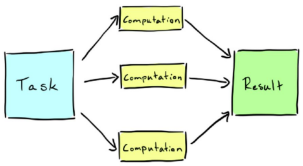
تمرین کامپیوتری سوم

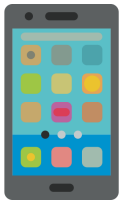
دانشکده مهندسی برق و کامپیوتر	**طراحی چندریسه‌ای**	سیستم‌های عامل - پاییز ۹۹
استاد:	مهلت تحویل:	مسئولان تمرین:
دکتر مهدی کارگهی	**ساعت ۲۳:۵۹ روز ۲۴ آذرماه ۱۳۹۹**	**محمّد مریدی، مبینا شاه‌بنده و غزل مینایی**

مقدمه



هدف از این تمرین آشنایی شما با مفاهیم اولیه **طراحی چندریسه‌ای**[1] یک مسئله است. در این تمرین شما به تحلیل داده‌هایی که از مشخصات و قیمت فروش گوشی‌های موبایل جمع‌آوری شده‌است می‌پردازید. پیشنهاد می‌شود در ابتدا به مطالعه پرونده[2] مربوط به **پیش‌زمینه** که در کنار این پرونده بارگذاری شده‌است پرداخته و سپس شرح تمرین را مطالعه فرمایید.

شرح تمرین



در این تمرین شما به تحلیل داده‌هایی که از مشخصات و قیمت فروش گوشی‌های موبایل جمع‌آوری شده‌است می‌پردازید. در ابتدا برنامه شما اقدام به خواندن و تجزیه مجموعه داده[3]ی ارائه شده می‌کند و آنها را در حافظه خود ذخیره می‌کند. پس از استخراج داده‌ها و ویژگی‌های آنها، برنامه اقدام به نرمال‌سازی[4] داده‌ها و در نهایت اقدام به تعیین طبقه قیمتی گوشی‌ها می‌کند. در این تمرین شما به دو روش این مسئله را پیاده‌سازی می‌کنید. همچنین در قالب گزارش کاری که در کنار این پرونده بارگذاری شده‌است، به مقایسه این روش‌ها و بررسی پیاده‌سازی‌های انجام‌شده می‌پردازید.

---

[1] Muti-Threaded Design
[2] File
[3] Dataset
[4] Data Normalization

استخراج ویژگی‌ها

در مجموعه داده تهیه دیده‌شده ستون price_range ستون هدف[5] و سایر ستون‌ها، معرف ویژگی‌های هر نمونه هستند. در ابتدا به استخراج ویژگی‌های مربوط به هر نمونه کرده و آن را در حافظه برنامه ذخیره کنید.

نرمال‌سازی داده‌ها

برای نرمال‌سازی داده‌ها لازم است که برای هر ویژگی مقدار کمینه[6] و بیشینه[7] هر ستون بدست آورده شود. در این قسمت برای تمام نمونه‌های موجود در مجموعه داده، با استفاده از رابطه‌ی زیر برای هر ستون عمل نرمال‌سازی داده‌ها را انجام می‌دهید و در ادامه تمرین از داده‌های نرمال‌شده استفاده می‌کنید.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

برای مثال فرض کنید برای یک نمونه مقدار موجود در ستون clock_speed آن برابر با 1.5 است. همچنین فرض شود که کمینه و بیشینه تمام نمونه‌ها برای این ستون به ترتیب برابر با 0.5 و مقدار 3 هستند. به این ترتیب مقدار نرمال‌شده این ستون برای این نمونه خاص به صورت زیر محاسبه می‌گردد:

$$x_{new} = \frac{1.5 - 0.5}{3 - 0.5} = 0.4$$

تعیین طبقه قیمتی گوشی‌ها

مدلی برای تعیین طبقه قیمتی گوشی‌ها بر اساس همین مجموعه داده از قبل آموزش دیده است و بردارهای وزن این مدل در اختیار شما قرار داده شده‌است. در این قسمت برای هر برای هر نمونه (سطر)، حاصل رابطه زیر را برای چهار طبقه مختلف قیمت بدست آورده و پس از پایان محاسبات برای هر نمونه، طبقه قیمتی را که بیشترین امتیاز برای آن حاصل شده است به عنوان پاسخ آن نمونه ذخیره می‌کنید.

$$y_c = W_c x + \beta_c$$

[5] Target: ستونی که قصد طبقه‌بندی آن وجود دارد

[6] Minimum

[7] Maximum

در این رابطه، W_c بردار وزن یکی از طبقه‌های قیمتی در بردارهای وزن که در اختیار شما قرار داده شده‌است، β_c مقدار $Bias$ آن، x هر نمونه و c طبقه قیمتی مورد بررسی است. همچنین در این رابطه ضرب میان بردار وزن و x از نوع ضرب داخلی[8] است.

محاسبه دقت

برای هر نمونه، طبقه قیمتی تعیین شده در قسمت قبل را با مقدار واقعی آن که در ستون price_range قرار دارد مقایسه می‌کنید. پس از اتمام مقایسه برای تمام نمونه‌ها، دقت مدل از رابطه زیر به دست آورید:

$$Accuracy = \frac{number\ of\ samples\ classified\ correctly}{number\ of\ all\ samples}$$

در انتها باید مقدار این دقت را تا **دو رقم اعشار** گزارش کنید.

پیاده‌سازی سری[9]

در این بخش از تمرین شما به پیاده‌سازی سری برنامه خواسته شده می‌پردازید. سعی کنید در این بخش از تمرین **بهترین پیاده‌سازی** که می‌توانید را از لحاظ **زمان اجرا** انجام دهید. پس از انجام این بخش از تمرین به کامل کردن بخش مربوطه در **گزارش کار** اقدام کنید.

پیاده‌سازی چندریسه‌ای

در این بخش از تمرین به موازی‌سازی اعمال صورت گرفته در توابعی که در بخش قبل به عنوان Hotspot[10] از آن‌ها یاد کردید می‌پردازید. تجزیه کردن پرونده‌ها و ذخیره‌سازی آنها در حافظه از اعمال زمان‌گیر در بسیاری از برنامه‌هاست که احتمالا از توابع مربوط به آنها (درکنار سایر توابع) به عنوان Hotspot های برنامه یاد کرده‌اید. برای موازی‌سازی این بخش می‌توانید مجموعه داده‌ای را که در اختیارتان قرار گرفته است به پرونده‌هایی کوچکتر تقسیم کرده و اعمال مربوطه را توسط چندین ریسه انجام دهید.

- **دقت کنید** که مجاز به تغییر ساختار مجموعه داده، از قبیل تغییر ستون‌های مربوطه **نیستید** و تنها می‌توانید نمونه‌های موجود در هر پرونده را بین پرونده‌های کوچکتر تقسیم کنید.
- **دقت شود** که در صورتی که تعداد ریسه‌هایی که برای این منظور در نظر می‌گیرید برابر با n باشد، اسامی مجموعه‌های داده کوچک شده باید بصورت (train_[0, n - 1].csv) باشد و استخراج هر پرونده را توسط یک ریسه انجام دهید.
  - برای مثال ۲ ریسه، این مجموعه‌ها بصورت train_0.csv و train_1.csv هستند.

[8] Dot Product

[9] Serial

[10] توابعی که در برنامه‌تان بیشترین زمان اجراها را به خود اختصاص می‌دهند.

- همچنین **دقت شود** که در برنامه خود تصور کنید که در هر پرونده شکسته شده، سطر اول نام مربوط به ستون‌ها آورده شده است.

- **توجه شود** که این بخش از تمرین باید به صورت **چندریسه‌ای** پیاده‌سازی گردد و سایر پیاده‌سازی‌ها قابل قبول **نیست**. پس از انجام این بخش از تمرین به کامل کردن بخش مربوطه در **گزارش کار** اقدام کنید.

ورودی و خروجی برنامه

نمونه اجرای برنامه با فرض اینکه پوشه مربوط به مجموعه‌های داده در کنار پرونده اجرایی شما قرار گرفته است در زیر آمده است:

نمونه اجرا
./PhonePricePrediction.out datasets/

قالب و نمونه خروجی این اجرای برنامه در زیر آمده است:

قالب خروجی
Accuracy: <accuracy_percentage>%

نمونه خروجی
Accuracy: 93.05%

نکات تکمیلی

- تمام خروجی‌های برنامه را در جریان خروجی استاندارد[11] چاپ کنید.
- **تضمین می‌شود** که ورودی‌هایی که به برنامه شما داده می‌شود صحیح هستند و نیازی به بررسی صحت ورودی توسط برنامه شما نیست.
- طراحی درست، کارایی[12] برنامه و شکستن برنامه به بخش‌های کوچکتر تأثیر زیادی در نمره‌ی تمرین دارد.
- **دقت شود** برای موازی‌سازی پروژه تنها مجاز به استفاده از کتاب‌خانه **PThread** هستید.

[11] Standard Output Stream

[12] Performance

نحوه‌ی تحویل

- **دقت کنید** که فایل آپلودی شما با نام OS_CA3_<SID1>[_SID2_SID3].zip حتما باید شامل **دو پوشه**[13] مجزا باشد که در یک پوشه پیاده‌سازی سری و در پوشه دیگر پیاده‌سازی موازی آورده شده است. در کنار پوشه‌ها گزارش کار خود را در قالب pdf قرار دهید.
    - برای مثال، نمونه فایل مورد قبول در زیر آمده است:

```
OS_CA3_810198999.zip
├── parallel
│   ├── main.cpp
│   └── makefile
├── report.pdf
└── serial
    ├── main.cpp
    └── makefile
```

- با توجه به مشکلات پیش آمده برای استفاده از GitLab توسط کاربران ایرانی، امکان استفاده از این سایت برای آپلود پروژه وجود ندارد. اما می‌توانید از سایت bitbucket به جای آن استفاده کنید و مانند GitLab پروژه خود را پیش ببرید. همچنین باید اکانت os_ta_fall2020@yahoo.com را به عنوان یکی از اعضا به مخزن خود اضافه کنید. در انتها نیز در محل متن آنلاین، آدرس مخزن و شناسه آخرین commit خود را آپلود کنید.
- **همچنین** در قسمت متن آنلاین محل تحویل، تعداد ریسه‌هایی که برای انجام عملیات‌های مربوط به خواندن فایل‌ها استفاده کردید را مشخص کنید.
- برنامهٔ شما باید در سیستم‌عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- **دقت کنید** که پروژه شما باید دارای Makefile باشد. همچنین در Makefile خود مشخص کنید که از استاندارد c++11 استفاده می‌کنید.
- نام فایل اجرایی شما که در کنار Makefile خود ساخته می‌شود باید PhonePricePrediction.out باشد.
- نیازی به قرار دادن فایل‌های مجموعه داده و وزن‌ها نیست، زیرا کدهای شما با مجموعه داده‌ی متفاوتی تست می‌شوند.
- **دقت کنید** که برنامه شما توسط آزمون‌های خودکار سنجیده می‌شود. به همین منظور به قالب‌های ذکر شده برای مجموعه‌های داده و فایل آپلودی دقت کافی را داشته باشید.

[13] Directory

- نکته‌هایی که در جلسه توجیهی تمرین گفته می‌شود و یا در فروم‌های مربوطه مطرح می‌شود بخشی از تمرین هستند؛ بنابراین به آن‌ها توجه داشته باشید.
- **توجه شود** پروژه باید برای دانشجویان *مهندسی کامپیوتر* به صورت **انفرادی** و برای دانشجویان *غیر مهندسی کامپیوتر* در قالب **گروه سه نفره** انجام شود.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.