Zharfa Job Application Task

دیتا

فایل gene.csv رو دانلود کنید:

- ستون PC1 تا P3 نمایش عددی از دی‌ان‌ای هر فرد است (به منظور مطالعه بیشتر الگوریتم PCA رو مطالعه کنید).
- ستون Lat و Long به ترتیب عرض و طول جغرافایی هر فرد می‌باشد.

تسک

یک اپ fastAPI با پایتون بنویسید که هنگام اجرا دیتای بالا را بخواند و بر اساس PCها یک الگوریتم HCD بزند و خروجی این الگوریتم را درون Redis ذخیره کند چون بعدا به آن نیاز پیدا خواهیم کرد. وقتی که این برنامه کامل شد اپ خود Dockerize کنید.

ساب‌تسک ۱

```
GET /gene/htree
```

درختی که از الگوریتم HCD ساخته شد را در قالب یک جیسون برگردانید (هر کلید جیسون نام خوشه [یک عدد تصادفی و متمایز برای هر خوش] و مقدار آن یک لیست از آیدی سمپل‌هایی که درون آن کلاستر قرار دارند):

```
{
  “data”: {
    “1”: [1, 2, 3, 4],
    “2”: [5, 6]
  }
}
```

ساب‌تسک ۲

```
GET /gene/filter
{
  “rectangle”: [x1, y1, x2, y2],
  “cluster_id”: <cluster_id>,
}
```

آیدی سمپل‌هایی که در کلاستر cluster_id قرار داشته باشند (از دیتای ردیس استفاده کنید) و همینطور از نظر جغرافیای در مستطیل ورودی باشند. برای فیلتر کردن دیتا از Redis استفاده کنید.

```
{
  “data”: [1, 2]
}
```

الگوریتم HCD

فرض کنید می‌خواهیم این الگوریتم را روی دیتای مربوطه بر اساس PCها بزنیم. بار اول همه سمپل ها را به تابع زیر میدهیم و این تابع دیتا را بر اساس شباهت به دو خوشه تقسیم میکند (خوشه ۰ و ۱). فرض کنید به طور تصادفی این دو خوشه را با آیدی ۱ و ۲ ذخیره میکنیم. در مرحله بعد این کار را یکبار برای خوشه صفر و یکبار برای خوشه یک انجام میدهیم تا زمانی که بیشتر از ۳۰ سمپل داشته باشیم. برای درک بیشتر می‌توانید درباره الگوریتم Hirarchical Clustering: Divisive مطاله کنید.

```
from sklearn.cluster import KMeans
def get_labels(data_pc):
  model = KMeans(n_clusters=2)
  model.fit(data_pc)
  Return list(model.labels_)
```