المحاضرة الرابعة: لغات برمجة الذكاء الاصطناعي وأطر عمله

تخصص هذه المحاضرة لتسليط الضوء على الأدوات التقنية الفعلية التي يستخدمها مطورو وعلماء الذكاء الاصطناعي، مع التركيز على لغات البرمجة الأكثر شيوعاً في هذا المجال ومنصات التعلم العميق التي تمكّن من بناء نماذج ذكية,

1. بايثون(Python) : لغة الذكاء الاصطناعي السائدة

تُعد بايثون (Python) حاليًا اللغة الأكثر هيمنة وشهرة في مجال الذكاء الاصطناعي والتعلم الآلي، وقد رسخت مكانتها بفضل مجموعة من الخصائص الفريدة التي تجعلها الخيار الأول للباحثين والمطورين على حد سواء، وتتميز Python بكونها لغة عالية المستوى(High-Level Language) ، وتستخدم بناء جمل (Syntax) واضحًا وبسيطًا يشبه إلى حد كبير اللغة الإنجليزية، مما يسهل عملية القراءة والكتابة والتعلم، وهذا يرفع من إنتاجية المبرمج(Programmer Productivity) ، مما يسمح للفرق بإنشاء النماذج وتجربتها بشكل أسرع بكثير مما تسمح به اللغات الأخرى التي تتطلب تفاصيل معقدة لإدارة الذاكرة، وهي ميزة حاسمة في عالم يتسم بالتحول الرقمي المتسارع، وتدعم Python مفهوم البرمجة الشيئية (Object-Oriented Programming - OOP) بشكل ممتاز، مما يسهل كتابة شفرات قابلة لإعادة الاستخدام (Reusable) ومرنة، وهي خاصية أساسية في بناء وحدات متكاملة للتعلم الآلي والشبكات العصبونية المعقدة,

السبب الجوهري لشعبية بايثون يكمن في نظامها البيئي الضخم واللامتناهي من المكتبات (Libraries) والأدوات الجاهزة المصممة خصيصاً للتعامل مع البيانات والرياضيات، ومن أبرز هذه المكتبات التي تشكل العمود الفقري للذكاء الاصطناعي:

- **NumPy:** توفر هياكل بيانات متقدمة (كالمصفوفات والأبعاد المتعددة) وأدوات رياضية عالية الكفاءة لمعالجة الأرقام، وهي ضرورية لجميع العمليات الحسابية المتجهة (Vector Operations) المطلوبة في خوارزميات التعلم العميق,
- **Pandas:** أداة قوية وفعالة لمعالجة وتحليل البيانات بشكل أسرع وأكثر مرونة، وتُستخدم بشكل أساسي في مرحلة هندسة البيانات (Data Engineering) لعمليات التنظيف والمعالجة المسبقة.
- **Scikit-learn:** مكتبة شاملة توفر مجموعة واسعة من خوارزميات التعلم الآلي الكلاسيكية، مثل خوارزميات التصنيف (Classification) والتجميع (Clustering) والانحدار(Regression) .
- **Matplotlibو Seaborn** : مكتبات مخصصة لـ تصوير البيانات (Data Visualization) بشكل رسومي، مما يساعد الباحثين على فهم الأنماط والنتائج التي توصل إليها النموذج بشكل فوري وواضح.

1. لغات برمجة أخرى في مجال الذكاء الاصطناعي

على الرغم من هيمنة بايثون، لا تزال هناك لغات برمجة أخرى تلعب أدواراً حيوية ومكملة في النظام البيئي للذكاء الاصطناعي، خاصة في المجالات التي تتطلب سرعة فائقة أو تكاملاً مع أنظمة التشغيل والعتاد:

أ. جافا(Java) :

تتميز جافا (Java) بكونها لغة مترجمة وقابلة للتنفيذ عبر المنصات(Platform Independent) ، وهي قوية بشكل خاص في بناء التطبيقات المؤسسية الضخمة وأنظمة التشغيل الخلفية(Backend Systems) ، ورغم أنها ليست الخيار الأول لتدريب النماذج البحثية الأولية، إلا أنها الخيار الأمثل لـ النشر (Deployment) والإنتاج في بيئات الشركات الكبيرة، وتُستخدم Java في بناء أنظمة معالجة البيانات الضخمة في الوقت الفعلي (Real-time Big Data Processing) والتطبيقات الموجهة للمستخدم، ولها مكتبات مخصصة مثل Deeplearning4j التي تسمح بتشغيل نماذج التعلم العميق على خوادمJava .

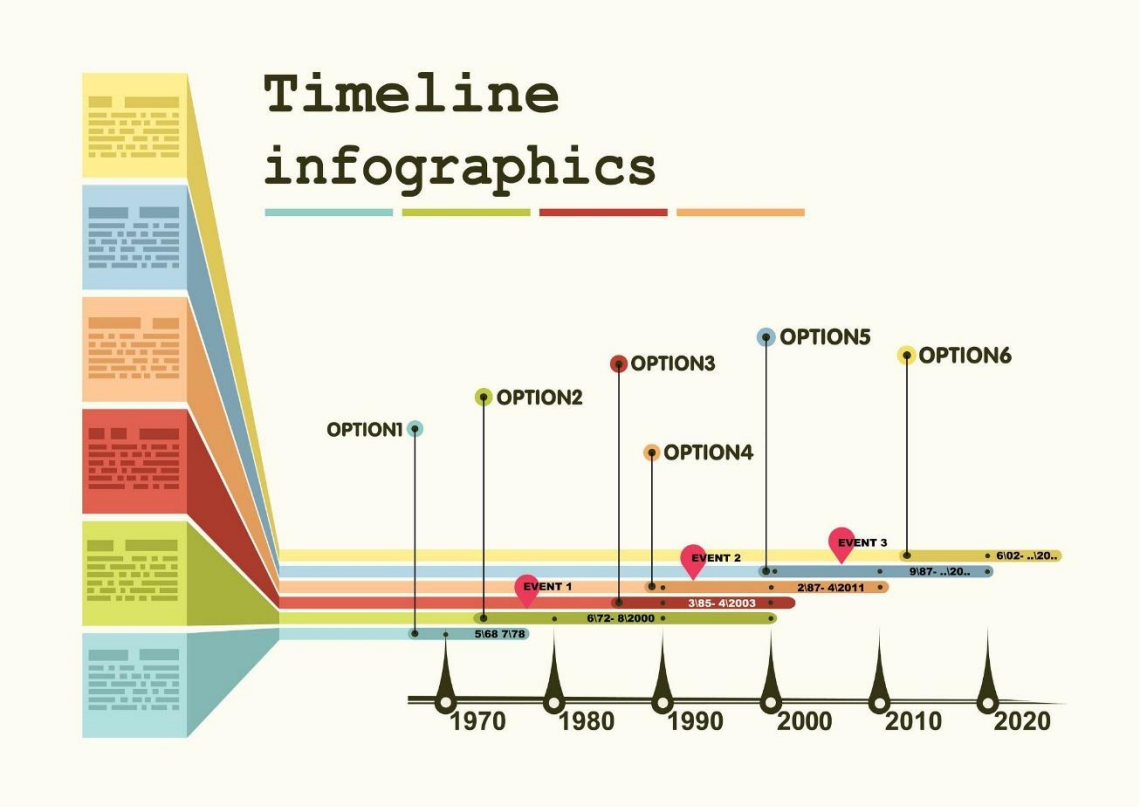
ب. سي بلس بلس(C++) :

تُعتبر C++من اللغات منخفضة المستوى نسبيًا التي تتميز بـ سرعة تنفيذها الفائقة وقدرتها على التحكم المباشر بالذاكرة والعتاد، وهي لغة أساسية لعدة أسباب:

- **الخوارزميات الأساسية:** يتم كتابة النواة الأساسية (Kernels) والعمليات الحسابية المُجهِدة داخل مكتبات بايثون الكبرى (مثل NumPy وأطر العمل) باستخدام C++ لضمان أعلى أداء ممكن.
- **الأنظمة المدمجة(Embedded Systems) :** تُستخدم C++ في برمجة الروبوتات والمركبات ذاتية القيادة (كما ذكرنا في المحاضرة 2) حيث يكون كل جزء من الثانية في التنفيذ مهمًا لسلامة القرار، وتُعد ضرورية في الرؤية الحاسوبية والتحكم الآلي.
- **الألعاب والرسوميات:** تُستخدم C++ على نطاق واسع في محركات الألعاب (Game Engines) التي تستخدم الذكاء الاصطناعي لتسيير سلوك الشخصيات غير اللاعبة(NPCs) ، وهذا يتطلب أعلى درجات الفاعلية(Efficiency) .

ج. لغات تاريخية (LISP و Prolog):

لغات مثل LISPو Prolog كانت اللغات الرائدة في الفترة المبكرة من الذكاء الاصطناعي الرمزي(Symbolic AI) ، ولا تزال ذات قيمة أكاديمية عالية في دراسة المنطق والاستدلال، وتُستخدم هذه اللغات في بناء الأنظمة الخبيرة Expert) Systems) التي تعتمد على القواعد والمنطق المبرمج صراحةً، وهي تمثل الجذور الفكرية للخوارزميات المعتمدة على الاستدلال بدلاً من التعلم الإحصائي،



الشكل 03: مخطط معلومات بياني (Infographic) على شكل خط زمني(Timeline)

تحليل مخطط "الخط الزمني للمعلومات البيانية"

أولا: العنوان والهدف

- **العنوان Timeline Infographics :** (مخطط معلومات بياني زمني).
- **الهدف:** يهدف المخطط إلى دمج كمية من المعلومات النصية (الممثلة في الشريط العمودي على اليسار) وربطها بنقاط زمنية محددة على محور أفقي، مما يتيح تقديم سرد تاريخي أو تتبع مراحل تطور فكرة أو مشروع.

ثانيًا: المكونات الرئيسية للمخطط

أ. المحور الزمني الأفقي(Horizontal Timeline) :

- **النطاق الزمني:** يمتد المحور الزمني من عام **1970** حتى عام **2020**، مع نقاط محددة لكل عقد (1980، 1990، 2000، 2010)، مما يوفر إطارًا واضحًا لقياس التقدم.
- **علامات التجزئة:** يتم تمييز السنوات الرئيسية بعلامات على شكل دبابيس أو أسهم تشير للأسفل، مما يؤكد الطبيعة التاريخية أو التطورية للبيانات المعروضة.

ب. منطقة البيانات المرجعية(The Source Data)

- **الشريط العمودي الأيسر:** يمثل مجموعة من الكتل الملونة المتراصة عمودياً، كل كتلة (من **الأسفل إلى الأعلى**) تحتوي على بيانات نصية كثيفة، وهذه الكتل هي المصدر الأساسي للمعلومات التي يتم تمثيلها على الخط الزمني.
- **الألوان:** يتم استخدام ألوان مختلفة (مثل الأزرق، الأخضر، الأحمر، البرتقالي) لتمييز كل كتلة بيانات، وهذا التمييز اللوني يُستخدم لربط كل معلومة بمكانها الصحيح على المحور الزمني.

ج. ربط البيانات بالأحداث(Events and Options)

- **الخطوط الملونة:** تنطلق من كتل البيانات الملونة على اليسار وتتجه نحو اليمين لترتبط بنقاط محددة على الخط الزمني، ويوضح كل خط تطور مجموعة معينة من البيانات أو المراحل.
- **نقاط التحديد(Options) :** يشار إلى نقاط التطور أو المراحل الرئيسية بـ **OPTIONS** (الخيار 1، الخيار 2، ...، الخيار 6)، وهذه النقاط تمثل إنجازات، أو قرارات، أو مفاهيم ارتبطت بلحظة زمنية معينة.
- **الأحداث(Events) :** تم تحديد ثلاث نقاط حمراء رئيسية بـ EVENT 1 و EVENT 2 و EVENT 3، وهذه الأحداث تمثل على الأرجح نقاط تحول هامة أو لحظات فارقة أثرت على مسار التطور، وعادةً ما ترتبط بتواريخ دقيقة (مثل 7/78 - 5/68 أو 6/2011 - 2/87).

ثالثًا: التعليق والتطبيق

يمكن تطبيق هذا المخطط:

- **الخط الزمني (1970–2020):** يمثل بشكل مثالي التطور التاريخي للذكاء الاصطناعي الذي تناولته في المحاضرة الثانية.
- **EVENT 1 / EVENT 2 / EVENT 3:** يمكن أن تمثل مؤتمر دارتموث 1956 (بداية النشأة)، و شتاء الذكاء الاصطناعي(AI Winter) ، و بداية عصر التعلم العميق(Deep Learning) ، كنقاط تحول محورية.
- **Options:** يمكن أن تمثل ظهور لغات برمجة مختلفة (Option 1 = LISP، Option 2 = C++، Option 3 = Python)، أو ظهور أطر عمل (TensorFlow وPyTorch).

إنها أداة بصرية فعالة جداً لتبسيط وعرض المعلومات المعقدة أو التطورات التاريخية الطويلة في شكل سهل الاستيعاب.

2. أطر العمل (Frameworks) ومنصات التعلم العميق:

لا يمكن بناء نماذج الذكاء الاصطناعي الحديثة دون استخدام أطر العمل (Frameworks) التي توفر الأدوات والبنية التحتية الجاهزة، وتقدم هذه الأطر مستوى من التجريد (Abstraction) يسمح للمبرمج بالتركيز على تصميم الشبكة بدلاً من كتابة كل عملية رياضية معقدة من الصفر، وهي الأدوات التي يتم التركيز عليها بشكل خاص في هذا المقرر:

- **TensorFlow:**

تم تطوير TensorFlow من قبل شركة جوجل، ويُعد أحد أكثر الأطر استخداماً عالمياً، ويتميز بقدرته على بناء نماذج التعلم العميق (Deep Learning) المعقدة، ويدعم مجموعة واسعة من التطبيقات من الرؤية الحاسوبية إلى معالجة اللغات الطبيعية(NLP) ، ويُعرف TensorFlow ببيئته الشاملة التي تسمح بـ النشر السهل (Easy Deployment) على منصات متعددة، بما في ذلك الأجهزة المحمولة (TensorFlow Lite) والمتصفحات(TensorFlow.js) ، مما يجعله مثاليًا لتحويل النماذج البحثية إلى تطبيقات رقمية عملية,

- **PyTorch:**

تم تطوير PyTorch بواسطة مختبر أبحاث الذكاء الاصطناعي التابع لـ فيسبوك(Meta) ، وقد اكتسب شعبية هائلة، خاصة في الأوساط البحثية والأكاديمية، ويُعرف PyTorch بمرونته العالية وسهولة استخدامه، لا سيما في عملية التصحيح (Debugging)، ويستخدم مفهوم الرسوم البيانية الحاسوبية الديناميكية (Dynamic Computation Graphs)، على عكس TensorFlow التقليدي، مما يجعله أكثر جاذبية للباحثين الذين يحتاجون إلى تجربة هياكل نماذج جديدة بسرعة.

- **Keras:**

في الأصل، كان Keras مكتبة مستقلة، لكنه الآن جزء متكامل من TensorFlow ، ويُعد Keras الخيار الأمثل للمبتدئين ولـ التطوير السريع(Rapid Prototyping) ، والهدف منه هو جعل عملية بناء الشبكات العصبونية بسيطة ومباشرة قدر الإمكان، وتستخدم واجهة Keras البرمجية مستوى عالٍ من التجريد، مما يسمح للمبرمج ببناء نموذج معقد ببضعة أسطر من الشفرة، وهذا يسرّع من دورات التطوير ويسهل على طلاب التخصصات التطبيقية إتقان بناء النماذج.

أداة الذكاء الاصطناعي	نوعها	اللغة الأم	الميزة الرئيسية	الاستخدام النموذجي
Python	لغة برمجة عالية المستوى	-	سرعة التطوير، سهولة القراءة، النظام البيئي الضخم للمكتبات.	البحث العلمي، تحليل البيانات، بناء واجهات النماذج.
C++	لغة برمجة منخفضة المستوى	-	سرعة التنفيذ، التحكم المباشر بالذاكرة والعتاد (الفاعلية).	الأنظمة المدمجة، الروبوتات، نواة مكتبات التعلم العميق.
TensorFlow	إطار عمل للتعلم العميق	**Python/C++**	شامل، موجه للنشر(Deployment) ، دعم واسع للمنصات.	تطبيقات الإنتاج، النشر على الهواتف والأجهزة الطرفية.
PyTorch	إطار عمل للتعلم العميق	**Python/C++**	مرونة عالية، رسوم بيانية ديناميكية، مفضّل في الأوساط البحثية.	الأبحاث المتقدمة، تطوير الخوارزميات الجديدة.
Keras	واجهة برمجية عالية المستوى	**Python/TensorFlow**	سهولة الاستخدام، **التطوير السريع**، مثالي للمبتدئين.	بناء النماذج الأولية بسرعة فائقة,

كخلاصة لما سبق: تتناول المحاضرة الرابعة الأدوات التقنية الأساسية التي تشكّل العمود الفقري لتطوير الذكاء الاصطناعي، وفي مقدمتها تبرز بايثون (Python) كلغة سائدة بلا منازع، وتعود هيمنة Python إلى عدة عوامل حاسمة، أبرزها كونها لغة عالية المستوى ذات بناء جمل (Syntax) بسيط وواضح، مما يرفع بشكل كبير من

إنتاجية المبرمج ويسمح بالتجريب السريع للنماذج، وهي ميزة لا تُقدّر بثمن في بيئات البحث والتطوير، كما تدعم Pythonالبرمجة الشيئية (OOP) بكفاءة، مما يسهل بناء مكونات برمجية قابلة لإعادة الاستخدام، ولكن السبب الجوهري لشعبيتها يكمن في نظامها البيئي الهائل من المكتبات المتخصصة والمُحسَّنة للأداء مثل NumPy، التي توفر الأدوات الرياضية الأساسية للمصفوفات والعمليات المتجهة (Vector Operations) الضرورية للتعلم العميق، و Pandasلمعالجة البيانات الأولية وتنظيفها في مرحلة هندسة البيانات، و Scikit-learnالتي تحتوي على خوارزميات التعلم الآلي الكلاسيكية جاهزة للاستخدام، بالإضافة إلى مكتبات Matplotlibو Seabornلتصوير البيانات وتحليل نتائج النماذج بشكل بصري فوري، مما يجعلها منصة متكاملة تغطي دورة حياة مشروع الذكاء الاصطناعي بالكامل من التحليل الأولي حتى بناء النماذج الأولية.

ومع ذلك، وعلى الرغم من تفوق بايثون في التدريب والبحث، تظل هناك لغات أخرى تلعب أدوارًا مكمّلة وضرورية، خاصة في سياقات **الإنتاج والسرعة الفائقة**، حيث تتميز لغة **جافا (Java)** بقوتها في بناء التطبيقات المؤسسية الضخمة وأنظمة التشغيل الخلفية(Backend Systems) ، كما تُعد مثالية لـ **النشر (Deployment)** في بيئات الشركات التي تتطلب الأداء العالي والقابلية للتشغيل عبر منصات متعددة(Platform Independent) ، وتُستخدم بشكل خاص في أنظمة معالجة البيانات الضخمة في الوقت الفعلي، أما لغة **سي بلس بلس(C++)** ، فهي تُعد الأداة الأهم لضمان **سرعة التنفيذ الفائقة** والتحكم المباشر بالعتاد، حيث يتم كتابة النواة الأساسية (Kernels) للعمليات الحسابية المُجهِدة داخل مكتبات بايثون الكبرى باستخدام C++ لزيادة كفاءتها القصوى، وتلعب دورًا حيويًا في **الأنظمة المدمجة (Embedded Systems)**مثل الروبوتات والمركبات ذاتية القيادة، حيث تكون الفاعلية الزمنية (Real-time Efficiency)هي مفتاح اتخاذ القرار الآمن، وإضافة إلى ذلك، تظل لغات تاريخية مثل **LISPو Prolog**ذات أهمية أكاديمية في مجال **الذكاء الاصطناعي الرمزي**، لكونها تستخدم في بناء **الأنظمة الخبيرة** المعتمدة على المنطق والاستدلال الصريح بدلاً من التعلم الإحصائي.

في سياق بناء نماذج التعلم العميق الحديثة، لا يمكن الاستغناء عن **أطر العمل (Frameworks)** التي توفر مستوى من **التجريد (Abstraction)** يسمح للمطور بالتركيز على تصميم الشبكة بدلاً من العمليات الرياضية الأولية، وتُعد هذه الأطر هي الأدوات الرئيسية التي يتم استخدامها في بناء النماذج الذكية، وأبرزها هو **TensorFlow**، الذي طورته جوجل ويتميز ببيئته الشاملة وتوجهه القوي نحو **النشر السهل (Easy Deployment)** على مختلف المنصات، بما في ذلك الأجهزة المحمولة والمتصفحات، مما يجعله مثاليًا لتحويل النماذج البحثية إلى تطبيقات إنتاج رقمية عملية، وعلى الجانب الآخر، يأتي **PyTorch**، الذي طورته ميتا (فيسبوك سابقاً)، واكتسب شعبية كبيرة في الأوساط البحثية والأكاديمية بفضل **مرونته العالية** واستخدامه لمفهوم **الرسوم البيانية الحاسوبية الديناميكية**، مما يسهل عملية التجريب والتصحيح(Debugging) ، وكمُكمِّل لهما، تُعد **Keras**واجهة برمجية عالية المستوى، وهي الآن جزء أساسي منTensorFlow ، وتتميز بكونها الخيار الأمثل للمبتدئين وللتطوير السريع(Rapid Prototyping) ، حيث تستخدم أعلى مستويات التجريد لتمكين بناء نماذج معقدة ببضعة أسطر من الشفرة، مما يسرّع من دورات التطوير ويجعل إتقان بناء النماذج أسهل للطلاب والمطورين، وهذا التنوع في الأدوات يضمن إيجاد الحل الأمثل لكل مهمة، سواء كانت بحثًا نظريًا(PyTorch) ، أو تطبيقًا يتطلب النشر الواسع(TensorFlow) ، أو نظامًا مدمجًا يتطلب السرعة القصوى.(C++)

تمارين وتطبيقات عملية:

التمرين 1: سيناريو الاختيار (لغة مقابل إطار عمل)

لنفترض أنك تعمل على مشروعين في نفس الوقت:

1. **المشروع أ:** تصميم وتدريب شبكة عصبونية لتصنيف الرسائل الجماهيرية (الوحدة 3).
2. **المشروع ب:** برمجة نظام تحكم لربط أجهزة استشعار الحرارة والضغط داخل روبوت صغير (Embedded System).

السؤال: ما هي اللغة/الإطار الأنسب لكل مشروع، ولماذا؟

1. **المشروع أ (تصنيف الرسائل):**

- **الأداة الأنسب:** Python مع إطار عمل PyTorch أوTensorFlow .
- **السبب:** لأن هذا المشروع يتطلب معالجة بيانات ضخمة، واستخدام مكتبات متقدمة للتعلم العميق، وتكون سرعة التطوير (Python)أهم من سرعة التنفيذ الخام,(C++) .

2. **المشروع ب (الروبوت):**

- **الأداة الأنسب,**C++ :
- **السبب:** لأن برمجة الأنظمة المدمجة تتطلب تحكمًا دقيقًا في العتاد والذاكرة لضمان **الفاعلية الزمنية** (Real-time Efficiency)والسرعة الفائقة في اتخاذ القرار، وهي من خصوصيات C++ التي يصعب تحقيقها بلغات مفسّرة,

التمرين 2: دور التجريد في Keras

عند استخدامKeras ، يقوم الطالب ببناء طبقات الشبكة العصبونية باستخدام أوامر بسيطة، بينما تقوم المكتبة داخليًا بترجمة هذه الأوامر إلى عمليات رياضية معقدة.

السؤال: ما هي الخاصية البرمجية (المذكورة في المحاضرة 1) التي يخدمها هذا التجريد؟ وما فائدته لطالب الاتصال الاستراتيجي؟

1. **الخاصية: العمومية (Generality) و التجريد,(Abstraction)**
2. **الفائدة للطالب:** يقلل Keras من الحاجة إلى الغوص في تفاصيل الرياضيات المعقدة لإدارة المصفوفات، مما يمكن الطالب من التركيز على تصميم النموذج(Architecture Design) ، أي اختيار الطبقات الصحيحة لإيجاد حل فعال للمشكلة الاتصالية، بدلاً من إضاعة الوقت في برمجة العمليات الرياضية الأولية.

التمرين 3: سؤال/ جواب

	السؤال	الجواب
1	ما هي اللغة الأكثر هيمنة وشهرة حاليًا في مجال الذكاء الاصطناعي والتعلم الآلي؟	اللغة هي بايثون**(Python)** .
2	ما هي الخاصية البرمجية الأساسية في Python التي تجعلها خيارًا مفضلاً لرفع إنتاجية المبرمجين؟	كونها لغة **عالية المستوى (High-Level Language)** ذات بناء جمل بسيط وواضح.
3	ما المكتبة الأساسية في Python المسؤولة عن توفير هياكل بيانات متقدمة (كالمصفوفات والأبعاد المتعددة) والعمليات الحسابية المتجهة؟	مكتبة **NumPy**.
4	ما اسم المكتبة التي تُستخدم بشكل أساسي في مرحلة هندسة البيانات (Data Engineering)لعمليات التنظيف والمعالجة المسبقة للبيانات الجدولية؟	مكتبة **Pandas**.
5	ما هو دور مكتبات Matplotlibو Seabornفي النظام البيئي لـ Python للذكاء الاصطناعي؟	تستخدم لـ **تصوير البيانات (Data Visualization)** بشكل رسومي لمساعدة الباحثين على فهم الأنماط والنتائج.
6	ما المكتبة الشاملة التي توفر مجموعة واسعة من خوارزميات التعلم الآلي "الكلاسيكية" مثل خوارزميات التصنيف والانحدار؟	مكتبة **Scikit-learn**.

7	لماذا تُستخدم لغة **جافا (Java)** بكثرة في مجال الذكاء الاصطناعي، رغم أنها ليست الخيار الأول لتدريب النماذج الأولية؟	تُفضل لـ **النشر (Deployment)** والإنتاج في البيئات المؤسسية الضخمة بسبب قوتها في أنظمة التشغيل الخلفية.
8	ما هي السمة الرئيسية للغة جافا التي تجعلها قوية في بناء التطبيقات المؤسسية الضخمة؟	كونها لغة **مترجمة وقابلة للتنفيذ عبر المنصات (Platform Independent)**.
9	ما هي اللغة منخفضة المستوى نسبيًا التي تتميز بـ **سرعة تنفيذها الفائقة** والتحكم المباشر بالذاكرة والعتاد؟	لغة **سي بلس بلس(C++)** .
10	كيف تساهم C++ في أداء مكتبات Python الكبرى مثل NumPy وأطر العمل؟	يتم كتابة **النواة الأساسية (Kernels)** والعمليات الحسابية المُجهِدة لتلك المكتبات باستخدام C++ لضمان أعلى أداء.
11	ما هو مجال التطبيق الذي تُعد فيه C++ ضرورية لبرمجة الروبوتات والمركبات ذاتية القيادة؟	**الأنظمة المدمجة.(Embedded Systems)**
12	اذكر مثالين على لغات "تاريخية" كانت رائدة في الفترة المبكرة من الذكاء الاصطناعي الرمزي؟	**LISP**و **Prolog**.
13	ما النوع من الأنظمة الذي لا يزال يُستخدم فيه LISP وProlog ، ويعتمد على القواعد والمنطق المبرمج؟	**الأنظمة الخبيرة(Expert Systems)** .
14	ما هو المصطلح الذي يصف توفير أطر العمل مستوى منه يسمح للمبرمج بالتركيز على تصميم الشبكة بدلاً من كتابة العمليات الرياضية؟	**التجريد(Abstraction)** .
15	ما هو إطار العمل للتعلم العميق الذي طورته شركة جوجل ويتميز بتوجهه نحو النشر السهل على منصات متعددة؟	**TensorFlow**.
16	ما هي الميزات التي يقدمها TensorFlow لتمكينه من النشر على الأجهزة المحمولة والمتصفحات؟	يوفر أدوات مثل **TensorFlow Lite**و **TensorFlow.js**.
17	ما هو إطار العمل الذي طوره مختبر أبحاث الذكاء الاصطناعي التابع لـ فيسبوك(Meta) ؟	**PyTorch**.
18	ما المفهوم الذي يستخدمه PyTorch ويجعله مفضلاً في الأوساط البحثية لمرونته العالية وتسهيل عملية التصحيح؟	**الرسوم البيانية الحاسوبية الديناميكية (Dynamic Computation Graphs)**.
19	ما هي المكتبة/الواجهة البرمجية التي تُعد الخيار الأمثل للمبتدئين ولـ **التطوير السريع(Rapid Prototyping)** ؟	**Keras**.
20	ما الخاصية البرمجية (المذكورة في المحاضرة 1) التي يخدمها مستوى التجريد العالي فيKeras ؟	**العمومية.(Generality)**
21	إذا كان المشروع يتطلب تصنيف الرسائل الجماهيرية (معالجة بيانات ضخمة)، فهل تُفضل Python أمC++ ؟	**Python**(مع إطار عمل مناسب) لأن سرعة التطوير أهم من سرعة التنفيذ الخام.
22	ما هي اللغة الأنسب لبرمجة نظام تحكم لربط أجهزة استشعار داخل روبوت صغير(Embedded System) ؟	**C++**.
23	لماذا تُعد C++ هي الأنسب لبرمجة الأنظمة المدمجة في الروبوتات؟	لأنها تتيح **التحكم الدقيق في العتاد والذاكرة** وتضمن **الفاعلية الزمنية(Real-time Efficiency)** .
24	ما هي الفائدة الرئيسية التي يحققها التجريد في Keras لطالب الاتصال الاستراتيجي غير المتخصص في الرياضيات؟	تقليل الحاجة للغوص في تفاصيل الرياضيات المعقدة، والتركيز على **تصميم النموذج(Architecture Design)** .
25	ما المصطلح الذي يصف استخدام Python كخيار أفضل من اللغات الأخرى التي تتطلب تفاصيل معقدة لإدارة الذاكرة؟	**إنتاجية المبرمج(Programmer Productivity)** .

26	ما هو دور Python في دعم مفهوم البرمجة الشيئية (Object-Oriented Programming - OOP)؟	تدعم OOP بشكل ممتاز، مما يسهل كتابة شفرات **قابلة لإعادة الاستخدام (Reusable)** ومرنة.
27	ما هي طبيعة البيانات التي تعالجها مكتبة Pandas بشكل أساسي في مرحلة هندسة البيانات؟	**البيانات الجدولية(Tabular Data)** .
28	لماذا تُعتبر C++ لغة أساسية في الرؤية الحاسوبية والتحكم الآلي؟	لأن كل جزء من الثانية في التنفيذ مهم لسلامة القرار، وتتطلب **سرعة تنفيذ فائقة**.
29	ما هو المجال الذي تُستخدم فيه C++ على نطاق واسع وتستخدم الذكاء الاصطناعي لتسيير سلوك الشخصيات غير اللاعبة(NPCs) ؟	**الألعاب والرسوميات(Game Engines)** .
30	اذكر بشكل عام الميزة الرئيسية التي تمنحها أطر العمل (Frameworks) لمطوري الذكاء الاصطناعي؟	توفر **الأدوات والبنية التحتية الجاهزة**)البنية التحتية الرياضية) لبناء النماذج المعقدة دون البدء من الصفر.