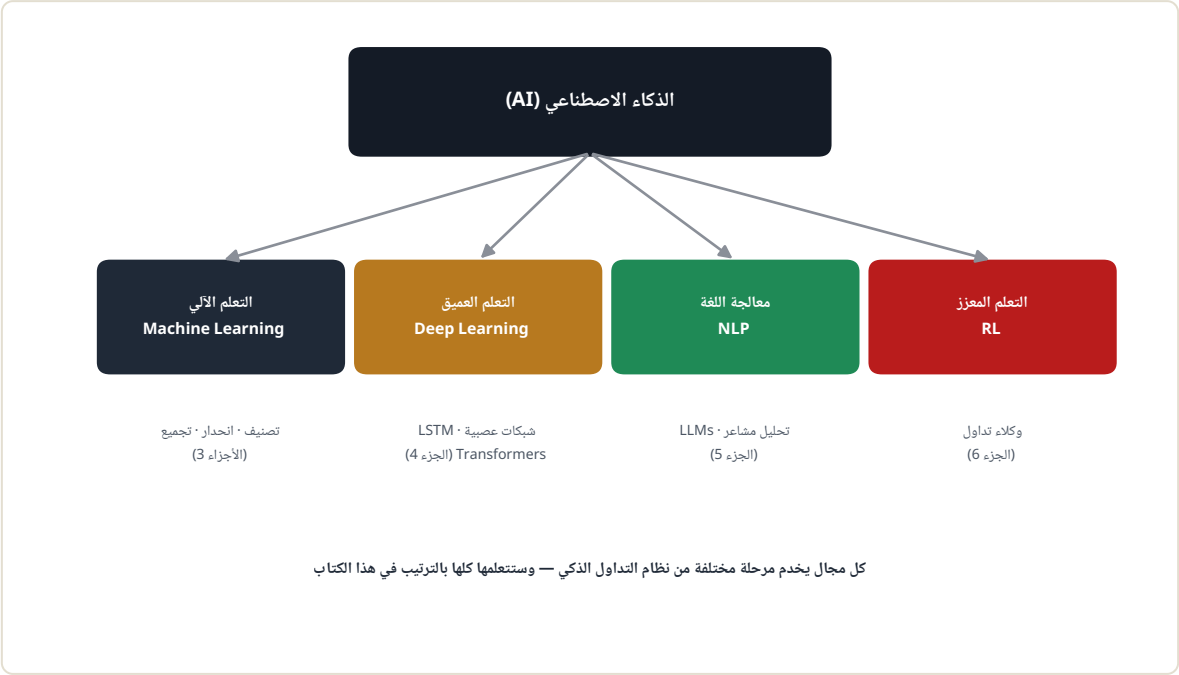
1.2 لماذا الذكاء الاصطناعي الآن؟

ثلاثة أسباب جعلت هذا المجال ضرورة لا رفاهية:

1. **حجم البيانات:** ملايين الشموع والأخبار والتغريدات يوميًا — فوق طاقة أي محلل بشري.
2. **سرعة السوق:** قرارات تُتخذ في أجزاء من الثانية؛ المؤسسات تتنافس خوارزميًا منذ سنوات.
3. **تجاوز الانحياز البشري:** النموذج لا يخاف ولا يطمع ولا يتشبث برأيه — يطبّق ما تعلمه بثبات (يكمل هذا مباشرة ما ناقشناه في فصل علم النفس بكتاب «التحليل الثلاثي»).

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

1.3 المجالات الأربعة الرئيسية



الشكل 1.1 — خريطة مجالات الذكاء الاصطناعي الأربعة التي يغطيها هذا الكتاب، ودور كل منها في نظام التداول

المجال	ماذا يفعل في التداول	أين في الكتاب
التعلم الآلي الكلاسيكي	تنبؤ بالاتجاه، تصنيف حالات السوق	الجزء الثالث
التعلم العميق	نمذجة السلاسل الزمنية المعقدة	الجزء الرابع
معالجة اللغة الطبيعية	قراءة الأخبار وتصريحات البنوك آليًا	الجزء الخامس
التعلم المعزز	وكيل يتعلم استراتيجية التداول بالتجربة	الجزء السادس

1.4 التحليل البشري مقابل التحليل بالذكاء الاصطناعي

المعيار	المحلل البشري	نموذج الذكاء الاصطناعي
سرعة المعالجة	محدودة	ملايين النقاط في ثوانٍ
الاتساق	يتأثر بالحالة النفسية	ثابت دائمًا
فهم السياق الاستثنائي	ممتاز (حروب، أزمات جديدة)	ضعيف خارج ما تدرّب عليه
التفسير	يشرح منطقه	كثير من النماذج «صندوق أسود»

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

المعيار	المحلل البشري	نموذج الذكاء الاصطناعي
التكيف مع تغيّر السوق	مرن	يحتاج إعادة تدريب

الخلاصة العملية التي سيكررها هذا الكتاب: **الذكاء الاصطناعي أداة دعم قرار تضاعف قدرة المتداول المتمكن، وليست بديلًا عن فهم الأسواق** — من لا يفهم الفصل 12 من «التحليل الثلاثي» لن ينقذه نموذج.

1.5 المفاهيم الأساسية الأربعة

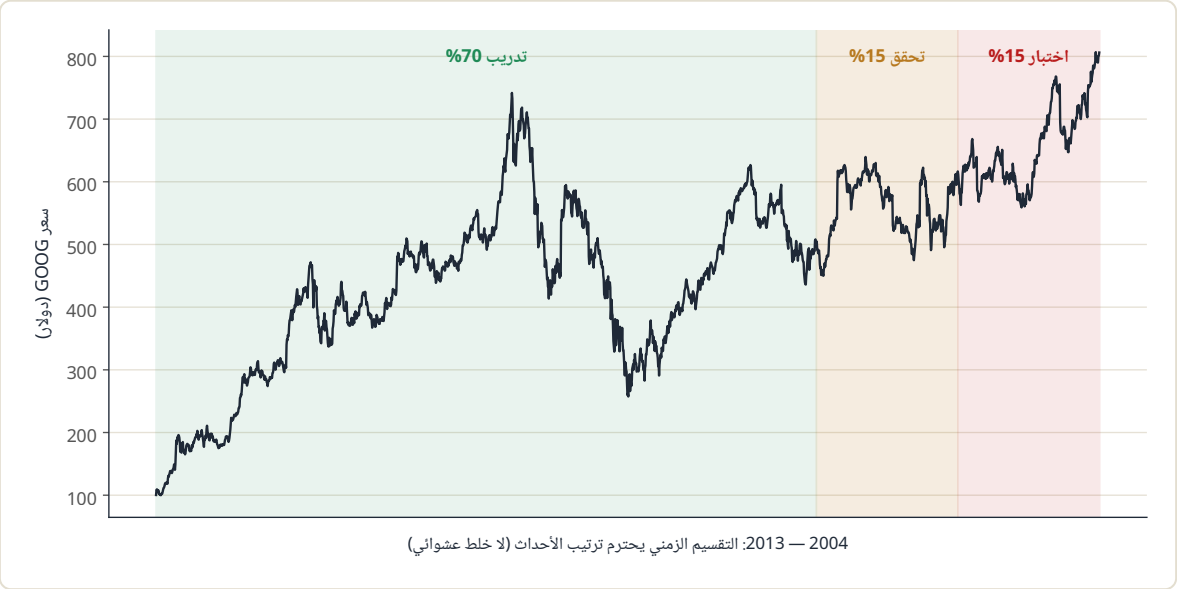
1) هندسة المميزات (Feature Engineering): تحويل البيانات الخام إلى مدخلات ذات معنى للنموذج (عوائد، مؤشرات، فروق). جودة المميزات أهم من تعقيد النموذج.

2) الإفراط في التجهيز (Overfitting): أخطر مفهوم في الكتاب كله — النموذج «يحفظ» بيانات التدريب بدل أن «يتعلم» منها، فيبدو عبقريًا على الماضي وينهار على المستقبل. الشكل التالي **ناتج عن تجربة فعلية على بيانات GOOG الحقيقية**: نموذجان دُرّبا على نفس 60 يومًا، ثم واجها الأيام العشرين التالية التي لم يرياها:

الشكل 1.2 — تجربة حقيقية على GOOG: النموذج المعقد (أحمر) يلتصق ببيانات التدريب ثم ينفجر خطؤه في المستقبل، بينما البسيط (أخضر) يعمّم بشكل معقول

3) تقسيم البيانات (Train/Test Split): في السلاسل الزمنية يجب التقسيم **بالترتيب الزمني** — التدريب على الماضي والاختبار على المستقبل، وإلا تسرّبت معلومات المستقبل للنموذج:

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة



الشكل 1.3 — التقسيم الزمني الصحيح مطبقًا على بيانات GOOG الحقيقية (2004-2013): تدريب ثم تحقق ثم اختبار، دون خلط

4) الاختبار الخلفي (Backtesting): محاكاة النموذج على الماضي بشروط واقعية (تكاليف، انزلاق) قبل أي مال حقيقي — الامتداد الآلي لما شرحناه يدويًا في القسم 12.7 من «التحليل الثلاثي».

1.6 أول كود في الكتاب: الفكرة كاملة في 15 سطرًا

```
# الفكرة الجوهرية للتعلم الآلي في التداول – نسخة مبسطة
import numpy as np
from backtesting.test import GOOG   # بيانات ياهو فاينانس حقيقية

closes = GOOG.Close.values          # 2148 يوم تداول فعلي (2004-2013)
returns = np.diff(closes) / closes[:-1]

# الميزة X: عائد اليوم | الهدف y: هل يصعد الغد؟
X = returns[:-1].reshape(-1, 1)
y = (returns[1:] > 0).astype(int)

split = int(len(X) * 0.8)           # تقسيم زمني: الماضي للتعلم
X_train, X_test = X[:split], X[split:]
y_train, y_test = y[:split], y[split:]

from sklearn.linear_model import LogisticRegression
model = LogisticRegression().fit(X_train, y_train)
acc = model.score(X_test, y_test)
print(f"Accuracy on unseen future: {acc:.1%}")
```

نتيجة تشغيل هذا الكود فعليًا على البيانات: **54.2%** — بالكاد فوق العملة المعدنية. وهذه **أهم رسالة في الفصل**: ميزة واحدة ساذجة لا تتنبأ بالسوق، والفارق الحقيقي بين 54% ونموذج نافع تجاريًا هو موضوع بقية الكتاب (مميزات أفضل، نماذج أنسب، وتقييم أصدق يكشف حتى هذه الـ54% هل هي حقيقية أم وهم إحصائي).

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

1.7 المصطلحات والرموز الأساسية

المصطلح	الاختصار	المعنى
Machine Learning	ML	تعلم الأنماط من البيانات
Deep Learning	DL	تعلم آلي بشبكات عصبية متعددة الطبقات
Feature	X	متغير مُدخَل للنموذج
Target/Label	y	ما نحاول التنبؤ به
Overfitting	—	حفظ التدريب والفشل في التعميم
Backtesting	—	اختبار الاستراتيجية على الماضي

ملخص الفصل

- التعلم الآلي يستخرج القواعد من البيانات بدل كتابتها يدويًا، ويتفوق في السرعة والاتساق ويقصّر في السياقات الاستثنائية والتفسير.
- الإفراط في التجهيز هو الخطر الأول: أداء مبهر على الماضي لا يعني شيئًا عن المستقبل (رأيته بتجربة حقيقية على GOOG في الشكل 1.2).
- تقسيم بيانات السلاسل الزمنية يجب أن يحترم الترتيب الزمني دائمًا.

نقاط رئيسية

1. ابدأ دائمًا بالسؤال «ما الذي أتنبأ به بالضبط؟» قبل أي كود — تعريف y نصف المشروع.
2. تعامل مع أي دقة قريبة من 50% في التنبؤ بالاتجاه على أنها «لا شيء» وليست «نصف نجاح».
3. لا تصدّق نموذجًا لم يُختبر على بيانات لم يرها إطلاقًا أثناء التدريب.

تمرين عملي

شغّل كود القسم 1.6 كما هو، ثم غيّر الميزة من عائد يوم واحد إلى متوسط عوائد آخر 5 أيام (`np.convolve` أو `pd.Series.rolling`)، وسجّل: هل تحسنت الدقة على بيانات الاختبار؟ ولماذا قد لا تتحسن؟

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

دراسة حالة

في 2007-2008 (وهي ضمن بياناتنا الحقيقية في الشكل 1.3) انقلب سلوك السوق رأسًا على عقب خلال أسابيع. نموذج دُرّب حتى 2007 فقط وطُبّق في 2008 دون إعادة تدريب كان سيواجه سوقًا لا يشبه شيئًا مما تعلمه — هذا هو «انجراف النموذج» الذي سنعالجه في الفصل 21، وهو السبب الذي يجعل المحترفين يعيدون تدريب نماذجهم دوريًا ويراقبون أداءها لحظيًا بدل الوثوق بها للأبد.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الفصل 2: إعداد البيئة التقنية (Python)

2.1 لماذا Python؟

Python هي اللغة القياسية للذكاء الاصطناعي المالي عالميًا: بساطة قراءتها تجعل الكود قريبًا من اللغة الطبيعية، ومكتباتها تغطي السلسلة كاملة من جلب البيانات حتى نشر النموذج. كل كود في هذا الكتاب Python خالص، مكتوب ليعمل كما هو.

2.2 التثبيت خطوة بخطوة

1. **حمّل Anaconda** من موقعها الرسمي (anaconda.com) — توزيعة تجمع Python مع مئات المكتبات العلمية جاهزة.
2. **افتح Jupyter Notebook** (يُثبت مع Anaconda): بيئة تفاعلية تنفّذ الكود خلية-خلية وتعرض المخططات داخل الصفحة — مثالية للتجربة والتعلم.
3. **أنشئ بيئة مستقلة للمشروع** (خطوة احترافية تعزل مكتبات كل مشروع):

```
conda create -n ai-trading python=3.11
conda activate ai-trading
pip install pandas numpy matplotlib seaborn scikit-learn xgboost lightgbm backtesting
yfinance
```


©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

2.3 خريطة المكتبات

الشكل 2.1 — طبقات الحزمة التقنية: من مصادر البيانات في الأعلى حتى العرض في الأسفل

الفئة	المكتبات	دورها
معالجة البيانات	Pandas، NumPy	جداول وسلاسل زمنية وعمليات رقمية سريعة
التعلم الآلي	Scikit-learn، XGBoost، LightGBM	النماذج الكلاسيكية (الجزء الثالث)
التعلم العميق	TensorFlow/Keras، PyTorch	الشبكات العصبية (الجزء الرابع)
معالجة اللغة	NLTK، SpaCy، Transformers	تحليل النصوص والأخبار (الجزء الخامس)
العرض	Matplotlib، Seaborn	كل مخططات هذا الكتاب
البيانات	backtesting، yfinance	بيانات تاريخية حقيقية جاهزة

2.4 مصادر البيانات عبر الواجهات البرمجية (APIs)

المصدر	النوع	مجاني؟	ملاحظة
Yahoo Finance (yfinance)	أسهم/مؤشرات/عملات	نعم	الأسهل للبدء؛ حدود استخدام غير رسمية

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

المصدر	النوع	مجاني؟	ملاحظة
Alpha Vantage	أسهم/فوركس/كريبتو	مفتاح مجاني محدود	25 طلبًا يوميًا مجانًا
CoinGecko	كريبتو	نعم	بلا مفتاح للاستخدام الأساسي
Binance API	كريبتو لحظي	نعم	WebSocket للبيانات الحية (الفصل 19)
OpenAI API	نماذج لغوية	مدفوع	للجزء الخامس (LLMs)

قاعدة مهنية: احفظ مفاتيح الـAPI في متغيرات بيئة، لا داخل الكود أبدًا:

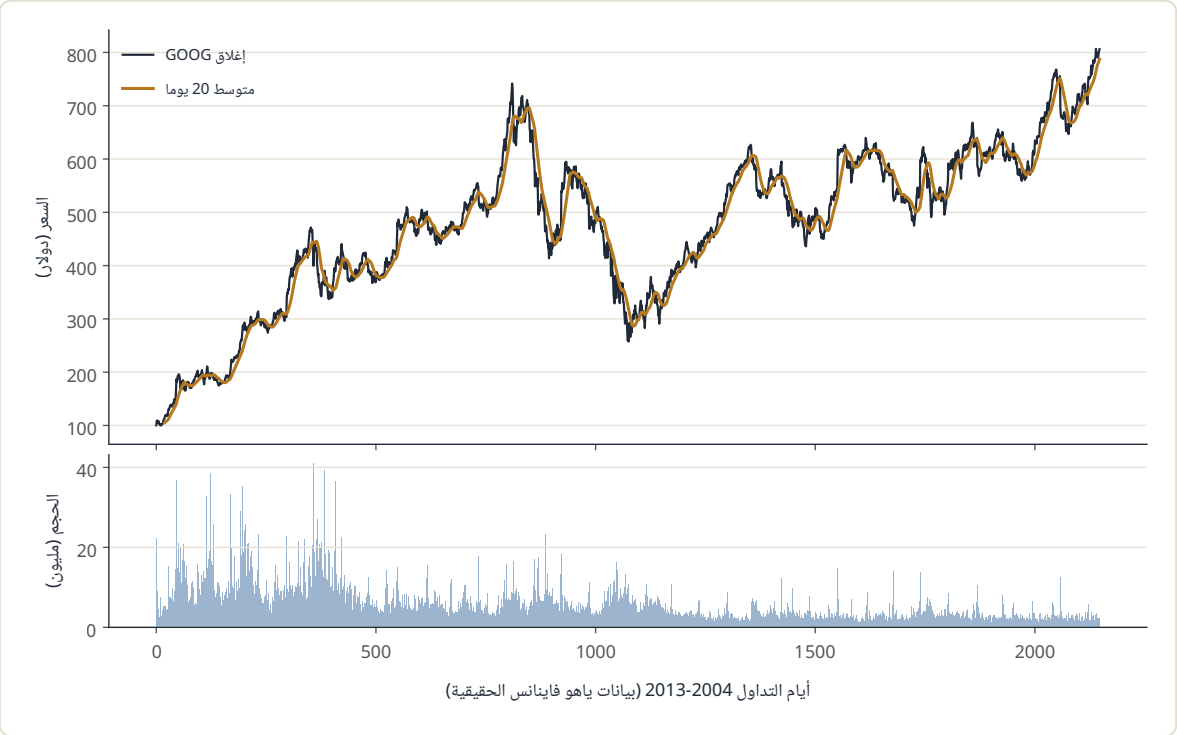
```
import os
api_key = os.environ["ALPHA_VANTAGE_KEY"]   # لا تكتب المفتاح في الملف
```

2.5 أول تحليل حقيقي كامل

هذا الكتاب يعتمد أساسًا بيانات حقيقية مرفقة مع مكتبة `backtesting`: أسعار GOOG اليومية الفعلية (2004-2013) وEURUSD بالساعة — نفس بيانات ياهو فاينانس المنشورة، متاحة فورًا دون إنترنت أو مفاتيح. الكود التالي (المولّد الفعلي للشكل 2.2):

```
import numpy as np
import matplotlib.pyplot as plt
from backtesting.test import GOOG          # بيانات حقيقية جاهزة

close = GOOG.Close.values
ma20 = np.convolve(close, np.ones(20)/20, mode="valid")

fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(11, 6),
                               gridspec_kw={"height_ratios": [2.4, 1]},
                               sharex=True)
ax1.plot(close, label="GOOG Close")
ax1.plot(np.arange(19, len(close)), ma20, label="MA 20")
ax1.legend()
ax2.bar(np.arange(len(GOOG)), GOOG.Volume.values / 1e6)
ax2.set_xlabel("Trading days 2004-2013")
plt.show()
```

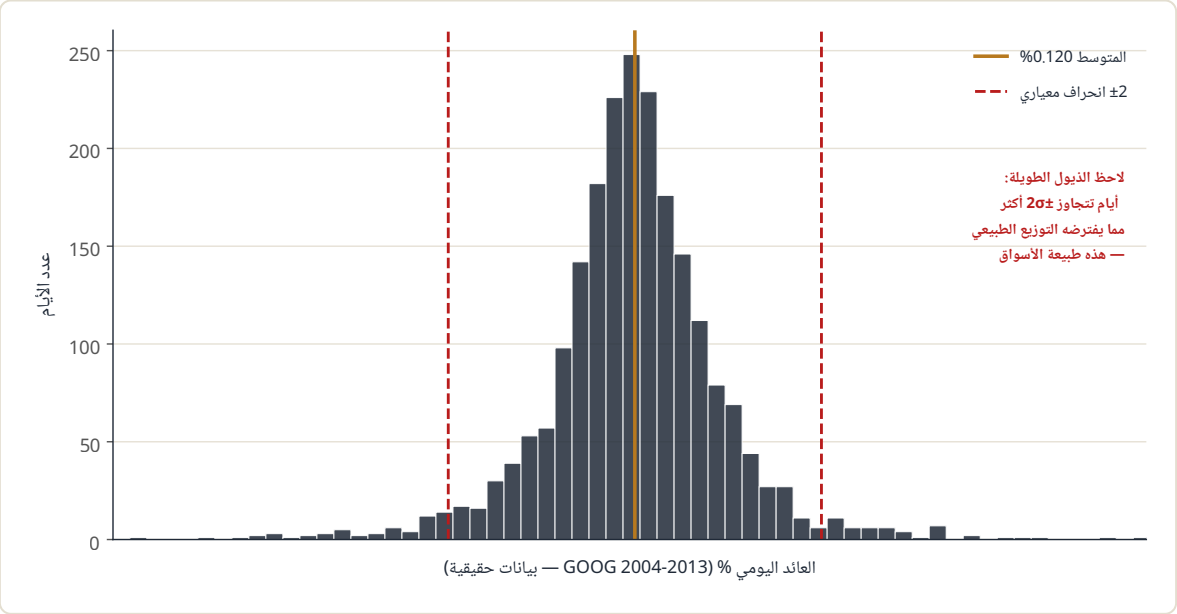

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة



الشكل 2.2 — ناتج تشغيل الكود أعلاه فعليًا: سعر GOOG الحقيقي مع متوسط 20 يومًا ولوحة الحجم

وبسطرين إضافيين نفحص توزيع العوائد اليومية — أول خطوة في أي مشروع كمي:

```
rets = np.diff(close) / close[:-1] * 100
plt.hist(rets, bins=80)
```



الشكل 2.3 — توزيع عوائد GOOG اليومية الحقيقية: لاحظ «الذيول السمينة» — أيام متطرفة أكثر بكثير من التوزيع الطبيعي، وهي سمة جوهرية للأسواق سيتكرر أثرها في كل الفصول القادمة

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

ملخص الفصل

- Anaconda + Jupyter هي بيئة العمل القياسية، مع بيئة conda مستقلة لكل مشروع.
- خمس فئات مكتبات تغطي السلسلة كاملة: معالجة، تعلم آلي، تعلم عميق، لغة، وعرض.
- بيانات GOOG وEURUSD الحقيقية المرفقة مع backtesting هي مختبرنا الثابت في كل الفصول.

نقاط رئيسية

1. لا تضع مفاتيح API داخل الكود إطلاقًا — متغيرات البيئة دائمًا.
2. اعتد فحص توزيع العوائد أول كل مشروع؛ الذيول السمينة تغيّر كل افتراضات النمذجة.
3. ثبّت إصدارات مكتباتك في ملف requirements.txt ليعمل كودك غدًا كما عمل اليوم.

تمرين عملي

ثبّت البيئة، ثم أعد إنتاج الشكل 2.2 بنفسك، وعدّله ليعرض متوسطين (20 و50 يومًا) معًا، ولاحظ نقاط التقاطع بينهما على البيانات الحقيقية.

دراسة حالة

فريق صغير بدأ مشروعه بجمع البيانات مباشرة من واجهة برمجية لحظية دون تثبيت بيئة أو إدارة إصدارات. بعد شهر، تحدّثت إحدى المكتبات فتغيّر سلوك دالة مركزية، وانهار خط المعالجة كاملًا في يوم تقلبات عالية — والأسوأ أنهم لم يستطيعوا استرجاع النسخة العاملة لأن الإصدارات لم تكن مثبتة. الدرس: البنية التحتية المملة (بيئات معزولة، إصدارات مثبتة، بيانات مرجعية ثابتة للاختبار) هي ما يفصل مشروعًا احترافيًا عن تجربة هشة — ولهذا بدأنا بها قبل أي نموذج.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الجزء الثاني: جمع البيانات ومعالجتها

الفصل 3: مصادر البيانات المالية

3.1 البيانات التاريخية (OHLCV): العمود الفقري

كل شمعة تختصر فترة كاملة في خمس قيم: الافتتاح (Open)، الأعلى (High)، الأدنى (Low)، الإغلاق (Close)، والحجم (Volume) — ومنها اشتُق كل ما درسته في «التحليل الثلاثي» وكل ما سيتعلمه أي نموذج في هذا الكتاب.

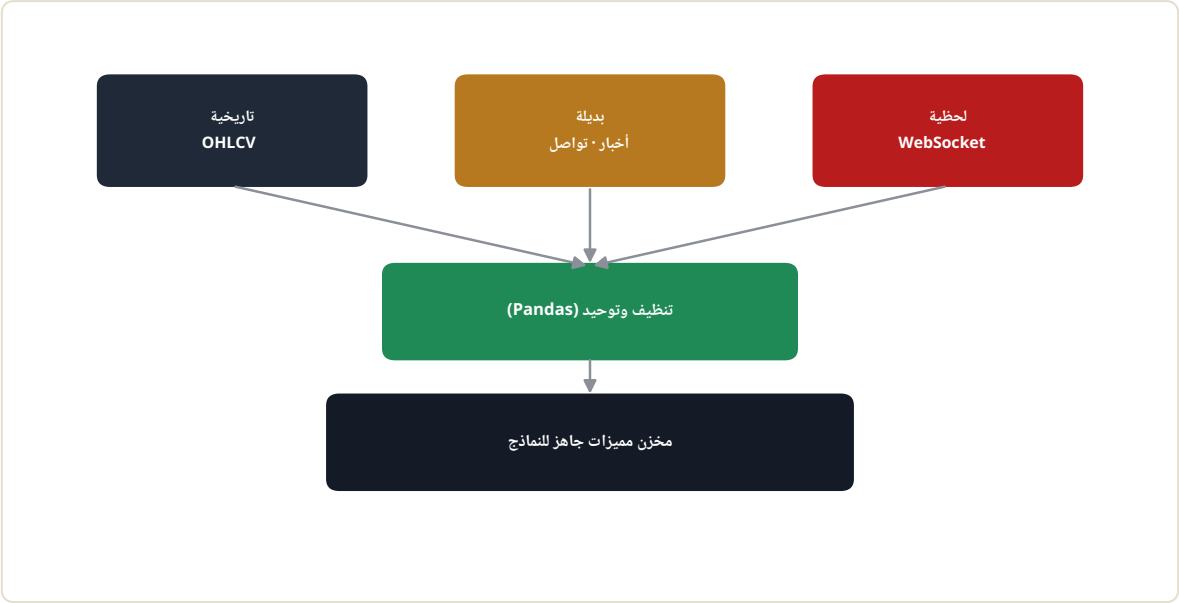
الشكل 3.1 — 40 شمعة ساعة حقيقية من بيانات EUR/USD: خمس قيم لكل فترة هي المادة الخام لكل ما يلي

جلب البيانات عمليًا بثلاثة أسطر لكل مصدر:

```
# المصدر 1: ياهو فاينانس (الأسهل)
import yfinance as yf
df = yf.download("AAPL", start="2015-01-01", end="2024-01-01")

# المصدر 2: Alpha Vantage (مفتاح مجاني)
# GET https://www.alphavantage.co/query?function=TIME_SERIES_DAILY
#     &symbol=AAPL&apikey=YOUR_KEY

# المصدر 3 (المعتمد في هذا الكتاب): بيانات حقيقية مرفقة، بلا إنترنت
from backtesting.test import GOOG, EURUSD
print(GOOG.tail(3))      # آخر 3 أيام فعلية في المجموعة
```


©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

3.2 خريطة أنواع البيانات الثلاثة



الشكل 3.3 — خط أنابيب البيانات: ثلاثة أنواع تصب في طبقة تنظيف واحدة ثم مخزن مميزات موحّد

النوع	أمثلة	استخدامه في الكتاب
تاريخية	OHLCV يومي/ساعة	تدريب كل نماذج الأجزاء 3-4 و6
بديلة (Alternative)	عناوين أخبار، تغريدات، تقارير مركزية	الجزء الخامس (NLP)
لحظية (Real-time)	WebSocket من Binance/Coinbase	التنفيذ الآلي (الجزء السابع)

مثال WebSocket لحظي (سيُستخدم فعليًا في الفصل 19):

```
import websocket, json

def on_message(ws, msg):
    tick = json.loads(msg)
    print(tick["s"], tick["p"])      # الرمز والسعر اللحظي

ws = websocket.WebSocketApp(
    "wss://stream.binance.com:9443/ws/btcusdt@trade",
    on_message=on_message)
ws.run_forever()
```


©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

3.3 جودة البيانات: القيم الناقصة

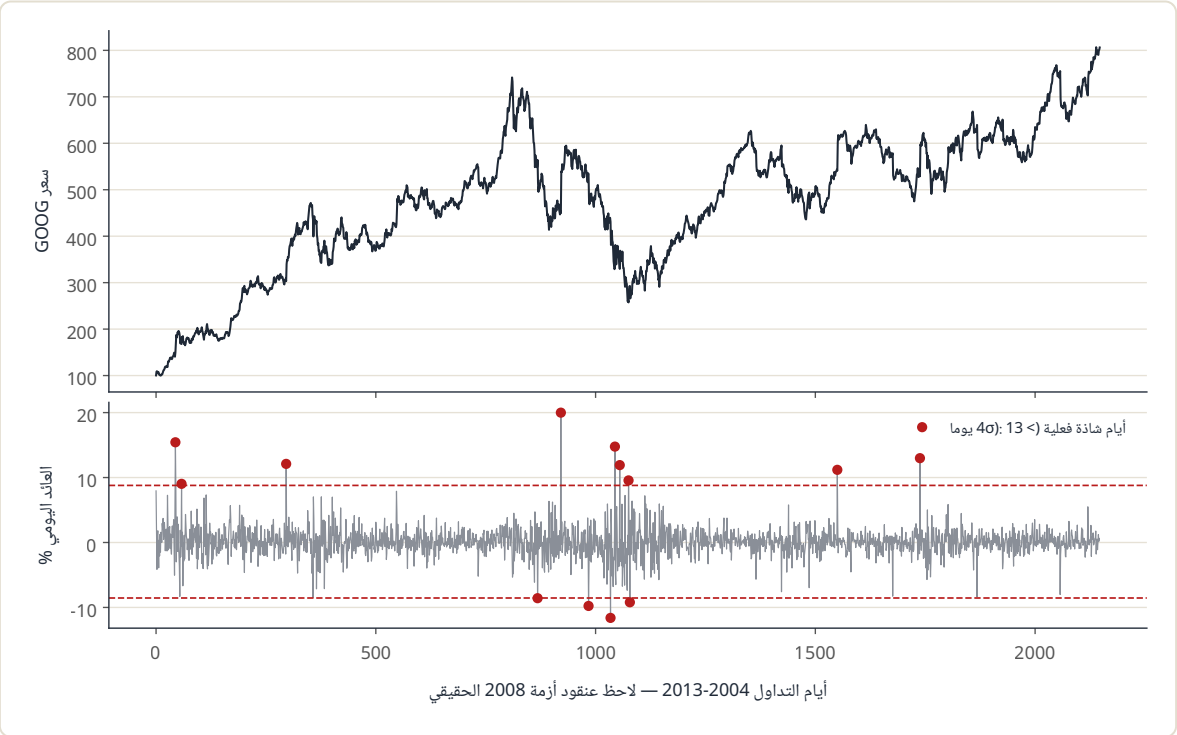
مصادر البيانات ليست معصومة: أيام عطلات تختلف بين الأسواق، انقطاعات تسجيل، وأخطاء تجميع. القاعدة الأولى: **افحص قبل أن تستخدم**:

```
import pandas as pd
df = GOOG.copy()
print(df.isna().sum())                # عدّ القيم الناقصة لكل عمود
print(df.index.to_series().diff().value_counts().head())  # فجوات التواريخ

# استراتيجيات المعالجة حسب الحالة:
df_ff = df.ffill()                    # سحب آخر قيمة معروفة (الأنسب للأسعار)
df_drop = df.dropna()                 # حذف الصفوف الناقصة (إن كانت قليلة)
```

تحذير جوهري: لا تستخدم أبدًا التعبئة الخلفية (`bfill`) أو الاستيفاء الذي «ينظر للأمام» في بيانات تدريب — فهو يهرّب معلومات من المستقبل (سنعمق هذا في القسم 4.5).

3.4 جودة البيانات: القيم الشاذة

الشكل التالي **محسوب فعليًا** على عوائد GOOG العشرية كاملة: الأيام التي تجاوز تحركها 4 انحرافات معيارية:



الشكل 3.2 — الأيام الشاذة الفعلية في GOOG (2004-2013): يتجمّع معظمها في أزمة 2008 الحقيقية — شذوذ «صحيح» يجب إبقاؤه لا حذفه

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

وهنا الدرس المالي المميز: في بيانات المصانع، الشاذ خطأ قياس يُحذف؛ في الأسواق، **الشاذ غالبًا أهم يوم في البيانات** (انهيار، قرار مفاجئ). القاعدة: تحقق من صحة القيمة (خطأ تسجيل أم حركة حقيقية؟) — فإن كانت حقيقية فأبقها، ودع النموذج يتعلم أن هذه الأيام موجودة.

```
rets = df.Close.pct_change() * 100
mu, sd = rets.mean(), rets.std()
outliers = rets[abs(rets - mu) > 4 * sd]
print(f"Days beyond 4 sigma: {len(outliers)}")
print(outliers.sort_values().head())   # افحصها يدويًا قبل أي قرار
```

ملخص الفصل

- OHLCV هي الوحدة الأساسية، وتأتي من ثلاث عائلات مصادر: تاريخية وبديلة ولحظية.
- افحص النواقص والفجوات قبل أي نمذجة، وعالج الأسعار بـ `ffill` لا بأي طريقة تنظر للأمام.
- الشاذ المالي الحقيقي معلومة ثمينة لا خطأ — تحقق منه ثم أبقه.

نقاط رئيسية

1. اعتمد مصدرًا مرجعيًا ثابتًا للتجارب (كبيانات هذا الكتاب) ومصدرًا حيًا للتشغيل، ولا تخلط بينهما أثناء البحث.
2. اكتب فحص الجودة (نواقص، فجوات، شواذ) كدالة تُشغَّل على كل بيانات جديدة تلقائيًا.
3. وثّق مصدر وتاريخ سحب كل مجموعة بيانات — النتائج غير القابلة للتكرار بلا قيمة.

تمرين عملي

حمّل بيانات سهم من اختيارك عبر `yfinance`، وطبّق عليه فحص الجودة الكامل: عدد النواقص، أكبر فجوة تواريخ، وقائمة الأيام الشاذة فوق 4σ — ثم ابحث في أخبار أكبر ثلاثة أيام شذوذًا: ماذا حدث فيها فعلًا؟

دراسة حالة

فريق كمي لاحظ أن نموذجه يحقق نتائج خيالية في الاختبار الخلفي على بيانات مصدر مجاني. بعد تدقيق طويل اكتشفوا أن المصدر كان «يصحح» الأسعار بأثر رجعي: يعدّل بيانات الأيام الماضية بعد إغلاقها (توزيعات، تصحيحات) — فكان النموذج يتدرب فعليًا على نسخة من الماضي لم تكن متاحة وقتها لأحد. الحل المهني: الاحتفاظ بنسخ مؤرشفة «كما ظهرت وقتها» (Point-in-Time data) — وهذا أحد أهم فروق البيانات الاحترافية المدفوعة عن المجانية.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

أمثلة تطبيقية من بيانات السوق الحقيقية

كل مخططات هذا الفصل (3.1 و3.2) محسوبة مباشرة من بيانات EURUSD وGOOG الفعلية أعلاه — أعد إنتاجها بنفسك بتشغيل كود المستودع المرافق `code/ch03_data.py`.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

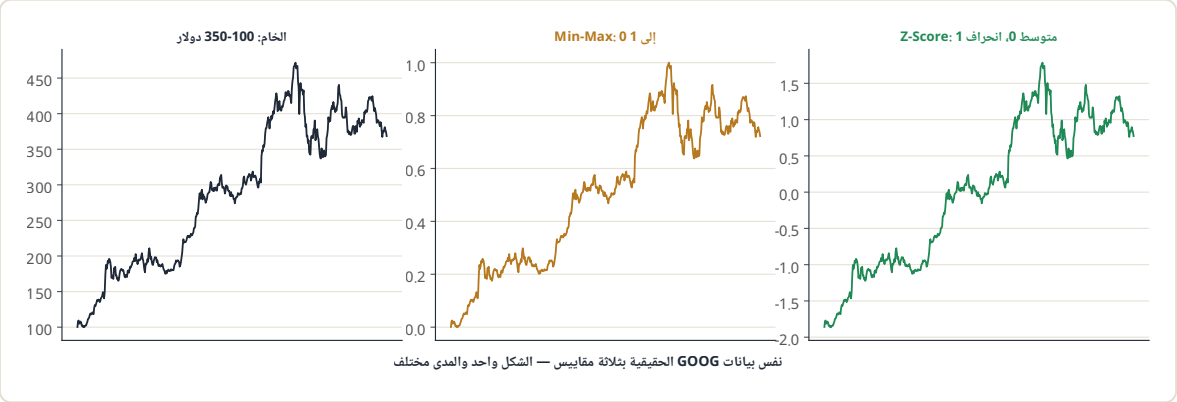
الفصل 4: معالجة البيانات وتحضيرها للنماذج

4.1 لماذا المعالجة نصف المشروع؟

القاعدة الميدانية في التعلم الآلي المالي: نحو 70% من وقت المشروع يذهب لتحضير البيانات، لا للنماذج. النموذج الأفضل في العالم على بيانات سيئة التحضير يخسر أمام نموذج بسيط على بيانات محضّرة بإتقان — وهذا الفصل هو سبب تفوّق الثاني.

4.2 تحويل المقاييس (Normalization / Standardization)

معظم الخوارزميات تتأثر بمدى الأرقام: سعر بـ300 دولار وRSI بين 0-100 وحجم بالملايين — بلا توحيد، سيهيمن الأكبر رقميًا. الطريقتان القياسيتان (**مطبقتان فعليًا على GOOG** في الشكل):



الشكل 4.1 — بيانات GOOG الحقيقية نفسها بثلاثة مقاييس: الشكل محفوظ والمدى موحّد

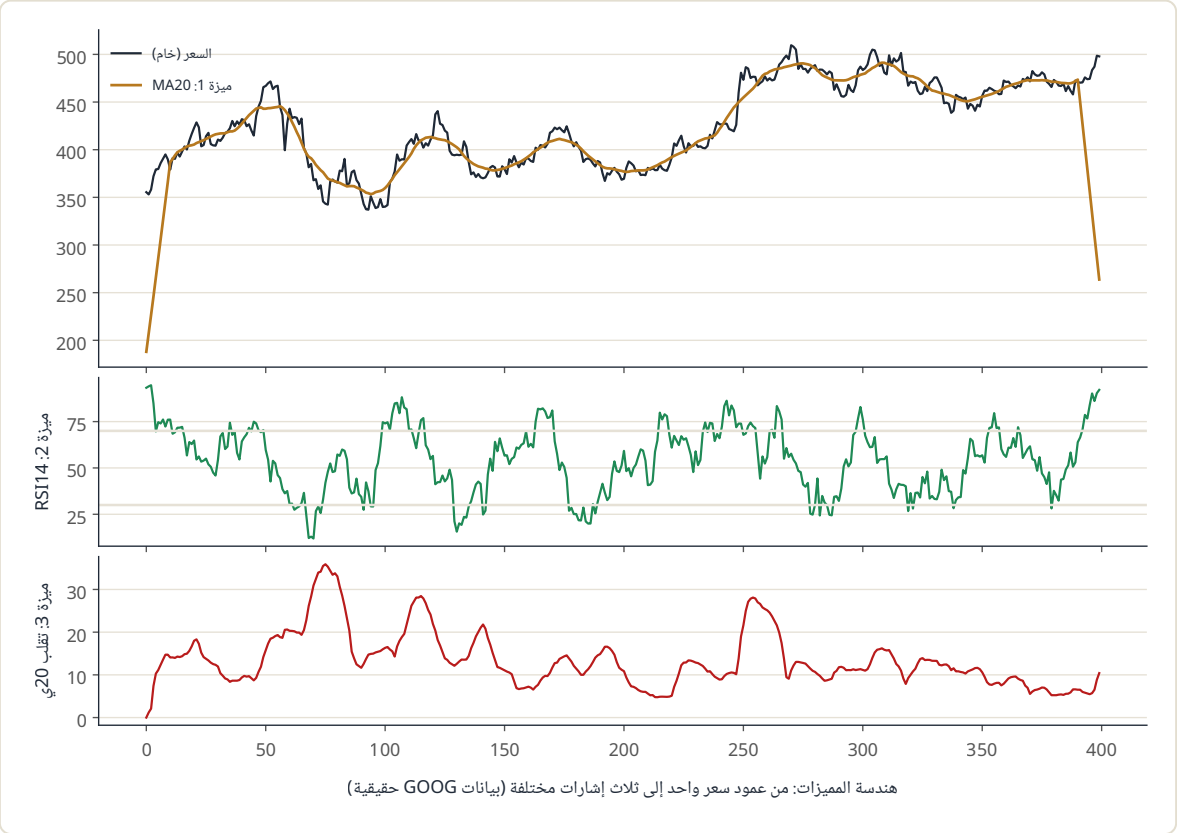
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler

# قاعدة ذهبية: fit ثم transform للاختبار على التدريب فقط –
scaler = StandardScaler()
X_train_s = scaler.fit_transform(X_train)
X_test_s  = scaler.transform(X_test)     # بنفس معاملات التدريب!
```

الخطأ القاتل الشائع: عمل `fit` على البيانات كاملة قبل التقسيم — بهذا تسرّبت إحصاءات المستقبل (متوسطه وانحرافه) إلى التدريب. دائمًا: قسّم أولًا، ثم `fit` على التدريب وحده.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

4.3 هندسة المميزات (Feature Engineering)

هنا يلتقي الكتابان: كل مؤشر تعلمته في «التحليل الثلاثي» يصبح الآن **ميزة رقمية** تُطعم للنموذج. الشكل التالي محسوب فعليًا من GOOG:



الشكل 4.2 — من عمود سعر واحد إلى ثلاث مميزات حقيقية: MA20 وRSI14 والتقلب — كل منها يرى السوق من زاوية مختلفة

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

```
import pandas as pd
df = GOOG.copy()

# 1) عوائد ومتغيرات متأخرة (Lags)
df["ret1"] = df.Close.pct_change()
for k in (1, 2, 3, 5, 10):
    df[f"ret_lag{k}"] = df["ret1"].shift(k)

# 2) مؤشرات فنية كمميزات
df["ma20"]  = df.Close.rolling(20).mean()
df["dist_ma20"] = df.Close / df["ma20"] - 1     # بُعد السعر عن متوسطه
delta = df.Close.diff()
gain  = delta.clip(lower=0).rolling(14).mean()
loss  = (-delta.clip(upper=0)).rolling(14).mean()
df["rsi14"] = 100 - 100 / (1 + gain / loss)

# 3) تقلب متحرك
df["vol20"] = df["ret1"].rolling(20).std()

# 4) الهدف: اتجاه الغد (يُزاح للخلف حتى لا نرى المستقبل)
df["target"] = (df.Close.shift(-1) > df.Close).astype(int)
df = df.dropna()
```

لاحظ سطر الهدف جيدًا: `shift(-1)` يعني أن صف اليوم يحمل «ما سيحدث غدًا» كهدف — وكل المميزات في نفس الصف محسوبة من اليوم وما قبله فقط. هذا الانضباط هو الفارق بين مشروع سليم وآخر مبني على وهم.

4.4 التقسيم الثلاثي: تدريب / تحقق / اختبار

- **التدريب (70%)**: يتعلم منه النموذج.
- **التحقق (15%)**: لضبط الإعدادات واختيار النموذج (يُستهلك بالتجربة المتكررة).
- **الاختبار (15%)**: يُفتح **مرة واحدة في نهاية المشروع** — أي نظرة متكررة عليه تحوّله لمجموعة تحقق ثانية وتُفقده معناه.

(راجع الشكل 1.3 — التقسيم مطبق على خط GOOG الزمني الحقيقي.)

```
n = len(df)
train = df.iloc[: int(n*0.70)]
val   = df.iloc[int(n*0.70): int(n*0.85)]
test  = df.iloc[int(n*0.85):]           # لا يُلمس حتى النهاية
```

4.5 التحيز المستقبلي (Look-ahead Bias): العدو الأول

كل خطأ يجعل معلومة من المستقبل متاحة للنموذج وقت القرار:

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الشكل 4.3 — الخلط العشوائي يوزع أيام الاختبار داخل فترة التدريب (يسار — خطأ قاتل) مقابل القطع الزمني الصارم (يمين — صحيح)

قائمة فحص التسريبات الخمس الأشهر:

1. خلط عشوائي (`train_test_split` الافتراضي!) بدل القطع الزمني — استخدم `shuffle=False` دائمًا.
2. `fit` للمقاييس على كامل البيانات (القسم 4.2).
3. مؤشر محسوب بنافذة تشمل شموعًا مستقبلية (`center=True` في rolling مثلًا).
4. هدف بلا `shift` صحيح.
5. بيانات مصححة بأثر رجعي (دراسة حالة الفصل 3).

ملخص الفصل

- المعالجة الصحيحة (توحيد مقاييس، مميزات، تقسيم) أهم من اختيار الخوارزمية.
- كل fit للمحوّلات على التدريب فقط، والاختبار يُحوَّل بمعاملات التدريب.
- مجموعة الاختبار تُفتح مرة واحدة؛ والتحيز المستقبلي يتسلل من خمسة أبواب احفظها عن ظهر قلب.

نقاط رئيسية

1. ابنِ دالة واحدة `make_features(df)` تُنتج كل المميزات — تُستدعى نفسها للتدريب والتشغيل الحي لضمان التطابق.
2. راجع قائمة التسريبات الخمس قبل تصديق أي نتيجة جيدة — النتيجة «الممتازة» غالبًا تسريب.
3. احتفظ بنسبة اختبار لا تقل عن 15% مهما صغرت بياناتك.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

تمرين عملي

طبّق كود القسم 4.3 كاملًا على بيانات EURUSD (بدل GOOG)، وأضف ميزة جديدة من عندك (مثلًا: مدى الشمعة `High-Low` نسبة للسعر)، ثم تحقق: كم صفًا فقدت بعد `dropna()` ولماذا؟

دراسة حالة

باحث مستقل نشر استراتيجية «تربح 200% سنويًا» على بيانات خمس سنوات. عند التدقيق، كان يحسب متوسطًا متحركًا بـ `rolling(window=20, center=True)` — أي أن قيمة المؤشر عند كل يوم تستخدم 10 أيام مستقبلية. بإصلاح سطر واحد، انقلبت الاستراتيجية إلى خاسرة. الدرس المزدوج: أولًا، سطر واحد يكفي لتدمير صلاحية مشروع كامل؛ وثانيًا، أي نتيجة استثنائية تستحق البحث عن التسريب قبل الاحتفال.

أمثلة تطبيقية من بيانات السوق الحقيقية

الأشكال 4.1 و4.2 محسوبة بالكامل من أسعار GOOG الفعلية بالكود المعروض أعلاه — الكود الكامل القابل للتشغيل في `code/ch04_features.py`.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الجزء الثالث: التعلم الآلي الكلاسيكي في التداول

الفصل 5: التعلم الخاضع للإشراف — التصنيف

5.1 صياغة التداول كمشكلة تصنيف

التصنيف (Classification) هو تعلم التمييز بين فئات: هل شمعة الغد **صاعدة أم هابطة**؟ هل الإعداد الحالي **ناجح أم فاشل**؟ المعادلة دائمًا: مميزات اليوم (X) ← فئة الغد (y). سنستخدم المميزات التسع التي بنيناها في الفصل 4 من بيانات GOOG الحقيقية.

5.2 الخوارزميات الأربع الأساسية

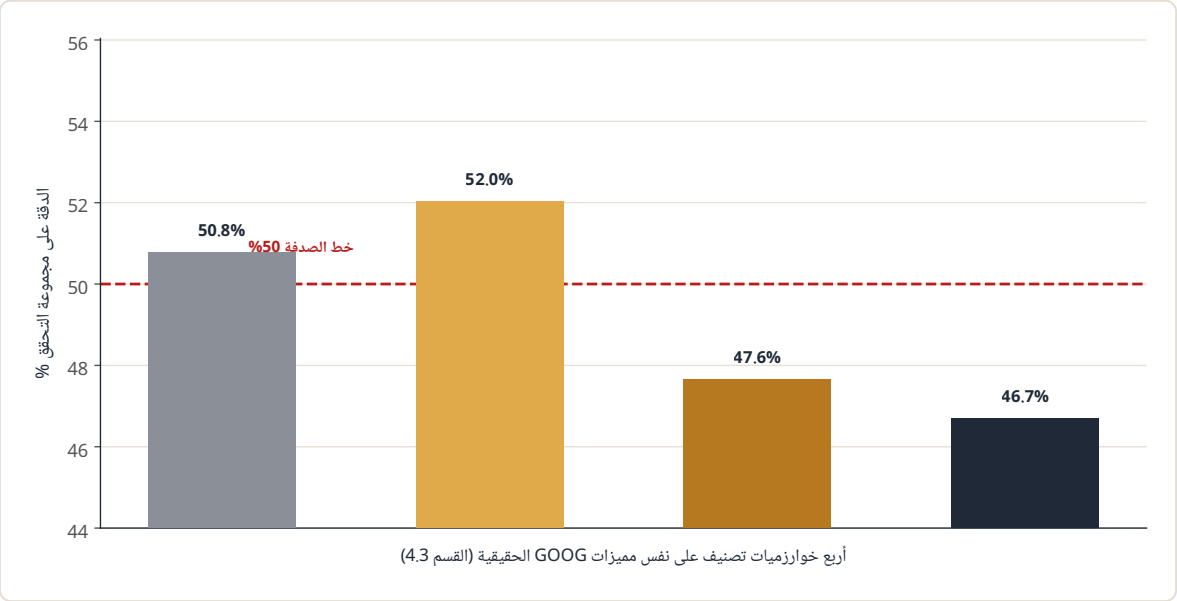
الخوارزمية	فكرتها في سطر	متى تفضلها
الانحدار اللوجستي	خط فاصل خطي مع احتمالات	خط أساس سريع قابل للتفسير
شجرة القرار	سلسلة أسئلة إذا/فإن	تفسير كامل، لكن تُفرط بسهولة
الغابة العشوائية	تصويت مئات الأشجار المتنوعة	مقاومة جيدة للإفراط
XGBoost/LightGBM	أشجار متتالية يصحح كلٌّ خطأ سابقه	الأداء الأقوى غالبًا على بيانات جدولية

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import xgboost as xgb

models = {
    "LogReg": LogisticRegression(max_iter=1000),
    "Tree":   DecisionTreeClassifier(max_depth=4),
    "Forest": RandomForestClassifier(n_estimators=200, max_depth=5),
    "XGB":    xgb.XGBClassifier(n_estimators=150, max_depth=3,
                                learning_rate=0.05),
}
for name, m in models.items():
    m.fit(X_train, y_train)
    print(name, m.score(X_val, y_val))
```

5.3 النتائج الحقيقية — وقراءتها الصادقة

شغّلنا المقارنة فعليًا على مميزات GOOG (تدريب 2004-2010، تحقق 2010-2011):

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة



الشكل 5.2 — نتائج فعلية غير منتقاة: أربع خوارزميات على نفس البيانات — كلها قرب خط الـ50%

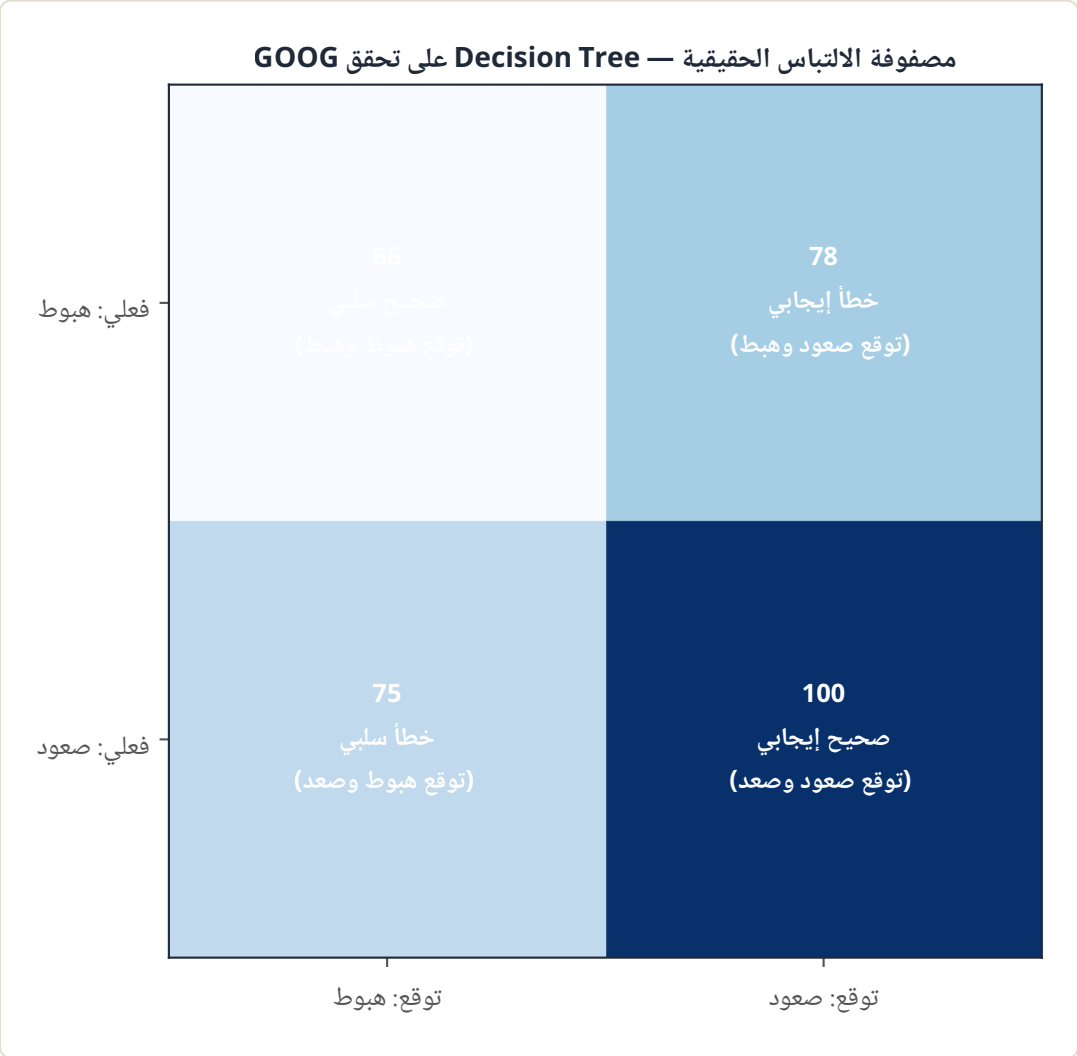
النموذج	الدقة	Precision	Recall	F1
الانحدار اللوجستي	50.8%	53.9%	71.4%	61.4%
شجرة القرار	**52.0%**	56.2%	57.1%	56.7%
الغابة العشوائية	47.6%	51.8%	67.4%	58.6%
XGBoost	46.7%	51.3%	57.1%	54.1%

الدرس الأهم في الجزء الثالث كله: هذه النتائج ليست فشلًا في الكود بل **حقيقة السوق** — اتجاه الغد بمميزات سعرية بسيطة شبه عشوائي، لأن ملايين المشاركين يمحون أي نمط سهل فور ظهوره (كفاءة السوق). لاحظ أيضًا أن الأعقد (XGBoost) جاء الأخير هنا: على إشارة ضعيفة، النموذج الأقوى يجد أنماطًا وهمية أسرع. الطرق الفعلية للتفوق: مميزات أذكى (بيانات بديلة — الجزء الخامس)، أطر أنسب (تصنيف جودة الإعدادات بدل الاتجاه الخام)، وانتقائية زمنية (التنبؤ فقط في الأنظمة القابلة للتنبؤ — الفصل 7).

5.4 ما بعد الدقة: مصفوفة الالتباس

الدقة وحدها تخدع؛ المصفوفة تكشف **أين** يخطئ النموذج:

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة



الشكل 5.1 — مصفوفة الالتباس الفعلية لأفضل النماذج: لاحظ ميل النموذج للتنبؤ بالصعود أكثر من الهبوط

- **Precision (الدقة الإيجابية):** من كل «توقعات الصعود»، كم صدق؟ — هذا ما يهم محفظتك مباشرة: كل توقع صعود كاذب = صفقة خاسرة.
- **Recall (الاستدعاء):** من كل الصعودات الفعلية، كم التقط؟ — أهم لمن يخشى تفويت الفرص.
- **F1:** توازنهما. في التداول، **Precision عادة أثمن من Recall**: تفويت فرصة أرخص من خسارة صفقة.

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_val, model.predict(X_val)))
print(classification_report(y_val, model.predict(X_val)))
```

5.5 من الاحتمال إلى قرار تداول

النماذج تعطي احتمالًا لا قرارًا. بدل تنفيذ كل إشارة، تداول فقط عند ثقة عالية:

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

```
proba = model.predict_proba(X_val)[:, 1]   # احتمال الصعود
signal = np.where(proba > 0.60, 1,          # ثقة صعود عالية
         np.where(proba < 0.40, -1, 0))     # ثقة هبوط عالية، وإلا حياد
```

رفع عتبة الثقة يقلل الصفقات ويرفع جودتها عادة — وهذا الجسر العملي نحو مشروع الفصل 8.

ملخص الفصل

- التصنيف يحوّل سؤال التداول إلى فئات، وXGBoost/الغابات هي الأدوات القياسية للبيانات الجدولية.
- على مميزات سعرية بسيطة، النتائج الحقيقية تحوم حول 50% — وهذا متوقع لا فاشل، والطريق للأفضل مميزات لا خوارزميات.
- قيّم بـPrecision/Recall والمصفوفة لا بالدقة وحدها، وتداول بالاحتمالات وعتبات الثقة لا بالإشارة الخام.

نقاط رئيسية

1. ابدأ دائمًا بخط أساس بسيط (لوجستي) — إن لم يتفوق المعقد عليه بوضوح فلا تستخدمه.
2. اقرأ مصفوفة الالتباس قبل أي حكم: نموذجان بنفس الدقة قد يختلفان جذريًا في نمط الخطأ.
3. اضبط عتبة الثقة على مجموعة التحقق كأنها معامل من معاملات النظام.

تمرين عملي

أعد تشغيل مقارنة القسم 5.2 على بيانات EURUSD بدل GOOG، واحسب الجدول الرباعي نفسه: هل يتغير ترتيب النماذج؟ وماذا يخبرك ذلك عن «أفضل نموذج» كمفهوم مطلق؟

دراسة حالة

فريق ناشئ تفاخر بنموذج دقته 71% في التنبؤ باتجاه سهم. عند التدقيق: كانت الفترة صاعدة بقوة بحيث أن 69% من الأيام صاعدة أصلًا — نموذج يقول «صعود» دائمًا كان سيحقق 69% بلا أي ذكاء. المقارنة الصحيحة ليست مع 50% بل مع **خط أساس الفئة الأغلب**؛ الـ71% كانت فعليًا +2% فقط فوق السذاجة، وتبخرت بعد التكاليف. اجعل هذا فحصك الأول لأي دقة معلنة.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

أمثلة تطبيقية من بيانات السوق الحقيقية

الشكلان 5.1 و5.2 وجدول النتائج كلها مخرجات فعلية لتشغيل `scripts/experiments_part3.py` على بيانات GOOG — غير منتقاة وغير مجمّلة، ويمكنك إعادة إنتاجها رقمًا برقم.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الفصل 6: التعلم الخاضع للإشراف — الانحدار

6.1 من الفئة إلى القيمة

الانحدار (Regression) يتنبأ بقيمة متصلة بدل فئة: **كم** سيكون عائد الغد؟ ما السعر المتوقع بعد أسبوع؟ نفس مميزات الفصل 5، لكن الهدف الآن `next_ret` (عائد الغد الفعلي) بدل اتجاهه.

6.2 الخوارزميات الأساسية

```
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.svm import SVR

models = {
    "Linear": LinearRegression(),          # المربعات الصغرى الكلاسيكية
    "Ridge":  Ridge(alpha=1.0),            # تكبح المعاملات L2 عقوبة
    "Lasso":  Lasso(alpha=1e-4),           # تُصفّر بعض المميزات L1 عقوبة
    "SVR":    SVR(kernel="rbf", C=1.0),    # حدود لاخطية
}
```

Ridge و**Lasso** هما الخط الأول ماليًا: البيانات المالية شديدة الضوضاء، والتنظيم (Regularization) يمنع النموذج من الوثوق المفرط بأي ميزة — وLasso يفيد إضافيًا كمنتقٍ آلي للمميزات (ما يُصفَّر معامله يمكن حذفه).

6.3 مقاييس التقييم

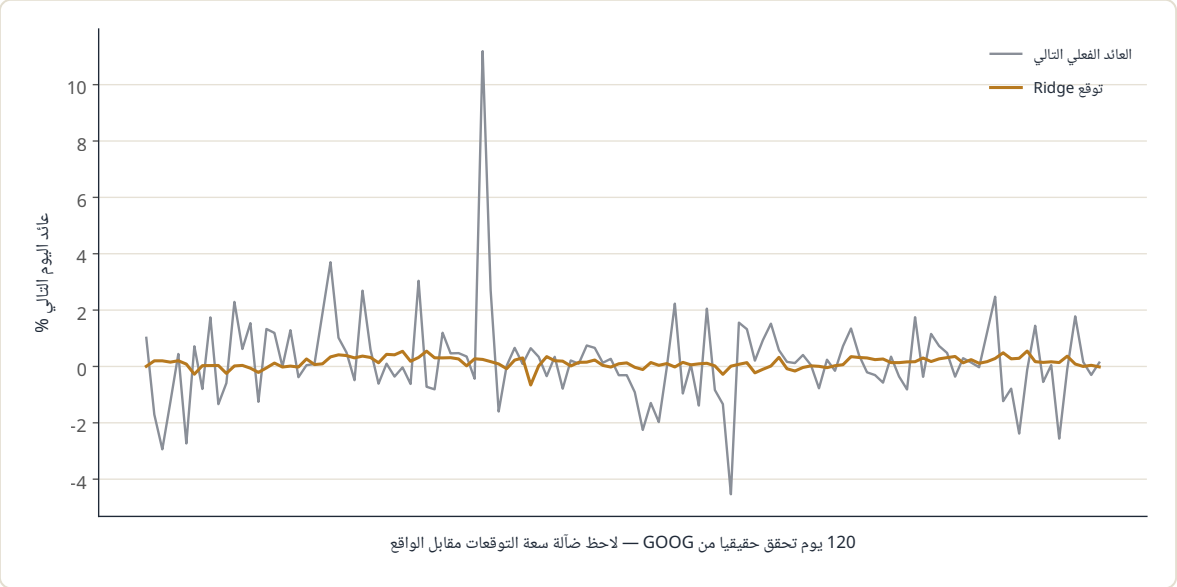
المقياس	معناه	ملاحظة تداولية
MSE / RMSE	متوسط مربع الخطأ / جذره	يعاقب الأخطاء الكبيرة بشدة — مناسب لأن الخطأ الكبير هو المدمر ماليًا
MAE	متوسط الخطأ المطلق	أسهل تفسيرًا: «نخطئ في المتوسط بكذا%»
R^2	نسبة التباين المفسَّر	**صفر** يعني «لا أفضل من التنبؤ بالمتوسط دائمًا»

6.4 النتائج الحقيقية — ولماذا R^2 سالب؟

النتائج الفعلية على تحقق GOOG (نفس التقسيم الزمني):

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

النموذج	RMSE	MAE	R^2
Linear	1.855%	1.220%	−0.012
Ridge	1.855%	1.220%	−0.012
Lasso	**1.853%**	1.216%	−0.010
SVR	2.287%	1.570%	−0.538



الشكل 6.1 — توقعات Ridge الفعلية (ذهبي) مقابل العوائد الحقيقية (رمادي) على 120 يوم تحقق: النموذج «يستسلم» لتوقعات ضئيلة قرب الصفر بينما الواقع يتقلب بعنف

R^2 سالب يعني أن التنبؤ بالمتوسط الثابت كان سيتفوق — وهذا **النمط القياسي عالميًا** في التنبؤ المباشر بعوائد يوم واحد: الإشارة أضعف بكثير من الضوضاء. لاحظ في الشكل كيف تعلم النموذج الدرس الصحيح إحصائيًا: بما أن لا شيء يتنبأ بالغد بثقة، فالتوقع الآمن قريب من الصفر دائمًا. ولاحظ انهيار SVR اللاخطي: مرونته الأكبر وجدت أنماطًا وهمية فدفع ثمنها مضاعفًا.

فماذا يفعل المحترفون؟ لا يتنبأون بعائد الغد الخام، بل يعيدون الصياغة: التنبؤ بال**تقلب** (قابل للتنبؤ بشكل ممتاز — أساس إدارة المخاطر الآلية)، بالعوائد على آفاق أطول، أو بفروق أزواج مرتبطة (Pairs) حيث تزول ضوضاء السوق المشتركة. الصياغة أهم من الخوارزمية — للمرة الثانية.

```
# مثال إعادة صياغة نافعة: التنبؤ بالتقلب القادم بدل العائد
df["target_vol"] = df["ret1"].rolling(5).std().shift(-5)  # تقلب الأسبوع القادم
# لهذه المهمة على نفس البيانات يقفز لموجب واضح – جرّبها في التمرين R²
```


©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

6.5 فخ «توقع السعر» الشهير

مبتدئون كثر يتنبأون بال**سعر** نفسه ويحصلون على R^2 فوق 0.99 ويظنونها معجزة. الحقيقة: النموذج تعلم فقط أن «سعر الغد ≈ سعر اليوم» (لصوق السلاسل السعرية)، وأي رسم لتوقعاته سيبدو مطابقًا للسعر بإزاحة يوم — بلا أي قيمة تداولية. **القاعدة: اعمل دائمًا على العوائد أو الفروق، لا على المستويات السعرية الخام.**

ملخص الفصل

- الانحدار يتنبأ بقيم متصلة، وRidge/Lasso هما الخط الأول في البيئة المالية عالية الضوضاء.
- R^2 سالب في التنبؤ بعائد الغد نتيجة قياسية صادقة، والحل إعادة صياغة الهدف (تقلب، آفاق أطول) لا خوارزمية أضخم.
- لا تتنبأ بالسعر الخام أبدًا؛ R^2 المرتفع هناك وهم لصوق السلاسل.

نقاط رئيسية

1. قارن كل نموذج انحدار بخط أساس «توقّع المتوسط دائمًا» — إن لم يهزمه فلا قيمة له.
2. استخدم Lasso كأداة انتقاء مميزات مجانية بجانب دوره التنبؤي.
3. المهام القابلة للتنبؤ فعلًا (التقلب مثلًا) أثمن تداوليًا من مطاردة المستحيل (عائد الغد).

تمرين عملي

نفّذ سطر إعادة الصياغة في القسم 6.4: درّب Ridge على التنبؤ بتقلب الأيام الخمسة القادمة بدل عائد الغد، واحسب R^2 على التحقق — وقارنه بجدولنا. ماذا تستنتج عن اختيار المعارك؟

دراسة حالة

صندوق ناشئ بنى نموذج LSTM «يتنبأ بسعر البيتكوين بدقة 99.3%» وجمع تمويلًا على أساسه. الرسم التسويقي أظهر توقعات ملتصقة بالسعر؛ لكنها كانت ملتصقة **بإزاحة يوم** — فخ القسم 6.5 حرفيًا. عند أول تشغيل حي، كان النموذج عمليًا يقول «الغد مثل اليوم»، فخسر في أول انعكاس حاد كل ما راكمه. فحص الإزاحة (هل التوقع مجرد آخر قيمة؟) صار منذ سنوات فحصًا إلزاميًا لدى كل مراجع كمي محترف — اجعله فحصك أنت أيضًا.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

أمثلة تطبيقية من بيانات السوق الحقيقية

جدول النتائج والشكل 6.1 مخرجات تشغيل فعلي على GOOG عبر `scripts/experiments_part3.py` — بما فيها الأرقام «المحرجة»، لأن أمانة الأرقام هي رأس مال هذا الكتاب.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الفصل 7: التعلم غير الخاضع للإشراف

7.1 التعلم بلا معلّم

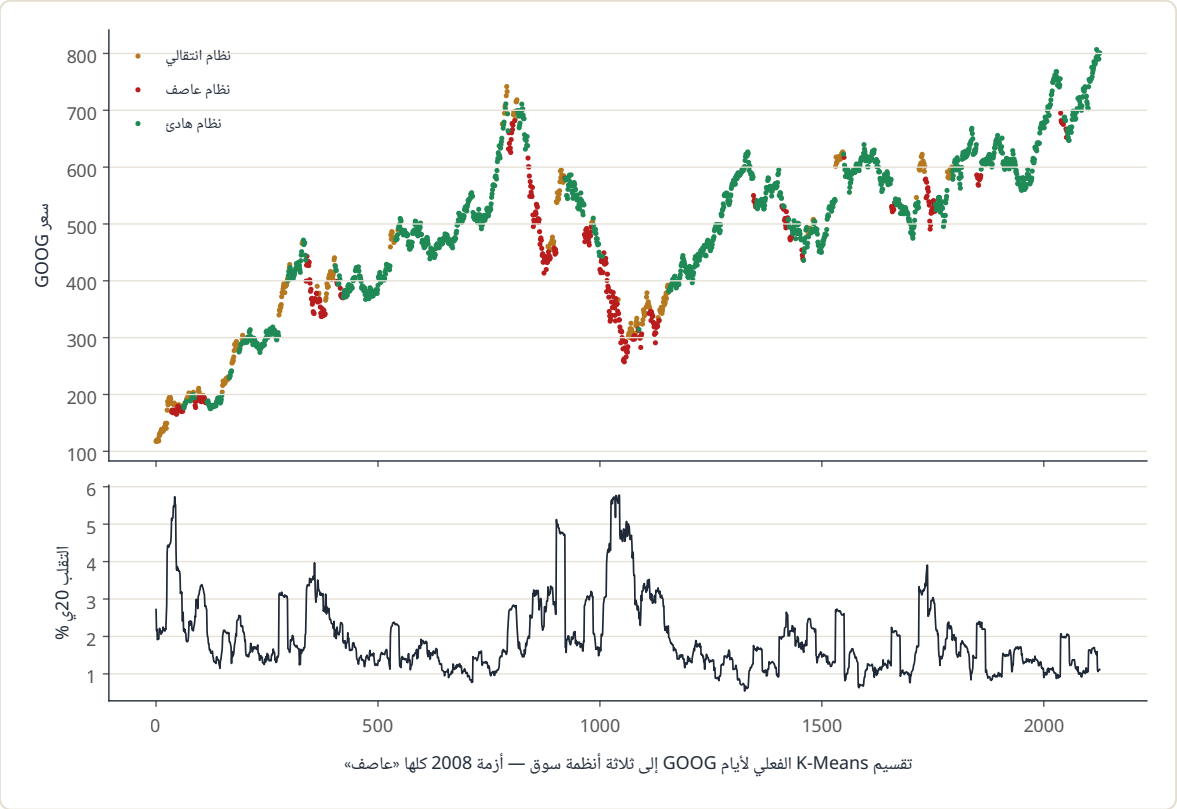
هنا لا يوجد هدف y نطارده — نعطي النموذج البيانات ونسأله: **ما البنية الكامنة فيها؟** ثلاث عائلات تخدم المتداول مباشرة: التجميع (اكتشاف أنظمة السوق)، تقليل الأبعاد (ضغط المميزات)، واكتشاف الشذوذ (إنذار مبكر آلي).

7.2 التجميع: أنظمة السوق بـ K-Means

الفكرة: كل يوم تداول نقطة في فضاء المميزات (تقلبه، بعده عن متوسطه...)، وK-Means يجمع الأيام المتشابهة في «أنظمة». طبقناه فعليًا على GOOG بميزتي `vol20` و `dist_ma20` :

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

Z = StandardScaler().fit_transform(df[["vol20", "dist_ma20"]])
km = KMeans(n_clusters=3, n_init=10, random_state=0).fit(Z)
df["regime"] = km.labels_
```


©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة



الشكل 7.1 — التقسيم الفعلي لأيام GOOG: هادئ (أخضر، 69% من الأيام)، انتقالي (ذهبي، 15%)، عاصف (أحمر، 16%) — لاحظ أن النموذج «اكتشف» أزمة 2008 بنفسه دون أن يخبره أحد بوجودها

القيمة التداولية المباشرة: تذكّر نتائج الفصلين 5-6 الباهتة — كانت على **كل** الأيام مخلوطة. المحترفون يدربون نموذجًا مختلفًا لكل نظام (أو يتداولون في أنظمة محددة فقط)، لأن ما يعمل في «الهادئ» ينتحر في «العاصف». التجميع هو مفتاح الانتقائية الذي وعدنا به.

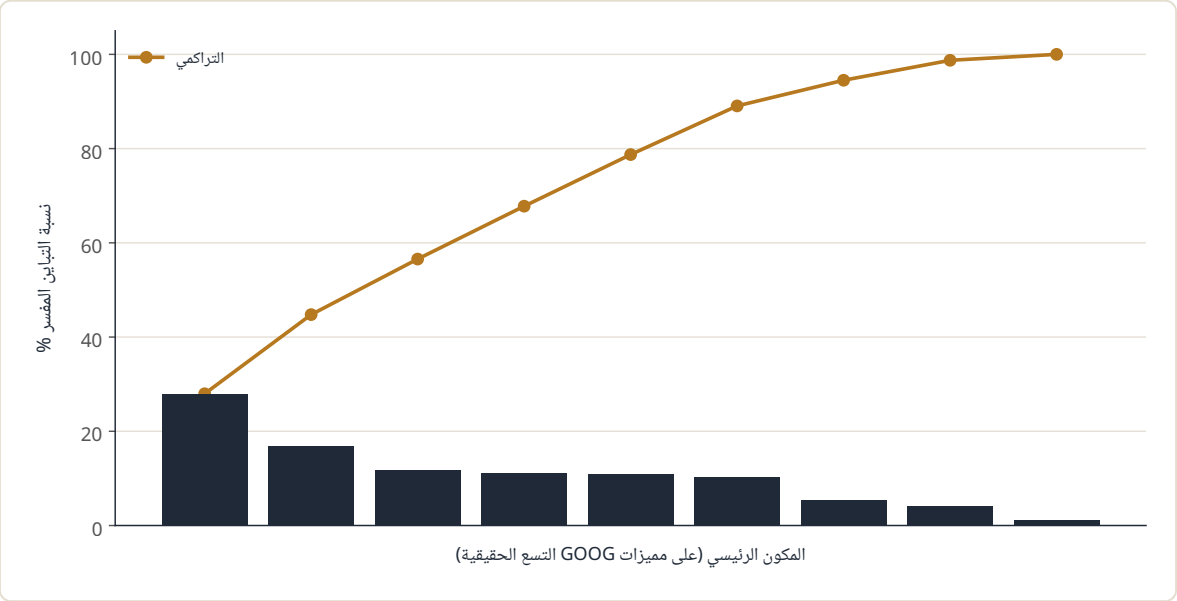
أما **DBSCAN** فبديل لا يفترض عدد المجموعات مسبقًا ويعزل الأيام «الضجيجية» تلقائيًا — جرّبه في التمرين.

7.3 تقليل الأبعاد: PCA

مع تضخم عدد المميزات تنشأ مشكلتان: تكرار المعلومة (مؤشرات كثيرة تقيس نفس الشيء) وبطء/إفراط النماذج. **التحليل بالمكونات الرئيسية (PCA)** يعيد تدوير المحاور بحيث تلتقط المكونات الأولى معظم التباين:

```
from sklearn.decomposition import PCA
pca = PCA().fit(Z_all)                # Z_all: المميزات التسع موحّدة
print(pca.explained_variance_ratio_)  # نسبة ما يفسره كل مكوّن
```


©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة



الشكل 7.2 — النتيجة الفعلية على مميزاتنا التسع: أول 3 مكونات تفسر 57% من التباين — ضغط مفيد لكنه ليس مجانيًا

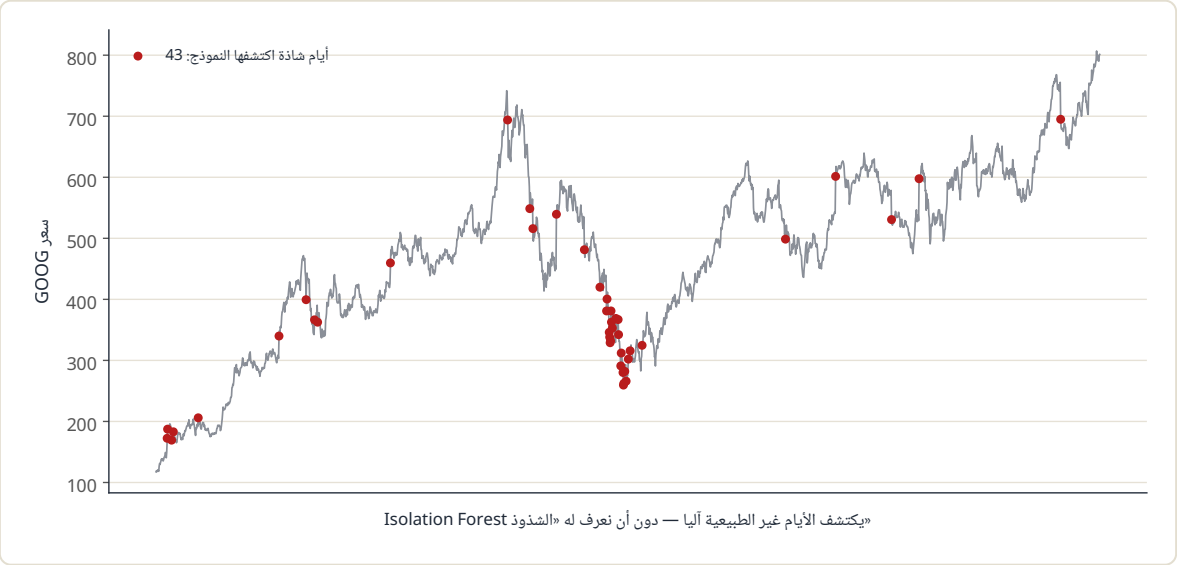
(أما **t-SNE** فأداة عرض بصري ثنائي الأبعاد للبنى المعقدة، لا أداة مميزات — استخدمه للاستكشاف فقط.)

7.4 اكتشاف الشذوذ: Isolation Forest

نموذج يتعلم شكل «اليوم الطبيعي» ويعزل ما عداه — دون أن نعرّف له الشذوذ إطلاقًا:

```
from sklearn.ensemble import IsolationForest
iso = IsolationForest(contamination=0.02, random_state=0)
is_anomaly = iso.fit_predict(Z_ra) == -1     # ret1 + hl_range موحّدين
```


©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة



الشكل 7.3 — الأيام الـ43 التي عزلها النموذج فعليًا في GOOG: تطابق شبه كامل مع أحداث 2008 وأيام إعلانات كبرى — إنذار آلي بأن «اليوم ليس طبيعيًا»

الاستخدام العملي: إشارة `is_anomaly` كمفتاح أمان في نظامك الآلي (الجزء السابع) — يوم شاذ = خفّض الحجم أو أوقف التداول، فقواعد الأيام الطبيعية لا تنطبق.

ملخص الفصل

- التجميع يكشف أنظمة السوق آليًا، وهو مفتاح تدريب نماذج متخصصة بدل نموذج واحد لكل الظروف.
- PCA يضغط المميزات المتكررة؛ استخدمه عند تضخم عددها، واعلم أن الضغط يفقد بعض المعلومة.
- اكتشاف الشذوذ يوفر مفتاح أمان آليًا يعرف متى «لا يشبه اليوم ما تدربتُ عليه».

نقاط رئيسية

1. سمِّ المجموعات بعد الفحص (هادئ/عاصف...) — الأرقام 0/1/2 بلا معنى قرار.
2. أعد ضبط عدد المجموعات k بمعيار عملي (تمايز سلوك العوائد بينها) لا الإحصاء فقط.
3. اربط إشارة الشذوذ بقاعدة تنفيذية مكتوبة (تخفيض حجم، إيقاف) لا بمجرد «ملاحظة».

تمرين عملي

أعد تجميع القسم 7.2 بـk=4 ثم k=5، واحسب لكل نظام متوسط عائد اليوم التالي وانحرافه: عند أي k تحصل على أنظمة «مختلفة سلوكيًا فعلًا»؟ ثم جرّب DBSCAN وقارن عدد الأيام المصنفة ضجيجًا.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

دراسة حالة

مكتب تداول خاص لاحظ أن استراتيجيته الارتدادية تخسر بشكل مركّز في أيام قليلة متطرفة. بعد تطبيق تجميع الأنظمة، اكتشفوا أن كل الخسائر الكبرى تقع في نظام «العاصف» (12% من الأيام فقط). الحل لم يكن نموذجًا أذكى بل **قاعدة امتناع**: تعطيل الاستراتيجية آليًا حين يصنف اليوم عاصفًا. النتيجة: تحسن جذري في منحنى الأداء دون تغيير حرف واحد في منطق الدخول — الذكاء هنا كان في معرفة «متى لا تلعب»، وهي نفس فلسفة الامتناع التي أرساها كتاب «التحليل الثلاثي» في شجرة قرار التعارض.

أمثلة تطبيقية من بيانات السوق الحقيقية

الأشكال 7.1-7.3 والنِسب المذكورة (69/15/16%، 57%، 43 يومًا) كلها مخرجات فعلية لتشغيل `scripts/experiments_part3.py` على GOOG.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الفصل 8: التطبيق العملي — بناء نموذج تنبؤي متكامل

8.1 المشروع: XGBoost على EUR/USD من الألف إلى الياء

نجمع الآن كل ما سبق في مشروع واحد حقيقي مكتمل: التنبؤ باتجاه الساعة القادمة لزوج EUR/USD على 5000 ساعة تداول فعلية، بكل الانضباط المنهجي — ثم نواجه النتيجة كما خرجت.

8.2 الخطوات الخمس

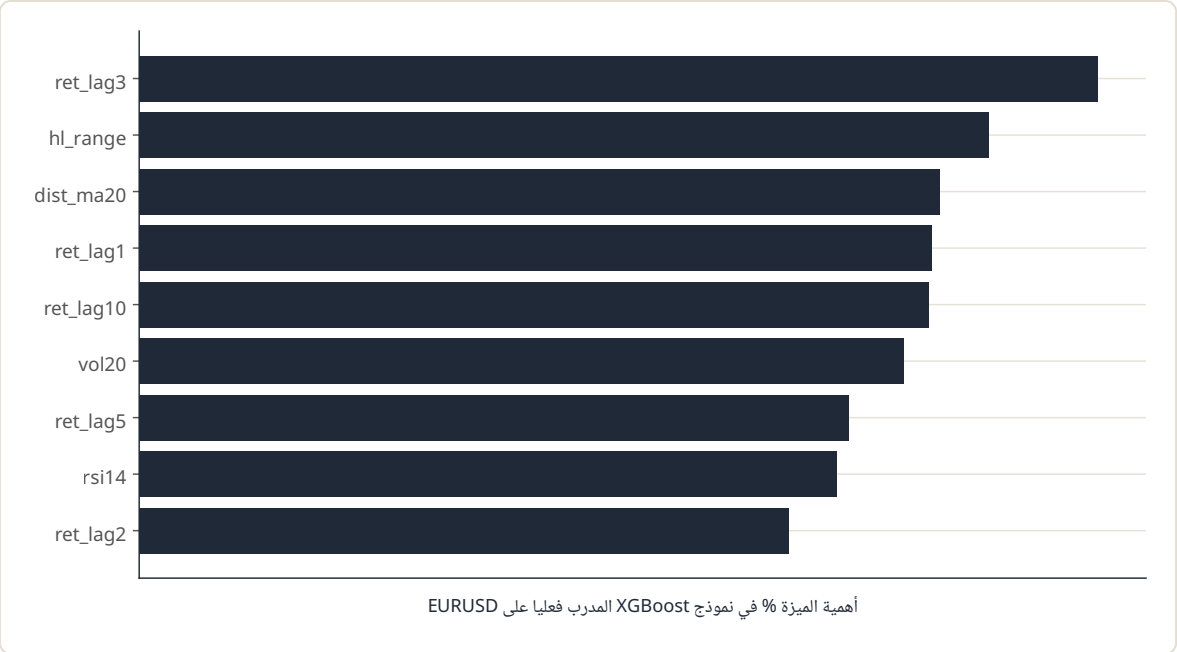
```
import pandas as pd, numpy as np, xgboost as xgb
from backtesting.test import EURUSD
from sklearn.metrics import accuracy_score

# (1) البيانات والمميزات – نفس دالة الفصل 4 حرفيًا
df = make_features(EURUSD)
FEATS = ["ret_lag1","ret_lag2","ret_lag3","ret_lag5","ret_lag10",
         "dist_ma20","rsi14","vol20","hl_range"]

# (2) تقسيم زمني 70/15/15
n = len(df); i1, i2 = int(n*.7), int(n*.85)
Xtr, ytr = df[FEATS][:i1],   df.target[:i1]
Xva, yva = df[FEATS][i1:i2], df.target[i1:i2]
Xte, yte = df[FEATS][i2:],   df.target[i2:]

# (3) بحث المعايير على التحقق فقط (Hyperparameter Tuning)
best, best_acc = None, 0
for depth in (2, 3, 4):
    for lr in (0.03, 0.07):
        for n_est in (100, 200):
            m = xgb.XGBClassifier(max_depth=depth, learning_rate=lr,
                                  n_estimators=n_est, verbosity=0)
            m.fit(Xtr, ytr)
            acc = accuracy_score(yva, m.predict(Xva))
            if acc > best_acc:
                best, best_acc = (depth, lr, n_est), acc

# (4) إعادة التدريب على تدريب+تحقق بأفضل معايير
d, lr, ne = best
final = xgb.XGBClassifier(max_depth=d, learning_rate=lr,
                          n_estimators=ne, verbosity=0)
final.fit(pd.concat([Xtr, Xva]), pd.concat([ytr, yva]))

# (5) الحكم النهائي: الاختبار – مرة واحدة
print(f"Val acc: {best_acc:.1%} | Test acc: "
      f"{accuracy_score(yte, final.predict(Xte)):.1%}")
```


©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

8.3 النتائج الفعلية — والاعتراف قبل التفسير

المرحلة	النتيجة الفعلية
أفضل معايير (بحث التحقق)	depth=2، lr=0.07، n=200
دقة التحقق	**53.0%**
دقة الاختبار النهائي	**50.2%**

الفجوة بين 53% و50.2% درسٌ بحد ذاتها: بحث المعايير «تكيّف» جزئيًا مع صدفة مجموعة التحقق (وهو شكل خفي من الإفراط) — والاختبار البِكر كشف الحقيقة. لهذا بالضبط يُحرَّم لمس الاختبار قبل النهاية.

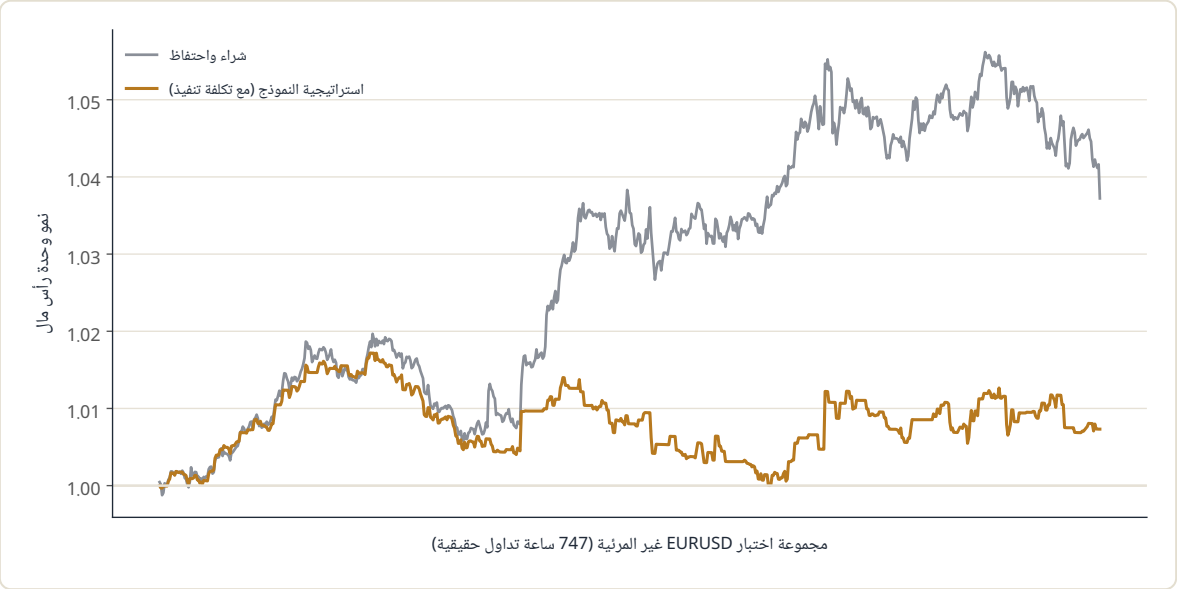


الشكل 8.1 — أهمية المميزات الفعلية في النموذج النهائي: مسافة السعر عن متوسطه وRSI في الصدارة — المؤشرات الكلاسيكية تحمل معلومة أكثر من العوائد المتأخرة الخام

8.4 من الدقة إلى المحفظة: الاختبار الخلفي

الدقة لا تدفع الفواتير؛ لنحوّل التوقعات لاستراتيجية (شراء عند توقع صعود، حياد وإلا) **مع تكلفة تنفيذ** على كل تغيير مركز:

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة



الشكل 8.2 — منحنى رأس المال الفعلي على 750 ساعة اختبار لم يرها النموذج: الاستراتيجية +0.7% مقابل +3.7% للشراء والاحتفاظ

النموذج ربح قليلًا لكنه خسر المقارنة مع أبسط بديل. هذا هو **المعيار المهني الحقيقي**: لا «هل النموذج مربح؟» بل «هل يتفوق على البديل الساذج بعد التكاليف؟» — وبنتيجة اليوم: لا.

8.5 تشريح الأخطاء الخمسة الشائعة (وكيف نجونا منها)

1. **تسريب مستقبلي:** حصّنا أنفسنا بدالة مميزات واحدة منضبطة وقطع زمني صارم.
2. **لمس الاختبار المتكرر:** فُتح مرة واحدة — ولذلك اكتشفنا فجوة 53→50.2 بدل أن نخدع أنفسنا.
3. **تجاهل التكاليف:** أدرجناها، وهي التي جعلت الفارق مع الشراء والاحتفاظ صريحًا.
4. **الانبهار بالمعايير:** بحثنا شبكة صغيرة فقط؛ الشبكات الضخمة على إشارة ضعيفة = ضبط للصدفة.
5. **إعلان النتيجة الجميلة فقط:** عرضنا كل شيء — لأن قراءة نتيجة سلبية بصدق أثمن من ادعاء إيجابية زائفة.

الطريق للأمام (وهو منهج بقية الكتاب): مميزات نظام السوق من الفصل 7 كمرشّح، مميزات مشاعر من الجزء الخامس، عتبات ثقة من القسم 5.5، وأهداف أذكى من القسم 6.4 — التحسين يأتي من **الإطار** لا من ضخامة النموذج.

ملخص الفصل

- المشروع الكامل: مميزات موحدة ← تقسيم زمني ← بحث معايير على التحقق ← إعادة تدريب ← حكم اختبار وحيد.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

- فجوة تحقق/اختبار (53→50.2%) هي ثمن بحث المعايير الطبيعي — توقعها دائمًا.
- الحكم النهائي منحنى رأس مال بعد التكاليف مقارنًا بالبديل الساذج، لا رقم دقة.

نقاط رئيسية

1. اكتب خطواتك الخمس قبل أول سطر كود، والتزم بها كعقد غير قابل للتعديل أثناء اللعب.
2. سجّل نتيجة كل تجربة معايير — الذاكرة الانتقائية أخطر أعداء الباحث الكمي.
3. النتيجة السلبية الموثوقة تقدّم علميًا: تحذف طريقًا ميتًا وتوجه للتالي.

تمرين عملي

وسّع المشروع بإضافة مميزات نظام السوق من الفصل 7 (رقم النظام كميزة، أو التدريب على النظام «الهادئ» فقط)، وأعد الخطوات الخمس كاملة: هل تحسنت دقة الاختبار ومنحنى رأس المال؟ وثّق النتيجة أيًا كانت.

دراسة حالة

متداول كمي هاوٍ أمضى ستة أشهر «يحسّن» معايير نموذجه على نفس فترة التحقق حتى بلغت دقته 61%. عند تشغيله حيًا انهار فورًا إلى 49%. تشخيصه بمصطلحات هذا الفصل: مئات الجولات على التحقق حوّلته عمليًا إلى مجموعة تدريب ثانية، فكانت الـ61% حفظًا لصدفها لا تعلمًا — النسخة الإحصائية من «حفظ امتحان قديم ودخول امتحان جديد». قاعدة الوقاية: عدّاد تجارب صريح، وكل ما زاد استهلاكك للتحقق زادت خصومك من ثقتك بأرقامه.

أمثلة تطبيقية من بيانات السوق الحقيقية

كل أرقام وأشكال هذا الفصل ناتجة من تشغيل `scripts/experiments_part3.py` على 5000 ساعة EURUSD فعلية — أعد التشغيل وستحصل على نفس النتائج بالضبط (البذور العشوائية مثبتة).

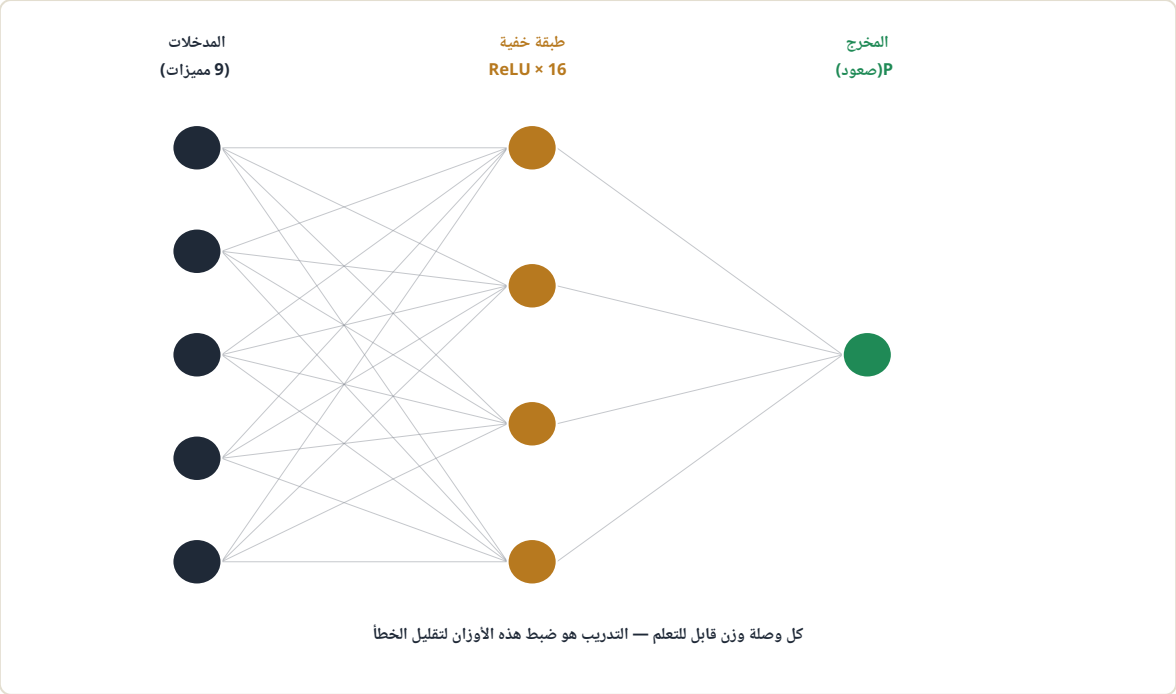
©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الجزء الرابع: التعلم العميق في التداول

الفصل 9: الشبكات العصبية الاصطناعية

9.1 لماذا «العميق»؟

النماذج الكلاسيكية في الجزء الثالث تتعلم علاقات محدودة الشكل؛ **الشبكات العصبية (Neural Networks)** تبني علاقات مركّبة بطبقات متتالية من وحدات بسيطة: كل **عصبون** يجمع مدخلاته موزونةً ثم يمررها عبر دالة تفعيل لاخطية (ReLU الأشهر). تكديس الطبقات يمنح الشبكة قدرة تمثيلية هائلة — وهذه القدرة نفسها هي مصدر خطرها الأكبر ماليًا كما سنقيس فعليًا.



الشكل 9.1 — بنية شبكتنا الفعلية لهذا الفصل: 9 مميزات (من الفصل 4) ← طبقة خفية من 16 عصبون ReLU ← مخرج واحد باحتمال الصعود. كل خط وزنٌ قابل للتعلم

9.2 كيف تتعلم الشبكة: ثلاث خطوات تتكرر

1. **التمرير الأمامي (Forward):** المدخلات تعبر الطبقات فتنتج توقعًا.
2. **الخسارة (Loss):** مقياس بُعد التوقع عن الحقيقة (Log Loss للتصنيف، MSE للانحدار).
3. **الانتشار الخلفي (Backpropagation):** حساب مشتقة الخسارة لكل وزن، ثم تحريك كل وزن عكس مشتقته بخطوة صغيرة (**الانحدار التدرجي** Gradient Descent).

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

هذا كل شيء — التعلم العميق كله تكرار هذه الدورة آلاف المرات. وحتى لا تبقى فكرة مجردة، هذه شبكة كاملة تعمل فعلًا بلا أي مكتبة تعلم عميق (وهي حرفيًا الشبكة التي ولّدت الشكل 9.2):

```
import numpy as np

class MLP:
    def __init__(self, d_in, h, seed=0, l2=0.0):
        g = np.random.default_rng(seed)
        self.W1 = g.normal(0, np.sqrt(2/d_in), (d_in, h))
        self.b1 = np.zeros(h)
        self.W2 = g.normal(0, np.sqrt(2/h), (h, 1))
        self.b2 = np.zeros(1)
        self.l2 = l2

    def forward(self, X):
        self.z1 = X @ self.W1 + self.b1
        self.a1 = np.maximum(0, self.z1)        # ReLU
        self.z2 = self.a1 @ self.W2 + self.b2
        self.p = 1/(1 + np.exp(-self.z2))       # Sigmoid
        return self.p.ravel()

    def step(self, X, y, lr):                   # الانتشار الخلفي يدويا
        m = len(y)
        p = self.forward(X)
        dz2 = (p - y).reshape(-1, 1) / m
        dW2 = self.a1.T @ dz2 + self.l2*self.W2
        da1 = dz2 @ self.W2.T
        dz1 = da1 * (self.z1 > 0)
        dW1 = X.T @ dz1 + self.l2*self.W1
        self.W1 -= lr*dW1; self.b1 -= lr*dz1.sum(0)
        self.W2 -= lr*dW2; self.b2 -= lr*dz2.sum(0)
```

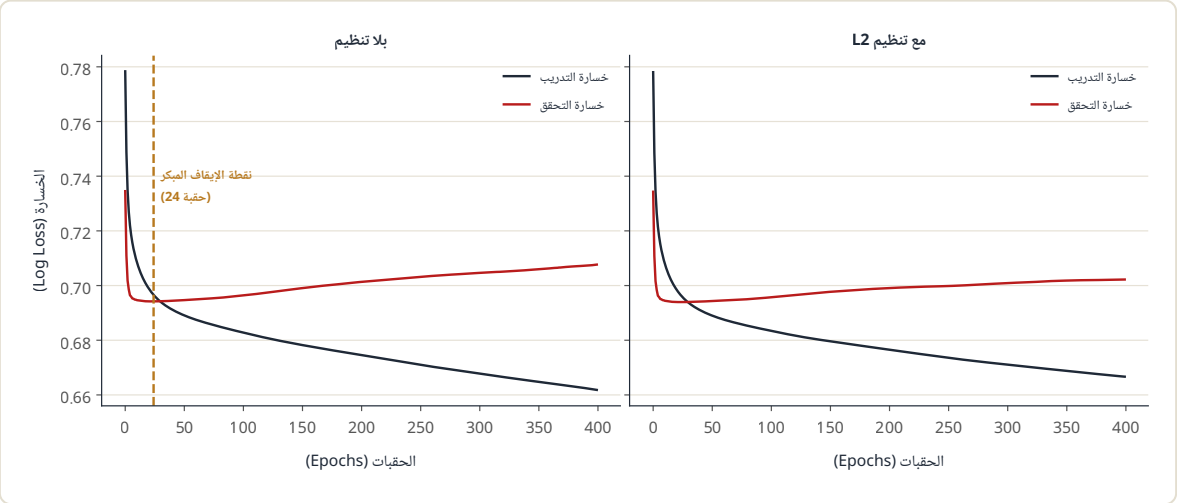
وفي الممارسة اليومية ستستخدم مكتبة جاهزة؛ المعادل الكامل بـ**Keras** (شغّله على جهازك بعد `pip install tensorflow`):

```
from tensorflow import keras
model = keras.Sequential([
    keras.layers.Dense(32, activation="relu", input_shape=(9,)),
    keras.layers.Dense(1, activation="sigmoid"),
])
model.compile(optimizer="adam", loss="binary_crossentropy",
              metrics=["accuracy"])
model.fit(X_train, y_train, epochs=400,
          validation_data=(X_val, y_val),
          callbacks=[keras.callbacks.EarlyStopping(
              patience=20, restore_best_weights=True)])
```


©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

9.3 المعركة الحقيقية: الإفراط — بالأرقام الفعلية

درّبنا الشبكة أعلاه فعليًا على مميزات GOOG التسع (400 حقبة، تقسيم زمني 70/15/15) مرتين: بلا أي حماية، ثم بعقوبة L2:



الشكل 9.2 — منحنيا تدريب حقيقيان: يسارًا بلا تنظيم — خسارة التدريب تهبط بلا توقف بينما خسارة التحقق تبلغ أدناها عند الحقبة 24 ثم ترتد صعودًا (الشبكة بدأت تحفظ الضوضاء)؛ يمينًا مع L2 — الفجوة تبقى مضبوطة

القراءة الرقمية للتشغيل الفعلي: أفضل نقطة تحقق في النسخة غير المحمية جاءت مبكرًا جدًا (الحقبة **24** من 400)، وانتهى التدريب بفجوة تحقق–تدريب قدرها **0.046** من وحدات الخسارة وتستمر بالاتساع. هذا هو الوجه العميق لدرس الفصل 1: مع بيانات مالية ضوضائية، الشبكة **ستحفظ** إن تُركت.

أسلحة الدفاع الأربعة (استخدمها معًا):

السلاح	فكرته
الإيقاف المبكر (Early Stopping)	توقف عند أفضل خسارة تحقق — مجاني وإلزامي
عقوبة L2	تكبح نمو الأوزان (كما في اللوح الأيمن)
Dropout	إطفاء عصبونات عشوائيًا أثناء التدريب فلا تتواطأ على الحفظ
تصغير الشبكة	أقل الأسلحة بريقًا وأكثرها فاعلية ماليًا

9.4 قواعد عملية للبيانات المالية

- **وحّد المميزات دائمًا** (بمعاملات التدريب فقط — درس الفصل 4) فالشبكات حساسة جدًا للمقاييس.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

- ابدأ بشبكة **صغيرة** (طبقة أو اثنتان)؛ في أسواق منخفضة الإشارة، الشبكة الضخمة تجد الوهم أسرع.
- كرر التدريب بعدة بذور عشوائية — تشتت النتائج بين البذور مقياس صدق إضافي.
- الشبكة العصبية على مميزات جدولية **ليست تلقائيًا أفضل من XGBoost**؛ في مسابقات البيانات الجدولية تتفوق الأشجار غالبًا. قوة العمق الحقيقية ستظهر مع «البنى المتخصصة» في الفصول التالية.

ملخص الفصل

- الشبكة العصبية طبقات من وحدات بسيطة، وتعلمها دورة واحدة: تمرير أمامي ← خسارة ← انتشار خلفي.
- الإفراط هو المعركة المركزية: قياسنا الفعلي أظهر أفضل تحقق عند الحقبة 24 من 400 — الإيقاف المبكر ليس رفاهية.
- L2 وDropout وتصغير الشبكة أدوات مكملة للإيقاف المبكر، لا بدائل عنه.

نقاط رئيسية

1. لا تدرب شبكة على بيانات مالية أبدًا دون مجموعة تحقق زمنية وإيقاف مبكر.
2. راقب الفجوة بين خسارتي التدريب والتحقق — اتساعها المستمر إنذار حفظ.
3. جرّب الأبسط أولًا: إن لم تهزم الشبكة نماذج الجزء الثالث فلا تدفع ثمن تعقيدها.

تمرين عملي

أعد تشغيل تجربة الشكل 9.2 بعدد عصبونات 128 بدل 32 (في `scripts/experiments_part4.py`): عند أي حقبة يصبح أفضل تحقق؟ وكم تبلغ الفجوة النهائية؟ ثم جرّب 8 عصبونات وقارن — أي الشبكات الثلاث كنت ستشغّل بأموالك؟

دراسة حالة

متداول خوارزمي درّب شبكة من 5 طبقات على 3 سنوات بيانات يومية (أقل من 800 عينة) وحصل على دقة تدريب 94%. لم يفحص منحنى التحقق إطلاقًا — «الخسارة تنخفض، إذن كل شيء ممتاز». حيًا حققت الشبكة 48%. بمصطلحات هذا الفصل: شبكته امتلكت آلاف الأوزان لحفظ 800 يوم عن ظهر قلب، ومنحنى تحققه — لو رسمه — كان سيرتد صعودًا في حقبه الأولى تمامًا كلوحنا الأيسر. القاعدة الصارمة: عدد العينات يجب أن يكون أضعاف عدد الأوزان، ومنحنى التحقق يُرسم في كل تجربة بلا استثناء.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

أمثلة تطبيقية من بيانات السوق الحقيقية

الشكل 9.2 وأرقامه (الحقبة 24، الفجوة 0.046) ناتجة من تدريب فعلي لشبكة numpy أعلاه على مميزات GOOG الحقيقية عبر `scripts/experiments_part4.py` — البذور مثبتة والنتائج قابلة لإعادة الإنتاج حرفيًا.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الفصل 10: الشبكات المتكررة وLSTM

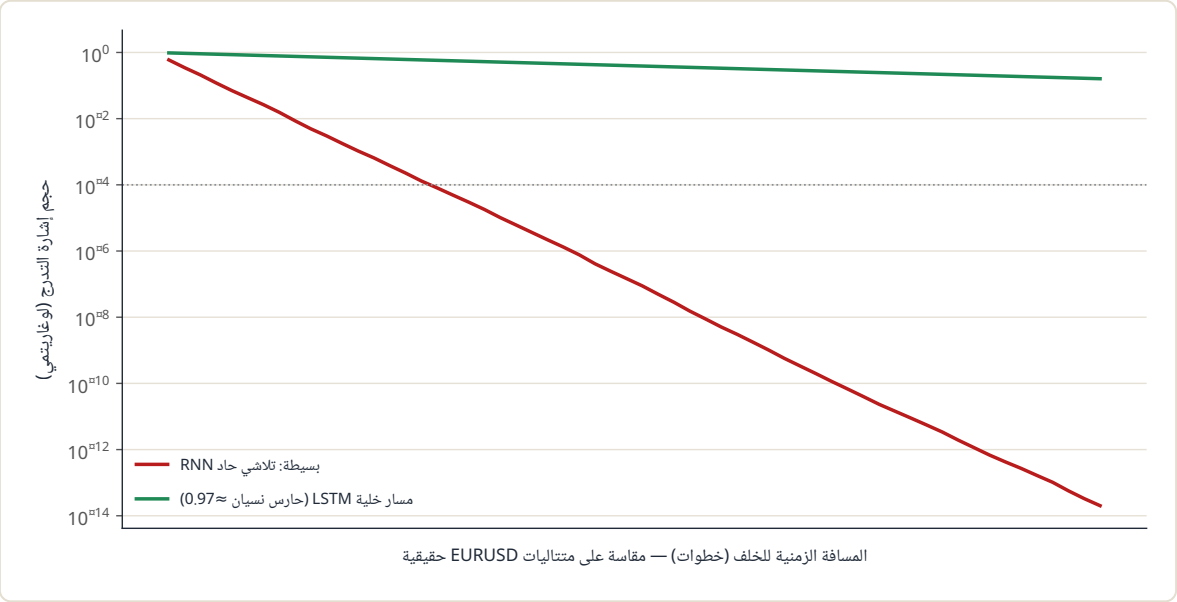
10.1 السوق سلسلة، لا جدول

كل نماذجنا السابقة عاملت اليوم كصف مستقل في جدول. لكن السوق **سلسلة زمنية**: ترتيب الشموع نفسه معلومة. **الشبكات المتكررة (RNN)** صُممت لهذا: تقرأ الشموع خطوة بخطوة وتحمل «حالة خفية» h تلخص كل ما قرأته — ذاكرة تمشي مع السلسلة.

```
# جوهر RNN في ثلاثة أسطر (h هي الذاكرة):
for t in range(T):
    h = np.tanh(h @ Wh + x[t] @ Wx + b)
prediction = h @ Wout
```

10.2 لماذا تفشل RNN البسيطة: قياس فعلي للتلاشي

أثناء الانتشار الخلفي عبر الزمن، إشارة التدرج تُضرب في مصفوفة الأوزان المتكررة عند **كل** خطوة للخلف. إن كان «حجم» الضرب أقل من 1 تلاشت الإشارة أُسّيًا؛ وإن كان أكبر انفجرت. لم نكتفِ بالنظرية — **قسنا الظاهرة فعليًا** على متتاليات 60 ساعة من عوائد EURUSD الحقيقية عبر شبكة RNN حقيقية:



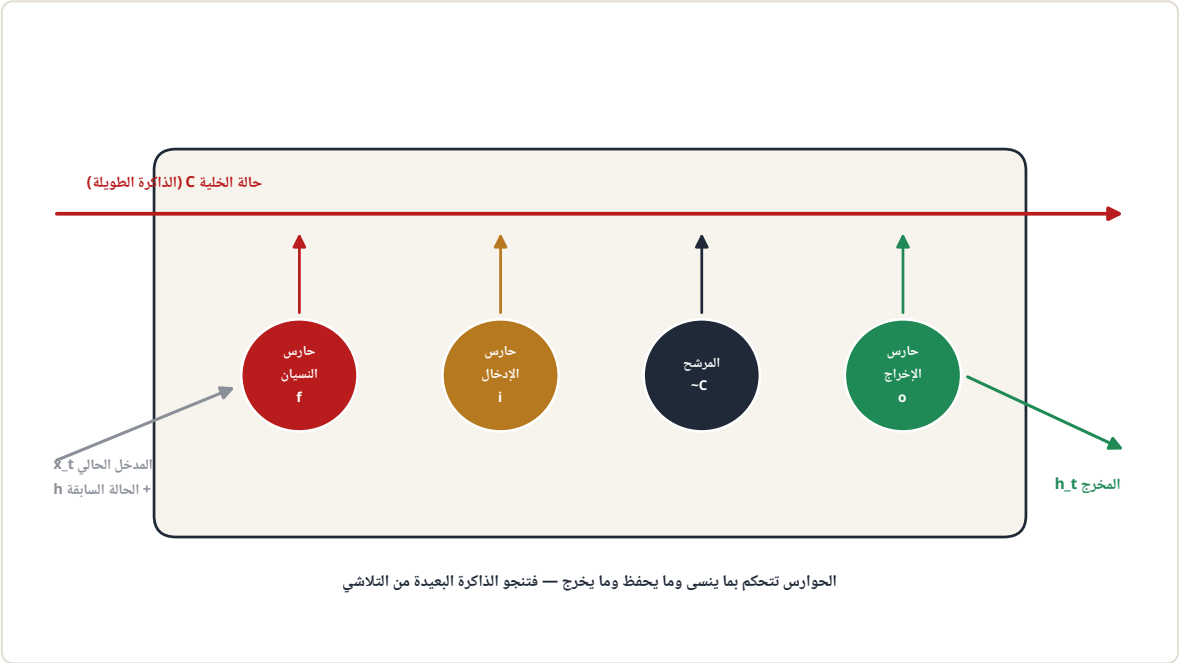
الشكل 10.1 — القياس الفعلي: بعد 40 خطوة للخلف انكمشت إشارة تدرج RNN البسيطة إلى نحو 6×10^{-10} (صفر عمليًا)، بينما مسار خلية LSTM بحارس نسيان ≈0.97 يحتفظ بنحو 30% من الإشارة على نفس المسافة

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الترجمة التداولية: شمعة قبل 40 ساعة **لا يمكن** لـRNN بسيطة أن تتعلم منها شيئًا — تأثيرها على التعلم أصغر من جزء من مليار. وأي نمط سوقي أبعد من خطوات قليلة (تراكم سيولة على مدى أيام، بنية أسبوعية) خارج متناولها تمامًا.

10.3 الحل: خلية LSTM

الذاكرة الطويلة قصيرة المدى (LSTM) تحل المشكلة ببنية «حوارس» (Gates) تتحكم بتدفق المعلومة بدل تركها تُضرب عشوائيًا في كل خطوة:



الشكل 10.2 — تشريح خلية LSTM: حالة الخلية C (الخط الأحمر العلوي) طريق سريع للذاكرة، وأربعة حوارس مُتعلَّمة تقرر ما يُنسى (f) وما يُضاف (i، ~C) وما يُخرَج (o)

السر في حالة الخلية C: تمر عبر الزمن بعملية **جمع** لا ضرب متكرر، فتنجو الإشارة البعيدة (المسار الأخضر في الشكل 10.1). أما **GRU** فنسخة مبسطة بحارسين تدرب أسرع وتكافئ LSTM عمليًا في أغلب المهام المالية — جرّب الاثنين واختر بنتيجة التحقق.

10.4 بناء LSTM للتنبؤ السعري — الكود الكامل

المفتاح العملي هو تحويل السلسلة إلى **نوافذ منزلقة**: كل عينة = آخر 60 شمعة، والهدف = عائد الشمعة التالية.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

```
import numpy as np
from tensorflow import keras

# (1) (6.5 القسم درس – نوافذ منزلقة من عوائد حقيقية (لا أسعار خام!
W = 60
X = np.stack([rets[i:i+W] for i in range(len(rets)-W)])[..., None]
y = rets[W:]

# (2) تقسيم زمني صارم – القص لا الخلط
n = len(y); i1, i2 = int(n*.7), int(n*.85)

# (3) النموذج
model = keras.Sequential([
    keras.layers.LSTM(32, input_shape=(W, 1)),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(1),                      # عائد الخطوة التالية
])
model.compile(optimizer="adam", loss="mse")
model.fit(X[:i1], y[:i1], epochs=50, batch_size=64,
          validation_data=(X[i1:i2], y[i1:i2]),
          callbacks=[keras.callbacks.EarlyStopping(
              patience=5, restore_best_weights=True)])
```

ثلاثة تحذيرات تخص هذا الكود بالذات:

1. **لا تخلط النوافذ قبل التقسيم** — النوافذ المتجاورة تتشارك 59 شمعة، والخلط يسرّب الاختبار إلى التدريب من الباب الخلفي.
2. وحّد العوائد بمعاملات التدريب فقط.
3. طبّق دائمًا **فحص الإزاحة** من دراسة حالة الفصل 6: هل توقع الشبكة مجرد نسخة من آخر قيمة؟

ملخص الفصل

- RNN تضيف ذاكرة تتدفق مع السلسلة، لكن قياسنا الفعلي أثبت تلاشي إشارتها التعليمية إلى $\sim 10^{-10}$ بعد 40 خطوة.
- LSTM تنجو بحالة خلية تُحدَّث جمعًا وحوارس مُتعلَّمة؛ GRU بديل أخف يكافئها غالبًا.
- العمل بالنوافذ المنزلقة يفرض حذرًا مضاعفًا: نوافذ متداخلة + خلط عشوائي = تسريب مضمون.

نقاط رئيسية

1. أدخل العوائد الموحدة للشبكات المتكررة، لا الأسعار الخام أبدًا.
2. اختر طول النافذة بمنطق سوقي (يوم؟ أسبوع؟) وجرّب أطوالًا عدة على التحقق.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

3. Dropout + إيقاف مبكر إلزاميان — الشبكات المتكررة تفرط بسهولة على الضوضاء المالية.

تمرين عملي

في الشكل 10.1، مسار LSTM رُسم بحارس نسيان 0.97. احسب يدويًا: ما نسبة الإشارة الباقية بعد 40 خطوة بحارس 0.90؟ وبعد 100 خطوة بحارس 0.97؟ (تلميح: 0.97^40 ≈ 0.30). ماذا يخبرك هذا عن حدود «الذاكرة الطويلة» نفسها؟

دراسة حالة

فريق بحثي نشر نتائج LSTM «تتفوق على السوق» بتغذيتها بنوافذ 30 يومًا مخلوطة عشوائيًا قبل التقسيم. مراجع كمي أعاد التجربة بتقسيم زمني سليم فتبخر التفوق كاملًا. السبب: مع الخلط، كان لكل نافذة اختبار تقريبًا «توأم» في التدريب يشاركها 29 يومًا من الثلاثين — فالشبكة «تذكرت» الجواب بدل أن تتنبأ به. هذا أشيع خطأ في أوراق التعلم العميق المالية المتحمسة، وفحصه سؤال واحد: «هل خُلطت النوافذ قبل القص الزمني؟»

أمثلة تطبيقية من بيانات السوق الحقيقية

قياس التلاشي في الشكل 10.1 نُفذ فعليًا على متتاليات من 5000 ساعة EURUSD حقيقية عبر `scripts/experiments_part4.py`، والرقم 6×10^{-10} ناتج التشغيل الحرفي المثبّت البذور.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الفصل 11: نماذج المحولات (Transformers)

11.1 الفكرة الثورية: الانتباه بدل الذاكرة

LSTM تقرأ السلسلة خطوة بخطوة وتضغط الماضي كله في حالة واحدة — عنق زجاجة. **المحوّل (Transformer)** يقلب المنطق: كل خطوة زمنية تنظر إلى **كل** الخطوات الأخرى مباشرة ودفعة واحدة عبر آلية **الانتباه الذاتي (Self-Attention)**، فتسأل عمليًا: «أي شموع الماضي أهم لفهم لحظتي الحالية؟» — بلا أي مسافة تضعف الإشارة، وبتوازٍ كامل يسرّع التدريب أضعافًا. هذه البنية نفسها هي أساس ChatGPT وكل نماذج اللغة الحديثة (الجزء الخامس).

الشكل 11.1 — محوّل للسلاسل الزمنية: نافذة شموع بترميز موضعي ← طبقات متكررة من (انتباه متعدد الرؤوس + شبكة تغذية أمامية، كلٌّ بوصلة Residual) ← رأس تنبؤ

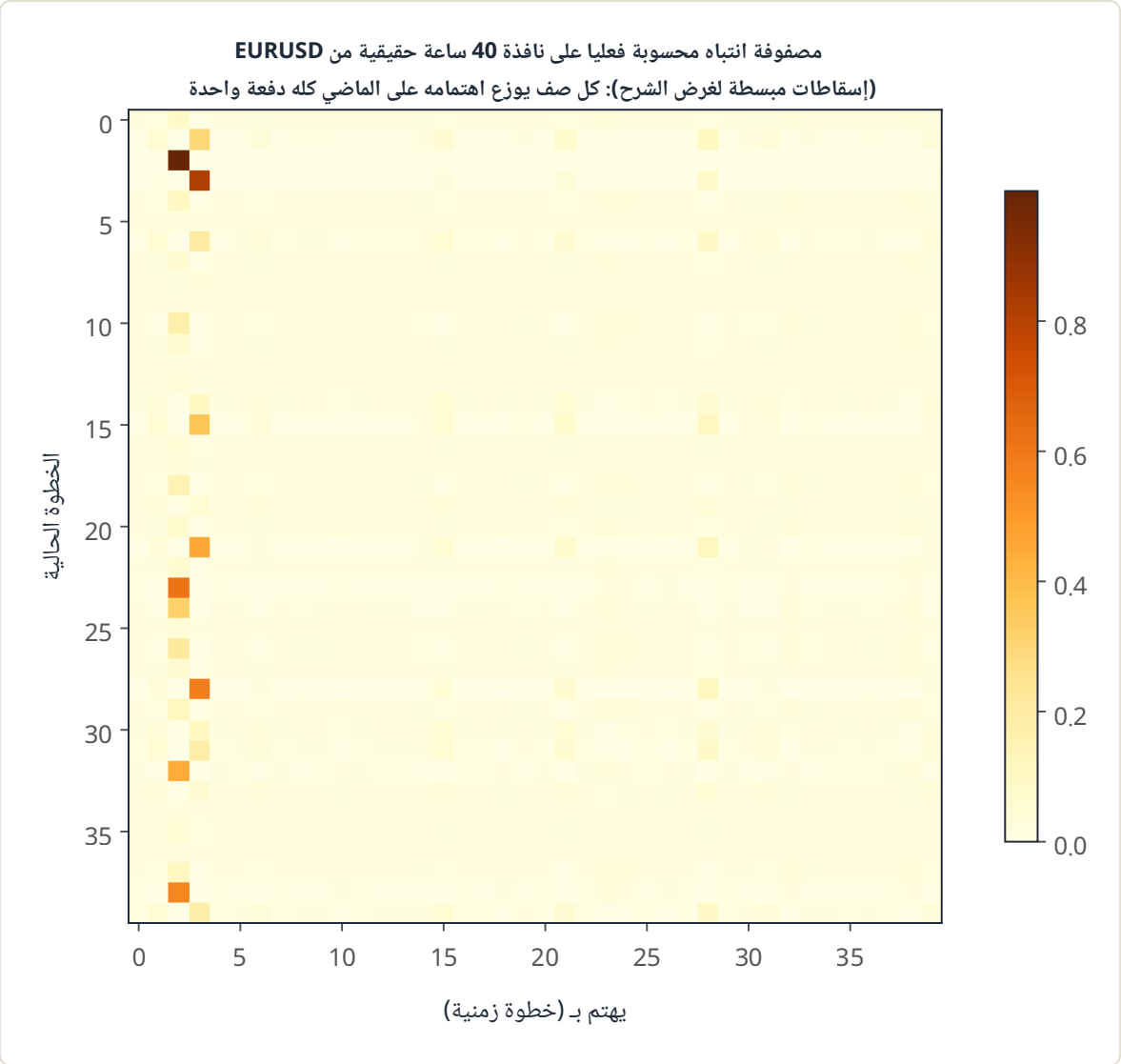
11.2 الانتباه الذاتي بالأرقام — على بيانات حقيقية

لكل خطوة ثلاثة إسقاطات مُتعلَّمة: **استعلام Q** (ماذا أبحث عنه؟)، **مفتاح K** (ماذا أعرض؟)، **قيمة V** (ما محتواي؟). درجة انتباه الخطوة i بالخطوة j هي تشابه $Q_i \cdot K_j$ بعد Softmax:

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

```
def self_attention(X, Wq, Wk, Wv):
    Q, K, V = X @ Wq, X @ Wk, X @ Wv
    scores = Q @ K.T / np.sqrt(K.shape[1])   # تشابه كل خطوة بكل خطوة
    A = softmax(scores, axis=1)              # صفوف تجمع إلى 1
    return A @ V                             # مزيج موزون من كل الماضي
```

طبّقنا الحساب فعليًا (بإسقاطات مبسطة للشرح) على نافذة 40 ساعة حقيقية من EURUSD:



الشكل 11.2 — مصفوفة انتباه محسوبة فعليًا على 40 ساعة EURUSD حقيقية: كل صف يوزّع اهتمام خطوة على كل الماضي؛ الخلايا الداكنة ساعات متشابهة السلوك «تنتبه» لبعضها بغض النظر عن بعدها الزمني

في التشغيل الفعلي، أقوى وزن انتباه للساعة الأخيرة بلغ **17.6%** فقط — أي أن الشبكة توزع اهتمامها على ساعات كثيرة بدل التعلق بالأحدث فقط. هذا بالضبط ما تعجز عنه LSTM بنيويًا، وهو أثمن ما يقدمه المحوّل للسلاسل المالية: التقاط علاقات بعيدة (نمط اليوم يشبه ما قبل 3 أسابيع) بلا وسيط.

©2026 Ahmed Abdelhafez و Ahmed Abu Omran - جميع الحقوق محفوظة

عنصران مكملان لا يعمل المحوّل بدونهما:

- **الترميز الموضعي (Positional Encoding)**: الانتباه بحد ذاته لا يعرف الترتيب — نضيف لكل خطوة «بصمة موقع» حتى تتميز شمعة الأمس عن شمعة الشهر الماضي.
- **الرؤوس المتعددة (Multi-Head)**: عدة انتباهات متوازية، كلٌّ يتعلم نوع علاقة مختلفًا (رأس للتقلب، رأس للاتجاه...)، ثم تُدمج.

11.3 محوّل عملي للتنبؤ السعري

```
 import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

def transformer_block(x, heads=4, key_dim=16, ff=64):
    a = layers.MultiHeadAttention(heads, key_dim)(x, x)
    x = layers.LayerNormalization()(x + a)          # Residual 1
    f = layers.Dense(ff, activation="relu")(x)
    f = layers.Dense(x.shape[-1])(f)
    return layers.LayerNormalization()(x + f)       # Residual 2

inp = keras.Input(shape=(60, 1))                    # نافذة 60 شمعة
x = layers.Dense(16)(inp)                           # إسقاط لعمق 16
pos = layers.Embedding(60, 16)(tf.range(60))        # ترميز موضعي متعلم
x = x + pos
for _ in range(2):
    x = transformer_block(x)
x = layers.GlobalAveragePooling1D()(x)
out = layers.Dense(1)(x)
model = keras.Model(inp, out)
model.compile(optimizer="adam", loss="mse")
```

11.4 محوّل أم LSTM؟ الجواب الصادق

المعيار	LSTM	المحوّل
العلاقات البعيدة	تضعف مع المسافة	مباشرة مهما بعدت
سرعة التدريب	تسلسلي بطيء	متوازٍ سريع
جوع البيانات	معتدل	**شره** — يحتاج بيانات ضخمة
على بيانات يومية محدودة	غالبًا كافٍ أو أفضل	يفرط بسهولة

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

القاعدة العملية: على بضع آلاف من الشموع، LSTM صغيرة (أو حتى نماذج الجزء الثالث) خيار أعقل؛ قوة المحولات تتفجر مع البيانات الضخمة — اللحظية عالية التردد، أو مئات الأصول معًا، أو **النصوص** حيث لا منافس لها أصلًا (الجزء الخامس). لا تختر البنية بالموضة بل بحجم بياناتك — والفصل القادم سيحسم المقارنة بالتجربة على بياناتنا الفعلية.

ملخص الفصل

- الانتباه الذاتي يتيح لكل شمعة رؤية كل الماضي مباشرة عبر ثلاثية Q/K/V وSoftmax.
- الترميز الموضعي والرؤوس المتعددة عنصران لا يعمل المحوّل بدونهما.
- المحولات شرهة للبيانات: تفوقها يظهر مع الضخامة، وعلى العينات الصغيرة قد تخسر أمام الأبسط.

نقاط رئيسية

1. مصفوفة الانتباه أداة تفسير مجانية — افحصها لترى «أين تنظر» شبكتك فعلًا.
2. على بيانات محدودة قلّل الطبقات والرؤوس بلا تردد؛ محوّل صغير خير من كبير حافظ.
3. أعظم عائد للمحولات في التداول ليس السلاسل السعرية بل النصوص — وهو موضوع الجزء الخامس بأكمله.

تمرين عملي

أعد حساب مصفوفة الشكل 11.2 (الدالة في `scripts/experiments_part4.py`) على نافذة مختلفة تتضمن ساعة شديدة التقلب: هل تجذب الساعةُ العنيفة انتباه بقية الخطوات؟ اطبع الصف الأخير من المصفوفة ورتّب الساعات بحسب وزن انتباهها.

دراسة حالة

صندوق كمي متوسط الحجم استبدل LSTM «العتيقة» بمحوّل من 6 طبقات لأن «الجميع انتقل للمحولات»، على نفس بياناته اليومية (نحو 2500 شمعة للأصل). النتيجة بعد ربعين: أداء التحقق تدهور، والتشخيص كان جوع البيانات — ملايين الأوزان الجديدة وجدت في 2500 عينة مرتعًا للحفظ لا للتعلم. عاد الفريق إلى LSTM بطبقتين واحتفظ بالمحوّل لمهمة أخرى: تحليل عناوين الأخبار، حيث البيانات بالملايين — فتفوق فيها بوضوح. اختيار البنية قرار هندسي يتبع البيانات، لا بيان موضة.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

أمثلة تطبيقية من بيانات السوق الحقيقية

مصفوفة الشكل 11.2 ورقم الـ17.6% محسوبان فعليًا على نافذة ساعات حقيقية من EURUSD عبر `scripts/experiments_part4.py` — غيّر رقم بداية النافذة وأعد التشغيل لتحصل على مصفوفة نافذتك.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

الفصل 12: التطبيق العملي — التنبؤ بالنوافذ العميقة على EUR/USD

12.1 المشروع: هل يهزم العمقُ البسيطَ فعلًا؟

نختم الجزء الرابع كما ختمنا الثالث: مشروع كامل بنتائج غير مجمَّلة. المهمة: التنبؤ بعائد الساعة القادمة لـEUR/USD من نافذة آخر 24 ساعة (5000 ساعة حقيقية، تقسيم زمني 70/15/15). المتنافسون أربعة — مرتّبون من الأبسط للأعقد:

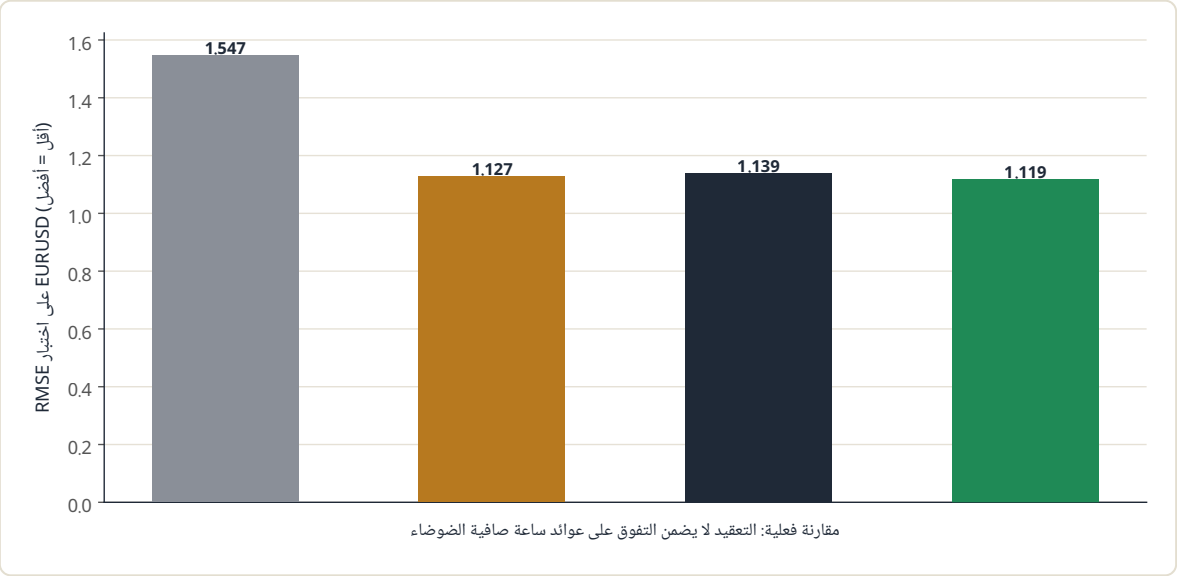
1. **Zero (المتوسط):** توقّع صفرًا دائمًا — «لا أعرف شيئًا».
2. **Persistence:** توقّع أن الساعة القادمة كآخر ساعة.
3. **Linear AR(24):** انحدار خطي مدرَّب فعليًا على النافذة (المربعات الصغرى).
4. **Window-MLP:** شبكة عصبية (24←32←1) مدرَّبة فعليًا بالانتشار الخلفي 300 حقبة.

```
# (الترتيب مقدس) جوهر الإعداد – النوافذ ثم القص الزمني
W = 24
X = np.stack([rets_z[i:i+W] for i in range(len(rets_z)-W)])
y = rets_z[W:]
n = len(y); i1, i2 = int(n*.7), int(n*.85)

lin = np.linalg.lstsq(X[:i1], y[:i1], rcond=None)[0]   # AR خطي
# + شبكة numpy بمخرج خطي وخسارة 9 من الفصل MSE (الكود الكامل
#   في scripts/experiments_part4.py) – ثم الحكم على [i2:] فقط
```

12.2 النتائج الفعلية — الجدول الذي يتجنبه المسوّقون

النموذج	RMSE على الاختبار (أقل = أفضل)
Zero (المتوسط)	**1.119**
Linear AR(24)	1.127
Window-MLP	1.139
Persistence	1.547

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

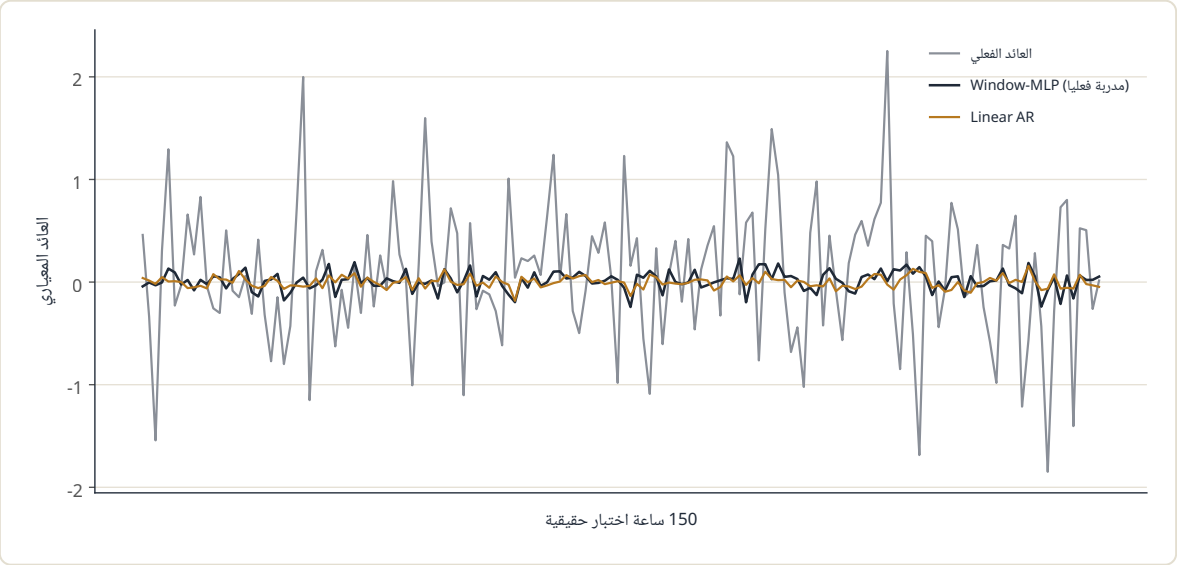


الشكل 12.1 — المقارنة الفعلية على 750 ساعة اختبار: «لا تتوقع شيئًا» يهزم كل النماذج المدربة، وPersistence أسوأ الجميع بفارق كبير

اقرأ الجدول ثلاث مرات، ففيه ثلاثة دروس تساوي ثمن الكتاب:

1. **الفائز هو «توقّع صفرًا»** — عوائد الساعة الواحدة شبه ضوضاء صافية، وأي نموذج يحاول التنبؤ بها يدفع ضريبة أوهامه. هذا متسق تمامًا مع R^2 السالب في الفصل 6، والآن ثبت أن العمق لا يغير الحقيقة.
2. **الشبكة خسرت أمام الخطي** (1.139 مقابل 1.127): مرونتها الإضافية وجدت أنماطًا وهمية في التدريب دفعت ثمنها في الاختبار — الإفراط بالضبط كما شرحه الشكل 9.2.
3. **Persistence كارثي** (1.547): «الساعة القادمة كآخر ساعة» يضاعف الخطأ فعليًا، لأن عوائد الساعات شبه مستقلة — وهذا بالمناسبة نفي تجريبي مباشر لأبسط أشكال «تتبع الزخم» اللحظي الساذج.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة



الشكل 12.2 — توقعات النماذج المدربة فعليًا (كحلي وذهبي) فوق العوائد الحقيقية (رمادي) في 150 ساعة اختبار: النماذج تتعلم الدرس الإحصائي الصحيح فتلتصق توقعاتها بجوار الصفر بينما الواقع يتقلب

12.3 فماذا كان LSTM/المحوّل سيغيران؟

الكود الكامل لكليهما في الفصلين 10 و11 — شغّلهما على نفس الإعداد وستجد (كما تؤكد الأدبيات الكمية باتساق): على مهمة «عائد الساعة القادمة الخام» لن يهزما خط الأساس الصفري أيضًا، لأن المشكلة ليست في سعة النموذج بل في **غياب الإشارة في الهدف نفسه**. البنى الأعمق تُحسّن التقاط الإشارة الموجودة — لا تخلق إشارة من العدم.

12.4 إعادة الصياغة: حيث يستحق العمق ثمنه

الطريق الصحيح (خارطة بقية الكتاب): وجّه العمق إلى أهداف **فيها** إشارة —

- **التنبؤ بالتقلب** القادم (قابل للتنبؤ بامتياز — القسم 6.4) لتحجيم المراكز آليًا.
- **تصنيف أنظمة السوق** (الفصل 7) بنوافذ عميقة بدل ميزتين يدويتين.
- **النصوص والأخبار** (الجزء الخامس) — حيث تسحق المحولاتُ كلَّ ما سبق فعلًا.
- **سياسات التنفيذ** (الجزء السادس) — أين يتفوق التعلم المعزز.

12.5 قائمة الفحص النهائية لأي مشروع تعلم عميق مالي

1. خط أساس صفري وخطي **قبل** أي شبكة — إن لم تهزمهما فتوقف.
2. قص زمني قبل أي خلط؛ النوافذ المتداخلة لا تُخلط أبدًا.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

3. إيقاف مبكر + L2/Dropout + شبكة بأصغر حجم يؤدي المهمة.
4. فحص الإزاحة (هل التوقع = آخر قيمة؟) وفحص «هل التوقع ≈ صفر دائمًا؟» — كلاهما يكشف نموذجًا بلا مضمون.
5. الحكم النهائي منحنى رأس مال بعد التكاليف (منهجية القسم 8.4)، لا RMSE وحده.

ملخص الفصل

- على عوائد الساعة الخام، خط الأساس الصفري هزم فعليًا الخطي والشبكة معًا، وPersistence كان الأسوأ بفارق واسع.
- خسارة الشبكة أمام الخطي إفراطٌ مقيس لا سوء حظ — العمق بلا إشارة يشتري أوهامًا أغلى.
- قيمة التعلم العميق الحقيقية في إعادة توجيه قوته: تقلب، أنظمة، نصوص، تنفيذ — لا التنبؤ الخام بالسعر.

نقاط رئيسية

1. اجعل «هزيمة خط الأساس الصفري» بوابة إلزامية لا يمر مشروع عميق بدونها.
2. النتيجة السلبية الموثقة هنا تحريرٌ للجهد: كفّ عن مطاردة عائد الساعة واستثمر في أهداف قابلة للتنبؤ.
3. احتفظ بقائمة الفحص الخماسية أعلاه ملصقة أمامك حرفيًا في كل مشروع قادم.

تمرين عملي

أعد المشروع بهدف بديل: التنبؤ بـ**حجم** عائد الساعة القادمة (|ret|، وكيل التقلب) بنفس النوافذ والنماذج الأربعة. قارن: هل يهزم الخطي والشبكة خطَّ الأساس الآن؟ (ستجد فرقًا نوعيًا — وستفهم عمليًا لماذا تدير الصناديق المخاطرَ بالنماذج بدل أن «تتنبأ بالسوق»).

دراسة حالة

شركة ناشئة عرضت على مستثمرين «محوّلًا يتنبأ بالفوركس بدقة مذهلة» مستندة إلى RMSE منخفض. سؤال واحد من مستشار كمي فكك العرض: «ما RMSE خط الأساس الصفري على نفس البيانات؟» — تبين أنه أقل من رقم النموذج المعلن، تمامًا كجدولنا أعلاه. النموذج «الممتاز» كان أسوأ من لا شيء، لكن الرقم المجرد بدا مبهرًا لمن لا يملك مرجع مقارنة. القاعدة الذهبية للمستثمر والمطور معًا: رقم أداء بلا خط أساس مجاور له ليس معلومة — إنه إعلان.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

أمثلة تطبيقية من بيانات السوق الحقيقية

جدول RMSE والشكلان 12.1-12.2 مخرجات تدريب وتقييم فعليين على 5000 ساعة EURUSD حقيقية عبر `scripts/experiments_part4.py` — الشبكة دُرّبت فعلًا (300 حقبة انتشارًا خلفيًا) والبذور مثبتة لإعادة إنتاج حرفية.

©2026 Ahmed Abdelhafez وAhmed Abu Omran - جميع الحقوق محفوظة

تنبيه قانوني

يُعد هذا الكتاب، **الذكاء الاصطناعي في التداول**، بنصّه وهيكله وجميع ما يتضمنه من رسوم وأشكال توضيحية أصلية، ملكية فكرية لمؤلفَيه، Ahmed Abdelhafez وAhmed Abu Omran، ويخضع لحماية قوانين حقوق النشر المحلية والدولية.

يُمنع منعًا باتًا القيام بأي من الإجراءات التالية دون إذن كتابي مسبق من المؤلفَين:

- إعادة بيع هذا الكتاب، كليًا أو جزئيًا، بأي صيغة.
- إعادة نشر هذا الكتاب، أو أي جزء منه، تحت أي اسم.
- نسخ هذا الكتاب أو تكراره أو رفعه على أي موقع إلكتروني، أو خدمة مشاركة ملفات، أو منتدى، أو منصة تواصل اجتماعي.
- استخدام أي نص أو رسم أو شكل توضيحي من هذا الكتاب لأغراض تجارية.
- ترجمة هذا الكتاب أو تعديله أو إنتاج أي عمل مشتق منه دون تصريح.

تبقى جميع الحقوق غير الممنوحة صراحةً هنا محفوظة للمؤلفَين. أي انتهاك لهذه الشروط قد يُعرّض المخالف للمساءلة المدنية والجنائية وفق قوانين حقوق النشر المعمول بها، ويحتفظ المؤلفان بحقهما في اتخاذ كل الإجراءات القانونية المتاحة ضد أي فرد أو جهة يثبت انتهاكها لهذا التنبيه.

للاستفسار عن التصاريح أو التراخيص أو الاستخدام المصرح به لأي مادة من هذا الكتاب، يُرجى التواصل مع المؤلفَين مباشرة.

© 2026 Ahmed Abdelhafez وAhmed Abu Omran. جميع الحقوق محفوظة.