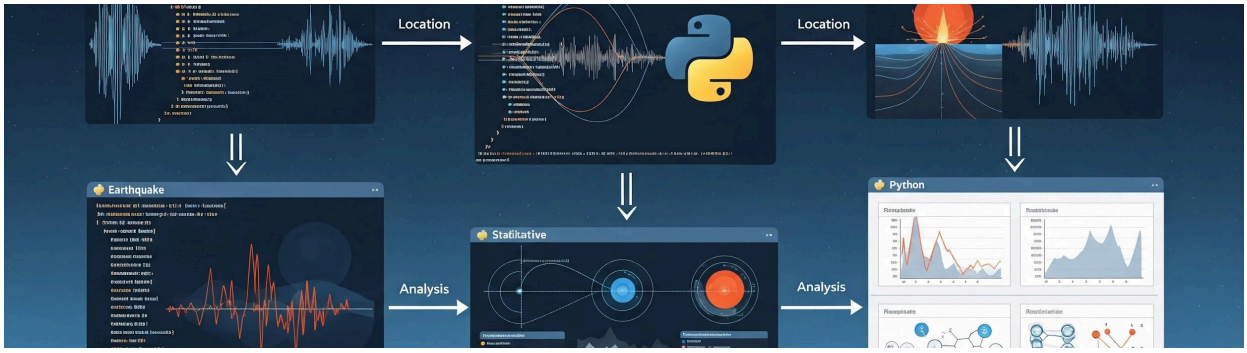

به نام خالق زیبایی‌ها

پروژه نهایی بوت‌کمپ پایتون آلفا | سری دهم بوت‌کمپ‌های کوئرا

تابستان ۱۴۰۴

تحلیل داده‌های زلزله در ژاپن

این پروژه با هدف مقایسه منابع و تحلیل داده‌های زلزله‌خیزی در ژاپن انجام می‌شود. برای کارایی بیشتر، از ترکیب ابزارهای مختلف استفاده می‌کنیم: SQL برای ذخیره و query سریع داده‌های حجیم، Pandas برای تحلیل سطح بالا و پردازش دسته‌ای داده‌ها، و NumPy برای محاسبات دقیق عددی و برداری. به‌طور خاص، SQL برای ذخیره و مدیریت داده‌های بزرگ و انجام query های اولیه بهینه‌سازی شده است، Pandas امکان پردازش انعطاف‌پذیر و تحلیلی در حافظه را با متدها و امکانات جامع فراهم می‌کند، و NumPy در محاسبات برداری و ریاضی سنگین عملکرد بسیار بالایی دارد. این ترکیب ابزاری به ما امکان می‌دهد از مزایای هر کدام بهره ببریم و تحلیل جامع و سریع‌تری داشته باشیم.

توجه کنید که:

- در این پروژه نیاز به نگهداری داده‌ها در قایل ها و دیتابیس دارید.
- هم‌چنین کد شما در همان محیط ترمینال اجرا می‌شود و نیاز به پیاده‌سازی رابط کاربری (UI) ندارید.

- استفاده از **گیت** در این پروژه الزامیست. شما می بایست از ابتدا با گیت کار کرده و ریپازیتوری پروژه خود را در گیت هاب بسازید. تمام تاریخچه فعالیت های شما در ریپازیتوری پروژه مشاهد می شود لذا نمی توانید صرفا در آخر کار کد های تکمیل شده را در گیت هاب قرار دهید. در انتها تنها لینک گیت هاب خود را داخل یک فایل متنی قرار دهید. (تنها فرمت قابل قبول همین است و بار گذاری فایل پایتون امکان پذیر نمی باشد)
- سعی کنید از یک ورژن مشخص پایتون و کتابخانه ها استفاده کنید و یک VENV مربوط به پروژه بسازید و کنار پروژه فایل requirements که شامل کتابخانه ها و .... را کنار فایل اصلی پروژه خود بسازید. همچنین یک فایل Read me در کنار پروژه می تواند دید بهتری به کد شما بدهد.
- برای شروع پروژه خود و دید بهتر نسبت به مسیر و تقسیم بندی بهتر وظایف استفاده از فلوچارت اجباری می باشد.
- در نهایت برای ارائه های خود یک فایل پاورپوینت ارائه آماده سازی کنید

در ادامه توضیحات پیاده‌سازی هر بخش برای شما آمده است.

۱. سناریوی رخداد زلزله

ژاپن همواره به‌عنوان یکی از زلزله‌خیزترین کشورهای جهان شناخته شده است. هدف این پروژه تحلیل جامع داده‌های زلزله یکماه ماه گذشته در ژاپن است تا بتوانید از صفر تا تحلیل نهایی، داده‌های واقعی را از سه منبع معتبر (EMSCو USGS و GEOFON) و دیتاستی که برای شما از زلزله های ژاپن تهیه کردیم جمع‌آوری، پردازش، تحلیل آماری و بصری‌سازی کنید و نتایج را گزارش دهید. تمرکز اصلی بر درک تفاوت منابع، کیفیت داده‌ها، تحلیل خطر و استخراج اطلاعات مفید خواهد بود.

---

۲. استخراج و گردآوری داده‌ها

منبع اول دریافت داده‌های زلزله از API USGS: از سرویس Earthquake Catalog API سازمان زمین‌شناسی آمریکا (USGS) استفاده کنید تا داده‌های زلزله‌های ژاپن را در بازه زمانی ماه گذشته

دریافت نمایید. می‌توانید پارامترهایی مانند محدوده جغرافیایی ، بازه زمانی و حداقل بزرگی را مشخص کنید و خروجی را به فرمت CSV ذخیره کنید.

```
import requests
from datetime import datetime, timedelta




end_date = datetime.today().date()
start_date = end_date - timedelta(days=30)


url = "https://earthquake.usgs.gov/fdsnws/event/1/query"
params = {
    "format": "csv",
    "starttime": str(start_date),
    "endtime": str(end_date),
    "minlatitude": 24,
    "maxlatitude": 46,
    "minlongitude": 123,
    "maxlongitude": 146,
    "minmagnitude": 1
}
```

```
response = requests.get(url, params=params)



with open("japan_earthquakes.csv", "w", encoding="utf-8") as f:

    f.write(response.text)
```

منبع دوم وب‌اسکریپینگ از GEOFON: از وب‌سایت موسسه GFZ Helmholtz Centre for Geosciences، جدول زلزله‌های ژاپن در ماه گذشته را Scrape کرده و داده های زلزله‌ها را به وسیله کتابخانه های BeautifulSoup و Request جمع آوری کرده جدول نهایی را به CSV تبدیل نمایید.

می توانید از لینک زیر جداول موجود در HTML این منبع داده مربوط به زلزله های این کشور را در دسترس داشته باشید:

GEOFON Japan Earthquakes Data

فایل‌های نهایی را با نام‌های JAPAN_USGS.csv و JAPAN_GEOFON.csv ذخیره کنید.

منبع سوم وب‌اسکریپینگ از EMSC: از وب‌سایت مرکز لرزه‌نگاری اروپا-مدیترانه (EMSC)، جدول زلزله‌های ژاپن در ماه گذشته را Scrape کرده و داده های زلزله‌ها را جمع‌آوری کنید. برای این کار می‌توانید از کتابخانه `Selenium` استفاده کرده و جدول نهایی را به فرمت CSV ذخیره نمایید.

EMSC Japan Earthquakes Data

منبع چهارم استفاده از دیتاست : با استفاده از دیتاستی که در اختیار شما داده شده عملیات پیش پردازش داده ها را انجام داده و داده ها را تمیز کنید و به عنوان منبع چهارم استفاده کنید.

DATASET JAPAN Eearthquakes data

فایل‌های نهایی را با نام‌های JAPAN_USGS.csv و JAPAN_GEOFON.csv و JAPAN_EMSC.csv JAPAN_DATASET.csv ذخیره کنید.

---

۳. پاک‌سازی و پیش‌پردازش داده‌ها با Pandas و اعمال ریاضیاتی و آماری با Numpy

Pandas برای تحلیل درون‌حافظه و کار با داده‌های جدول‌بندی‌شده (DataFrame) بسیار مناسب است و امکانات فراوانی برای مرتب‌سازی، فیلتر کردن، گروه‌بندی، و ترکیب داده‌ها دارد. با Pandas می‌توان داده‌های خام را تمیز کرد (مثلاً حذف یا پر کردن مقادیر گمشده)، ستون‌های جدید تولید کرد، نوع داده‌ها را تبدیل نمود، و به‌راحتی نمودارهای مقدماتی رسم کرد. همچنین می‌توانیم از ویژگی‌هایی مانند merge() و pivot_table() استفاده کنیم تا جدول‌ها را مانند SQL به یکدیگر متصل کنیم یا جداول محوری برای مقایسه مقادیر بسازیم.

- 1. با استفاده از pandas، چهار فایل CSV حاصب از مرحله قبل را بخوانید.
- 2. تعداد ردیف‌ها و ستون‌ها را برای هر فایل چاپ کنید (shape).
- 3. نوع داده‌ها را بررسی و ستون‌های تاریخ و عددی را به datetime و float تبدیل کنید.
- 4. مقادیر گمشده (NaN) را پیدا کرده و حذف یا جایگزین کنید.
- 5. ستون جدیدی به نام Month از روی time بسازید.
- 6. ستون Category بسازید که شدت زلزله را به صورت طبقه‌بندی‌شده (کمتر از 4 = ضعیف، 4 تا 6 = متوسط، بیشتر از 6 = شدید) نشان دهد.
- 7. گروه‌بندی داده‌ها بر اساس Month و Category انجام داده و میانگین و تعداد زلزله‌ها را محاسبه کنید.
- 8. ستون جدیدی به نام region یا area_name از ستون place استخراج کنید که نشان‌دهنده منطقه وقوع زلزله باشد (مثلاً Tokyo, Honshu, Near East Coast of Japan). این کار باید برای تمام داده ها انجام شود.
- 9. داده‌ها را بر اساس منطقه گروه‌بندی کرده و تحلیل‌های آماری زیر را با استفاده از متد groupby انجام دهید:
    1. تعداد زلزله در هر منطقه (با .size() یا count())
    2. میانگین بزرگی و عمق زلزله‌ها در هر منطقه
    3. حداکثر بزرگی یا عمق ثبت‌شده در هر منطقه
    4. رسم نمودار میله‌ای از تعداد زلزله‌ها در هر منطقه با استفاده از خروجی groupby

کتابخانه NumPy یک ابزار قدرتمند برای محاسبات عددی و برداری است. این کتابخانه با استفاده از آرایه‌های nبعدی (ndarray) عملیات ریاضی مانند ضرب برداری، تجمع (sum)، ضرب داخلی (dot)، و دیگر توابع خطی را با کارایی بسیار بالا انجام می‌دهد. مزیت اصلی NumPy قابلیت «بردارسازی» (vectorization) است که امکان اجرای عملگرها و توابع ریاضی روی کل آرایه‌ها را بدون نیاز به حلقه‌های پایتون فراهم می‌کند.

به این ترتیب، عملیات مانند ضرب دو آرایه یا اعمال توابع ریاضی (مانند np.sqrt یا np.log) مستقیماً در سطح C/Fortran اجرا شده و بسیار سریع‌تر از اجرای مشابه با حلقه‌های پایتون خواهد بود. در پروژه زلزله‌ها می‌توان از NumPy برای انجام محاسبات دقیق آماری و جغرافیایی (مثل محاسبه فواصل مکانی بین زلزله‌ها یا تغییر مقیاس داده‌ها) استفاده کرد.

تسک‌ها و چالش‌های عملی بخش NumPy:

- ساخت آرایه‌های NumPy از داده‌های خام و استفاده از آنها برای محاسبه‌ی ویژگی‌های عددی. به‌طور مثال، با استفاده از فرمول برداری فاصله اقلیدسی بین مختصات دو نقطه (آرایه‌های x و y) می‌توان فاصله بین دو محل وقوع زلزله را محاسبه کرد:

  dist = np.sqrt((x2 - x1)2 + (y2 - y1)2)

  حال از شما خواسته می شود با استفاده از فرمول بالا فاضله کانونی هر کدام از زلزله های رخ داده تا توکیو پایتخت ژاپن را محاسبه کنید. و در ستونی در فایل های داده خود ذخیره کنید

- انجام محاسبات آماری و ریاضی روی آرایه‌ها: مثل محاسبه میانگین، واریانس یا صدیق (percentile) بزرگی زلزله‌ها با توابع NumPy (np.mean, np.std, np.percentile) بدون نیاز به حلقه.

۴. ذخیره‌سازی داده‌ها در پایگاه‌داده SQL

SQL توانایی انجام سریع عملیات روی داده‌های حجیم را دارد و برای محاسبه آمار کلی (مثل تعداد و میانگین) و ترکیب جدول‌ها بسیار مناسب است. به‌عنوان مثال می‌توانیم با SQL تعداد زلزله‌ها را در هر سال و هر کشور محاسبه کنیم یا با استفاده از عمل JOIN، جدول رویدادهای زلزله را با جدول مختصات جغرافیایی یا جمعیتی ترکیب کنیم تا تحلیل‌های مکانی انجام دهیم. SQL همچنین امکان فیلتر کردن سریع داده‌ها (مثلاً بر اساس بزرگی بالای یک مقدار مشخص) و انجام محاسبات مقدماتی را فراهم می‌کند.

تسک‌ها و چالش‌های عملی پیشنهادی بخش SQL:

ایجاد جداول در دیتابیس: در یک پایگاه داده واقعی، جدولی به نام Earthquakes ایجاد کنید که ستون‌هایی مانند تاریخ/زمان، عرض جغرافیایی، طول جغرافیایی، بزرگی (Magnitude)، ژرفا (Depth)، کشور، منبع داده (مثلاً USGS ESMC و یا Dataset یا GEOFON) و... را داشته باشد. وارد کردن دیتافریم به دیتابیس MySql برای مثال :

```
import pandas as pd
from sqlalchemy import create_engine

# 1. Read CSV file
df = pd.read_csv("data.csv")  # Replace with your CSV file path

# 2. Create a database connection
# MySQL:
engine =
create_engine("mysql+pymysql://user:password@host:port/database_
name")
```

```
# 3. Insert data into SQL table
df.to_sql(
    name="your_table_name",
    con=engine,
    if_exists="append",
    index=False,
    chunksize=1000
)
```

1. در یک پایگاه داده واقعی (MySQL یا PostgreSQL)، یک جدول با نام earthquakes طراحی و ایجاد کنید که شامل ستون‌های زیر باشد:

id (کلید اصلی، خودافزاینده)

time (تاریخ و زمان وقوع زلزله)

latitude, longitude (مختصات جغرافیایی)

depth (عمق)

magnitude (بزرگی)

region (نام منطقه استخراج‌شده از place)

source (منبع داده: USGS یا GEOFON)

2. با استفاده از pandas.to_sql داده‌های تمیزشده را از DataFrame به جدول SQL وارد کنید. برای عملکرد بهتر، از chunksize استفاده کرده و 'if_exists='append را مشخص نمایید.

3. نوشتن کوئری‌های SQL برای تحلیل:

- محاسبه تعداد کل زلزله‌ها به تفکیک ماه و منطقه:

```
SELECT region, EXTRACT(MONTH FROM time) AS month, COUNT(*) FROM
earthquakes GROUP BY region, month
```

- محاسبه میانگین بزرگی در هر منطقه و منبع:
- استخراج ۱۰ زلزله شدید اخیر:
- محاسبه بیشترین و کمترین عمق در هر منطقه:
- حذف داده‌های خارج از محدوده یا مشکوک:
- به‌روزرسانی داده‌های ناقص:
- (اختیاری): تعریف شاخص (Index) برای ستون‌های پرجستجو مانند time, region یا magnitude برای بهبود سرعت کوئری‌ها

---

۵. مصورسازی داده‌ها

از کتابخانه‌های تصویری Python نظیر Matplotlib برای نمایش نتایج استفاده کنید. نمودارهای مورد نیاز عبارتند از:

- هیستوگرام : برای توزیع بزرگی زلزله ها در هر شهر (مثلا تفکیک شده بر اساس هر شهر یا منابع)
- نمودار خطی : روند زمانی شمارش زلزله ها و میانگین بزرگی زلزله ها به ازای هر هفته یا هر روز
- نمودار پراکندگی (Scatter): بزرگی عمق زلزله یا بزرگی در برابر زمان
- نمودار جعبه ای (Boxplot): مقایسه توزیع بزرگی با عمق زلزله ها
- هیت مپ (Heatmap):نقشه حرارتی توزیع جغرافیایی زلزله ها با استفاده از طول و عرض جغرافیایی زلزله های رخ داده و همچنین نقشه حرارتی زلزله های رخ داده با استفاده از فاصله کانونی زلزله تا توکیو (امتیازی)

---

۶. تست‌نویسی و اعتبارسنجی

- برای اطمینان از صحت مراحل انجام شده، تست‌های واحد (Unit Tests) بنویسید. این تست‌ها می‌توانند موارد زیر را پوشش دهند:
- بررسی صحت دریافت داده (مثلاً تعداد رکوردها یا صحت ساختار جدول خروجی CSV)
- اطمینان از حذف مقادیر ناقص (چک کنید ستون‌های کلیدی فاقد مقدار خالی باشد)
- صحت نوع داده‌ها (برای مثال اطمینان از این‌که مقادیر بزرگی عددی هستند)
- محاسبات آماری (اعلام خطا در صورتی که میانگین یا انحراف معیار به شکل غیرمنتظره‌ای متفاوت باشد)
- صحت درج داده‌ها در پایگاه داده (مثلاً تعداد ردیف‌ها بعد از INSERT برابر انتظار باشد)

---

۷. نتیجه‌گیری و تحلیل نهایی

ویژگی‌های در ژاپن: با توجه به تحلیل‌های انجام‌شده، تفاوت‌های شاخص بین زلزله‌های مختلف را بررسی کنید مثلا اکثر این زلزله ها دارای چه عمقی هستند و چه شدتی دارند و با توجه به اینکه هرچه عمق این زلزله ها کمتر و شدت آن ها بیشتر باشد زلزله خطرناک نر است زلزله های ثبت شده را بر اساس خطرناک بودن رتبه بندی کنید

تسک‌ها:

1. بررسی این‌که آیا تمام منابع رخدادهای کوچک را پوشش می‌دهد؟ چه تفاوتی بایکدیگر دارند؟

2. بررسی اینکه زلزله‌های شدید معمولاً در چه عمقی رخ داده‌اند؟

3. رتبه‌بندی خطرناک‌ترین زلزله‌ها (شدت بالا و عمق کم).

4. مقایسه تعداد زلزله‌های بزرگ در هر منبع.

5. نتیجه‌گیری علمی درباره رفتار زلزله‌ای ژاپن در ماه اخیر.(اختیاری)

6. پیشنهاد برای ترکیب منابع برای بهینه سازی کیفیت داده‌ها.(اختیاری)

موفق باشید =)))