

## پروژک‌های موقعیت شغلی هوش مصنوعی و Back-end Developer:

### بخش اول: هوش مصنوعی

#### پروژک اول: روش‌های کلاسترینگ

قیمت close منهای open به open کندل هر روز را نرخ بازدهی روزانه می‌نامیم (با احتساب علامت). با یکی از پکیج‌های مرسوم کلاسترینگ نرخ بازدهی روزانه بیت کوین را در سه کلاستر طبقه‌بندی نمایید.

خروجی کلاستر را در سه حالت

نرخ بازدهی بزرگ منفی (۱-)

نرخ بازدهی کم و حدود صفر (۰)

نرخ بازدهی بزرگ و مثبت (۱+)

کلاس‌بندی می‌کنیم.

#### پروژک دوم: تحلیل داده

همبستگی نرخ بازدهی بیت کوین را با نرخ بازدهی نفت و طلا جداگانه به دست آورید. برای این کار می‌توانید از دیتابیس یاهو و پکیج‌های گیت‌هاب استفاده کرده و اصلاحات خود را روی آن اعمال نمایید.

حال سعی نمایید اثر لگ زمانی یا تأخیر زمانی را وارد کرده و همبستگی‌ها را مجدد بررسی نمایید. چند لگ زمانی مثبت و منفی را در نظر بگیرید. چه نتیجه‌ای می‌گیرید؟

#### پروژک سوم: یادگیری ماشین

می‌خواهیم یک ماشین پیش‌بینی کلاستر نرخ بازدهی بیت کوین بسازیم، به گونه‌ای که کلاستر هر روز را از داده‌های کلاسترینگ ۴ روز قبل متوالی بسازیم. ابتدا با الگوریتم‌های آموزش، ماشین خود را آموزش داده و سپس تست بگیرید.

## بخش دوم: بک‌اند

**نکته:** این تسک‌ها باید با زبان پایتون و یا Node.js انجام شوند. Node.js به پایتون اولویت دارد.

### پروژک اول:

(۱) با استفاده از پکیج ccxt یا API داکيومنت Binance، کل داده‌های candlestickهای روزانه را برای جفت‌ارزهای BTC/USDT از سال ۲۰۱۷ از سایت Binance دریافت کنید.

(۲) داده‌ها را در یک دیتابیس MongoDB با ساختار مناسب ذخیره کنید.

(۳) با استفاده از GraphQL، دو کوئری برای دریافت داده‌های ذخیره‌شده بر روی دیتابیس بنویسید به نحوی که کاربر با تعیین جفت‌ارز موردنظر بتواند داده‌های مرتبط را دریافت کند. کوئری‌ها باید از آدرس localhost:4000/graphql قابل اجرا باشند.

### پروژک دوم:

دو collection برای signals و decisions در MongoDB داریم.

هر سیگنال دارای signal\_number (شماره‌ی سیگنال) و status (وضعیت فعلی سیگنال) است.

در سیگنال، status می‌تواند یکی از موارد open، target و یا stop باشد.

```
Signal {
  signal_number
  status
}
```

تصمیم یا decision دارای analyst (یوزرنیم فردی که در مورد سیگنال تصمیم‌گیری کرده است)، signal (سیگنالی که در مورد آن تصمیم گرفته شده است) و analyst\_decision (تصمیم آنالیست در مورد سیگنال که می‌تواند confirm و یا reject باشد) است.

```
Decision {
  analyst
  signal
  analyst_decision
}
```

تصمیمی درست است که تارگت و یا استاپ شدن سیگنال را پیش‌بینی کرده باشد.

میخواهیم رتبه‌ی آنالیزها را بر اساس درصد تصمیم درست (نسبت تصمیمات درست به کل تصمیم‌ها) به دست آوریم. توجه کنید که در صورتیکه سیگنال open باشد نمی‌توان تصمیم آنالیزها را درست یا غلط تلقی کرد.

تابعی بنویسید که اطلاعات decision ها و signal ها را از دیتابیس استخراج کند، سپس لیست مرتب‌شده‌ای از { آنالیزها: تعداد تصمیمات درست، تعداد کل تصمیمات } را برگرداند.

### **پروژک سوم:**

- (۱) برای پروژک دوم، user story ها، acceptance criteria ها و user acceptance test ها را بنویسید.
- (۲) یونیت تست‌ها را نوشته و در فولدر tests در کنار src قرار دهید.

**کد خود را برای پروژک‌های اول تا سوم این بخش بر روی یک ریپازیتوری در گیت‌هاب قرار دهید. سعی کنید clean coding را رعایت کرده و از standardjs در جاوااسکریپت و PEP8 در پایتون استفاده کنید.**

### **پروژک چهارم (بخش امتیازی):**

در مورد کپی‌تریدینگ (Copy Trading) مطالعه کنید. سیستمی را طراحی کنید که در آن چند Master و چند Slave (یا Subscriber) وجود داشته باشند. اسلیوها می‌توانند به یک یا چند مستر سابسکرایب کنند و مدیریت بخشی از دارایی خود را در اختیار یک مستر بگذارند. همچنین مسترها می‌توانند به یک یا چند مستر دیگر سابسکرایب کنند (اسلیو مستر دیگر شوند).

- (۱) برای این سیستم Class Diagram و Domain Model را رسم کنید.
- (۲) طراحی دیتابیس به چه صورت خواهد بود؟

طراحی خود را در نرم‌افزاری مانند draw.io رسم کنید.

موفق باشید (:)