# مقدمه انجام کار گروهی:

در ابتدا خواستیم که یک پروژه در databriks ایجاد کنیم اما بدلیل عدم دسترسی به فیلتر شکن همه اعضا گروه نتوانستیم به این سایت دسترسی داشته باشیم پس تصمیم بر این شد که بصورت لوکال کار را پیش ببریم و کد را در github برای دسترسی همه اعضا به اشتراک بگذاریم.

Databricks یک شرکت نرم افزاری آمریکایی است که توسط سازندگان آپاچی اسپارک تأسیس شده است. Databricks یک پلت فرم مبتنی بر وب برای کار با Spark ایجاد می کند که مدیریت خودکار خوشه و نوت بوک های سبک IPython را ارائه می دهد.

این شرکت که در زمینه داده های بزرگ(BigData) فعالیت می کند، عمده ابزار هایی که توسعه می دهد را به صورت عمومی منتشر کرده است.

انبارهای داده (data warehouses) برای داده های ساختاریافته ایده آل هستند، اما برای رسیدگی به داده های بدون ساختار، داده های نیمه ساختار یافته و داده هایی با تنوع، سرعت و حجم بالا، دریاچه های داده (Data Lakes) معرفی شده اند.

پلتفرم Databricks Lakehouse بهترین ویژگی های دریاچه های داده و انبارهای داده را برای ارائه قابلیت اطمینان، حاکمیت قوی، انعطاف پذیری و پشتیبانی از یادگیری ماشینی، ترکیب می کند.

و اما apache zeppelin یک پلتفرم جهت مصور سازی داده است. در واقع این نوت بوک مکانی برای تمام نیاز های شماست از جمله: کشف داده ها، تجزیه و تحلیل داده ها و تجسم داده ها.

Apache Spark یک سیستم محاسباتی خوشه ای سریع و همه منظوره است که APIهای سطح بالا را که از نمودارهای اجرای عمومی پشتیبانی می کند در جاوا، اسکالا، پایتون و R و یک موتور بهینه سازی شده ارائه می دهد. Apache Spark در Zeppelin پشتیبانی می شود.

مقایسه zeppelin و Databricks:

|  |  |  |
| --- | --- | --- |
|  | zeppelin | Databricks |
| **setup** | خود میزبان(Self-hosted) | خود میزبان یا کاملا مدیریت شده |
| **JUPYTER COMPATIBILITY** | ندارد | سازگار با ژوپیتر(JUPYTER) |
| **PROGRAMMING LANGUAGES** | Python, SQL, Spark | JUPYTER |
| **DATA VISUALIZATION** | ناشناخته | مصور سازی با UI |
| **COLLABORATIVE EDITING** | مبتنی بر فایل یا نا همزمان | بلادرنگ |
| **PRICING** | رایگان | رایگان و پولی |
| **LICENSE** | منبع باز | اختصاصی |

**مراحل نصب و راه اندازی spark :**

ابتدا ابزارهای زیر را به ترتیب نصب کردیم:

Java.SE.Development.Kit.17.0.5.x64

Git-2.39.1-64-bit

GitHubDesktopSetup-x64

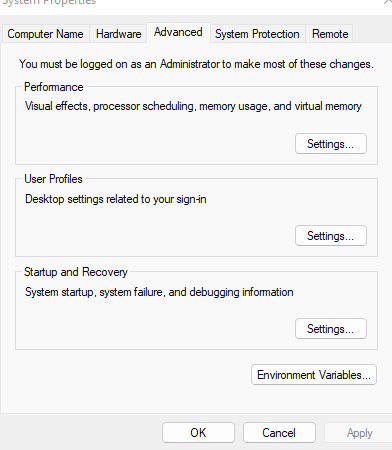
vscode

فایل زیپ زیر را فقط در مسیر درایو c استخراج کرده و به آن دست نمیزنیم:

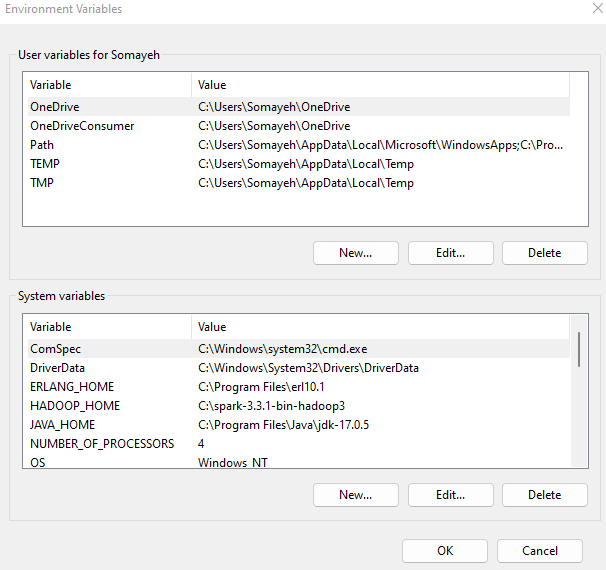
spark-3.3.1-bin-hadoop3.rar

سپس مسیرهای پیش فرض را در EnvironMent Variables اضافه میکنیم برای این کار مراحل زیر را انجام میدهیم:

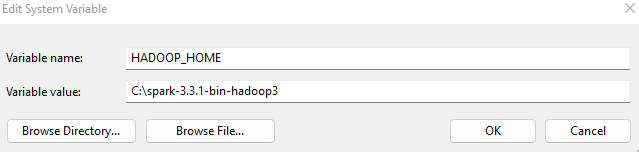
ابتدا system properties را باز میکنیم

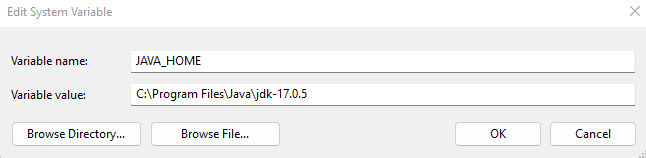


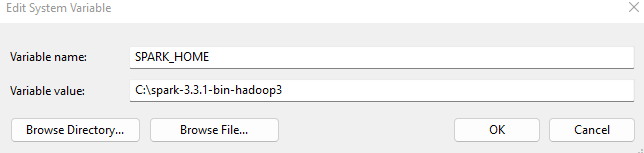
روی دکمه Environment Variables کلیک کرده در قسمت پایین دکمه new را میزنیم:



و سپس مسیر ها را به صورت زیر اضافه میکنیم:







نکته variable value ها با توجه به مسیر نصب برای هر کس ممکن است متفاوت باشد.

برای استفاده از پوسته اسپارک در Command Prompt دستور spark-shell را اجرا میکنیم و سپس دستور pip install pyspark را در همین محیط، برای نصب pyspark اجرا میکنیم.

سپس برای استفاده از کد موجود در github از آن یک شبیه سازی(Clone) در درایو c ایجاد میکنیم. با اجرای دستور cd c:/ در Command Prompt خط دستور(Command Line) را به درایو c منتقل میکنیم و دستور زیر را در آن اجرا مکنیم:

Git clone ssh address in github for this project

سپس برای استفاده از کد شبیه سازی شده در درایو c دستورات زیر را به ترتیب در Command Prompt اجرا میکنیم:

c:\>cd spark\_project

c:\spark\_project>code .

و برای استفاده از github desktop دستور زیر را اجرا میکنیم:

c:\spark\_project>github .

در نهایت بدلیل اینکه انجام مصور سازی(Visualization ) بصورت محلی منجر به خروج از چهارچوب اسپارک می­شد به ناچار از دو پلت فرم معرفی شده فوق یعنی Dtatabricks که بر پایه متدهای کلان داده در بستر اسپارک انجام می­شود و Apache Zeppelin استفاده کردیم.