به نام خدا

خلاصه کوتاه و نکات

پروژه مرتب سازی و تمیزکردن دیتاست ها

مهدی میر

تیر ۱۴۰۱

خلاصه

- دیتای متغیر های یکسان (مثلا قیمت سهام روزانه) از بین فولدرها و دیتاست های مختلف بعضا تکراری استخراج، تمیز و با یکدیگر مرج شده است.
  - به عنوان مثال دیتای قیمت سهام در پروژه ها و فولدر های متفاوت با فرکانس های متفاوت، پوشش سهم و پوشش زمانی متفاوت موجود بود که تا حد امکان با یکدیگر ادغام و تمیز شده اند.
- از تکرار دیتا در دیتاست های نهایی جلوگیری شده است.
  - مثلا دیتای اندازه بازار سهم ها فقط در یک جا آمده است. برای استفاده از آن در جاهای دیگر و قرار دادن نظیر به نظیر آن در دیتاست های مختلف کافی است که با استفاده از ستون های تعیین کننده (Index Columns) با دیتای مورد نظر مرج شود.
- جلوگیری از این کار به منظور سادگی بیشتر دیتاست ها و جلوگیری از پیچیده شدن دیتاها و کمتر بودن تعداد ستون ها در یک دیتاست می باشد. کمتر بودن دیتا هم به سادگی و هم به کم حجم شدن فایل ذخیره شده می انجامد. همچنین اینکار بعضا به خاطر پوشش متفاوت زمانی یا سهمی دیتاهای متفاوت می باشد.
  - به عنوان مثال در مورد اطلاعات قیمت دلار در یک دیتاست بازه زمانی بیشتری وجود داشت که اگر با دیتاست دیگر ادغام میشد تعداد زیادی مقدار None به ازای سلول های خالی در دیتاست نهایی تولید میشد که جالب و مناسب نیست.
- دیتاست ها به منظور سادگی بیشتر و سهولت استفاده و دسته بندی تا حد ممکن و بهینه از هم تفیک شده اند.
  - مثلا دیتای تعداد کل سهام یک شرکت از دیتای قیمت روزانه جدا شده است.

- اینکه یک دیتاست شامل دیتای پایه ای است و از روی دیتاست های دیگر همین مجموعه دیتا بدست آمده است یا نه مشخص شده است. که در اکسل گزارش کلی به تفکیک هر دیتاست گزارش شده است.
  - به عنوان مثال دیتای مارکت کپ از روی دیتای قیمت نهایی تعدیل شده هر سهم در هر روز و همچنین دیتاست تعداد سهام آن شرکت به صورت روزانه بدست آمده است و در نتیجه به عنوان دیتای "بدست آمده (Derived)" علامت گذاری شده است.
  - مشخص کردن دیتای پایه ای (Baseline) در مقابل دیتای (Derived) این مزیت را ایجاد می کند که میدانیم برای آپدیت کردن دیتا لازم است کدام دیتاست ها از سورس های مورد نظر دیتای جدید را دریافت کنند و آپدیت بشوند و دیتاهای بدست آمده از آن ها با کد قابل آپدیت کردن است که مرحله بعد از آپدیت دیتای پایه ای می باشد.
- دیتاها در فولدرها به صورت هرمی از کل به جز دسته بندی شده اند.
- دیتای غیر حجیم (زیر ۲۰ هزار خط) به صورت فایل اکسل و بالای ۲۰ هزار خط به خاطر جاگیری کمتر و هم چنین سرعت خواندن و نوشتن بسیار بالاتر نسبت به اکسل با فرمت Parquet ذخیره شده اند.
  - در مواردی که دیتا با فرمت Parquet ذخیره شده است یک سمپل ۵ هزارتایی از دیتا با فرمت اکسل برای دسترسی و مشاهده سریع دیتا در کنار دیتا قرار داده شده است.
  - فرمت اکسل با تعداد خط خیلی بالا اصولا بهینه نیست علاوه بر حجم بالا، در صورت ذخیره سازی دیتای حجیم به صورت اکسل، برای باز کردن و خواندن و نوشتن و کار با عموما کامپیوتر ها به مشکل و کندی بر میخورند.
  - در کل مجموعه فقط از این دو فرمت ذخیره سازی دیتا استفاده شده است و از تعدد فرمت جلوگیری شده است. فرمت های دیگر نظیر CSV تماما به یکی از این دو فرمت که بهینه تر هستند تبدیل شده اند.
- در کنار هر دیتاست (فولدرهای نهایی) یک فایل META.json وجود دارد که در مورد همان دیتا اطلاعات کلی نظیر توضیح کوتاه یک خطی و شروع و پایان زمانی دیتا و فرکانس دیتا موجود است.
  - این اطلاعات به صورت یک جا در اکسل گزارش کلی هم برای هر دیتاست وجود دارد.
- اسم ستون های یکسان در دیتاست های مختلف یکی شده است.
  - نه تنها اسم ستون بلکه فرمت ذخیره سازی مقادیر یکسان به صورت فرمت مشخص و یکسانی بین همه دیتاست ها رعایت شده است.
  - مثلا تمامی ستون هایی که مقادیر آن شامل یک تاریخ شمسی است با اسم JDate و به صورت استرینگ با فرمت یکسان به صورت "1399-08-21" ذخیره شده اند.

- در مورد فرمت ذخیره سازی ستون های دیتا ملاحظات زیادی از نظر فنی و کاربردی لحاظ شده است و وسواس زیادی خرج شده است.
    - مثلا برای تاریخ ها از استرینگ استفاده شده است زیرا اولا تاریخ یک متغیر کاردینال نیست و ثانیا با وجود خط فاصله در تاریخ با یک نگاه به دیتا و ستون مشخص می شود که با تاریخ سر و کار داریم علاوه بر این خواندن آن با خط فاصله راحت تر است. و نهایتا در صورت ذخیره سازی مثلا به صورت 13990821 وقتی به صورت اکسل یا فرمت های دیگر ذخیره می شود و در خلال کد زنی بعضا به طور ناخواسته و اتوماتیک توسط مثلا اکسل به عدد تبدیل می شود و مانند عدد با آن رفتار می شود و ممکن است حتی به صورت عدد اعشاری به صورت تصادفی و ناخواسته توسط سیستم ذخیره بشود و مثلا بعد از آخرین رقم همه ی ستون دیتای تاریخ با ۴ تا صفر بعد از اعشار ذخیره شده باشند.
- به عنوان گزارش کلی از ساختار فولدری دیتا ۳ فایل PDF که ساختار درختی فولدرها را به صورت نمودار درختی نشان می دهد قرار گرفته است که می‌تواند برای آگاهی از نحوه دسته بندی دیتاها و همچنین اعمال تغییرات لازم مورد استفاده قرار گیرد.
- بعضی دیتاست ها دیتاست های کمکی هستند که برای استفاده از دیتاست های اقتصادی کاملا مورد نیاز و ضروری می باشند اما در دسته دیتاست های کمکی قرار می‌گیرند و دیتای اقتصادی در نظر گرفته نشده اند. دیتای اقتصادی را دیتایی در نظر گرفته ام که مستقیما متغیر عددی اقتصادی یا زمانی در آن بکار رفته باشد.
- بعضی دیتاها ساختار دیتاست ندارند که به صورت جدا دسته بندی شده اند.
- بعضی دیتاها به دلیل مشکلات خود دیتا یا به دلیل اینکه تمیز کردن و مرج کردن و مرتب سازی آن ها نسبت به دیتاهای دیگر خیلی پیچیده تر و زمان برتر بوده اند جدا قرار گرفته اند که لازم است این ها هر کدام بررسی و به دیتاست های تمیز شده به مرور زمان اضافه شوند.
- دیتاهای تمیز شده و مرتب شده لزوما به معنای کاملا تمیز و پرفکت بودن دیتا نیستند بلکه هر دیتا مشکلات و مسائل خودش را دارد که باید هر کدام جدا بررسی و انجام شود.
- لازم است که ورژن قبلی دیتا هم به عنوان بکاپ هم به هدف ورژنینگ نگه داری شود.
    - این کار بدلیل احتمال اشتباه در مرج دیتاها هم لازم است. ممکن است جایی اشتباهی کرده باشم که بعدا توسط استفاده تخصصی از دیتا مشخص شود که لازم است به ورژن قبلی دیتا مراجعه شود و مقادیر صحیح دوباره استخراج شوند.
    - دیتای خام فقط در ورژن قبلی وجود دارد و در ورژن جدید دیتای خام فقط دیتای تمیز شده و نهایی موجود است.
- در دیتاست های نهایی هیچ کدام شامل دیتای خام نیستند.
    - این کار اولا به خاطر ساده تر بودن ساختار دیتاها می باشد.

- دوما اغلب دیتاست های نهایی از چند دیتای خام و نیمه تمیز شده یا از مرج کردن چند دیتاست مختلف بدست آمده اند و اینکه دیتای خام مربوط به کدام دیتاست می باشد دیگر بی معنی است چون ساختار عوض شده است.

نکات اکسل گزارش

- در اکسل گزارش، اسم مجموعه دیتا به همراه آدرس آن در فولدر دیتاست ها گزارش شده است. همچنین به ازای هر دیتاست نهایی موارد زیر موجود است:
  - توضیح تک خطی دیتا
  - دیتای پایه ای بودن در مقابل دیتای بدست آمده (Baseline VS Derived)
  - تاریخ شروع و پایان دیتا
  - فرکانس دیتا
  - تعداد خطوط دیتا (یا تعداد مشاهدات)
  - تعداد ستون های دیتا
  - اسم نرمالایز شده ستون های دیتا. جهت اینکه با یک نگاه به صورت تقریبا دقیق متوجه شد دیتا شامل چه مواردی است.

نکات نمودار درختی دسته بندی دیتاها

- نمودار درختی دسته بندی دیتاها از روی ساختار فولدری دیتاها کشیده شده است و دقیقا مطابق ساختار فولدری و دسته بندی انجام شده می باشد.
- گره های نهایی نمودار Terminal Nodes فولدرهایی هستند که شامل دیتاست نهایی هستند و دیگر زیر فولدر ندارند.
- صرفا جهت خیلی عریض نشدن نمودار درختی سطح بالاتر دسته بندی بعضا زیر سطح پایین تر از دسته بندی مجاور قرار گرفته اند که با رنگ بندی نمودار جهت تشخیص سریع به این مساله پرداخته شده است.
- گره های نهایی که شامل دیتاست ها هستند به صورت مستطیل و گره های بالاسری که صرفا اسم فولدرها و نشانه و اسم دسته بندی هستند به صورت بیضی در نمودار نمایش داده شده اند.
- گره های غیر نهایی که نشانگر اسم دسته و سطح دسته بندی هستند در هر سطح رنگ یکسان با یکدیگر دارند که همرنگ یال های متصل کننده آن ها به دسته بندی بالاسری می باشد. همچنین رنگ سطوح مختلف با هم متفاوت است.
- گره های نهایی علاوه بر مستطیل شکل بودن بر اساس فرکانس خودشان رنگ بندی شده اند که در کنار نمودار رنگ فرکانس ها ذکر شده است مثلا فرکانس روزانه دارای رنگ نارنجی می باشد.

- دیتاست های نهایی که دیتای پایه ای هستند دارای رنگ پر رنگ از رنگ فرکانس خود هستند اما اگر دیتاست بدست آمده باشند دارای رنگ کم رنگ شده همان رنگ فرکانس خودشان هستند همچنین خط دور آن ها نقطه چین شده است.
- نمودار علاوه بر عکس موجود در پوشه توسط لینک زیر قابل مشاهده است. لینک دارای قابلیت زوم بیشتر و بهتر دیدن نمودار را فراهم می‌کند.
    - لینک

بعضی پیشنهادات برای ادامه، غنی سازی و گسترش کار

- دیتاست هایی که موفق به تمیز شدن آن نشده ام توسط افراد دیگر و در قالب پروژه های جدا تمیز و به مجموعه دیتاهای تمیز اضافه بشوند.
- کیفیت دیتا در دیتاست های تمیز شده بالاتر برود.
- کدهای مربوط به استخراج و تمیز سازی با استفاده از ابزارهای روز مثل گیت‌هاب جمع آوری و دسته بندی و ساختار بندی بشود. مشخص باشد هر کد مربوط به کدام دیتا و کدام بخش جمع آوری و پردازش دیتا می باشد. بهتر است یک سرور برای اجرای کدها اختصاص داده شود و علاوه بر مشخص کردن پیش نیاز های هر پروژه ی کد، روی آن سرور دقیقا قابل اجرا باشد برای استفاده های آینده، آپدیت کردن یا اعمال اصلاحات دوباره روی دیتا در آینده.
- سورس هر دیتا به طور دقیق مشخص بشود.
- در مورد هر دیتاست با استفاده از مثلا Google data studio گزارش های کمی و کیفی ساخته بشود که به راحتی قابل آپدیت کردن بعد از آپدیت شدن دیتا باشد.
- میزان پوشش هر دیتاست از دیتایی که گزارش می کند مشخص شود نسبت به کل جامعه آن دیتا.
- سوراخ ها و خالی بودن دیتاست ها مشخص بشوند و نقطه ضعف های هر دیتابیس به صورت سیستماتیک مشخص بشود.
- دیتاهای کمکی (Helpers) کاملا به صورت پابلیک در دسترس دانشجویان قرار بگیرد چون وقت زیادی از هر کس خواهد گرفت که نتیجه اقتصادی و ریسرچی هم ندارد اما در هر کاری لازم است.
    - همچنین به صورت اپن سورس با استفاده از مکانیزم مثلا گیت‌هاب دیتاهای کمکی بعد از انجام هر پروژه توسط دانشجویان کامل تر و غنی تر و آپدیت تر بشوند. (استفاده از کارهای دستی انجام شده در پروژه های مختلف و حفظ و انتقال آن به نسل بعدی پروژه ها و دانشجویان)
- برای استفاده دانشجویان در سطوح مختلف دسترسی های متفاوت تعریف بشود و مثلا یک ورژن از نمودار درختی برای دیتاهایی که صرفا اعلام داشتن آن ها موردی ندارد به صورت عمومی منتشر

بشود و برای دسترسی بیشتر بسته به دیتا و شرایط کار و پروژه به صورت موردی دسترسی های گوناگون تعریف و اعمال بشود.

- به صورت دوره ای از دیتا بکاپ با فشردگی بالا و حجم پایین گرفته شود.