تقرير TAXI TRIP DATA (BIG DATA~10GB):TP02

الهدف

- يهدف هذا العمل التطبيقي إلى استكشاف الطرق الفعّالة للتعامل ومعالجة مجموعات بيانات CSV ضخمة (أكثر من 5 جيجابايت) باستخدام لغة Python
- تتجاوز مجموعات البيانات الكبيرة بسهولة سعة ذاكرة النظام، مما يؤدي إلى انهيار النظام أو ظهور أخطاء نفاد الذاكرة عند محاولة قراءتها مباشرة.
- يُقارن هذا التجريب بين ثلاث طرق رئيسية لقراءة وتحليل ملفات CSV كبيرة بكفاءة.
- Dataset: /kaggle/input/taxi-trip-data/2018_Yellow_Taxi_Trip_Data.csv
  Size: 9.71 GB | Rows: 112,234,626

الطرق المطبقة

Pandas with chunksize

تعتمد هذه الطريقة على قراءة ملف CSV على شكل أجزاء صغيرة (Chunks)
بدل تحميل مجموعة البيانات بالكامل في الذاكرة دفعة واحدة، تقوم Pandasبمعالجة عدة آلاف من الصفوف في كل مرة، مما يساعد على الحفاظ على انخفاض استهلاك الذاكرة(RAM)

DASK

تتيح مكتبة Dask تنفيذ المعالجة المتوازية باستخدام عدة أنوية للمعالج (CPU cores)
تقوم بقراءة ملفات CSV الضخمة على شكل أقسام (partitions) وتُجري العمليات بشكل مؤجل (Lazy evaluation) حتى يتم استدعاء الدالة ().compute
يساهم هذا الأسلوب في تسريع معالجة الملفات الكبيرة خصوصًا على الأنظمة التي تحتوي على معالجات متعددة.

Pandas with Gzip Compression

يتم أولًا ضغط مجموعة البيانات إلى تنسيق **.gz** باستخدام مكتبة **gzip** ثم تُقرأ بواسطة Pandas
يساعد الضغط على تقليل مساحة التخزين المستخدمة ولكنه قد يزيد قليلاً من زمن القراءة بسبب عملية فك الضغط أثناء التحميل.

النتائج

Method	File Size (GB)	Rows Read	Time Taken (s)	Memory Used (GB)	Notes
Pandas + Chunksize	9.71	112,234,626	486.50	1.30	Sequential read, very low memory
Dask	9.71	112,234,626	307.57	2.84	Fast, parallel processing
Pandas + Gzip	~3.4 (compressed)	112,234,626	389.68	2.84	Smaller size, moderate speed

التحليل

الطريقة	الإيجابيات	السلبيات
Pandas + Chunksize	-استهلاك منخفض جدًا للذاكرة -سهل التنفيذ - يعمل على الأجهزة محدودة الموارد	-بطيء مع مجموعات البيانات الكبيرة جدًا -لا يدعم المعالجة المتوازية -يتطلب التكرار اليدوي على الأجزاء
Dask	-سريع وقابل للتوسّع -يستخدم عدة أنوية للمعالج (CPU cores) -يتعامل بكفاءة مع البيانات الضخمة	-يستهلك ذاكرة أكبر -إعداد أكثر تعقيدًا قليلًا
Pandas + Gzip	-حجم ملف أصغر - يوفر مساحة التخزين - سهل الدمج والاستخدام	-أبطأ في القراءة بسبب فك الضغط - يستهلك طاقة معالجة (CPU) أكبر

الخاتمة

لكل طريقة نقاط قوتها حسب الموارد المتوفرة والأهداف المرجوة:

- طريقة Pandasمع chunksize مثالية للأجهزة ذات الذاكرة المحدودة منخفضة
- طريقة Daskالأفضل عندما تتوفر السرعة والمعالجات متعددة الأنوية(multi-core CPUs)
- ضغط الملفات باستخدام Gzip ممتاز لتوفير مساحة التخزين.

إن التعامل الفعّال مع ملفات CSV الضخمة يتطلب موازنة بين السرعة واستهلاك الذاكرة ومساحة التخزين.
يُعدّ الدمج بين Dask والضغط (compression) الحل الأمثل لتحقيق أداء متوازن في مشاريع البيانات الضخمة الواقعية.