



دنیای صنعت

نشریه تخصصی انجمن علمی مهندسی کامپیوتر

دانشکدگان فارابی دانشگاه تهران

شماره پانزدهم

آبان ۱۴۰۲

در این شماره خواهید خواند

- شما در برابر امنیت اطلاعات خود مسئول نیستید
- گپ و گفتی دوستانه با دکتر کاظم فولادی قلعه
- هوش مصنوعی و گرافیک: ادوبی فایرفلای
- دانشگاه قبول شدم حالا چیکار کنم
- آشنایی با regex در پایتون
- پدرِ مادرِ زبان ها



شناسنامه

دانشگاه صادر کننده مجوز:

دانشگاه تهران

زمینه انتشار:

علمی، تخصصی

تاریخ انتشار:

۱۴۰۲/۰۸/۳۰

نوع انتشار:

دو ماهنامه

صاحب امتیاز:

انجمن علمی مهندسی کامپیوتر

دانشکدگان فارابی دانشگاه تهران

مدیر مسئول:

سید علی فقیه موسوی

سردبیر:

سید محمد جعفری

گرافیکست، صفحه آرا و طراح جلد:

سید علی فقیه موسوی

ویراستار ادبی:

محمد امین عارفی نیا

کارشناس نشریات:

مریم شریفی دانا

هیئت تحریریه:

سید محمد جعفری - سید علی فقیه موسوی -

علی رادمرد - سیده فرگل ناظم زاده - سعید رزاقی

ایمیل: cesadonyayesefroyek@gmail.com

تلگرام: @cesa_pr ایتا: @donya_sefroyek

دنیای
صفحه

سخن آغازین

به نام خداوند قلم

جهان صفر و یک، جهان بودن یا نبودن!

به دنیای صفر و یک خوش آمدید! در دنیای صفر و یک کوچکترین و کمترین چیزها هم به وسیله صفرها و یکها ساخته می‌شوند. به نوعی بودن یا نبودن‌ها قرار است که ما را به سمت اهدافمان پیش ببرد. در دنیای صفر و یک اما امروز بودن‌ها از نبودن‌ها پیشی گرفته، تعداد یک‌ها رفته رفته بیشتر شده و این به معنی آن است که دنیای صفر و یک بار دیگر به جریان افتاده.

قطار دنیای صفر و یک اکنون به ایستگاه پانزدهم رسیده و این مهم را مرهون تلاش و زحمات کسانی است که هدایت آن را به دست داشتند. اولین شماره از نشریه دنیای صفر و یک در دومین دوره از انجمن علمی مهندسی کامپیوتر دانشکدگان فارابی و به همت انجمن وقت و گروهی از دانشجویان علاقه مند منتشر شد. پس از سپری شدن دوره دوم نشریه، نوبت به دوره سوم رسیده که دوره تحول در نشریه بود. ایده های نو و پشتکار تیم نشریه قبل دست به دست هم داد تا ساختار و شکل کلی نشریه دستخوش تغییراتی شود که در نهایت موجب رضایت مخاطبان و اساتید بود. تجربه دوره سوم و شنیدن ضعفها و قوتها و ارتباط بیشتر با ورودی‌های جدید رشته به کمک ما آمدند تا دوره چهارم را هر چه بهتر از دوره های قبل شروع کنیم. امید است که از پس این مسئولیت مهم به خوبی بر بیاییم.

از تمامی دوستان و عزیزانی که در دوره‌های قبل خصوصا دوره سوم نشریه زحمت کشیدند به خصوص محمد آزاد که به نوعی احیاگر نشریه بود، کمال تشکر و قدردانی رو دارم و از خدای منان برای ایشان طلب سلامتی و بهروزی دارم.

چهارمین دوره نشریه دنیای صفرویک از همان ابتدا درگیر اتفاقات غیر منتظره‌ای شد که روند کار را کمی کند می‌کرد. ولی رفته رفته به لطف خدا و با همت بچه‌های انجمن و سردبیری سید محمد جعفری موتور چهارمین دوره از نشریه دنیای صفرویک در دل هفتمین دوره از انجمن علمی دانشجویی مهندسی کامپیوتر دانشکدگان فارابی دانشگاه تهران، روشن شد. این نشریه از نظر ساختار کلی شباهت‌های زیادی به نشریه های دوره سوم دارد اما در جزئیات تغییراتی داشته که آن را نسبت به شماره های قبل خواندنی تر میکند.

در پایان از تمامی کسانی که در طول انتشار این شماره از نشریه همراه ما بودند، علی رادمرد، محمدمین عارفی‌نیا، سیده فرگل ناظم زاده، سعید رزاقی و محمدرضا آرمانیپور تشکر میکنم و امیدوارم که در آینده هم ایشان را در کنار نشریه دنیای صفر و یک ببینیم.

ارادتمند شما

سید علی فقیه موسوی

فهرست

يك كلام حرف حساب

۱ گپ و گفتی دوستانه با دکتر کاظم فولادی

روزد روزگار رایانه

۴ پدرِ مادرِ زبان ها

بخش ویژه

۶ دانشگاه قبول شدم حالا چکار کنم؟

تخصص

۱۰ آشنایی با Regex در پایتون

۱۲ الگوهای طراحی یا Design Patterns

۱۴ هوش مصنوعی و گرافیک : ادوبی فایرفلای

۱۶ شما در برابر امنیت اطلاعات خود مسئول نیستید!

۱۸ تکنیک Score Matching در هوش مصنوعی

۲۰ معرفی نرم افزار: نرم افزار Notion

کلاکت

۲۲ Black Mirror: White Christmas

دیگه چه خبر

۲۴ مروری بر اخبار تکنولوژی

مسابقه

۲۶ جدول کلمات متقاطع، اختلاف بین تصاویر



نویسنده: سعید رزاقی
مصاحبه: محمد رضا آرمانیپور

گفتگوی با دکتر کاظم فولادی قلعه

ریاست محترم دانشکده مهندسی
دانشکده‌گان فارابی دانشگاه تهران

از خودتان برایمان بگویید. در چه دانشگاه‌هایی تحصیل کرده‌اید؟ گرایش‌های شما در مقاطع کارشناسی، ارشد و دکتری چه بودند؟

من متولد سال ۱۳۶۰ در تهران هستم. برای دوره کارشناسی در رشته مهندسی کامپیوتر گرایش نرم‌افزار در دانشگاه یزد قبول شدم و لیسانسم را در سال ۱۳۷۹ تا ۱۳۸۳ در دانشگاه یزد گرفتم.

همان سال ۸۳ هم فوق لیسانس مهندسی کامپیوتر گرایش هوش ماشین و رباتیک در دانشکده فنی دانشگاه تهران قبول شدم. تهران آمدم و فوق لیسانس را در دو سال خواندم. سال ۸۵ هم دکترای همان دانشگاه تهران و همین رشته قبول شدم و تا سال ۱۳۹۲ ادامه دادم تا مقطع دکترای فارغ التحصیل شدم.

دوره دانشجویی همواره همراه با چالش‌ها و بالا و پایین‌های بسیار است. درباره چالش‌ها و موانعی که در دوران دانشجویی بر سر راه شما قرار داشتند بفرمایید.

یکی از مشکلات ما این بود که منابع عموماً همه چاپی بودند و پی‌دی‌اف آن‌ها به این راحتی گیر نمی‌آمد. اگر شما پی‌دی‌اف پیدایم کردید مثل گنج بود. تقریباً سه سال اول هیچ کتاب الکترونیکی پیدا نمی‌شد. همچنین پیدا کردن کتاب زبان اصلی مثل یک معجزه بود و بعضی از کتاب‌ها در کتابخانه هم پیدا نمی‌شدند و اگر یک نفر پیدا می‌کرد همه می‌گرفتند و از آن کپی می‌کردند. هنوز هم من تعداد زیادی کتاب زبان اصلی دارم که کپی هستند.

در حال حاضر در کدام حوزه تحقیقاتی و کاری مشغول فعالیت هستید؟ دلیل انتخاب این حوزه چه بوده است؟

الان حوزه تحقیقاتی اصلی من سایبرنتیک است. و هر چیزی که به سایبرنتیک مربوط شود به طور خاص هوش مصنوعی، که رشته تحصیلی من هم بوده و بقیه زمینه‌هایی که به علم کامپیوتر از این حیث مربوط است.

من یک مقدار تنوع کاریم زیاد است ولی جهت مند است و همه این‌ها در یک پلن با هم مرتبط است. یک مقدار به خاطر این است که خود سایبرنتیک تنوع موضوعی دارد. از نظر روشی هم، حجم زیادی از روش‌ها در آن قابل استفاده است. برای همین دستم روی موضوعات پژوهشی که کار می‌کنم خیلی باز است و بیشتر هم تا الان انتخاب موضوعات را بر اساس نیازها انجام دادم، یعنی متوجه شدم که چیزهایی کاربردی دارد. این قاعده که، این کاری که دارد انجام می‌شود

حتماً مشکلی یا گرفتاری را از کسی یا مجموعه‌ای یا از کشور رفع بکنند نیز برای من مهم بوده است.

یعنی یکی از دلایل انتخاب سایبرنتیک این بوده که برای کشور یک کاری کرده باشید؟

اصلاً برای مردم جهان و برای کشور و برای هر چیزی که فکر کنید. مطمئنم یکی از نیازهای جدی‌ای است که دیگران هم کمتر به آن می‌پردازند. به این علت انتخابش کردم.

در همین راستا هم درباره آزمایشگاه پژوهشی فضای سایبر هم یک توضیحی می‌دهید که در چه مسیری هستید و چه کارهایی انجام می‌دهید؟

آزمایشگاه فضای سایبر اولین آزمایشگاه این دانشکده و مجموعه کل فارابی بود که دقیقاً با نگاه نیاز محوری تاسیس شد. ما یک پایگاهی به عنوان نهاد پژوهشی که این فعالیت‌های علمی ذیل آن انجام بشود نیاز داشتیم. از نظر ساختاری آزمایشگاه پاسخگوی اینجا بود. ما دقیقاً با همین نیت پیشنهاد آن را دادیم و پس از تصویب، تاسیس شد.

آزمایشگاه در حال حاضر تقریباً در تمام زمینه‌های پایه و بنیادین مربوط به دانش سایبرنتیک و کاربردهای آن که عموماً در فضای سایبر دیده می‌شود فعالیت انجام می‌دهد. الان در حال کار روی پرونده‌ها و پروژه‌های پژوهشی هستیم. به طور خاص برگزاری رویدادهای تخصصی حوزه فضای سایبر نیز از کارهای ویژه

آزمایشگاه است که مهم ترین شان کنفرانس ملی فضای سایبر است که هر ساله برگزار می شود.

مبحث یادگیری عمیق هم جزو جدیدترین مباحث هوش مصنوعی است. ما سال ۹۶ تقریباً همزمان با آزمایشگاه فضای سایبر آزمایشگاه یادگیری عمیق را هم تأسیس کردیم. جنس فعالیت در آزمایشگاه فضای سایبر و یادگیری عمیق باهم متفاوت است. بحث این آزمایشگاه بیشتر تکنولوژی محور است. برای نمونه: تولید محصولات نرم افزاری در حوزه هوش مصنوعی که عموماً از تکنیک های یادگیری عمیق استفاده می کنند.

در آستانه ۱۰ سالگی دانشکده مهندسی دانشکده گان فارابی هستیم. با توجه به اینکه چندین سال از حضور شما در این دانشکده می گذرد عملکرد دانشکده رو چطور ارزیابی می کنید. به عنوان ریاست دانشکده چه آینده ای رو برای دانشکده مهندسی می بینید؟

سوال جالبی است و به اندازه خود این دانشکده فراز و نشیب دارد. این جا هم در این ده سالی که گذشته فضای آرامی نبوده به طوری که تقریباً دوبار دانشکده تا مرز انحلال پیش رفته و خیلی هم جدی بوده. حالا باید یک روزی درباره تاریخ و تاریخچه اش بنشینیم و صحبت کنیم. خودش یک قصه مفصل است.

اگر بخواهم اجمالاً بگویم در همین فراز و نشیب ها و این اتفاقات به تعبیری نگران کننده، هم برای دانشجوی و هم برای اساتید، یک پیشرفت قابل توجه و یک جریان رو به رشد همیشه دیده می شد. از کیفیت دانشگاه گرفته تا کیفیت دانشجویان و کارهای پژوهشی، از جمله پایان نامه ها، پروژه ها، مقالات، کتاب ها و...

برآورد من این است که رو به رشد بوده با ارقام و اعداد می شود این را اثبات کرد. برای مثال: کیفیت فارغ التحصیل ها مهم است چرا که شما بهترین دانشجو ها را هم بگیرید مهم آن است که وقتی آن ها فارغ التحصیل می شوند در چه سطحی هستند، و من فکر می کنم آمار بیکاری در فارغ التحصیلان ما نزدیک به صفر است. این از شاخصه هایی هست که نشان می دهد دانشکده رو به پیشرفت بوده است. البته در این ده سال هم پیشرفت هایی داشته ایم. برگزاری رویداد های ملی و بین المللی بخصوص در این چهار سال گذشته، می توان گفت تقریباً جایگاه دانشکده را در کل کشور خیلی تغییر داده است. ما همیشه گفتیم دانشکده ما ابعادش خیلی بزرگ نیست. از نظر تعداد هیئت علمی و اساتید در کل دانشکده های دانشگاه تهران، شاید ماز نظر ابعاد کوچک ترین و یا عضو کوچک ترین ها باشیم. ولی میزان اثر گذاری و فعالیت ها خیلی بالاتر از اندازه دانشکده ماست. هر چند که دور بودن از مرکز خیلی از مزایا را از ما می گیرد و ما نمی توانیم آن ها را به راحتی داشته باشیم.

حالا اینکه می گوید در آینده چه خواهد شد، پیش بینی من شرطی است. نمی خواهم یک چیز قطعی بگویم. اگر روندی که تا الان طی شده قدر دانسته شود و مجموعه اعضای دانشکده که شامل اساتید و دانشجویان هستند، همه را عرض میکنم، این شرایط را قدر بدانند و با این پشتوانه جلو بروند، دهه دوم دانشکده یعنی از الان تا سال ۱۴۱۲ انشاء الله یک دهه کاملاً درخشان در تاریخ دانشکده خواهد شد. چون تقریباً زیر ساخت های اصلی فراهم شده و برنسازی هم صورت گرفته. هر چند نیاز هایی هم وجود دارند، اما زیر ساخت های به خصوص نرم افزاری تهیه شده است. در فضای علمی مهم تر از سخت افزار، نرم افزار است. شما باید آن بنیان های فکری را آماده کرده باشید که این ها وجود دارد. حتی اگر ما رشته های دیگری هم اضافه نکنیم و با همین رشته های موجود جلو برویم، در فضای پژوهش و ارتباط با صنعت اتفاق های خوبی را می توانیم رقم بزنیم. ولی لازمه آن این است که خودمان را باور کنیم و شدت فعالیت ها را افزایش بدهیم و با جدیت جلو برویم. ولی اگر به خاطر فضای یأس و ناامیدی که بیرون از دانشگاه و در خیلی از جاها شکل گرفته که تابع مسائل اقتصادی هم است، به خاطر این مشکلات بگوییم ما هم دیگر نمی خواهیم کار کنیم و بی حوصله شویم، احتمالاً این دهه، دهه بدتری از دهه قبل خواهد شد و این باعث می شود که خیلی از کارهایی که قبلاً می توانستیم انجام بدهیم را دیگر نتوانیم انجام بدهیم.



باتوجه به اینکه شما تا مقاطع بالای دانشگاهی تحصیل کرده اید و مشغول به فعالیت آکادمیک هستید، به چه افرادی پیشنهاد می کنید که تحصیلات خود را تا مقاطع ارشد و دکتری ادامه دهند؟ این ادامه تحصیل، چه تأثیری در زندگی و آینده شغلی آنها ایجاد می کند؟

سوال خیلی خوبی است چون من حداقل جزو کسانی هستم که خیلی توصیه به ادامه تحصیل نمی کنم مگر برای کسانی که می دانم به یک انگیزه خاصی می خواهند درس بخوانند. یک بخشی از بخش های آموزش، مثلاً تا دیپلم عمومی شده است. یعنی دیگه هر کسی در این دنیای مدرن بخواد زندگی کند باید یک دیپلم گرفته باشد.

بچه هایی که می خواهند ادامه بدهند با هدف اینکه مثلاً در آینده کار علمی بکنند، نمیگم لزوماً هیئت علمی شدن، ادامه دادنشان فقط برای مدرک نیست می خواهند یک چیزی یاد بگیرند و در فضای فکری باشند و مطالعه کنند و به این کار ها علاقه دارند. این ها باید ادامه تحصیل بدهند یعنی اگر این ها زود جذب بازار کار روتین بشوند بعضی از استعداد هایشان بروز نمی کند و ممکن است بعداً پشیمان شوند. ولی بعضی ها دقیقاً برعکس هستند و از ابتدا حتی راهشان را مشخص کردند و می دانند می خواهند به بازار کار بروند و یک تجربه عملی کسب کنند. البته همانطور که عرض کردم تحصیلات عمومی چاره ای نیست الان تحصیلات عمومی تا لیسانس هست. یعنی اگر الان کسی کمتر از لیسانس مدرک داشته باشد در فضای اجتماعی یک مقداری نقص تلقی می شود که آن هم چیز درستی نیست ولی متأسفانه مانند خیلی چیز های دیگر که درست نیست و جا افتاده این هم درست نیست. ولی واقعاً تحصیلات تکمیلی را فقط باید به کسانی توصیه کرد که انگیزه های جدی علمی دارند. یا در خودشان نبوغی می بینند که می خواهند نبوغ را ظاهر کنند و یا خیلی کنجکاو هستند و سؤالاتی دارند که حتماً باید در فضای علمی به آن پاسخ داده شود، بنابراین به آن باید مانند یک شغل نگاه کرد، شغل کار علمی که نیاز به تحصیلات تکمیلی دارد.

انگیزه را در کنار استعداد قرار می دهید و یا انگیزه داشتن کافی است؟ چون بعضی ها هستند که از همان اول درشان خوب است و حتی در دانشگاه هم درس برایشان خیلی مهم است و به کار خیلی اهمیت نمی دهند و هستند کسانی که تا سال چهارم به سراغ کار نمیروند، این ها خوب طبیعتاً استعداد بیشتری دارند. اما خب یکسری از بچه ها هستند که ممکن است استعداد علمی نداشته باشند ولی انگیزه علمی را داشته باشند. چه توصیه هایی برای آنها دارید؟

اول روشن کنیم ببینیم آن انگیزه بیشتر برای مدرک است یا آن موضوع علمی؟ چون خیلی وقت ها انگیزه مدرک است. مدرک هم به هر حال جذابیت خاص خودش را دارد. خیلی ها حتی در تحصیلات تکمیلی، با انگیزه مدرک جلو می آیند. اما اگر قصه مدرک را کنار بگذاریم من فکر می کنم انگیزه و استعداد یک ارتباط مستقیم باهم دارند. یعنی هر کس استعداد در زمینه ای دارد انگیزه هم در آن زمینه دارد.

همانطور که گفتید در حال حاضر درصد اندکی از دانشجویان کامپیوتر علاقه به فعالیت آکادمیک دارند و بیشتر دنبال مهارت برای بازار کار هستند این مسئله رو ناشی از چه چیزی می بینید؟

مهندسی شان به درد کار نمی خورد. اینطور متوجه می شوند که مشکل آن ها در آن رشته ها حل می شود. بله این ها ممکن است در یک شرکت و کارخانه جوابگو باشد ولی من نگاهم خیلی وسیع تر است. و نگاهم به جهان است. جهان مشکلاتی دارد که این مشکلات را آدم های علمی باید بتوانند حل کنند. حالا سوال این است که آیا با این رشته ها و ترکیب این رشته ها این سطح از مشکلات قابل حل هستند؟ پاسخ من منفی است. چون این ترکیب رشته ها دید محدود می دهد. دید جامع نمی دهد که شما بتوانید آن پدیده را تحلیل کنید و نسبت به آن اشراف پیدا کنید. من علم دیگری غیر از سایبرنتیک را نمی شناسم، اینکه نمی شناسم یعنی دانشمندان باید در آن رشته همچنین ادعایی کرده باشند و بگویند با دانش ما می شود یک کارهایی کرد. هیچ دانشی همچنین ادعای کلانی ندارد. اما از ابتدا دانشمندان سایبرنتیک حرف های خیلی بزرگی زده اند و ادعاهای خیلی بزرگی کرده اند از جمله اینکه بخواهند کل دنیا را تحت سلطه خودشان بگیرند. این علم همچنین ادعایی دارد پس با این ظرفیت ممکن است همچنین کارهایی انجام داد.

اگر بخواهید به دانشجویان جوان یک توصیه بکنید چه چیزی می فرمایید؟

من یک جمله ای را خیلی دوست دارم بگویم و بعضی وقت ها به عنوان حرف آخر به بچه ها می زنم این است که بچه ها خودتان را ارزان نفروشید، شما خیلی گران هستید. در حقیقت یک طوری عمل کنید که هر وقت خواستید پرونده خودتان را بررسی کنید، بین خودتان و خدای خودتان احساس کنید که آن عملکرد، عملکرد درستی بوده است.

اتفاقا در حوزه های تخصصی ما که جامعه خیلی به آن نیازمند است، بعضی وقت ها این نیازها، نیاز مردم نیست، نیازهای یک طبقه خاصی هست که می شود گفت در کانون های ثروت و قدرت قرار دارند. ما آن آدمی نشویم که نیازهای این کانون های قدرت و ثروت را پاسخ بدهیم و بعدا که حساب خود را انجام می دهیم، ببینیم که عمرمان را مفت فروخته ایم.

تغییر می کند و یک بعد تغییرات هم این است که دیگر با استانداردهای علوم گذشته مسائل جدید را نمی شود حل کرد. اصلا دسته بندی هایی که الان هست پاسخگوی مسائل جدید نیست. الان وقتی یک مسئله پیش می آید، همزمان که ابعاد مهندسی دارد ابعاد پزشکی، ابعاد محیط زیستی، ابعاد فرهنگی و اجتماعی دارد. تقسیم رشته ها در گذشته بر مبنای این بود که فرض می کردند می شود این ها را از هم جدا کرد و هر کسی در تخصص خودش کار را انجام دهد. اما الان این کار را نمی توان انجام داد. الان یکی از مشکلات بچه های ما هم همین است، که با توانایی مهندسی پیش می آیند و می گویند که ما می خواهیم یک پروژه مانند دیجی کالا یا اسنپ ایجاد کنیم و پروژه موفقی هم شود. بعد که باهم صحبت می کنیم متوجه می شویم که این توانایی لزوما برای کسی که فنی است، نیست.

حالا خیلی ها فکر می کنند که یک آدم مدیریتی باید این ها را انجام بدهد ولی تجربه ما نشان داده است که این رشته های تقسیم بندی شده الان، برای این تیپ مسائل جدید پاسخگو نیستند، یعنی فرد مدیریتی می خواهد یک کاری انجام دهد که حالا باز نواقص فنی دارد. یک رشته ای باید باشد که بتواند این ها را تحت مدیریت خودش در بیاورد و یعنی قطعا ترکیب آن رشته ها هم جواب نمی دهد. برای اینکه هر رشته اصول خاصی و استانداردهایی برای خودش دارد، شما اگر بخواهید این ها را ترکیب کنید روش ها و استانداردهایشان باهم تضاد پیدا می کنند. ولی اگر به سایبرنتیک به عنوان یک شاخه علمی نگاه کنیم از ابتدا تلاشش برای از بین بردن مرزها بوده است. من معتقدم تا زمانی که شما نتوانید در سایبرنتیک از نظر علمی مستقل شوید و متخصص در این زمینه نداشته باشیم، خیلی از مشکلات ما حل نمی شود. دوباره در پرانتز برای تاکید بیشتر بگویم، چون همه فکر می کنند مشکلات الان اقتصادی است پس اقتصاددان ها باید حل کنند ولی هیچ اقتصاددانی حتی در حوزه نظری هم نمی تواند نظریه کاملی ارائه دهد و خودشان هم نگران اند که یک وقت نظراتشان اجرا شود و وضعیت بدتر شود.

بسیاری هستند که لیسانس خود را در رشته کامپیوتر میگیرند و برای ارشد به سراغ رشته مدیریت میروند در برخی از کشورهای دیگر هم این مورد زیاد رخ میدهد.

بله به طور خاص به گرایش ام بی ای در رشته مدیریت روی می آورند. بیشتر مهندس ها این گرایش را خواندند و از گرایش های جذاب مدیریت هم هست.

خودتان برای دانشجویان توصیه می کنید؟

با چه هدفی؟ دو هدف وجود دارد بعضی اوقات از رشته مهندسی فرار می کنند و می خواهند ادامه تحصیل بدهند. اما بعضی ها احساس می کنند که دانش

این یک چیز کاملا طبیعی است. اتفاقا هر چقدر این نرخ بالاتر برود و نسبت کسانی که فضای کاری را ادامه می دهند به کسانی که فضای آکادمیک را دنبال میکنند بیشتر شود ما داریم به واقعیت نزدیک تر می شویم. اگر قرار است کسانی که تحصیلات تکمیلی را ادامه می دهند بعدا دوباره به بازار کار بیایند، بخشی از عمرشان را تلف کرده اند. به این باید فکر کنیم که موفقیت در بازار کار به تعداد سال هایی که در آنجا کار کرده اید و حجم روابطی که تولید شده بستگی دارد. ببینید اینها نکات دقیقی است که اگر به آن نگاه کنیم افزایش نرخ علاقه به فضای کار نسبت به آکادمیک توجیه پذیر است و چیز بدی هم نیست. ولی از یک جهت هم ما باید یک سیستمی داشته باشیم که کسانی هم که علاقه مند به ادامه تحصیل هستند و استعداد آن را هم دارند درست هدایت شوند و راهشان را درست پیدا کنند.

اگر الان به جای یک دانشجوی ورودی جدید مهندسی کامپیوتر در مقطع کارشناسی بودید، از چه امکانات یا فرصت هایی استفاده می کردید؟ همان راه قبلی را ادامه می دادید؟

ممکن است مثلا در انتخاب بعضی از موضوعات در مطالعه، یک تجدید نظر بکنم ولی فکر می کنم به صورت کلی راه درستی را رفته ام. این را خدمتان می گویم که من هوش مصنوعی را از همان سال اول لیسانس انتخاب کردم. وقتی متوجه شدم در ارشد چگونه می توان کامپیوتر را ادامه داد، این قسمت خیلی برایم جذابیت داشت و درس ها و کتاب هایش را مطالعه کردم و واقعا فهمیدم به این رشته علاقه دارم و خیلی جدی دنبالش کردم. واقعا پشیمان نیستم. الان که هوش مصنوعی خیلی مهم شده است خیلی از آدم های قدیمی که مثلاً از ۲۰ یا ۲۵ سال پیش ما را می شناسند می گویند که شما چگونه فهمیدید که هوش مصنوعی در آینده اینقدر مهم می شود؟ ما آن موقع نمی دانستیم که واقعا چه اتفاقی قرار است برای هوش بیفتد ولی خود موضوع اینقدر جذابیت داشت.

شاید اگر آن زمان دانشی که الان به دستش آوردیم را داشتیم و رشته ای به نام سایبرنتیک وجود داشت، من سایبرنتیک را انتخاب می کردم. به نظر من هوش مصنوعی ذیل سایبرنتیک معنا ندارد که خیلی کمتر هم بهش پرداخته شده است.

گرایش اینجینی داریم؟

الان نداریم ولی اگر فضای علمی همچنین رشته ای ایجاد کرده بود، به خیلی از کسانی که می خواستند به دانشگاه بروند، نه فقط کسانی که به فضای مهندسی علاقه دارند، پیشنهاد می کردم که از سایبرنتیک ورود کنند. به خاطر ویژگی ها و ظرفیت هایی که این دانش دارد. ما داریم وارد یک فضایی می شویم که همه چیز



پدر مادر زبان ها

دنيس مك آليستر ريچي (Dennis MacAlistair Ritchie) در ۹ سپتامبر سال ۱۹۴۱ در محله برانکسویل شهر نیویورک به دنيا آمد. پدر او، آليستر ای. ريچي، از دانشمندان آزمایشگاه های بل (Bell Labs) و همچنين یکی از نویسندگان نظريه طراحی

مدارهای سوئیچینگ بود. او در همان دوران کودکی به همراه خانواده به شهر سامیت در ایالت نیوجرسی مهاجرت کرد و دوران دبیرستان خود را در همان شهر گذراند. وی تحصیلات آکادمیک خود را در هاروارد ادامه داد. در سال ۱۹۶۳ موفق به دریافت مدرک کارشناسی در رشته فیزیک شد و در سال ۱۹۶۸ از پایان نامه دکتری خود با موضوع «پیچیدگی محاسباتی و ساختار برنامه» تحت نظارت پاتریک سی فیشر دفاع کرد.



سیده فرگناظرمزاده
دانشجو مهندسی کامپیوتر
دانشکده فاریاب دانشگاه تهران
Fargol.nazemzadeh@ut.ac.ir

ريچي در سال ۱۹۶۷ در مرکز تحقیقاتی علوم رایانه آزمایشگاه های بل که یکی از مشهورترین مراکز نوآوری دیجیتال در آن زمان بود، شروع به کار کرد. در آنجا ريچي روی پروژه Multics کار می کرد و در همان پروژه با کن تامسون (Kenneth Lane Thompson) آشنا شد. Multics در واقع مخفف شده «Multiplexed Information and Computing Service» به معنی «سرویس اطلاعاتی و محاسباتی چندگانه» می باشد، پروژه های که توسط MIT و با همکاری جنرال الکتریک و آزمایشگاه های بل در سال ۱۹۶۴ آغاز شد و یکی از نخستین سیستم عامل های اشتراک زمانی بود.



۴

شماره پانزدهم آبان ۱۴۰۲

Multics پروژه های بلند پروازانه بود که نوآوری های بسیاری را به همراه خود آورد و به همین دلیل کم کم تبدیل به پروژه های بسیار پیچیده شد. سرانجام در سال ۱۹۶۹، آزمایشگاه های بل به همین خاطر از این پروژه کناره گیری کرد اما ریچی و تامسون تصمیم گرفتند با استفاده از تجربیاتی که از این پروژه به دست آورده بودند روی پروژه جدید و کوچک تری کار کنند. تامسون یک مینی کامپیوتر PDP-۷ قدیمی پیدا کرد و سپس با کمک ریچی و افراد دیگر توانست سیستم عامل خودش را بر روی آن بسازد. در سال ۱۹۷۰، با پیشنهاد برایان کرنیهان (Brian Wilson Kernighan) که یکی از همکاران این پروژه بود، نام «Unics» را برای پروژه انتخاب کردند که نوعی بازی با کلمات Multics بود. (Multiplexed Information and Computing Service) و نام آن به «Unix» تغییر یافت.

سیستم عامل یونیکس

سیستم عامل یونیکس در ابتدا یک سیستم تک وظیفه ای بود، یعنی در یک زمان تنها یک برنامه یا فرآیند در آن قابل اجرا بود. ویژگی های چند وظیفه ای، چند کاربر و قابل انتقال بودن، بعدها به یونیکس اضافه شدند. این سیستم به زبان اسمبلی نوشته شده بود و به همین علت قابلیت انتقال نداشت. در آن زمان، ریچی و تامسون به این نتیجه رسیدند که برای تکمیل سیستم به زبان سطح بالاتری نیاز دارند. پس در سال ۱۹۷۰، تامسون با کمک ریچی زبان برنامه نویسی سیستمی «B» را بر مبنای ویژگی های زبان «Basic Combined Programming» (BCPL) Language نوشت و توسعه داد. سرانجام آنها تصمیم گرفتند یونیکس را روی مینی کامپیوتر PDP-۱۱ پیاده کنند که به دنبال این کار، متوجه کاستی ها و نقاط ضعف زبان B در سیستم جدید شدند.

UNIX®

ريچي در سال ۱۹۷۲ شروع به تغییر و بهبود زبان B کرد که در نهایت منجر به ساخت زبان جدیدی به نام «C» شد و در سال ۱۹۷۳، نسخه ۴ یونیکس که به زبان C بازنویسی شده بود، منتشر شد. این بازنویسی انقلابی در زمینه تکنولوژی و فناوری در آن زمان ایجاد کرد، زیرا تصور بر این بود که سیستم های عامل باید فقط با زبان اسمبلی نوشته شوند. تا این زمان برنامه ها و سیستم های عامل کامپیوتری محدودیت های زیادی داشتند و هر کدام تنها روی سخت افزار خاصی اجرا می شدند. بازنویسی یونیکس به زبان C، مسیر را برای قابل انتقال بودن آن هموارتر کرد اما با این حال، هنوز کدهای وابسته به PDP-۱۱ زیادی در آن وجود داشت و برای این کار مناسب نبود. در نهایت اولین انتقال یونیکس به سیستمی دیگر در سال ۱۹۷۷ و در نسخه ۶ آن انجام گرفت.

در سال ۱۹۷۳، ریچی و تامسون در همایشی، یک مقاله در مورد یونیکس ارائه کردند که بسیار مورد توجه قرار گرفت و افراد زیادی، خواهان دریافت

فعالیت های شغلی دیگر او

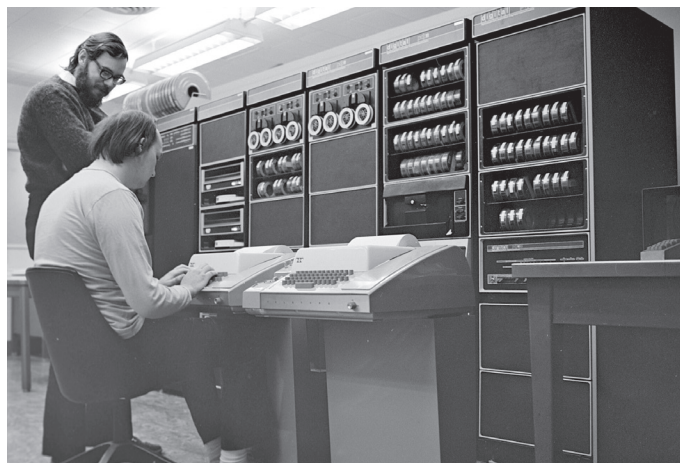
از فعالیت های دیگر ریچی می توان به همکاری او با جیمز ریدز (James Reeds) و رابرت موریس (Robert Morris) در پروژه توسعه سیستم رمزنگاری روی ماشین سایفر M-۲۰۹ آمریکا اشاره کرد. این سیستم توانایی رمزگشایی پیام هایی با حداقل ۲۰۰۰-۲۵۰۰ کلمه را داشت. پس از توسعه این سیستم و مذاکره با آژانس امنیت ملی، تصمیم بر این شد که آن را به دلیل اینکه قابلیت اجرا شدن روی ماشین های رمزگشایی کشور های دیگر را نیز داشت، منتشر نکنند. او همچنین در توسعه سیستم های عامل «Plan ۹» و «Inferno» و زبان «Limbo» همکاری داشته است.

افتخارات

در سال ۱۹۸۳، ریچی و تامسون جایزه تورینگ را برای توسعه نظریه سیستم های عامل عمومی و خصوصا پیاده سازی سیستم عامل یونیکس دریافت کردند. همچنین هر دو آنها به خاطر ایجاد سیستم عامل یونیکس و زبان C در سال ۱۹۹۰ مدال «IEEE Richard W. Hamming» را از مؤسسه مهندسان برق و الکترونیک (IEEE) و در سال ۱۹۹۹ به طور مشترک مدال ملی فناوری سال ۱۹۹۸ را از بیل کلینتون، رئیس جمهور وقت ایالات متحده آمریکا، دریافت کردند و در سال ۱۹۹۷ نامشان در موزه تاریخ کامپیوتر ثبت شد. در سال ۲۰۰۵ مؤسسه تحقیقات صنعتی جایزه «IRI Achievement Award» خود را برای کمک او به علم و فناوری و به طور کلی به جامعه با توسعه سیستم عامل یونیکس، به ریچی اعطا کرد و سرانجام در سال ۲۰۱۱ همراه با تامسون، جایزه «Japan Prize» را به خاطر توسعه سیستم عامل یونیکس دریافت کردند.

سال های آخر

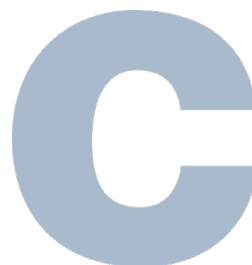
ریچی تمام سال های کاری خود را در آزمایشگاه های بل گذراند و در نهایت در سال ۲۰۰۷، وقتی که به عنوان رئیس بخش تحقیقات نرم افزار سیستم در «Lucent Technologies» مشغول به کار بود، اعلام بازنشستگی کرد. او در سال های آخر زندگی خود درگیر سرطان و بیماری قلبی شده بود و سلامتش در وضعیت نامناسبی قرار داشت. سرانجام دنیس ریچی در ۱۲ اکتبر سال ۲۰۱۱ و در ۷۰ سالگی از دنیا رفت. تلاش ها و دستاوردهای ریچی نقش کلیدی ای در پیشرفت تکنولوژی ایفا کرد به طوری که امروزه می توانیم تاثیر میراث او را در ابعاد مختلف زندگیمان حس کنیم.



یک نسخه از این سیستم عامل شدند. آنها به خاطر محدودیت هایی قادر به ارائه یونیکس به عنوان یک محصول تجاری نبودند پس آن را به صورت یک بسته سورس کد به همراه لایسنس توزیع کردند. دریافت کنندگان سورس کد توانایی دستکاری آن را داشتند و به همین دلیل یونیکس در مجامع دانشگاهی طرفداران زیادی پیدا کرد. این موضوع باعث شد شرکت های بزرگی از این سیستم عامل حمایت کنند تا روند رشد و توسعه آن سریع تر شود.

در دهه ۹۰ سیستم های یونیکس و شبه یونیکس (Unix-like) محبوبیت زیادی کسب کردند و به سیستم عامل منتخب بیش از ۹۰ درصد از ۵۰۰ ابر کامپیوتر برتر جهان تبدیل شدند. شبه یونیکس به سیستم عاملی گفته می شود که رفتاری مشابه رفتار یک سیستم یونیکسی دارد، در حالی که الزامات مطابق با مورد تایید هیچ نسخه ای از استاندارد «Single UNIX Specification» نیست. از شناخته شده ترین های این سیستم های عامل می توان به «Linux» و «BSD» اشاره کرد. امروزه از سیستم های عامل شبه یونیکس به صورت گسترده در سرور های مدرن، ورک استیشن ها (Workstation) و تلفن های همراه استفاده می شود. تجربه ساخت یونیکس و توسعه آن، باعث به وجود آمدن اصولی در برنامه نویسی شد که از آن ها با نام فلسفه یونیکس (Unix Philosophy) یاد می شود. این فلسفه بر ساخت کد های ساده، کوتاه، واضح و شفاف، ماژولار و قابل توسعه تاکید دارد که امکان به روز رسانی و تغییر آن توسط توسعه دهندگانی غیر از سازندگان اصلی آن، فراهم باشد.

زبان C



با عرضه یونیکس به دنیا، زبان C هم شناخته شد و برنامه نویسان به منابعی نیاز داشتند که نحوه استفاده صحیح و مؤثر از این زبان را یاد بگیرند. پس در سال ۱۹۷۸، ریچی و کرنیهان نسخه اول کتاب «زبان برنامه نویسی C»

(The C Programming Language) را منتشر کردند که با نام «K&R» هم شناخته می شود. این کتاب به منبع متعارفی برای زبان C تبدیل شد. در سال ۱۹۸۹ مؤسسه استاندارد ملی آمریکا استاندارد C را با عنوان «Programming Language C» تصویب کرد که این نسخه از زبان C اغلب به عنوان ANSI C، استاندارد C یا بعضی اوقات C۸۹ شناخته می شود همچنین در سال ۱۹۹۰، این استاندارد توسط سازمان بین المللی استاندارد سازی (ISO) تایید شد. استاندارد فعلی زبان C با نام C۱۸ شناخته می شود و در سال ۲۰۱۸ منتشر شده است.

C زبانی همه منظوره، ساخت یافته، رویه ای و همچنان یکی از پرکاربردترین زبان های برنامه نویسی است. امروزه از این زبان به صورت گسترده در توسعه سیستم های عامل، پایگاه داده ها مانند MySQL، سیستم های نهفته (Embedded systems)، طراحی کامپایلرها، توسعه برنامه های مرتبط با گرافیک و... استفاده می شود. بسیاری از زبان های پر کاربرد حال حاضر مانند Python، JavaScript، Java، C#، C++ و... به طور مستقیم یا غیرمستقیم از زبان C وام گرفته اند، به طوری که اغلب آن را مادر همه زبان های برنامه نویسی می دانند.

دانشگاه قبول شدم حالا چیکار کنم؟

همونجوری که از اسم مطلب مشخصه، این رو برای شمایی نوشتیم که تازه وارد دانشگاه شدین، قطعاً الان کلی سوال توی ذهنتون هست از اینکه چه اتفاقاتی قراره توی دانشگاه بیوفته و چیکار قراره کنید و مهم‌تر از همه، تهش قراره چی بشه؟

من خودم دانشجوی کامپیوتر هستم اما خب این مطلب فقط مختص کامپیوتری‌ها نیست، خیلی از مواردی که گفته شدن تقریباً عمومی هستن و به کار همه میان، اما خب چون رشته خودم کامپیوتره بیشتر تمرکز توی این مطلب روی همین رشته هست. اینجا همچنین ممکنه مواردی باشه که به درد ورودی‌های قدیمی‌تر هم بخوره و برای اون‌ها هم مفید باشه. بیشتر میخوام به عنوان یه کسی که خودش درگیر بعضی مسائل بوده، یه سری نکاتی که میدونم برای خیلی‌هاتون سوال هست رو براتون توضیح بدم. نکته مهمی که پیش از شروع باید بهتون بگم اینه که بیشتر مواردی که اینجا میگم نظرات و تجربیات خودم هستن، من هم همین سوالات رو داشتم و چیزی که قراره بخونید، جواب‌هایی هستن که من برای سوالاتم پیدا کردم و تصمیم گرفتم با شما به اشتراک بذارم تا یه مقدار کم و کیف قضیه دستتون بیاد و خودتون بتونید بگردید دنبال جواب‌های مخصوص به خودتون.



عرفان صابری

دانشجو مهندسی کامپیوتر
دانشکده فابری دانشگاه تهران
erfansaberiow@gmail.com

قبل از اینکه ادامه بدیم، اول میخوام قبولی در دانشگاه رو بهتون تبریک بگم؛ گذشتن از سد کنکور کار هر کسی نیست و همینکه تونستید ازش عبور کنید و وارد دانشگاه بشید باید به خودتون افتخار کنید. مسئله مهم رد شدن از این سد بود، حالا هر کس با یه نتیجه‌ای از این مرحله خارج شده، یکی دانشگاه دولتی، یکی دانشگاه آزاد، یکی شهر تهران قبول شده، یکی شهرستان و ... نکته مهمی که باید بدونید اینه که قبول شدن یا نشدن در یک دانشگاه خاص دلیل بر پیشرفت کردن یا شکست خوردن یک شخص نمیشه. معمولاً هیچ سازمان یا شرکتی یه دانشجو رو بخاطر اینکه فلان دانشگاه معروف و خفن درس خونده استخدام نمیکنه و هیچکس هم بخاطر اینکه یک نفر دانشگاه خوبی درس نخونده اون نفر رو رد نمیکنه. قطعاً هر دانشگاهی مزایا و معایب خودش رو داره و قرار نیست چشممون رو به روی این حقیقت ببندیم، اما در عمل همه چیز به شخص شما بستگی داره، نه دانشگاهتون.

اولین سوالی که ممکنه براتون پیش بیاد اینه که توی دانشگاه قراره چه اتفاقی بیوفته؟ جواب: تقریباً همون اتفاقاتی که توی دبیرستان میوفتاد؛ فقط توی مدرسه شما سر کلاس بودید و معلم میومدن بهتون درس میدادن، اینجا توی دانشگاه استاد سر کلاس هستن و شما تشریف میبرید خدمت استاد. عموماً کلاس‌ها به این صورته که استاد یک اسلایدی رو براتون میارن، از روی اون اسلاید براتون میخونن و توضیح میدن و شما باید سعی کنید درس رو یاد بگیرید و اگر زرنک باشید، جزوه هم بنویسید که در طول ترم و همچنین شب امتحان، منبع برای مطالعه داشته باشید. گاهی اوقات هم میتونید استاد رو راضی کنید که اون اسلایدها رو در اختیارتون بذارن که بعداً از روی اون‌ها مطالعه کنید، که خب خیلی وقت‌ها نمیتونه جای جزوه رو بگیره چون اسلایدها معمولاً توضیحات خیلی کامل و جامعی ندارن و اینکه خیلی پیش میاد که استاد، مثال و تمرینی خارج از اون اسلایدها براتون میارن و حل میکنن و شما باید اونا رو یادداشت کنید، چون احتمال زیادی هست که توی امتحان بیان. بعضی اساتید هم لطف میکنن و جزوه آماده بهتون میدن که خب تعدادشون خیلی زیاد نیست و واقعاً باید خوش‌شانس باشید که چنین فرصتی براتون پیش بیاد.

اما گفتیم جزوه، مگه توی دانشگاه کتاب برای خوندن نداریم؟ جواب: کتاب‌های دانشگاهی مثل کتاب‌های دبیرستان نیستن، یه زمانی دبیرستان بودیم و تنها منبع درس هم همون کتابی بود که اول سال دستمون میدادن، اما توی دانشگاه هر استادی برای درس خودش چند تا منبع ممکنه داشته باشه که هر کدوم برای خودشون یه پا دانشگاهن! علاوه بر اون کنار این منابعی که استاد برای درس معرفی میکنن، منابع دیگه‌ای هم وجود دارن که شما اگر علاقه داشتید میتونید دنبالشون برید و بررسیشون کنید. بیاید یه مثال بزنیم که یه مقدار قضیه رو بهتر درک کنید: کتاب ریاضیات گسسته دبیرستان رو یادتونه؟ با مقدمه و شناسنامه و صفحه خالیا و جلد کتاب سر جمع شاید ۱۰۰ صفحه نمیشد، اما حالا توی دانشگاه، درس ریاضیات گسسته (بعضی جاها ساختمان گسسته میگن)، دو تا منبع اصلی معمولاً داره: کتاب ریاضیات گسسته روزن با بیش از ۱۰۰۰ صفحه محتوا و کتاب ریاضیات گسسته گریمالدی با همون حدودای ۱۰۰۰ صفحه. بقیه درسا هم وضعشون اگه بدتر نباشه، بهتر هم نیست. پس قدر اون جزوه‌ای که دارید مینویسید رو بدونید وگرنه شب امتحان یهو میبینید شما موندید و چند تا منبع هر کدوم بالای هزار صفحه! حالا خر بیار و باقالی بار کن.



زبان اصلی رو بذارید جلوتون و کتاب ترجمه شده رو هم بذارید کنارش، از کتاب زبان اصلی یه صفحه رو بخونید و سعی کنید بفهمید چی میخواد بهتون بگه و بعد با کتاب ترجمه شده مقایسه کنید و ببینید درست فهمیدید یا نه. دیدن دوره و فیلم و سریال و انیمیشن و ... هم با زیرنویس انگلیسی شاید خوب باشه، کم کم هم میتونید وابستگیتون به زیرنویس رو کمتر کنید و از یه جایی به بعد دیگه استفاده نکنید ازش. شنیدن پادکست هم میتونه خوب باشه. خیلی ها گوش کردن آهنگ رو پیشنهاد میکنن که خب من خیلی موافق نیستم چون خواننده برای اینکه ریتمش جور بشه گاهی از سر و ته جملات میزنه یا کج و کوله تلفظ میکنه و این میتونه باعث بشه که شما هم اشتباه یاد بگیرید، ولی خب به عنوان تفریح خیلی هم باحاله، آهنگی که گوش میدید رو یه دور متن ترانهش رو هم بخونید (بزنید lyrics فلان آهنگ گوگل براتون میاره). اگر هم هیچ پیش زمینه ای ندارید (حتی در حد دبیرستان) میتونید برید کلاس زبان ثبت نام کنید تا مقدمات رو یاد بگیرید و البته اینم بگم که کلاس فقط برای مبتدی ها نیست، توی هر سطحی هستید اگر تونستید برید خیلی هم خوبه فقط به شرط اینکه کلاس درست حسایی باشه و خودتون هم تلاش کنید برای یادگیری.



از بحث زبان یه کم فاصله بگیریم، برگردیم سراغ همون مسائل دانشگاه. یه سوالی که خیلی ها دارن اینه که همونجوری که توی دبیرستان نمرات خوب میگرفتیم، اینجا هم میتونیم بگیریم؟ جواب کوتاه: خیر. جواب بلند: بستگی داره؛ اولاً که نمره دست استاد هست و اگر استاد سختگیر باشه قطعاً نمره خوب گرفتن سخته. اول از همه بهتون بگم که بلد بودن یک درس شرط کافی برای گرفتن نمره خوب نیست؛ شما باید فعال باشید، تمرین هایی که استاد بهتون میدن رو حل کنید، امتحان رو خوب بدید، پروژه ها رو خوب انجام بدید، سر کلاس مشارکت داشته باشید و اینجور موارد. خیلی از اساتید به حضور و غیاب اهمیت میدن و هر جلسه حضور غیاب میکنن، خیلی ها هم ممکنه حضور غیاب نکنن اما خب شما رو به چهره و اسم بشناسن و بدونن که کلاس براتون اهمیت داشته و شرکت کردید یا اهمیت نداشته و کلاس رو پیچوندید. بعضی اساتید هم واقعاً اهمیتی به این قضیه نمیدن، ما حتی استادی داشتیم که میگفتن خودشون در دوره دانشجویی هیچ کلاسی رو شرکت نمیکردن چون خودشون با مطالعه بیرون از کلاس خیلی بهتر یاد میگرفتن و خب ترجیح

یه مسئله مهمی که برای خیلی ها خصوصاً توی ترم اول پیش میاد اینه که منابعی که استاد معرفی میکنن رو باید بخریم؟ جواب: اگر قراره کلکسیون جمع کنید یا کتابخونه راه بندازید برید بخرید، اگر نه مطمئن باشید اون کتاب رو نه میتونید بخونید نه اصلاً خریدنش به صرفه هست از لحاظ مالی و حجم کاغذی که داره مصرف میشه برای چاپشون. پس چرا اساتید منبع معرفی میکنن؟ برای اینکه شما احياناً اگر سر کلاس یه مطلبی رو یاد نگرفتید، برید یه سر کتابخونه اون کتابی که استاد معرفی کردن رو قرض بگیرید و اون قسمتی که متوجه نشدید رو از روی کتاب بخونید؛ یا به لطف پیشرفت تکنولوژی میتونید برید فایل کتاب رو از اینترنت پیدا و دانلود کنید و اون قسمت رو ازش مطالعه کنید. گاهی اوقات هم اساتید موقع تمرین دادن آدرس بهتون میدن که فلان صفحه فلان کتاب، فلان شماره سوال رو برید حل کنید، و اینجا هم باز همون راهکارهایی که یه کم بالاتر گفتیم رو میتونید استفاده کنید.



سوال مهم دیگه ای که هست اینه که آیا این کتاب ها ترجمه شده هستند یا باید انگلیسی بخونیم؟ جواب: معمولاً کتاب های معروف تر ترجمه فارسیشون موجود هست، اما خیلی اوقات این ترجمه ها کیفیت خوبی ندارن و به عنوان یک فارسی زبان، احتمالاً از اون کتاب انگلیسی خیلی چیز بیشتری متوجه میشین تا ترجمه اون کتاب. طبعاً استثناهایی هم وجود دارن و ترجمه هایی هم وجود دارن که ارزش خوندن دارن، اگر نیاز داشتید میتونید تهیه کنید و بخونید، اما خب این نکته رو مدنظر داشته باشید که شما برای اینکه بتونید توی رشته تون موفق باشید، باید همیشه از لحاظ علمی و فنی به روز باشید؛ این به روز بودن هم فقط با مسلط بودن به زبان انگلیسی و مطالعه منابع انگلیسی زبان بدست میاد. بعداً میبینید که همه منابعی که شما نیاز به خوندنشون دارید ترجمه شده نیستن و دائماً مجبورید با منابع انگلیسی سر و کله بزنید، پس چه بهتر که زودتر شروع کنید به یادگیری و پیشرفت کردن در این زبان تا حداقل بتونید مسیرتون رو ادامه بدید.

صحبت از یادگیری زبان شد، سوالی که هست اینه که چطور زبان یاد بگیریم؟ اگر هیچی انگلیسی بلد نباشیم چی؟ جواب: تنها راه پیشرفت توی زبان انگلیسی اینه که باهاش درگیر باشید. اگر میخواید خوندنتون قوی بشه، باید بخونید؛ اگر میخواید نوشتنتون قوی بشه، باید بنویسید و همینطور درباره صحبت کردن و شنیدن که مهارت های اصلی هستن توی هر زبانی. یه راه خوب برای اینکه مهارت خوندنتون رو تقویت کنید اینه که کتاب

درس‌های مهم دیگه‌ای هم توی دانشگاه میخونید از جمله ساختمان‌داده، طراحی الگوریتم، پایگاه‌داده و ... که خب اینجا همیشه دونه دونه اسم برد و توضیح داد ولی خب تقریبا همون مواردی که بالاتر گفتیم اینجا هم صدق میکنن.

در نهایت هم راجع به بحث درس و نمره بگم که خیلی وقت‌ها تلاش میکنید و واقعا نتیجه‌بخش نیست، اینجا اون درس رو اصطلاحا «میوفتید» یا «رد میشید» یا اسامی دیگه که خب اتفاقیه که میوفته در هر صورت. اصلا افتادن برای مرده! (با عرض پوزش از فمینیست‌های عزیز، اصطلاحه دیگه، شما بگید برای زنه، کی به کیه). درس‌هایی که میوفتید رو میتونید ترم‌های بعد دوباره اخذ کنید یا معرفی به استاد بگیرید و از اینجور کارها. البته خب سعی کنید زیاد درس‌ها رو نیوفتید چون اونجوری یهو میبینید به جای ۸ ترم، ۱۰-۱۲ ترم درس خوندید و هنوز هم مدرک نگرفتید و بدتر از اون، ممکنه معدلتون خیلی پایین بیاد و مشروط بشید که اونهم دردسرهای خودش رو داره. توی دانشگاه ما اگر معدلتون زیر ۱۲ بشه شما اون ترم رو مشروط شدید و ترم بعد اجازه ندارید بیشتر از یه تعدادی واحد بردارید و اینکه اگر دو سه ترم مشروط بشید باهاتون خداحافظی میکنن و از دانشگاه اخراجتون میکنن. راجع به این مسئله برید پرسید و ببینید قوانین دانشگاه شما چطوری و صد البته، سعی کنید اصلا کارتون به اینجا نکشه. برعکس این قضیه هم هست؛ اونهایی که نمرات بالایی بگیرن اصطلاحا معدل الف میشن. اینجا معدل بالای ۱۷ الف حساب میشه و اونهایی که الف شدن میتونن به جای ۲۰ واحد، ۲۴ واحد درس بگیرن. اگر برای تموم کردن درستون و خلاص شدن از دانشگاه عجله دارید این قضیه کمکتون میکنه.

یه نکته آخر رو هم راجع به بحث کلاس‌ها بهتون بگم؛ نکته مثبتی که دانشگاه داره اینه که میتونید توی کلاس اساتید دیگه به عنوان مهمان شرکت کنید. یعنی مثلا به فلان میحث از فلان رشته علاقه دارید، میرید میشینید سر کلاس اون درس. معمولا کسی کاریتون نداره ولی خب شاید بهتر باشه از استاد اون درس اجازه بگیرید قبلش.

شرکت در کلاس تنها کاری نیست که توی دانشگاه میتونید انجام بدید. **سوال پیش میاد: توی دانشگاه چه کارایی میشه کرد؟ جواب:** فعالیت‌های متنوع در حوزه مختلف. از کارای علمی و فنی و پژوهشی بگیر تا کار هنری و ادبی و خیریه. توی دانشگاه‌ها گروه‌های مختلفی با اهداف و ارزش‌های شما هم‌راستا باشن و در خیلی‌هاشون میتونن با اهداف و ارزش‌های شما هم‌راستا باشن و در نتیجه فرصت خوبی باشن برای اینکه کنار درس خوندن، کارهای مفید دیگه‌ای هم انجام بدید. نکته مهم اینکه تقریبا همه این گروه‌ها دانشجویی هستن و افرادی مثل خودتون اونجا دارن فعالیت میکنن، پس شما هم اگر علاقه داشتید میتونید باهاشون همراه بشید.

اولین گزینه‌ای که پیش روتون هست انجمن‌های علمی هستن، معمولا هر رشته‌ای یک انجمن علمی داره، عده‌ای افراد که تصمیم داشتن به پیشرفت علمی و فنی جامعه‌شون کمک کنن و دور هم جمع شدن تا به سمت این هدف حرکت کنن. انجمن‌های علمی

میدادن وقتشون سر کلاس تلف نشه. برگردیم عقب‌تر، اونجایی که گفتم بلد بودن درس شرط کافی نیست، اگر بخوام دقیق‌تر بگم، بلد بودن درس حتی شرط لازم هم برای پاس کردن بعضی درس‌ها نیست. بعضی درس‌ها رو واقعا همیشه کاریشون کرد، فقط با توکل به خدا و تلاش و کوشش میشه یه نمره بالای ده گرفت و از اون درس خلاص شد. خیلی‌ها خیلی درس‌ها رو پاس میکنن و حتی نمره بالا میگیرن، بدون اینکه واقعا چیزی از اون درس بدونن. پس حواستون باشه اگر نمره پایین از یه درسی گرفتید اون دلیل بر ناتوانی یا عدم دانش شما نیست، شما اون «عدد» نیستید و اون عدد هم لزوما بیانگر توانایی و سطح سواد شما نیست.



اما حالا که این بحث مطرح شد، سوال پیش میاد که پس چرا میایم دانشگاه اگر قراره درس‌ها رو یاد نگیریم؟ **جواب:** قرار نیست درس‌ها رو یاد نگیرید. اتفاقا خیلی از درس‌ها هستن که توی این رشته فوق‌العاده مهمن. درسته که درس‌های غیرمفید و بیخود هم توی چارت درسی وجود دارن (که البته ممکنه برای خیلی‌های دیگه مفید باشن)، اما این دلیل نمیشه درس‌های مهم رشته رو هم نادیده بگیریم. در هر حوزه‌ای از رشته کامپیوتر که بخواید کار کنید، یکسری درس‌ها براتون حیاتی خواهند بود. درس «مبانی کامپیوتر و برنامه‌سازی» و درس «برنامه‌سازی پیشرفته» که ترم اول و دوم میخونید تقریبا مهمترین دروسی هستن که در سال اول دانشگاه باهاشون روبرو میشید. توی این درس‌ها شما مهارت برنامه‌نویسی رو یاد می‌گیرید. خیلی‌ها میگن ((من که میخوام فلان حوزه از کامپیوتر برم و برنامه‌نویسی به دردم نمیخوره)) یا اینکه ((من بعدا خودم دوره میبینم چون درسای دانشگاه قدیمین و به درد نمیخوره)) اما اینطور نیست؛ در جواب گروه اول باید بگم که برنامه‌نویسی تقریبا توی اکثر حوزه‌های کامپیوتر نیازتون میشه، این رو میتونید از افراد موفق اون حوزه پرسید. راجع به اینکه افراد موفق رو چطوری پیدا کنیم و کی دقیقا موفقه هم باید یه صحبتی کنیم سر فرصت. در جواب گروه دوم هم جواب‌های مختلفی میشه داد. اول: شما که در هر صورت باید این درس رو پاس کنید، باید سر کلاس بیاید و تمرین حل کنید و پروژه بنید، پس چرا همینجا یاد نگیرید؟ اگر قراره دوره جدا نگاه کنید خب حین گذروندن این درس دوره رو ببینید، اینجوری خیلی عالی میشه اتفاقا. دوم: خیلی از مباحثی که توی این درس‌ها گفته میشه رو توی دوره‌ها ممکنه پیدا نکنید چون جزو سرفصل اون دوره‌ها نیستن، ولی خب دونستشون ضروریه. پس به این درس‌ها اهمیت بدید.

کل زندگیشون رو پای این میذارن که هدف زندگیشون رو پیدا کنن و احیاناً اگر هم این هدف رو پیدا کنن، دیگه زمانی برای رسیدن بهش ندارن. اینجوری هم نیست که اگر یک حوزه‌ای رو انتخاب کردید تا آخرش مجبور باشید همون رو ادامه بدید، از این شاخه به اون شاخه پریدن بده ولی تلف کردن وقت برای پیدا کردن والاترین هدف غایی و نهایی (که ممکنه هیچوقت پیدا نشه) از اون هم بدتره.

در ادامه صحبت‌های قبلی اضافه کنم، هر زمان که انتخاب کردید میخواید توی چه حوزه‌ای کار کنید، نیاز دارید که مسیر درست حرکت در اون حوزه رو هم پیدا کنید. لزوماً تنها یک مسیر برای رسیدن به هدفتون وجود نداره، اما خب مسیرهای مختلف میتونن شما رو زودتر یا دیرتر و با کیفیت‌های مختلف به هدفتون برسونن. میتونید با سرچ کردن توی اینترنت، نقشه راه (Roadmap) های مختلفی برای حوزه‌ای که قصد دارید در اون کار کنید رو پیدا کنید. همچنین میتونید از افراد حرفه‌ای اون حوزه سوال کنید که راهنماییتون کنن که از کجا باید شروع کنید و چجوری باید پیش برید. پیدا کردن یه منتور مناسب میتونه فرصت خوبی باشه که اگر براتون پیش بیاد کلی جلو میندازتون.

در نهایت هم بگم، مطالعه غیردرسی کنید؛ همه چیز درس و دانشگاه نیست. کتاب بخونید، مستند ببینید، سخنرانی‌های افراد مختلف رو گوش کنید، پادکست‌ها و کانال‌های مختلفی که حس میکنید بهشون علاقه دارید رو دنبال کنید و سعی کنید رشته‌های دیگه و هنرهای دیگه رو هم دنبال کنید. تک بعدی بودن هیچوقت خوب نیست، اگر به زندگی افراد مطرح و تاثیرگذار در حوزه‌های مختلف نگاه کنید میبینید که اکثر اونها توی حوزه‌های مختلفی مطالعه و فعالیت داشتن، از فلسفه و جامعه‌شناسی گرفته تا موسیقی و نقاشی و ... این فرصت رو از خودتون نگیرید و زمانی رو برای اینجور فعالیت‌ها و مطالعات هم در برنامه خودتون قرار بدید.

امیدوارم مواردی که براتون نوشتم مفید بوده باشن و تونسته باشم کمکی (هرچند کم) بهتون کرده باشم، چه در جهت حل شدن سوالاتی که احتمالاً داشتید و چه در جهت پیدا کردن مسیرتون و رسیدن به اهدافتون. واقعیت اینه که خیلی از آدمایی که اومدن دانشگاه هنوز نمیدونن دارن چیکار میکنن، حتی اونهایی که میدونن هم گاهی فراموش میکنن که واقعاً دارن چیکار میکنن. پس «لطفاً هول نشید» و عجله نکنید، تلاش کنید که تجربه کسب کنید و مهمتر از همه، به بقیه هم کمک کنید و مطمئن باشید نتیجه این کمک‌ها رو بعداً توی زندگیتون میبینید. ممنونم که تا اینجا همراه من بودید؛ امیدوارم همیشه توی زندگیتون موفق و سربلند باشید!

بین‌رشته‌ای هم داریم که فرصت خوبی هستن برای آشنایی و فعالیت در رشته‌های متفاوت با رشته خودتون. اعضای انجمن‌های علمی معمولاً توی دانشگاه کارگاه‌ها و دوره‌های آموزشی برگزار میکنن، سمینار و وبینار و گفت‌گو و ... میذارن و تلاش میکنن بین دانشگاه و صنعت ارتباط ایجاد کنن. گزینه بعدی تشکلات سیاسی و اجتماعی هستن. گزینه بعدی هم کانون‌های فرهنگی هستن که در حوزه‌های مختلف فرهنگی و هنری مشغول فعالیتن از جمله نویسندگی و عکاسی و نقاشی و ... تا هلال احمر و کارآفرینی و اینجور موارد. کانون‌های خیریه هم در کنار اینها وجود دارن که تماماً توسط دانشجویها اداره میشن و اگر دغدغه کمک و کار خیر دارید میتونن فرصت خوبی باشن که تاثیرات خوبی روی جامعه‌تون بذارید.

جدای از این مواردی که بالاتر گفتم (که توسط دانشجویها اداره میشن) میتونید با آزمایشگاه‌ها و اساتید دانشگاهتون هم همکاری کنید و کار پژوهشی کنید، مقاله بنویسید، پروژه بنزید و کارهای مشابه انجام بدید. بسته به هدفی که برای آینده‌تون دارید، دانشگاه فرصت‌های خوبی بهتون میده که بتونید برای آینده‌تون تجربه خوبی کسب کنید، رزومه مناسبی بسازید و با افراد زیادی ارتباط بگیرید که بعدها بتونید در حوزه‌های مختلف به همدیگه کمک کنید و حتی شاید همکار بشید.



اما صحبت از هدفتون برای آینده شد. یه نکته مهمی رو باید بدونید، اینکه صرفاً گرفتن مدرک فایده‌ای براتون نداره. شما باید کار یاد بگیرید، تجربه پیدا کنید و کسب علم کنید تا بتونید به اهدافی که دارید برسید. مسیرهای زیادی رو میشه برای ادامه زندگی حرفه‌ای انتخاب کرد؛ از کار آکادمیک و ادامه تحصیل گرفته تا ورود به بازار کار و ... باید بررسی کنید و ببینید به چه حوزه‌ای علاقه دارید و چه شغلی برای شما مناسب‌تره و برنامه‌تون برای آیندتون چیه و بر این اساس، در دوران تحصیل هم فعالیت‌های مرتبط انجام بدید. مثلاً اگر میخواید به عنوان برنامه‌نویس وارد بازار کار بشید، باید درس‌های مرتبط با نرم‌افزار رو خوب یاد بگیرید و در کنار اونها دوره‌های آموزشی دنبال کنید و اگر تونستید پروژه بنزید و کارآموزی برید. متأسفانه دنیای ما اینجوریه که مهم‌ترین تصمیمات زندگیمون رو زمانی باید بگیریم که کمترین تجربه رو داریم، بنابراین سعی کنید از تجربیات افراد دیگه استفاده کنید و تحقیق کنید، حوزه‌های مختلف رو امتحان کنید و حوزه‌ای که به نظرتون براتون مناسب‌تره رو پیدا کنید؛ البته حواستون باشه که این بررسی و تحلیل زیاد هم طول نکشه، خیلی‌ها



همان طور که گفته شد از ریجکس می توان برای جستجو، جایگزینی و تطابق الگو استفاده کرد. در این راستا به معرفی سه تابع این کتابخانه می پردازیم:

تابع search :

```
txt="some text"
x=re.search("your pattern",txt)
```

خروجی قطعه کد بالا شامل یک رشته و یک تاپل است که به ترتیب عبارت تطابق داده شده و محل شروع و پایان عبارت مذکور را در صورت وجود بر می گرداند. اگر جستجو نتیجه ای نداشته باشد و عبارت مورد نظر در متن نباشد خروجی None خواهد بود.

حال فرض کنیم می خواهیم وجود یک عبارت یا یک الگو را در رشته دلخواه بررسی کنیم. برای مثال فرض کنیم می خواهیم ببینیم آیا کلمه "Python" در یک رشته وجود دارد یا خیر:

```
import re

string = "Python is fun"

match = re.search('Python', string)

if match:
    print("pattern found inside the string")
else:
    print("pattern not found")

# Output: pattern found inside the string
```

تابع findall :

```
txt="some text"
x=re.findall("your pattern",txt)
```

دستور بالا یک لیست شامل تمامی عبارت هایی که در متن با الگو تطابق دارند را خروجی می دهد.

برای مثال فرض کنیم می خواهیم همه اعداد استفاده شده در یک رشته را

Regex در پایتون



عطیه سادات طباطبایی
دانشجوی علوم کامپیوتر
دانشگاه تهران
atiehtaba82@gmail.com

Regex، مخفف عبارت Regular Expression، دنباله ای از کاراکترها است که یک الگوی جستجو را تعریف می کنند. ریجکس یک ابزار قدرتمند برای پردازش متن و تطبیق الگو است.

زبان های برنامه نویسی مانند java، Python و Perl از ریجکس پشتیبانی می کنند. این کتابخانه معمولاً در عملیات جستجو و جایگزینی استفاده می شود.

ریجکس در واقع ترکیبی از کاراکترهای خاص است که الگویی را تعریف می کنند که با مجموعه خاصی از کاراکترها مطابقت دارد.

در ادامه ابتدا به معرفی قواعد الگوسازی ریجکس می پردازیم و سپس چند مثال را با هم می بینیم.

در جدول زیر برخی از پرکاربردترین قواعد این کتابخانه را مشاهده می کنید:

General Tokens	
.	Any character
\n	Newline character
\t	Tab character
\s	Any whitespace character (Including \t, \n, etc)
\S	Any non-whitespace character
\w	Any word character (Upper/lowercase letters, 0-9, _)
\W	Any non-word character
\b	Word boundary (Matches between characters)
\B	Non-word boundary
^	The start of a line
\$	The end of a line
\\	The literal character "\"

Quantifiers	
a b	Match either "a" or "b"
?	Match either "a" or "b"
+	One or more
*	Zero or more
*?	Zero or more, but stop after first match
{N}	Exactly N number of times (Where N is number)
{N, M}	From N to M number of times (Where N and M are numbers)

برای مثال "[A-Za-z]+" یک الگو است که با هر رشته ای که فقط از حروف تشکیل شده مطابقت دارد.

برای استفاده از ریجکس در زبان پایتون از دستور زیر استفاده می کنیم.

```
import re
```

که نشان دهنده سال است و دو جفت دوددد که به ترتیب نشان دهنده ماه و روز هستند و اعداد هر قسمت با کاراکتر "/" از هم جدا شده اند مانند ۲۰۲۳/۰۵/۱۰. می خواهیم اگر کاربر تاریخ تولدش را مطابق الگوی خواسته شده وارد نکند ارور دریافت کند. خواهیم داشت :

```
import re
user_birthdate=input()
x=re.search("[0-9]{4}/[0-9]{2}/[0-9]{2}",user_birthdate)

if not x:
    print("pleas enter your birthdate correctly")
```

اعتبار سنجی آدرس ایمیل:

فرض کنید از کاربر خواسته ایم تا آدرس ایمیل خود را وارد کند. می خواهیم ببینیم آیا عبارتی که کاربر وارد کرده می تواند یک آدرس ایمیل باشد؟

برای این کار ابتدا یک الگو تعریف می کنیم. اگر الگوی تعریف شده با عبارت وارد شده توسط کاربر مطابقت داشت رشته ورودی می تواند یک آدرس ایمیل باشد در غیر این صورت عبارت وارد شده معتبر نخواهد بود.

برای تعریف الگو ابتدا به ساختار یک آدرس ایمیل دقت می کنیم. در یک آدرس ایمیل ابتدا مجموعه ای از کاراکترها که می توانند شامل عدد، حروف و یا کاراکترهای دیگر باشند کنار هم قرار می گیرند سپس علامت "@" می آید و پس از آن نیز مجددا مجموعه ای از کاراکترها می آید و در آخر نیز دامنه سایت که با "." شروع می شود :

```
import re
addres=input()
x=re.search("[A-Za-z0-9._-]+@[A-Za-z0-9._-]+\.[A-Za-z0-9._-]+",addres)
if x:
    print("valid email address")
else:
    print("invalid email address")
```

دقت کنید چون "." جزء قواعد ریجکس است و برای تطابق هر کاراکتر دلخواه به شمار می رود، اگر بخواهیم از آن به عنوان خود کاراکتر "." استفاده کنیم قبل از آن از علامت "\" استفاده می کنیم.

سخن پایانی :

کار با ریجکس در نگاه اول می تواند دشوار به نظر برسد اما با تمرین می توانید از قابلیت های این کتابخانه قدرتمند بهره ببرید.

برای مطالعه بیشتر می توانید به آدرس https://www.w3schools.com/python/python_regex.asp رجوع کنید.

استخراج کنیم؛ با توجه جدول بالا الگوی مورد نظر ما "[۰-۹]+" خواهد بود بنابراین طبق دستور زیر عمل می کنیم:

```
import re

string = 'The first release candidate of Python 3.12 was offered on 6 August 2023.'
pattern = '[0-9]+'

result = re.findall(pattern, string)
print(result)

# Output: ['3', '12', '6', 2023]
```

تابع sub :

```
txt="some text"
x=re.sub("your pattern","new string",txt)
```

با استفاده از دستور بالا می توانید یک عبارت جدید را جایگزین الگوی تعریف شده کنید.

برای مثال فرض کنیم می خواهیم همه فاصله های یک رشته را حذف کنیم. برای این کار باید همه فاصله ها را با یک رشته خالی جایگزین کنیم:

```
import re

string = 'Hello World!'
# matches all whitespace characters
pattern = '\s+'
# empty string
replace = ''

new_string = re.sub(pattern, replace, string)
print(new_string)

# Output: HelloWorld!
```

حال برای درک بهتر کاربرد regex چند مثال را با هم بررسی می کنیم

مطابقت فرمت تاریخ (yyyy/mm/dd):

فرض کنیم از کاربر خواسته ایم تا تاریخ تولدش را مطابق فرمت بالا وارد کند. فرض می کنیم فقط فرمت بالا را به عنوان تاریخ می پذیریم (چهار عدد

یک کلاس ابسترکت (می توان از اینترفیس نیز با شرایط مختلف استفاده کرد) داریم که یک سری اتریبیوت ها و متدهای عمومی را دارا است.

```
public abstract class Sandwich {

    public String name;
    public int price;

    public String preparing(){
        return "One " + this.name + " heard!";
    }

    public String ready(){
        return "One " + this.name + " coming up!";
    }

}
```

چند کلاس کانکرت (concrete classes) کلاس هایی که نوعشان مشخص است) را با اتریبیوت ها و متدهای خاص، به همراه سازنده داریم.

```
public class HotDog extends Sandwich {
    public HotDog(){
        this.name = "Hot Dog";
        this.price = 20;
    }
}

public class Falafel extends Sandwich{
    public Falafel(){
        this.name = "Falafel";
        this.price = 10;
    }
}
```

یک کلاس سازنده اشیا (کارخانه) وجود دارد که با توجه به پارامتر ارسال شده، نوع شیء درخواستی را تشخیص داده، و یک شیء جدید با آن نوع می سازد.

```
public class SandwichFactory{
    public Sandwich makeSandwich(String type){
        if (type.equals("Hot Dog")){
            return new HotDog();
        } else if (type.equals("Falafel")){
            return new Falafel();
        }
        else{
            System.out.println("Sorry, we don't have that sandwich.");
            return null;
        }
    }
}
```

Design Patterns

الگوهای طراحی

پروژه های بزرگ معمولاً نیازمند نظم بالایی هستند. دلایل این امر واضح است: عیب یابی سریع، قابلیت توسعه پذیری بالا و سریع، و خوانا و قابل فهم بودن کد برای مهندسین جدید. اگر بنا باشد در هر پروژه استانداردهای متفاوتی برای برقرار کردن نظم اعمال شود، وفق پیدا کردن برای افراد جدید بسیار سخت و زمان بر خواهد بود؛ همچنین شکل گیری ارتباط مؤثر بین افراد نیز به دلیل متفاوت بودن این استانداردها دشوار خواهد بود. «دیزاین پترن» ها، یا الگوهای طراحی، اینجا به کمک ما می آیند. این الگوها، پاسخ هایی بهینه و آزموده، در برابر مسائلی که در برنامه نویسی شیء گرا بروز میکنند هستند. به همین دلیل، خیلی از شرکت ها به «دیزاین پترن» ها اهمیت زیادی می دهند، و حتی برخی دانستن آن ها را برای ورود به شرکت الزامی می دانند، چرا که نقش بسیار مؤثری در یکپارچه سازی و کاهش زمان توسعه ایفا می کنند. در ادامه به معرفی دو پترن ساده می پردازیم.



سید محمد جعفری
دانشجو و مهندس کامپیوتر
دانشکده فاریاب دانشگاه تهران
Smjaffry@ut.ac.ir

اول: Factory Method

فرض کنید شما صاحب یک ساندویچی هستید. از ساعت ۱ بعد از ظهر که آغاز به کار می کنید، سفارش های متعددی برایتان ارسال می شود. در حال حاضر شما فقط ساندویچ فلافل و هات داگ عرضه می کنید. اما از آنجایی که شما خیلی خفن هستید، می خواهید در آینده ساندویچ های بیشتری در منو قرار بدهید. چون تازه کار هستید، هنوز پول کافی برای استخدام یک برنامه نویس جهت طراحی برنامه مدیریت سفارش ندارید؛ از طرفی شما برنامه نویس قدری هستید، پس تصمیم می گیرید که این سامانه را خود تهیه کنید. برای اینکه یک کد تمیز و خوشگل بنویسید که نوع ساندویچ سفارشی را بگیرد و یک شیء از آن نوع ساندویچ بسازد چه ایده ای دارید؟ آیا می توانید هنگام اجرای برنامه اشیا مختلفی بسازید؟ اگر قرار باشد ساندویچ های دیگری به منو اضافه شود، چقدر به کارتان اضافه می شود؟

مشخصاً انجام دادن این کار سخت نیست، می توانیم با چند عبارت شرطی که ورودی را رصد میکند اشیا مختلفی بسازیم، اما این کار انعطاف برنامه را کاهش می دهد. بر خلاف روش ساده، پترن Factory Method یا روش کارخانه در واقع کار ساختن شیء های متعدد را ساده می کند. این پترن با اتکا بر مفاهیم ارث بری و کلاس های ابسترکت، تعیین نوع شیء را بر عهده زیر کلاس (کلاس فرزند) می گذارد. برای این مثال، ما یک کلاس ابسترکت با نام Sandwich می سازیم. حالا هر کدام از ساندویچ های ما این کلاس ابسترکت را پیاده سازی می کنند. اثر این کار موقعی بیشتر به چشم می آید که ساندویچی شما گسترش پیدا کند، و ساندویچ های بیشتری به منو اضافه شود.

روش کار این پترن ساده است. به موارد زیر توجه کنید:

را بصورت میخ گرد برگرداند. اگر نصف قطر قاعده را محاسبه کنیم و آن را با شعاع حفره مقایسه کنیم، می توانیم درباره قرار گرفتن یا نگرفتن میخ در حفره اظهارنظر کنیم. نصف قطر را می توانیم با استفاده از رابطه فیثاغورث محاسبه کنیم که مقدار آن، به شرطی که اندازه ضلع a باشد برابر است با:

$$\text{Radius} = \frac{1}{2} * a * \text{sqrt}(2)$$

ساختار برنامه را در ادامه مشاهده می کنید. توجه کنید که `SquarePeg` یک `Adapter` میخ مربعی دریافت می کند و آن را به یک میخ گرد تبدیل می کند. حالا می توانیم از این میخ گرد برای چک کردن قرارگیری آن در حفره استفاده کرد.

```
public class RoundPeg {
    public int radius;

    public RoundPeg(int radius){
        this.radius = radius;
    }
}

public class RoundHole {
    public int radius;

    public RoundHole(int radius){
        this.radius = radius;
    }

    public boolean fits(RoundPeg peg){
        return this.radius > peg.radius;
    }
}

public class SquarePeg {
    public int width;

    public SquarePeg(int width){
        this.width = width;
    }
}
```

```
public class SquarePegAdapter extends RoundPeg {
    public SquarePeg peg;

    public SquarePegAdapter(SquarePeg peg){
        super(peg.width);
        this.peg = peg;
    }

    public int getRadius(){
        return (int) (peg.width * Math.sqrt(2) / 2);
    }
}
```

اکنون ما یک کلاس با نام `SandwichFactory` داریم که کار آن ساختن انواع مختلف ساندویچ است. می توان تصور کرد که این کلاس نقش آشپز را ایفا می کند. ما به او به فارسی سفارش مان را میگوییم، مثلاً می گوییم: «هات داگ»، و او یک هات داگ واقعی به ما تحویل میدهد! حال برنامه ای که برای اجرا مینویسیم را ببینید:

```
public static void main(String[] args) {
    SandwichFactory sandwichFactory = new SandwichFactory();
    System.out.println("What kind of sandwich do you want?");
    Sandwich sandwich = sandwichFactory.makeSandwich(userInput());
    System.out.println(sandwich.preparing());
    System.out.println(sandwich.ready());
}
```

یکی از اهداف این روش، مشخص کردن نوع شی در زمان اجرا است. میتوانستیم بصورت `if-else` در کلاس `main` هم کد این برنامه را بنویسیم، اما قواعد زیادی را باید زیر پا می گذاشتیم. در واقع ما ساخت اشیا را به کلاس `SandwichFactory` محول کردیم تا کاربر را درگیر جزئیات ایجاد شی نکنیم. علاوه بر این، دیگر لازم نیست برنامه اجرا را دستخوش تغییر کنیم، بلکه تغییرات را در کلاس `SandwichFactory` اعمال می کنیم. این تغییرات میتواند شامل ساختن کلاس کانکرت جدید، حذف کلاس، و یا تغییر پارامتر لازم برای ایجاد یک شی باشد.

دوم: Adapter

ممکن است شما نیز هنگام خریدن وسایل برقی متوجه شده باشید که کابل برق هر وسیله ای با پریز خانه ای ما سازگار نیست. البته ما خیلی شگفت زده نمی شویم و برای حل مشکل، یک تبدیل خریداری می کنیم تا بتوانیم از وسیله استفاده کنیم. مشابه این مسأله در برنامه نویسی نیز ممکن است پیش آید. فرض کنید برنامه ای نوشته اید که در خروجی اطلاعات را بصورت یک فایل تکست ذخیره می کند. یک برنامه نیز از گیت هاب دانلود کرده اید که اطلاعات را پردازش می کند. مسأله این است که برنامه پردازش اطلاعات، فقط فایل اکسل می پذیرد. شما اکنون میتوانید کد خود را تغییر دهید، یا یک برنامه بنویسید که ورودی تکست گرفته و خروجی اکسل می دهد.

برای نشان دادن روش کار این الگو، که یک الگوی ساختاری محسوب می شود، از مثال یک اسباب بازی استفاده می کنیم. یک تخته داریم که در آن حفره ی گردی وجود دارد که شعاع آن مشخص است. حال می خواهیم یک سری میخ با قاعده مربعی را در این حفره ها قرار دهیم. اندازه قاعده ی میخ ها متفاوت است، و ما باید تشخیص دهیم کدام میخ ها در این حفره قرار می گیرد. واضح است که هر میخ گردی که شعاع قاعده ی آن کمتر از شعاع حفره باشد در آن قرار می گیرد، اما درباره میخ های مربعی که برایشان شعاع تعریف نشده چه باید کرد؟ درست است که میتوان یک اتریبیوت با نام شعاع در کلاس میخ مربعی در نظر گرفت اما شاید ما فقط یک بار از این صفت استفاده کنیم، یا شاید لازم باشد برای اشکال مختلفی این تبدیل را انجام دهیم، این امور نشان می دهد که اتریبیوت نوشتن راه حل مناسبی نیست. الگوی `Adapter` در ورودی یک نوع متغیر دریافت میکند و در خروجی اطلاعاتی که از آن متغیر لازم داریم را با نوع مختلفی برمی گرداند.

میخواهیم یک مبدل برای میخ های مربعی بسازیم، تا به ما معادل آن میخ

هوش مصنوعی و گرافیک

هوش مصنوعی در گرافیک نیز مانند بسیاری از حرفه‌ها جایگاه خود را پیدا کرده است، به گونه‌ای که طراحان و شرکت‌های بزرگ هنگام تولید نرم افزارهای گرافیکی به دنبال استفاده و بهره‌وری حداکثری از آن هستند.

به این نکته توجه داشته باشید که هوش مصنوعی نمی‌تواند جایگزین خلاقیت انسان شود. به طور کلی خلاقیت انسان ناشی از الگوهای خاصی نیست که توسط هوش مصنوعی قابل تقلید باشد؛ و همان‌طور که می‌دانید تاثیر هوش مصنوعی بیشتر در مورد بهینه‌سازی و افزایش سرعت است.

هوش مصنوعی می‌تواند حجم وسیعی از داده‌ها را تجزیه و تحلیل کند و تنظیمات طراحی را پیشنهاد دهد. سپس یک طراح می‌تواند تنظیمات را بر اساس آن داده‌ها انتخاب و تأیید کند. در واقع هوش مصنوعی می‌تواند به عنوان مجری، تسريع کننده یا به عنوان تحلیل‌گر بر اساس الگوهای موجود به طراحان گرافیکی کمک بسیاری بکند.

ادوبی (Adobe) بالاخره دست به کار شد

پس از مدت‌ها بعد از ورود هوش مصنوعی به دنیای نرم‌افزارها به خصوص نرم‌افزارهای گرافیکی ادوبی نیز دست به کار شد. ادوبی به عنوان پرچم‌دار تولید نرم‌افزارهای گرافیکی (مانند ایلستریتور، فتوشاپ، پریمیر و...) انتظارات بالایی را متوجه خود می‌دید و بالاخره دست به رونمایی از هوش مصنوعی جدید خود یعنی فایرفلای (Firefly) زد.

امروزه در حال ورود به دنیایی هستیم که در آن انسان قادر خواهد بود دیدگاه خلاقانه خود را به سادگی با توصیف آنچه می‌خواهد با کلمات خود یا با یک حرکت ساده زنده کند. دنیایی که در آن به دلیل هوش مصنوعی مولد، مکالمه بین سازنده و رایانه در به چیزی طبیعی‌تر، شهودی‌تر و قدرتمندتر تبدیل خواهد شد.



چيست؟ Adobe Firefly

Adobe Firefly ابزاری مبتنی بر هوش مصنوعی است که در دسته هوش مصنوعی مولد قرار می‌گیرد. این ابزار از مجموعه‌ای از داده‌ها برای پردازش استفاده می‌کند.

Adobe Firefly دارای یک سیستم سریع است که در آن شما به سادگی کلمات یا توضیحات مورد نظر خود را تایپ کرده و سیستم بر اساس ورودی شما تصاویری را تولید می‌کند. فایرفلای راه‌های جدیدی برای ایده‌پردازی، ایجاد و برقراری ارتباط ارائه می‌کند، به شما کمک کند به سرعت و به آسانی تصاویر و جلوه‌های متنی با کیفیت بالا تولید کنید. همچنین به طور قابل توجهی تولید کارهای خلاقانه را با استفاده از هوش مصنوعی مولد بهبود می‌بخشد.

فایرفلای به نوعی توسعه فناوری ادوبی است چرا که ادوبی معتقد است که مردم باید قدرت داشته باشند تا ایده‌های خود را دقیقاً همان‌طور که تصور می‌کنند عملی کنند.

هوش مصنوعی و گرافیک:

Adobe Firefly

از گذشته تا به الان همواره با پیشرفت علم و تکنولوژی، کسب و کارها نیز تحت تاثیر قرار گرفته‌اند. به گذشته که برگردیم می‌بینیم از زمان ورود اولین ماشین‌ها در حوزه کار و صنعت کسب و کارها به سمت و سوی صنعتی شدن پیش رفته، و بهره‌مندی خود را از کامپیوترها به حداکثر رسانده‌اند.



سید علی فقیه موسس و دانشجو مهندسی کامپیوتر
دانشکده فوارای دانشگاه تهران
faghih.mousavi@ut.ac.ir

امروزه بازار بهره‌گیری از هوش مصنوعی بسیار داغ است. هوش مصنوعی به سرعت در حال فراگیر شدن است و جای خود را درون مشاغل گوناگون پیدا می‌کند. طراحی گرافیک که حرفه‌ای نوپا محسوب می‌شود، نیز از تاثیر هوش مصنوعی جان سالم به در نبرده است. در ادامه به نقش هوش مصنوعی در طراحی گرافیک می‌پردازیم و یکی از جدیدترین مدل‌های هوش مصنوعی مولد را با هم بررسی می‌کنیم.

هوش مصنوعی مولد (Generative AI) چیست؟

هوش مصنوعی مولد گونه‌ای از هوش مصنوعی است که می‌تواند داده‌های متنوعی نظیر تصاویر، ویدئو، صداها، متن‌ها و مدل‌های سه‌بعدی را تولید کند. Generative AI با یادگیری الگوها از داده‌های موجود دست به تولید خروجی‌های جدید و منحصر به فرد می‌زند. این هوش مصنوعی قادر به تولید محتوایی بسیار واقعی و پیچیده است که خلاقیت انسان را تقلید می‌کند و آن را به ابزاری ارزشمند برای بسیاری از مشاغل از جمله طراحی گرافیک تبدیل کرده است.

هوش مصنوعی مولد در سرتاسر جهان پراکنده شده و شیوه برقراری ارتباط، کار و خلاقیت انسان را دگرگون کرده است. به عنوان نمونه، ChatGPT، و فراگیر شدن آن با نزدیک به ۱۰۰ میلیون کاربر خود گواهی بر پذیرش سریع و تاثیر گسترده این فناوری پیشرفته است.



معرفی بخش‌های فعال وب اپلیکیشن فایرفلای

Text to Image: با استفاده از توضیحات تصاویری که به دنبال آن هستید را درخواست کرده و فایرفلای آنها را برای شما تولید می‌کند. شما می‌توانید با استفاده از فیلترها این درخواست را محدودتر کنید.



Generative Fill: شما را قادر می‌سازد تا با استفاده از یک قلم مو، قسمت‌هایی از پس‌زمینه تصویر را حذف کنید و سپس با استفاده از اعلان‌های متنی، پس‌زمینه جدیدی را رنگ کنید یا تولید کنید.



Text Effects: فایرفلای علاوه بر اینکه در ایجاد تصاویر واقعی و منحصر به فرد برای الهام بخشیدن یا استفاده در پروژه‌های خود قدرتمند است، کاربردهای جالبی برای ایجاد متن و فونت نیز دارد.

Generative Recolor: در این بخش شما می‌توانید فایل‌های spg خود را آپلود کنید و با استفاده از درخواست متنی از فایرفلای بخواهید تا وکتور آپلود شده را با رنگ‌های جدیدی که شما درخواست کرده‌اید به شما نشان دهد.

هوش مصنوعی و ادوبی

اگر مدتی است که از محصولات Adobe استفاده می‌کنید، ممکن است نام Sensei را شنیده باشید. Sensei هوش مصنوعی Adobe است که در سال‌های اخیر در برخی نرم‌افزارهای Adobe با هدف بهبود بهروری و حذف برخی از کارهای پیچیده یا تکراری گنجانده شده است.

برای مثال، ابزار Select Subject در فتوشاپ، به کمک این هوش مصنوعی ترکیب عکس را بسیار آسان‌تر و کاربردی‌تر کرده است.

Adobe Sensei که با مجموعه داده عظیمی از تصاویر و متن آموزش داده شده، Firefly را تقویت می‌کند. این هوش مصنوعی فایرفلای را قادر می‌سازد تا تصاویر واقعی و منحصر به فرد تولید کند و نیازهای فردی کاربران را برآورده سازد.

چرا ادوبی فایرفلای؟

شاید مهم‌ترین حسن فایرفلای در نظر گرفتن مسئله حق مالکیت برای تصاویر تولیدی آن باشد.

بسیاری از هوش مصنوعی‌های حوزه گرافیک مانند Midjourney، Stable Diffusion و DALL-E2 توسط ادوبی مورد تحقیق قرار گرفته‌اند، چرا که آنها بدون در نظر گرفتن مالکیت یا حق نسخه‌برداری، از تصاویر موجود در اینترنت استفاده می‌کنند، اما Firefly بر روی تصاویر منبع باز و محتوای Adobe Stock آموزش داده شده است.

ادوبی می‌گوید این کار Firefly را از نظر تجاری ایمن می‌کند.

چگونه از فایرفلای استفاده کنیم

هوش مصنوعی فایرفلای در سرویس‌های Adobe Creative Cloud، Adobe Express و Adobe Experience Cloud در دسترس قرار گرفته است. تمامی کاربران نرم‌افزارهایی مانند فتوشاپ، ایلاستریتر و غیره می‌توانند با فرمان‌های متنی از فایرفلای بخواهند کارهای مختلف را برایشان انجام دهد.

فایرفلای همچنین به صورت وب اپلیکیشن نیز در دسترس کاربران است. که به کاربران اجازه می‌دهد در اپلیکیشن‌های Adobe Creative Suite، و به طور رایگان و محدود (برای رفع محدودیت نیاز به تهیه اشتراک است) به آن دسترسی داشته باشند.

برای استفاده از وب اپلیکیشن فایرفلای لازم است که:

اتصال پایدار به اینترنت داشته باشید.

یک اکانت در Adobe داشته باشید.

در کشوری باشید که Adobe Firefly در آن موجود است.

با توجه به وضعیت اینترنت در ایران و در دسترس نبودن سایت ادوبی در ایران لازم است برای استفاده از فایرفلای از نرم‌افزارهای تغییر آی پی با آی پی‌های ثابت استفاده کنید و یا از دی ان اس بهره ببرید.

You have to try it to believe it

شما در برابر امنیت اطلاعات خود

مسئول نیستید!



علی رادمرد

دانشجو مهندسی کامپیوتر

دانشکده فاریابی دانشگاه تهران

aliradmard5@gmail.com

در عصر هوش مصنوعی، دانستن اینکه اطلاعات شخصی شما کجا مورد استفاده قرار گرفته است غیر ممکن است.

بارد بر اساس داده‌های جمع‌آوری شده بدون موافقت یا حتی بدون اطلاع از آن‌ها، آموزش داده شده‌اند. در بدترین حالت، سبک‌های آموزشی، مقدار زیادی از اطلاعات دیجیتال را جذب و ترکیب کرده و به عنوان مواد اولیه برای هوش مصنوعی مورد استفاده قرار می‌گیرند. در حالی که شرکت‌های فناوری در حال تلاش برای گنجاندن هوش مصنوعی در هر محصول قابل‌تصور هستند، از موتورهای جستجو گرفته تا بازی‌ها و ابزارهای نظامی، امکان ندارد بدانیم این خروجی به کجا می‌رود یا چگونه ممکن است تفسیر شود. فروشندگان داده (Data brokers) داده‌های زیادی را از وب استخراج کرده‌اند و پرونده‌های بزرگی برای مردم تشکیل داده‌اند اما خروجی این داده‌ها برای یک فرد عادی، یا موتورهای جستجو به طور رایگان قابل دسترسی نیست. دسترسی گسترده به هوش مصنوعی، احتمال نقض حریم شخصی را افزایش می‌دهد و افراد بیشتری در خطر مواجهه با عواقب مضر این عمل قرار می‌دهد.



مجموعه‌های زیادی که توسط هوش مصنوعی استفاده می‌شوند، بدون شک حاوی اطلاعاتی در مورد افراد هستند که توسط آنها ارائه نشده، ساخته نشده، یا حتی تصویری از وجود آنها نبوده است. سوابق عمومی مرتبط با ازدواج‌ها، رهن مسکن و مشخصات رای‌دهندگان همگی در این مجموعه‌ها قرار دارند، همچنین خبرها، اطلاعات کارمندان و صفحات ویکی‌پدیا. این مجموعه همچنین شامل میلیون‌ها عکس و ویدیو می‌شود؛ به عنوان مثال، Dall-E بر روی تصاویری آموزش داده شد که از رسانه‌های اجتماعی، موتورهای جستجو و سایت‌ها جمع‌آوری شده‌اند. بنابراین، اگر در پس‌زمینه یک عکس از سال ۲۰۰۷ در فلیکر (flickr) باشید، تصویر شما ممکن است برای آموزش یک الگوریتم استفاده شود. به نظر نمی‌رسد که کسی بداند چه چیزی وارد این مجموعه داده می‌شود، و هیچ راهی برای نظارت یا کنترل آن وجود ندارد. وقتی چت‌جی‌پی‌تی یک زندگینامه نادرست در مورد من می‌نویسد، من نمی‌دانم اطلاعات غلط از کجا نشأت گرفته، همانطور که نمی‌دانم اطلاعات صحیح از کجا آمده است. ما عادت داریم که حریم شخصی را به عنوان کنترل فردی بر روی اطلاعات بدانیم، اما اگر حتی منشأ اطلاعات شخصی‌تان را هم ندانید، امکان نظارت بر نحوه استفاده از آن وجود ندارد.

در سال ۲۰۱۰، مارک زاکربرگ در مراسم اهدای جوایز TechCrunch به حضار گفت که جوانان، به ویژه کاربران شبکه‌های اجتماعی، دیگر به حریم شخصی اهمیت نمی‌دهند. او گفت: «مردم واقعاً راحت‌تر شده‌اند، نه تنها اطلاعات بیشتری و انواع مختلفی از آن‌ها را به اشتراک می‌گذارند، بلکه با صراحت بیشتر و با افراد بیشتری این کار را می‌کنند. این قاعده اجتماعی فقط چیزی است که با گذر زمان تکامل یافته است.» این بیان نمایانگر یک باور متداول است که نقض حریم شخصی وقتی رخ می‌دهد که افراد اطلاعات شخصی خود را آشکار می‌کنند. به عبارت دیگر، وقتی چیزی در ردیت یا تیک‌تاک وایرال می‌شود یا عکس ارسال شده به یک طرفدار فاش می‌شود، در ابتدا، اشتباه از طرف کسی است که آن را منتشر کرده است. این مدل از مسئولیت فردی بسیار مقاوم و ماندگار است، اما کاملاً اشتباه است و در عصر هوش مصنوعی بی‌فایده است.



هوش مصنوعی مفهوم مسئولیت فردی در حفظ حریم شخصی را کاملاً نابود می‌کند، زیرا نمی‌توانید دسترسی این الگوریتم‌ها به اطلاعات شما را کنترل کنید یا ببینید که آن‌ها چه کارهایی با آن انجام می‌دهند. ابزارهایی مانند چت‌جی‌پی‌تی، دال-ای و گوگل

اطلاعاتی که در یک موضوع ارائه می‌شود، می‌تواند کاملاً بازسازی و بازمخلوط شود، معنی آن را تغییر داده و با نقض آنچه که فیلسوف هلن نیسنوم آن را «یکپارچگی موضوعی» می‌نامد، مواجه شود. چگونه کسی می‌تواند از این جلوگیری کند؟



علاوه بر این، هوش مصنوعی مولد می‌تواند انواع مختلفی از نقض‌های حریم شخصی خلاقانه را ممکن سازد. من نتوانستم ChatGPT را متقاعد کنم که آدرس خانه دوستم را به من بدهد، اما با خوشحالی توضیح داد چگونه می‌توان آدرس خانه‌ای را آنلاین پیدا کرد. دیپ فیک‌ها مسئله دیگری است؛ چه چیزی می‌تواند از تقلید سبک، صورت یا حتی سخنان شخص دیگر حریم خصوصی را بیشتر نقض کند؟ من تعدادی ضبط قدیمی از صدای خودم را به ابزاری به نام Descript آپلود کردم و چند ساعت بعد یک نسخه سنتز شده از صدای خودم داشتم که می‌توانستم از آن برای گفتن هر چیزی استفاده کنم (free trial). در حالی که برخی از ابزارهای معروف هوش مصنوعی مولد حاوی محافظ حریم شخصی هستند، تا نقض‌های حریم شخصی خیلی جدی‌تری را جلوگیری کنند، برخی دیگر ندارند. هوش مصنوعی‌ها به داده‌ها دسترسی دارند و اگر با ناآرامی و بی‌وجدانی استفاده شوند، می‌توانند شبکه اجتماعی کامل یک شخص را نقشه‌برداری کنند، وضعیت مالی یا مشکلات سلامتی او را تشکیل دهند و شناسایی کنند و آیا در مقابل کلاهبرداری‌های اینترنتی می‌توانند از خود محافظت کنند یا خیر.

با اینکه از دوره‌ی هیجانی بودن هوش مصنوعی آگاه هستیم، اما تفاوتی بین هوش مصنوعی مولد و دیگر فناوری‌هایی با پیامد عمیق بر حریم شخصی وجود دارد. پلتفرم‌های اجتماعی توسط الگوریتم‌های مشخص کنترل می‌شوند و ما نحوه عملکرد آنها را می‌فهمیم ولی هیچ کس حتی افرادی که در حوزه ساخت و توسعه هوش مصنوعی مولد تحقیق می‌کنند و برنامه می‌نویسند، نحوه عملکرد دقیق مدل‌های زبان بزرگ (LLM) را نمی‌فهمد. آنها با نرخ تغییر و نوآوری بسیار متفاوتی تغییر می‌کنند و می‌توانند توسط افرادی با استانداردهای اخلاقی متفاوت از خودمان مورد استفاده قرار گیرند. هوش مصنوعی مولد نقاط ضعف مدل قدیمی حفظ حریم شخصی ما را نشان می‌دهد؛ وقت این است که ما بی‌فایده بودن آن را در حفظ حریم شخصی تشخیص دهیم و به سوی یک مدل جدید حرکت کنیم.

آنتروپولوژیست‌ها و دانشجو‌های حقوقی سال‌هاست که می‌دانند حریم شخصی نمی‌تواند توسط افراد کنترل شود، بخشی به دلیل اینکه ما اطلاعات را در داخل شبکه‌ها به اشتراک می‌گذاریم. به عبارت دیگر، مردم درباره یکدیگر صحبت می‌کنند، هم در دنیای مجازی و هم در دنیای واقعی. راهی آسان برای تعیین محدودیت‌ها وجود ندارد؛ می‌توانید از دوستان خود بخواهید که عکس‌های فرزندان شما را در اینستاگرام نگذارند یا از شما در تیک‌تاک یاد نکنند، اما حریم شخصی شما، هر چقدر هم که سعی نکنید، نقض می‌شود. نقض‌های حریم شخصی شبکه‌ای اغلب به دلیل این اتفاق می‌افتد که اطلاعاتی که در یک محیط با قوانین و انتظارات خاص ارائه شده‌اند به جاهای دیگر منتقل شده و به شکل متفاوتی تفسیر می‌شوند. تیک‌تاک‌های ساخته شده برای مخاطبان کویر (queer) به منبعی برای کمپین‌های ضد تغییر جنسیت تبدیل می‌شوند؛ سخنرانی‌های سیاسی ارائه شده برای مخاطبان همدرد زمانی که توسط مخالفان تماشای شوند، به محتوایی کاملاً بی‌منطق تبدیل می‌شود.

فناوری‌های جدید به طور فزاینده‌ای حریم شخصی شبکه‌ای را تهدید می‌کنند. به عنوان مثال، شجره نامه قانونی به پلیس این امکان را می‌دهد که با بررسی شواهد ژنتیکی جمع‌آوری شده از خویشاوندان دور اشخاص مظنون را شناسایی کند. شما می‌توانید انتخاب کنید که از Ancestry.com استفاده نکنید، اما نمی‌توانید از یک فامیل دور که احتمالاً حتی اسمش را هم نشنیدید جلوگیری کنید که همین کار را انجام دهد. داده‌های بزرگ، که از مجموعه‌های داده‌ای عظیم به همین روش‌ها استفاده می‌کنند، به طور متداول دوستان، خویشاوندان و حتی آشنایان دور را نیز به هم مرتبط می‌کنند، که وقتی وارد الگوریتم‌های پیش‌بینی پلیسی یا الگوریتم‌های ارزیابی ریسک می‌شود، مشکل‌آفرین می‌شود. افراد نمی‌توانند اقدامی برای جلوگیری از چنین تجاوزاتی به حریم شخصی بکنند.



هوش مصنوعی مولد نگرانی‌های حریم شخصی شبکه‌ای را تشدید می‌کند؛ از انجام هر گونه «کار شخصی»، روش‌ها و استراتژی‌هایی که برای حفظ حریم شخصی استفاده می‌کنیم، باز می‌دارد. و خروجی‌های هوش مصنوعی مولد تولیدی به طرقی کاملاً از منبع اصلی خود جدا می‌شوند که تا قبل از این تصور نمی‌شد. خود همین که پیام‌های شخصی به راحتی فاش می‌شوند مشکل ساز است، چه برسد به اینکه از کل محتوای ردیت (reddit) به عنوان یک داده برای شعرهای رباعی و مقاله‌های غیر حرفه‌ای دانشگاه استفاده شود.

تبعیض شیء ها و رسم مرزهای تصمیم گیری می پردازند. دسته دیگر که Generative ها هستند، وظیفه شان این است که یک داده ای را تولید کنند. برای اینکه این داده جدید تولید شود باید مدل هوش مصنوعی توانایی یادگیری یک توزیع احتمال یا توزیع یادگیری را یاد بگیرد. برای مثال انسان یک مدل Generative است، اگر از او خواسته شود تا نقاشی یک خانه را بکشد بدون آنکه خانه ای را ببیند می تواند آن را رسم کند لذا نقاش یک دانشی از شکل و شمایل یک خانه دارد. پس هدف این مدل ها یادگیری توزیع داده است. یک مثال دیگر برای مدل های Generative، مدل های GAN هستند که به طور غیرمستقیم در حال یادگیری توزیع احتمال اند اما در واقع قصد نمونه برداری یا Sampling از آن توزیع احتمال را دارند. حال سوالی مطرح میشود که چطور امکان دارد مدل یک توزیع را یاد بگیرد؟

ابتدا باید بررسی کنیم مفهوم توزیع چیست؟ دیدگاه آماری اشاره می کند که اگر بخواهیم توزیع احتمالاتی یک واقعه را به دست بیاوریم به این واقعه یک عدد نسبت می دهد که اگر n بار آزمایش این واقعه تکرار شود نسبت تعداد بار به وقوع پیوستن این واقعه به n احتمال آن است. حال اگر این فرایند برای هر واقعه انجام شود توزیع احتمال داده به دست می آید. مدل های هوش مصنوعی در بخش طبقه بندی تاکنون قادر به تخمین مقدار این احتمال بوده اند اما آیا امکانش هست که تابع توزیع احتمال را برای هر مقدار ورودی تخمین زد؟

تابع انرژی مفهومی است که در علوم مختلف مانند فیزیک، شیمی، و بیولوژی به کار می رود. این تابع به ما می گوید که یک سیستم یا شیء، چقدر انرژی دارد. در واقع، می توانیم آن را به عنوان "مقداری" که به اندازه گیری انرژی یک موجود یا سیستم کمک می کند، تصور کنیم. وقتی یک سیستم تغییر می کند، تابع انرژی می تواند به ما نشان دهد که چگونه انرژی درون آن تغییر می کند. تابع انرژی در شبکه های عصبی معمولاً به عنوان یک معیار برای اندازه گیری عملکرد شبکه استفاده می شود. این تابع به ما می گوید که شبکه چقدر در پیش بینی نتایج، درست عمل می کند. وقتی شبکه عصبی با داده های ورودی کار می کند، تابع انرژی مقداری مشخص را اختصاص می دهد که نشان می دهد چقدر پیش بینی های شبکه از واقعیت می تواند اختلاف داشته باشد. هدف ما در آموزش شبکه عصبی این است که تابع انرژی را به حداقل برسانیم، به این معنا که پیش بینی های شبکه به نتایج واقعی نزدیک تر شوند. برای راحتی کار در پیاده سازی، تابع توزیع را به صورت یک تابع نرمال انرژی تبدیل می کنیم. نکته قابل توجه می تواند این باشد که به جای تابع توزیع در فرمول، می توانیم یک شبکه عصبی را بگذاریم.

این فرمول مشهور که از توزیع احتمال به نام "تابع توزیع احتمال گشتی" (Gibbs distribution) مشتق شده است، به ما نشان می دهد که انرژی یک شبکه عصبی به عنوان توانایی

Score Matching

در هوش مصنوعی

تکنیک Score Matching یک ابزار قدرتمند در آمار و احتمالات است که به ما کمک می کند تا داده ها را بهتر درک کرده و مدل های احتمالاتی را بهینه تر سازیم. این تکنیک به ما این امکان را می دهد که با تطابق توزیع های احتمالاتی مدل های مختلف به داده های مشاهده شده، مدل مناسبی را ایجاد کنیم. اهمیت Score Matching در این است که به

ما این قدرت را می دهد تا به طور دقیق تر و بهبود یافته تری توزیع های احتمالاتی را برای داده ها مدل کنیم، که این موضوع در مسائلی مانند یادگیری ماشین، تحلیل داده ها، و پیش بینی احتمالی پدیده ها بسیار مهم است. به طور ساده، Score Matching به ما امکان می دهد داده های پیچیده را بهتر درک کنیم و مدل هایمان را بهبود بخشیم، که این امر به تحقیقات بیشتر و بهترین استفاده از داده های ما منجر می شود.

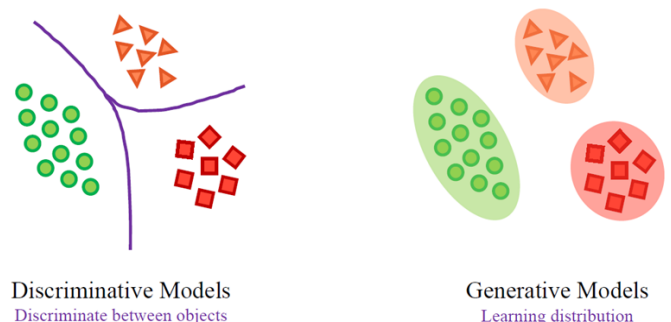


پویا ارده خانی

دانشجو مهندسی کامپیوتر
دانشکده فاریاب دانشگاه تهران
pouya.ardehkhani@ut.ac.ir

برای فهم بهتر این تکنیک، ابتدا مدل های هوش مصنوعی را باید به دو گروه تقسیم کنیم. گروه اول Discriminative ها و گروه دوم Generative ها هستند.

Models: Discriminative vs. Generative

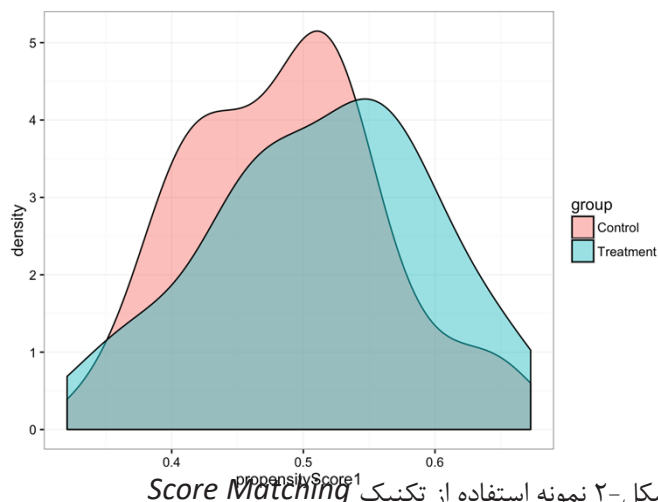


Discriminative Models
Discriminate between objects

Generative Models
Learning distribution

شکل-۱

Discriminative به معنای جدا کردن دو یا چند چیز از هم دیگر است، به عبارتی دیگر، انگار بین شیء ها یک مرز مشخصی را تعیین میکنند. در اصل این مدل ها تنها به



شکل ۲- نمونه استفاده از تکنیک Score Matching

در این تکنیک شما تلاش می‌کنید تا تغییرات در نحوه عملکرد تابع توزیع را درک کنید. امتیازها (Scores) معیاری هستند که به شما کمک می‌کنند این تغییرات را بسنجید. شما مقدار امتیازها را بین داده‌های واقعی و داده‌های تخمینی مدل مقایسه می‌کنید. اگر امتیازها مشابه باشند، تخمین مدل هوش مصنوعی خوب است. مدل نیز یاد می‌گیرد که چگونه توزیع داده‌ها را بهتر تخمین بزند. این تخمین باعث می‌شود که یک تخمین از تابع توزیع از داده‌ها به وجود آید همانند آنچه در شکل ۲ می‌بینید. این تکنیک به مدل‌های هوش مصنوعی کمک می‌کند تا بدون نیاز به محاسبات پیچیده، توزیع متغیرهای تصادفی را بهتر بفهمد و در مسائل مختلفی استفاده و نمونه سازی کند.

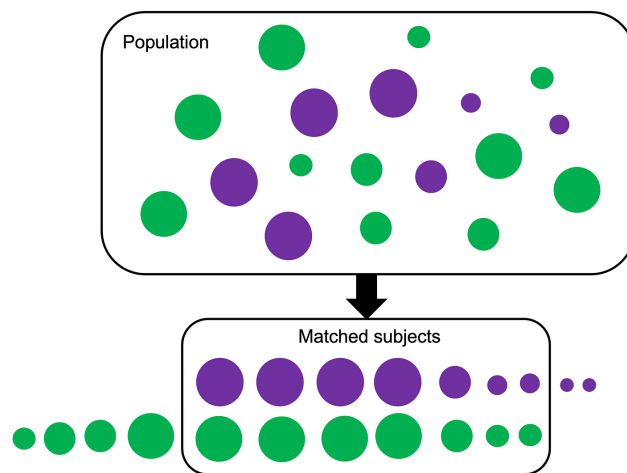
استفاده از تکنیک Score Matching در شبکه‌های عصبی بسیار اهمیت دارد. یکی از اصولی‌ترین مسائل در آموزش شبکه‌های عصبی، تنظیم پارامترها به منظور بهبود کارایی مدل است. این تکنیک به ما امکان می‌دهد توزیع ویژگی‌ها در داده‌ها را به دقت تخمین بزنیم، بدون نیاز به محاسبه مستقیم توزیع. با داشتن این اطلاعات، می‌توانیم در فرآیند آموزش شبکه‌ها، از روش‌های بهبود کارایی مانند روش‌های بهینه‌سازی تطبیقی بهره‌وری کنیم تا شبکه بهبودیافته‌ای را ایجاد کنیم. این می‌تواند منجر به آموزش سریع‌تر و دقیق‌تری در شبکه‌های عصبی شود.

علاوه بر این، Score Matching به ما کمک می‌کند توزیع‌های احتمالی پشت مشتقات ویژگی‌ها را تخمین بزنیم. این اطلاعات می‌توانند در فرآیند انتخاب ساختار مدل و مدیریت مشکلات آموزشی مانند مشکل کاهش گرادینت‌ها (vanishing/exploding) کمک کنند. در نتیجه، استفاده از تکنیک Score Matching به افزایش عملکرد و پایداری شبکه‌های عصبی کمک می‌کند و می‌تواند در تعمیق درک ما از رفتار شبکه‌ها و بهبود توانایی آن‌ها در مسائل مختلف مفید باشد.

شبکه در توجه به داده‌ها و نمونه‌های ورودی را اندازه‌گیری می‌کند. در واقع، این فرمول نسبت توزیع احتمالی به انرژی را نشان می‌دهد و با تعیین مقادیر مناسب برای ثابت عدد، ما می‌توانیم توزیع احتمالی را به انرژی متناظر با آن مرتبط کنیم. این تبدیل از توزیع به انرژی به ما کمک می‌کند تا برای مسائل مختلفی مانند یادگیری ماشینی و پردازش سیگنال، مدل‌های بهتری ایجاد کنیم و داده‌های ورودی را بهبود ببخشیم.

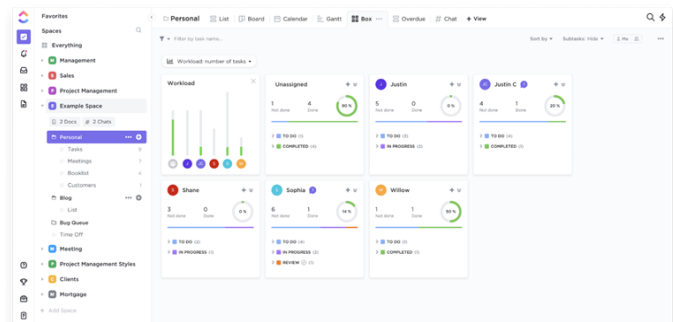
همچنین این فرمول ارتباطی را نمایان می‌کند که در شبکه‌های عصبی برای محاسبه انرژی یک وضعیت خاص استفاده می‌شود. در اینجا، "انرژی تابع" نشان‌دهنده مقدار انرژی مرتبط با وضعیت خاصی در شبکه عصبی است. این مقدار بر اساس خروجی شبکه برای آن وضعیت و تقسیم بر یک ثابت عدد محاسبه می‌شود. از تابع انرژی برای ارزیابی کیفیت پیش‌بینی‌های شبکه عصبی استفاده می‌شود، به طوری که مقدار کمتر انرژی به نتایج دقیق‌تر نزدیک‌تر است. این فرمول به ما کمک می‌کند تا عملکرد شبکه را ارزیابی کنیم و در صورت لزوم، شبکه را بهبود دهیم.

حال یک سوال دیگر مطرح می‌شود که آیا این عدد ثابت به راحتی قابل به دست آوردن است؟ خیر، محاسبات این عدد به قدری زیاد است که به نظر مشکل می‌آید. پس باید به تخمین زدن این عدد بپردازیم. اینجا تکنیک Score Matching مطرح می‌شود.

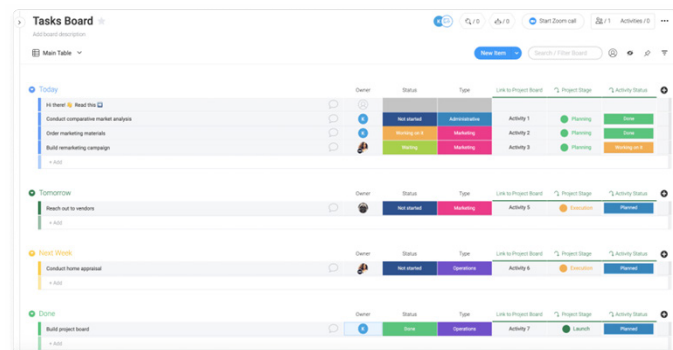


ایده اصلی پشت تکنیک Score Matching این است که می‌خواهد توزیع یک متغیر تصادفی را بدون نیاز به محاسبه مستقیم توزیع آن تخمین بزند. به جای محاسبه توزیع، این تکنیک از امتیازها (Scores) استفاده می‌کند که نشان‌دهنده تغییرات تابع توزیع مورد نظر در مورد پارامترهای آن است. با مقایسه امتیازها بین داده‌های مشاهده شده و داده‌های تولید شده از مدل تخمینی، می‌توان توزیع مورد نظر را تخمین زد. در متغیر تصادفی و احتمالات، امتیاز (Score) به تغییرات لگاریتم توزیع احتمالی متغیر تصادفی نسبت به پارامترهای آن توزیع اشاره دارد. به عبارت دیگر، امتیاز نشان‌دهنده تاثیر تغییرات پارامترهای توزیع بر توزیع احتمالی متغیر تصادفی است.

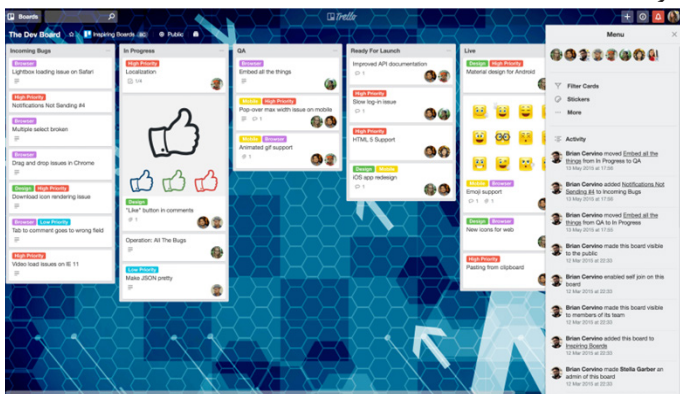
و به ندرت برای کارهای شخصی استفاده می‌شود. این برنامه به Monday.com شباهت زیادی دارد اما در کنار آن دارای ویژگی‌های برجسته‌ای مانند whiteboard است که برای کارهای تیمی کاربرد زیادی دارد و همچنین automation آن نسبت به Monday.com ساده تر می‌باشد. این برنامه را می‌توانید به صورت رایگان دانلود و استفاده کنید اما اشتراک‌های دیگر این برنامه قابلیت‌های بیشتری به شما می‌دهند که می‌توانند در مدیریت پروژه‌های تیم‌های بزرگتر به شما کمک کنند.



Monday.com: این برنامه کاملاً شبیه به Clickup است و تنها قابلیت‌های ویژه آن dashboard تمیزتر و دقیق‌تر و آنالیز بهتر داده‌ها و فعالیت‌های هر یک از افراد تیم و پروژه است. نکته‌ای که شاید برای برخی از کاربران اهمیت داشته باشد، استقرار این شرکت در اسرائیل است. اگر با این موضوع مشکل دارید می‌توانید از نرم‌افزار Clickup استفاده کنید.



Trello: تقریباً تمام قابلیت‌های نوشتن در این نرم‌افزار موجود است اما با customizability کمتر و فضای کار راحت‌تر. اگر فکر می‌کنید کار با نوشتن خیلی سخت است و یا به جای اینکه وقتتان را صرف انجام کارهای شخصی یا تیمتان کنید، مشغول کار با خود نرم‌افزار هستید، پیشنهاد میکنم از Trello استفاده کنید. البته در نظر داشته باشید که Trello قابلیت‌های کمتری نسبت به نوشتن دارد و ممکن است بعضی کارها با نوشتن بهتر مدیریت شوند.



معرفی نرم افزار: Notion

دانشگاه‌ها شروع شده‌اند و باید کم‌کم برای پروژه‌ها و تکالیف دانشگاه زمان مشخصی بگذارید تا روی هم جمع نشوند. ممکن است در کنار دانشگاه، کارهای دیگری هم داشته باشید که اولویت آنها کمتر از دانشگاه نباشد.



علی رادمرد

دانشجو مهندسی کامپیوتر
دانشکده فابریک دانشگاه تهران
aliradmard5@gmail.com

برای مدیریت بهتر این فعالیت‌ها، می‌توانید از نرم‌افزارهای مدیریت زمان و برنامه استفاده کنید که علاوه بر برنامه ریزی کارهای شخصی، برای فعالیت‌ها و برنامه‌های تیمی هم به کار می‌روند. این گونه برنامه‌ها در مبنا شبیه به هم هستند ولی چیزی که یک برنامه را برجسته می‌کند، قابلیت‌های منحصر به فرد آن است.

نرم‌افزارهای مشابه و تفاوت‌های آن‌ها
نرم‌افزارهای زیادی در این حوزه وجود دارند که Notion، ClickUp، Monday.com و Trello از معروف‌ترین آن‌ها هستند.

Notion: از این نرم‌افزار برای یادداشت برداری، مدیریت فعالیت‌ها و برنامه‌ریزی پروژه‌های شخصی و تیمی استفاده می‌شود. Notion قابلیت شخصی سازی بالایی دارد و قوه تخیل، تنها مانع شماسست؛ از طراحی پایگاه‌های داده و همگام‌سازی کردن با برنامه‌های معروف مثل Google Calendar و ... گرفته تا فرمول نویسی و کار با git. از نقاط قوت این برنامه می‌توان به مدیریت کارهای تیمی به صورت همزمان، کامنت نویسی و استفاده از template های آماده شده توسط کاربرهای دیگر اشاره کرد. با دیدن این قابلیت‌ها ممکن است فکر کنید که کار کردن با نوشتن خیلی سخت است و یا اینکه customize کردن را به انجام task ها و پروژه‌ها اولویت دهید، ولی همه این موارد بستگی به خود شخص و سطح تمرکزش روی هدف اصلی اش دارد. تفاوت اساسی نوشتن با بقیه برنامه‌های مشابه، رایگان بودن آن است اما اگر قصد دارید کاری حساس تر و در مقیاس بالاتری انجام دهید، می‌توانید به payment plans های داخل وبسایت نوشتن مراجعه کنید. این برنامه در پلتفرم های ویندوز، مک، وب، اندروید و iOS قابل استفاده است.

Tasks & Issues

Click on a card to view details.
Click on 'Board by Status' to toggle between views.

Name	Team	Assignee	Status	Options
Button blue hover color	Design	Zeno Wu, Leslie Jensen		
Desktop push notifications	Engineering	Chet Corcos		
Add avatars	Design	Garrett Fidalgo	In Progress	
Fix keyboard shortcuts	Engineering	Andrea Lim	Next Up	January 3,
Offline mode polish	Engineering	Garrett Fidalgo	In Progress	January 8,
Rename archive to trash	Engineering	Jenne Nguyen	Completed	

Clickup: این نرم‌افزار ابزاری برای مدیریت فعالیت‌ها، برنامه ریزی پروژه‌های تیمی و مدیریت هدف می‌باشد. محوریت عملکرد این برنامه، کارهای تیمی است

Trello، Github و ... دارند. این انعطاف پذیری به کاربران اجازه می‌دهد تا صفحاتی پویا و تعاملی ایجاد کنند.

پایگاه‌های داده رابطه‌ای:

نوشن به کاربران اجازه می‌دهد تا پایگاه‌های داده ایجاد و به یکدیگر متصل کنند. این قابلیت نوشن را به یک ابزار قدرتمند برای مدیریت پروژه، سازماندهی محتوا و پیگیری داده‌ها تبدیل کرده است. شما می‌توانید روابط بین داده‌ها را ایجاد کنید، داده‌ها را فیلتر و مرتب کنید و نمایه‌های سفارشی بسازید.

تمپلت‌ها و پیش‌تعریف‌ها:

نوشن تمپلت‌ها و پیش‌تعریف‌های گسترده‌ای برای موارد مختلف ارائه می‌دهد که می‌توانند به عنوان نقطه شروع برای ایجاد صفحات یا پایگاه‌های داده جدید استفاده شوند.

تخته‌کانبان:

نوشن یک نمای تخته‌کانبان ارائه می‌دهد که مدیریت وظایف و پروژه‌ها را آسان می‌کند. این قابلیت برای تیم‌ها و مدیریت پروژه‌ها بسیار مفید است و با روش agile تفاوت دارد.

ادغام با رسانه:

نوشن از ادغام (embed) رسانه‌ها از جمله تصاویر، ویدیوها و فایل‌های دیگر مستقیماً پشتیبانی می‌کند.

همکاری و به اشتراک گذاری:

نوشن امکان همکاری در زمان واقعی در اسناد و پایگاه‌های داده را فراهم می‌کند. این قابلیت به تیم‌ها اجازه می‌دهد که بدون هیچ مشکل و تعللی با هم کار کنند. همچنین کنترل‌های مجوز دقیقی برای به اشتراک گذاری با افراد، تیم‌ها یا عموم فراهم می‌کند.

دسترسی آفلاین:

نوشن دسترسی آفلاین به محتوای شما را فراهم می‌کند تا حتی زمانی که به اینترنت متصل نیستید، بتوانید از برنامه استفاده کنید. این ویژگی برای کاربرانی که به داده‌های خود حتی در حال حرکت هم نیاز دارند، بسیار مفید است.

سازگاری با چند پلتفرم:

نوشن در چندین پلتفرم مانند وب، دسکتاپ (Windows – macOS) و موبایل (Android – iOS) در دسترس است.

جستجو و فیلترهای پیشرفته:

نوشن برای دسترسی راحت‌تر و دقیق‌تر به اطلاعات حتی در پایگاه‌های داده بزرگ، قابلیت جستجو و فیلتر کردن پیشرفته را ارائه داده است تا بتوانید بر اساس پروژه، تاریخ، شخص مسئول، درصد تکمیل، تگ و ... به راحتی اطلاعات مورد نظر را پیدا کنید.

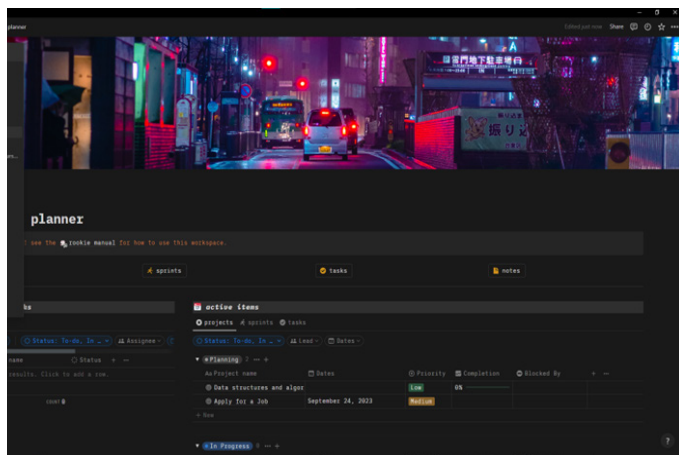
Public API:

نوشن با ارائه Public API، این امکان را به کاربران می‌دهد تا از این برنامه در کنار ابزارهای دیگر، به راحتی استفاده کنند و همچنین هوشمندسازی (automation) را راحت‌تر می‌کند.

قابلیت‌های نوشن و آشنایی با محیط کار

نوشن نسبت به بقیه برنامه‌های مدیریت زمان، قابلیت‌های بیشتری دارد که ممکن است یادگیری نحوه کار با برنامه را کمی سخت کنند. اما خوشبختانه نوشن راه‌های آسان‌تری را هم برای کار کردن با برنامه در اختیار کاربران گذاشته است. اخیراً نوشن با معرفی هوش مصنوعی‌اش، قابلیت‌های خود را گسترش داده است که از جمله این توانایی‌ها می‌توان به خلاصه نویسی، ترجمه، درست کردن گرامر، عوض کردن tone گفتار و ... اشاره کرد. یکی دیگر از ویژگی‌هایی که نوشن را برجسته می‌کند، تولید پایگاه‌های داده و ربط دادن این پایگاه‌های داده به یکدیگر است که در پروژه‌های تیمی بسیار پرکاربرد است.

مهم‌ترین ویژگی نوشن، template هایی است که هم داخل خود برنامه وجود دارند و هم آن‌هایی که توسط کاربران ساخته شده‌اند که به دو صورت رایگان و پولی در اختیار عموم قرار داده شده‌اند. اگر قصد دارید از این template ها استفاده کنید، می‌توانید در youtube کلمه نوشن را سرچ کنید تا افرادی که این template ها را ساخته‌اند پیدا کنید و از این قابلیت استفاده کنید.



برای اینکه بیشتر با فضای کاری نوشن آشنا شوید، به شما پیشنهاد می‌کنم که حتماً یک بار نوشن را نصب کرده و یک تمپلت مناسب پیدا کنید. این تمپلت از کانال Cajun Koi Academy در یوتیوب برداشته شده که می‌توانید آن را با یک جستجو پیدا کنید.

به طور کلی قابلیت‌های نوشن شامل:

محیط کار انعطاف‌پذیر:

نوشن یک ابزار چند منظوره برای بهبود سازماندهی، مدیریت پروژه و به اشتراک گذاری دانش است که فضایی را به کاربران تخصیص می‌دهد و آنها می‌توانند این فضا را بر اساس نیازهای خود سفارشی کنند. این برنامه به تمپلت‌های از پیش تعریف شده محدود نیست و امکان خلاق بودن را به کاربران می‌دهد تا بر اساس سلیقه‌شان، فضای مورد نظر خود را بسازند.

بلوک‌ها:

نوشن از رویکرد ماژولار "بلوک‌ها" استفاده می‌کند که انواع مختلفی مانند متن، جدول، پایگاه‌های داده، Task list، Code، File، صوت، تصویر، فیلم، فرمول، embed کردن پی‌دی‌اف، گوگل درایو، فیگما، Jira، Gitlab،



سید محمد جعفری
زینب قدوسی زاده
سید علی فقیه موسوی

خلاصه داستان

ذهن خود را گرم کنید

ساختن یک فیلم ترسناک آن هم بدون وجود قاتلی زنجیره‌ای که پس از کشتن قربانی هایش، انگشتانشان را برای یادگاری باخود می‌برد، شاید به نظر غیرممکن برسد. صنعت فیلم ترسناک این روزها برای همه ما تکراری و خسته کننده شده است. اما خبر خوب این است که با دیدن این اپیزود از سریال Black Mirror، تمامی احساسات ترس، انزجار، ظلم، بی اخلاقی و... را به میزان خوبی تجربه می‌کنید. اعصاب خود را آماده کنید که قرار است کارگردان با آن بازی کند. کش و قوس‌ها، اتفاقات، انسان‌ها همه و همه آماده‌اند تا ذهن شما را به چالش بکشند.

آیا چیزی به نام واقعیت وجود دارد؟

آیا وجود ما به بعد فیزیکی‌مان خلاصه می‌شود؟ یا اینکه خاطرات‌مان و تفکرات‌مان ما را تعریف می‌کند؟ حالا فکر کنید که از خاطراتتان یک کپی تهیه شود، که این کپی تعدادی خط کد است، آیا می‌توانید بگویید که او نیز به مانند شماست؟ آیا به بردگی کشیدن او، و استفاده از او به عنوان یک نرم افزار ظالمانه است؟ او می‌تواند ببیند، بشنود، اثری ایجاد کند، حتی می‌تواند ترس و درد را هم تجربه کند! به راستی چه چیزی باعث می‌شود که بین خودمان و این کپی تمایز قائل شویم؟ مگر ما هم تکه‌ای کد ژنتیکی نیستیم؟ قضاوتش بر عهده خودتان.

روز به روز، تکنولوژی‌های جدیدتری وارد زندگی ما می‌شوند. ممکن است شما نیز بر این عقیده باشید که سود این تکنولوژی‌ها بر آفاتش غلبه می‌کند، یا اینکه نقض حریم شخصی و اطلاعات شخصی، بزرگترین قیمتی است که باید برای استفاده از این پیشرفت‌های عظیم بپردازیم؛ ولی Black Mirror آمده تا از ما بخواهد یکبار دیگر این مقایسه را انجام دهیم، ولی این بار با عینک اخلاق؛ یکبار هم که شده از «من» فاصله بگیریم و به «ما» بپردازیم، ببینیم این پیشرفت چه عواقبی برای جامعه ما به همراه دارد. درک بالای سازندگان اثر از روان و تفکر انسان، نبوغ سینمایی به کار رفته و مهم‌تر از همه سوالات مهم فلسفی و وجودی که اثر مطرح می‌کند، مجموعه‌ای بی‌مثل و نوین را به ارمغان آورده است؛ اثری که شاید از معدود مواردی باشد که باعث شده نتفلیکس همچنان چیزی برای رو کردن داشته باشد.

کریسمس سفید اولین اپیزود ویژه‌ی ۹۰ دقیقه‌ای سریال Black Mirror است و تمام تلاشش را به کار می‌بندد تا در این ۹۰ دقیقه، بیشتر از حد معمول، ما را از زندگی سیر کند. به جای اینکه مثل همیشه یک خط داستانی را دنبال کنیم، سه مسیر را دنبال می‌کنیم که همه در نهایت به یکدیگر متصل شده و به سرانجامی خوفناک منجر می‌شوند.

Black Mirror

White Christmas

کریسمس سفید

IMDB: ۹,۱/۱۰ Rotten Tomato: ۹۰/۱۰۰

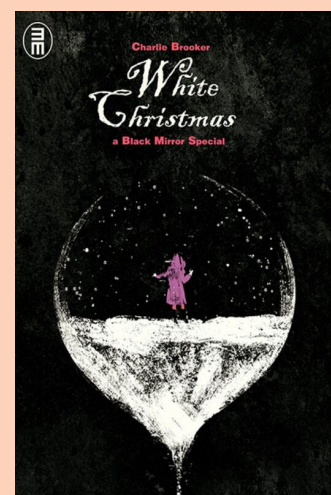
ژانر: درام، رازآلود، هیجان انگیز

کارگردان: کارل تیبتس Carl Tibbets

بازیگران اصلی: جان هم Jan Hamm، ریف اسپال Rafe Spall، اونا چاپلین Oona Chaplin



قیزر رسمی فیلم





نقاط منفی:

دور بودن از واقعیت:

پرداختن به موضوع‌های از این دست که با تکنولوژی سر و کار دارند مخصوصاً تکنولوژی‌هایی که هنوز به بهره‌برداری عمومی نرسیده‌اند کار آسانی نیست. همان قدر که این موضوع می‌تواند یک فیلم را خاص کند و مخاطب را به وجد بیاورد، می‌تواند ارتباط مخاطب با فیلم را هم قطع کند. تکنولوژی‌هایی که در این قسمت نشان داده شده بود، همتایی در زندگی روزمره ما ندارند. در واقع نزدیک‌ترین نمونه‌ی آن، بلاک یا مسدود شدن توسط یک فرد در شبکه‌های اجتماعی است. تفاوت شدت این دو موضوع بسیار زیاد است، به طوری که به مرور زمان و پیش رفتن فیلم، ارتباط بین این دو موضوع کمرنگ و کمرنگ‌تر میشود. در مقایسه با دیگر قسمت‌های این سریال، این تفاوت بیشتر محسوس میشود.

قیف از کجا آمد؟

با پیشروی فیلم، متوجه نقص‌هایی در یکپارچگی داستان میشویم: برای مثال، در داستان اول هنگام مرگ دوست «مت»، زهر بواسطه یک قیف به او خورانده شد، ولی گویا این قیف بصورت معجزه‌آسایی در صحنه قرار گرفت، زیرا که پیش از استفاده، هیچ اثری از آن نبود. برای مثال دیگر، در انتهای فیلم، هنگامی که «مت» توسط پلیس در لیست افراد خطرناک برای جامعه قرار می‌گیرد، برایمان سوال میشود که او چگونه دارد افراد سازمان پلیس را می‌بیند و چرا آن‌ها برایش مسدود سازی نشده‌اند؟ یا اینکه او چگونه قرار است لوازمات زندگی خود را بدون ارتباط داشتن با افراد جامعه فراهم سازد؟ چگونه باید غذا و پوشاک تهیه کند؟ سوالاتی از این قبیل، با اینکه کم بودند و خیلی حائز اهمیت نیستند، اما در برخی از صحنه‌ها بی پاسخ ماندند.

«پاتر» خنده بر لب داشتند. در این بین فقط «پاتر» بود که با این کپی‌ها احساس همدردی کرد. شاید بخاطر این بود که او واقعا انسان بدی نبود، یا حداقل بدتر از دیگران نبود.

استفاده از روش‌های مبتکرانه برای انتقال مفاهیم: وارد نکردن جزئیات اضافه مانند روش کار تکنولوژی‌های مورد بحث، یکی از خصوصیات سریال Black Mirror است. این قسمت هم استثنا نیست. این فیلم برای نشان دادن مفهوم بلاک (مسدودسازی) از روشی بسیار جالب استفاده کرده است. با زدن دکمه بلاک، هرگونه تعامل بین افراد غیر ممکن میشود و آن دو نفر دیگر قادر به دیدن و شنیدن یکدیگر نیستند و حتی اگر بخواهند هم دیگر نمی‌توانند معذرت خواهی کنند. با همه این‌ها، شاید جالب‌ترین نکته این باشد که ممکن است دیگر نتوانند همدیگر را ببینند، یا به هم پیام دهند، اما ابتدایی‌ترین نوع انتقال اطلاعات همچنان کارآمد است؛ منظور همان مداد و کاغذ است. ما که خود تا وقتی «پاتر» صحبت از نامه‌نگاری نکرد، اصلاً این روش به ذهنمان خطور هم نکرد، و این نشان دهنده اتکای بیش از حد بر تکنولوژی برای ارتباط با یکدیگر است، ارتباطی که اکنون بسیار سست و شکننده به نظر می‌رسد.

مرگ تدریجی حریم شخصی:

نکته مهم دیگری که فیلم به آن اشاره می‌کند، تعرض بیش از پیش تکنولوژی به حریم شخصی ماست. البته که این قسمت به صراحت قسمت‌های دیگر مانند «Crocodile» به این بحث اشاره نکرده، اما به صورت زیرپوستی این نکته مهم را گوشزد می‌کند. چرا باید برای اعتراف‌گیری از کپی شخص استفاده کرد؟ اصلاً این کار درست است که شما بدون اجازه فرد از او کپی تهیه کنید؟ یعنی این کپی در این حد «انسان» است که اعتراف آن به منزله اعتراف یک انسان واقعی باشد و باعث مجازات یک انسان شود؟ اگر اینطور باشد، محبوس کردن او بردگی نیست؟

قضایات با خودتان:

انتهای فیلم و آنچه شخصیت‌های داستان در نهایت از سر می‌گذرانند واقعا شما را متعجب می‌کند که چقدر سزاوار مجازات‌هایی هستند که دریافت می‌کنند؟ فیلم تصمیم‌گیری نهایی در مورد درستی یا نادرستی اتفاقات را بر عهده خودتان می‌گذارد. آیا یک کد کپی شده از ذهن شما سزاوار دلسوزی را دارد؟ آیا قطع کردن ارتباط یک فرد با تمامی جهان اطراف (حتی اگر توسط قانون پذیرفته شود) اخلاقی است؟

نقطه اوج فیلم از بین بردن مرز بین واقعیت و مجاز در ذهن ماست. به گونه‌ای که ذهن به طور ناخودآگاه برای دختر داخل کوکی ناراحت می‌شود و تحت تأثیر قرار می‌گیرد در حالی که در واقع آن تنها یک کد است و وجود انسانی ندارد. اگرچه بخش داستانی فیلم نتوانسته آنچنان که باید مخاطب را با خود درگیر کند ولی با وارد کردن تکنولوژی این خلا، به نوعی پوشانده شده و شاهکار کارگردان همین از بین بردن مرز بین دو دنیا در ذهن ماست. کوکی‌ها ساختارهای ذهن مخاطب را می‌شکنند و ذهن او را به چالش می‌کشند.

ریسمان مستحکم داستان:

فیلم در مجموع داستان خطی‌ای داشت، با اینکه از فلش‌بک استفاده شده بود، اما فلش‌بک‌ها نیز خود ساختار خطی داشتند. این فلش‌بک‌ها بخش اصلی داستان را تشکیل می‌دادند، و طوری طراحی شده بودند که در ابتدا به نظر بیاید که ارتباط چندانی با هم ندارند، و صرفاً جهت معرفی شخصیت‌ها به کار گرفته شده‌اند، اما هرچه به پایان فیلم نزدیک شدیم، دیدیم که داستان همگی حول محورهای خاصی می‌چرخند. فیلم در مواقع متعدد به بیننده نشانه‌هایی میدهد، که باعث می‌شود بیننده نیز به اتفاقاتی که در حال افتادن است، مشکوک شود؛ از اسم فیلم گرفته - که روز رقم خوردن جنایت را اعلام می‌کند - تا لحظات عجیبی مانند لحظه‌ای که «پاتر» یادش نمی‌آید در این کلبه برای چه کاری آمده است؛ و ساعت دیواری که نهایتاً همه چیز را به هم گره می‌زند.

کشمکش درونی «پاتر»:

پیش‌تر هم اشاره کردیم که داستان خوب، به هیچ وجه باعث ضعف زیبایی سینمایی نشده. کارگردان از پالت رنگی و نور با تبحر بالا استفاده کرده. مثلاً هنگامی که حرف زدن «مت» و «پاتر» را مشاهده می‌کنیم، نور آفتاب فقط نصف چهره پاتر را می‌پوشاند، این امر نشان دهنده درگیری درونی «پاتر» است و تصویری از روح آزرده او را به نمایش می‌گذارد. نکته جالب‌تر اینکه انگار «پاتر» تنها فرد داستان است که اندکی انسانیت در او باقی مانده است؛ «مت» به صورت روزانه به ساختن برده‌های تخم مرغی می‌پرداخت، و مشتریان هم بدون سوالات چندانی این امر را می‌پذیرفتند. حتی کارآگاه‌های پلیس هم هنگام شکنجه دادن کپی

نگاهی بر اخبار دنیای صفر و یک

در دو ماه گذشته

گوگل سرچ ۲۵ ساله شد: گوگل سرچ ۲۵ سال پیش در چنین روزی راهاندازی شد. این محصول گوگل از زمان راهاندازی، زندگی میلیاردها نفر در سراسر جهان را تغییر داده است. گوگل برای جشن تولد گوگل سرچ، ویدیویی کوتاه و همچنین یک Doodle به همراه شماره ۲۵ منتشر کرده است.



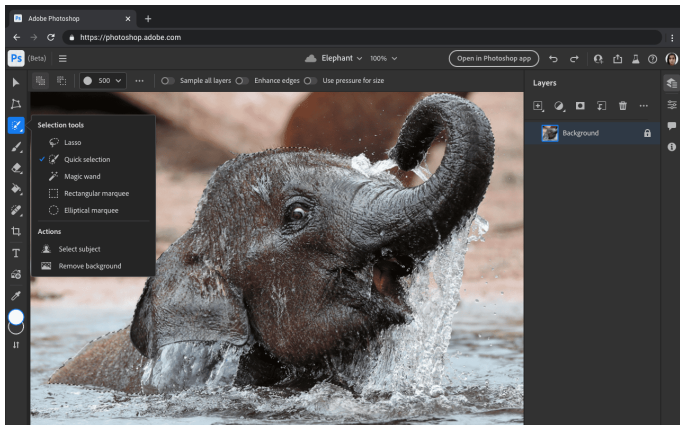
شرکت هوش مصنوعی xAI که امسال توسط ایلان ماسک بنیان‌گذاری شد خبر از آماده‌سازی یک هوش مصنوعی مولد جدیدی بنام Grok AI داده است: طبق گفته تیم دست‌اندرکار این هوش مصنوعی، Grok AI پاسخ برخی از سؤالاتی که هوش‌های مصنوعی مشابه از پرداختن به آن‌ها خودداری می‌کنند را نیز می‌دهد و نسبتاً ابزار باج‌رأت‌تری محسوب می‌شود. در حال حاضر تست بتای این محصول در حال انجام است.

پایان کار اپلیکیشن Google Podcasts در سال ۲۰۲۴: کاربران به یوتیوب موزیک مهاجرت کنند گوگل قصد دارد تا سال ۲۰۲۴ برنامه Google Podcasts را تعطیل کرده و به‌عنوان بخشی از برنامه مهاجرت به سرویس YouTube Music، کاربران را به این برنامه هدایت کند. اخیراً این شرکت اعلام کرد که امکانی را به پادکست‌ها اضافه می‌کند تا کاربران فیدهای RSS را تا پایان سال ۲۰۲۳ در یوتوب آلود کنند.



منا از هوش مصنوعی تصویرساز Emu رونمایی کرد: منا از یک هوش مصنوعی جدید برای ساخت تصاویر به‌نام Emu رونمایی کرده است که می‌خواهد با میدج‌رنی و DALL-E رقابت کند. این ابزار سرعت بالایی دارد و در مدت زمان کوتاهی می‌تواند تصاویر مدنظر کاربران را بسازد.

ادوبی نسخه تحت وب فتوشاپ را با ابزارهای هوش مصنوعی برای عموم کاربران عرضه کرد: پس از نزدیک به دو سال، بالاخره ادوبی نسخه وب فتوشاپ را از حالت بتا خارج کرد و حالا عموم کاربران به آن دسترسی دارند. این نسخه از ابزارهای هوش مصنوعی مانند Generative Fill و Generative Expand بهره می‌برد. این ابزارها از بیش از ۱۰۰ زبان از جمله فارسی پشتیبانی می‌کنند و تصویر نهایی نورپردازی و پرسپکتیو یکسان با عکس اصلی خواهد داشت. نسخه تحت وب فتوشاپ بخشی از طرح‌های پولی با حداقل هزینه ماهانه ۹۹.۹ دلار محسوب می‌شود و به صورت رایگان در اختیار کاربران قرار نمی‌گیرد.



معرفی سری جدید کامپیوترهای شرکت Apple مجهز به تراشه‌های M۳، M۳ Pro، M۳ Max: در ادامه موضوع تراشه‌ها، کوالکام نیز با تراشه‌ی Snapdragon X Elite به جنگ تراشه‌های پر قدرت اپل خواهد رفت. این تراشه از پردازنده Qualcomm Oryon CPU بهره می‌برد و طبق گفته کوالکام، پر قدرتمندترین تراشه‌ای است که این شرکت تا به حال برای یک کامپیوتر ساخته است. باید ببینیم از بین اپل و کوالکام، کدام‌یک سربلند از این رویارویی خارج خواهد شد.

سری ریلمی ۱۱ پرو در ایران معرفی شد: شرکت ریلمی گوشی‌های سری ریلمی ۱۱ پرو را در ایران معرفی کرد. این رویداد در تاریخ یکشنبه ۲۱ آبان و در محل سالن کوه نور هتل پارسین اوین برگزار شد. ریلمی ۱۱ پرو پلاس میان‌رده‌ای جذاب از ریلمی محسوب می‌شود که با دوربین اصلی ۲۰۰ مگاپیکسلی می‌تواند تجربه لذت‌بخشی از عکاسی را در اختیار شما بگذارد. این گوشی در حال حاضر حدود ۱۷ میلیون تومان قیمت دارد.



مایکروسافت ظاهراً آفیس ۲۰۲۴ را رایگان عرضه می‌کند: فروش کامل نرم‌افزار بدون نیاز به اشتراک پولی. مایکروسافت در یک اقدام غافلگیرکننده، قصد دارد سال آینده نسخه مستقل و لایسنس دائمی آفیس ۲۰۲۴ را منتشر کند. پیش‌نمایش این نسخه هم‌اکنون از طریق کانال «DevMain» قابل دسترسی است و کارمندان مایکروسافت در حال آزمایش آن هستند. این نرم‌افزار احتمالاً در سه‌ماهه سوم یا چهارم سال ۲۰۲۴ منتشر خواهد شد، و از مدل فروش اشتراک پولی خبری نخواهد بود.



اپل «سیری» را با هوش مصنوعی متحول می‌کند: اپل در حال به‌روزرسانی Siri با یک مدل زبانی بزرگ (LLM) است. اپل دستیار هوشمند خود را با یک LLM اختصاصی و نه LLM شرکت‌های دیگر مانند Llama ۲ شرکت متا، به‌روزرسانی می‌کند. انتظار می‌رود نسخه جدید سیری سال آینده میلادی در رویداد WWDC ۲۰۲۴ معرفی و به‌صورت استاندارد روی آیفون ۱۶ و نسل‌های بعدی آیفون نصب شود.

یوتوب استفاده از برچسب روی محتواهای خلق‌شده با هوش مصنوعی را اجباری می‌کند: یوتوب طی ماه‌های آینده از محتواسازان می‌خواهد استفاده از هوش مصنوعی در ویدیوهای خود را مشخص کنند که با وجود آن، یک برچسب روی این دسته از ویدیوها نمایش داده خواهد شد. یوتوب به‌طور خاص این برنامه را برای «محتواهای تغییر یافته یا مصنوعی که واقع‌گرایانه هستند» و توسط ابزارهای هوش مصنوعی ساخته شده‌اند، معرفی کرده است. اگر محتواسازان به هر دلیلی محتوای خود را به‌درستی برچسب‌گذاری نکنند، یوتوب می‌تواند آن را حذف کند، درآمدزایی افراد را متوقف سازد یا حساب سازنده را تعلیق کند.



اسپاتیفای به‌زودی به قابلیت نمایش خودکار متن پادکست‌ها مجهز می‌شود: اسپاتیفای به‌زودی، قابلیت جدیدی را برای تولید کنندگان محتوای پادکست ارائه می‌کند که به موجب آن به‌صورت خودکار برای پادکست متن زیرنویس تولید و نمایش داده می‌شود. این ویژگی، پادکست‌ها را برای کاربران قابل دسترسی‌تر می‌کند و درست مانند قابلیت نمایش متن آهنگ‌ها، به شنوندگان اجازه می‌دهد تا براساس متن به قسمت‌های مختلف یک برنامه منتقل شوند.



قلم ۹۹ دلاری S Pen Creator Edition سامسونگ وارد بازار شد: سال‌هاست که قلم S یکی از مزیت‌های محصولات سامسونگ نسبت به دیگر رقبای اندرویدی محسوب می‌شود. اکنون این شرکت نسخه «Creator Edition» قلم S Pen را که مدتی پیش معرفی شده بود، بالاخره روانه بازار آمریکا کرده است. این قلم کاربردهای گوناگونی در نوشتن و نقاشی دارد. درحالی‌که سامسونگ می‌گوید این قلم «پیشرفته‌ترین قلم S» و با حساسیت بالاست، اما درواقع فاقد بسیاری از ویژگی‌های هوشمندی است که سامسونگ طی سال‌ها به قلم S Pen اضافه کرده. به عنوان مثال، این قلم از قابلیت Air Commands پشتیبانی نمی‌کند، چون در آن باتری تعبیه نشده است.



دستگاه Ai Pin با قابلیت‌های هوش مصنوعی: گجتی که می‌خواهد جایگزین گوشی هوشمند شود یک استارت‌آپ مستقر در سن‌فرانسیسکو به نام Humane از اولین محصول خود با نام Ai Pin رونمایی کرد. این دستگاه به‌عنوان یک گوشی هوشمند بدون نمایشگر و البته پوشیدنی یاد شده است. از جمله کاربردهای این دستگاه می‌توان به گرفتن عکس‌های ۱۳ مگاپیکسلی (امکان ضبط ویدیو در آینده فراهم خواهد شد) و ارسال پیام‌های متنی اشاره کرد. همچنین Ai Pin همراه با یک دستیار مجازی هوش مصنوعی مبتنی بر فناوری مایکروسافت و OpenAI ارائه می‌شود. این دستگاه با یک پردازنده نامشخص از کوالکام عرضه می‌شود و وزن آن ۳۴ گرم است. قیمت این دستگاه ۶۹۹ دلار است که البته یک اشتراک ماهانه ۲۴ دلاری نیز باید برای آن خریداری کرد.

جدول کلمات متقاطع

حروف داخل جدول انگلیسی میباشند.

۱- شخصیت اصلی فیلم "The Social Network":

۲- ساختار داده‌ای که میتواند از فضای غیر پیوسته در حافظه استفاده کند:

۳- دنباله‌ای از کاراکترها که کنار هم قرار میگیرند:

۴- از محبوب‌ترین دیستروهای لینوکسی:

۵- نام زبان برنامه نویسی:

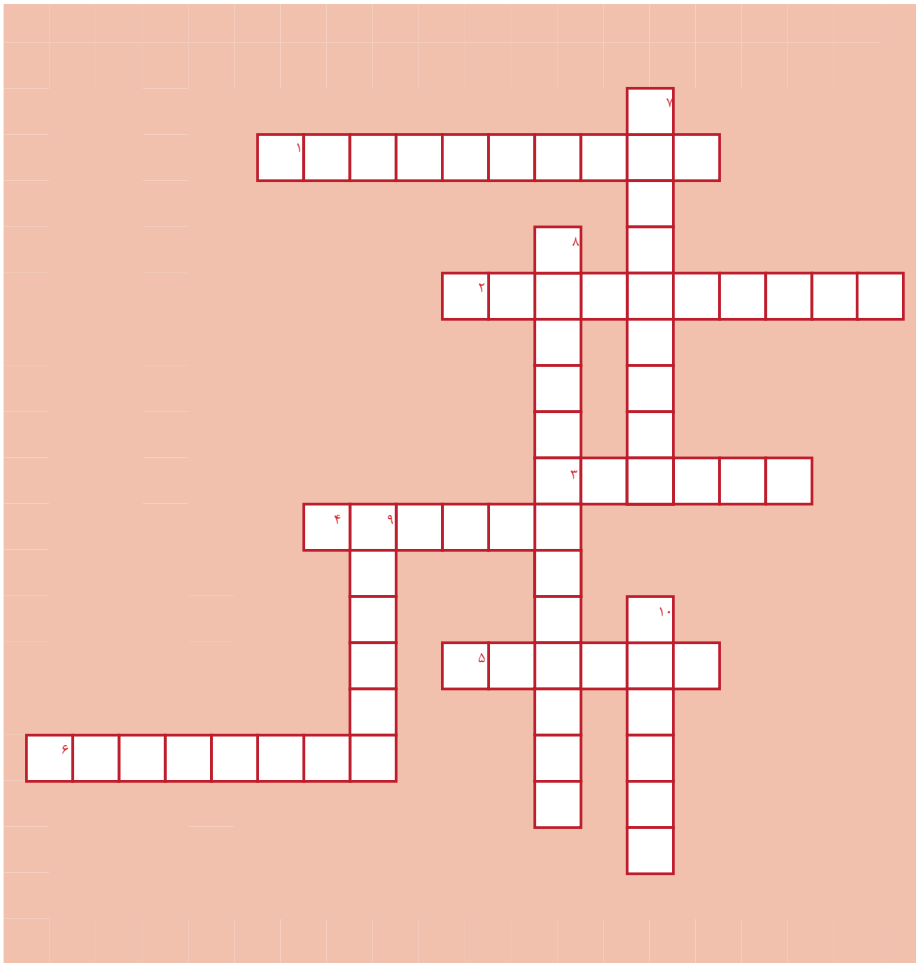
۶- دیجیتال نیست:

۷- قلب و مغز کامپیوتر:

۸- پنهان کردن اطلاعات یک کلاس از برنامه‌های خارج از آن کلاس:

۹- نام کوچک سازنده ++C:

۱۰- ابزاری برای ساختن کانتینر از برنامه‌ها برای اجرا بر هر پلتفرمی:



اختلاف بین تصاویر

تعداد اختلاف هایبین تصاویر: ۱۰ عدد

تصویر مقابل تصویر اتاق کار یکی از دانشجویان مقطع کارشناسی مهندسی کامپیوتر می باشد که در زمینه علوم داده (Data Science) فعالیت دارد.

دو تصویر مقابل در دو زمان مختلف از روز گرفته شده اند. شما قرار است که به او کمک کنید تا ۱۰ مورد اختلاف بین دو تصویر رو پیدا کند.



پاسخ های خود را میتوانید از طریق ایمیل نشریه : cesadonyayesesfroyek@gmail.com تلگرام: @cesa_pr ایتا: @donya_sefroyek ارسال کنید.

اسامی برندگان مسابقه این شماره از نشریه در شماره بعدی اعلام خواهند شد و هدایایی از طرف انجمن علمی کامپیوتر دانشکدگان فارابی به ایشان اعطا خواهد شد.

کلام آخر

از همه شما مخاطبان عزیز نشریه دنیای صفر و یک، دعوت می‌کنیم که مطالب و مقالات خود را در حوزه های مربوط به مهندسی کامپیوتر شامل موضوعات تخصصی رشته و یا موضوعات بین رشته ای برای ما ارسال کنید. مقالات پس از داوری توسط داوران انتخاب شده و پس از بررسی نهایی به مرحله انتشار می‌رسند.

مقالات خود را می‌توانید با دو شیوه زیر ارسال کنید:

۱. وبسایت نشریه: ابتدا در سایت نشریه (cesasj.ut.ac.ir) ثبت نام کنید. پس از تکمیل اطلاعات کاربری از قسمت ارسال مقالات، مقاله خود را ارسال کنید.

۲. از طریق تلگرام: فایل اصلی مقاله به همراه مشخصات نویسندگان را به آیدی (@A_1i0_64) ارسال کنید. گفتنی است که از سمت نشریه گواهی پذیرش و انتشار مقالات به نویسندگان اعطا خواهد شد و مقالات منتشر شده از طریق وبسایت نشریه به صورت دو زبانه قابل رهگیری خواهند بود.

مهلت ارسال مقالات برای نشریه شانزدهم: ۱۴۰۲/۹/۳۰

زمان اعلام نتایج داوری مقالات نشریه شانزدهم: ۱۴۰۲/۱۰/۱۰

زمان انتشار نشریه شانزدهم: ۱۴۰۲/۱۰/۳۰

اگر علاقه مند به عضویت در کارگروه‌های مختلف نشریه اعم از: کارگروه تخصصی، کارگروه اخبار، تیم کلاکت و... یا بخش‌های اجرایی نشریه مانند ویراستاری، صفحه آرایی و... هستید، نیز می‌توانید از طریق راه‌های ارتباطی زیر با ما در ارتباط باشید.

امیدواریم از مطالعه این شماره از نشریه دنیای صفر و یک لذت برده باشید. پذیرای نظرات، انتقادات و پیشنهادات شما هستیم. شما می‌توانید از طریق راه‌های زیر با ما در ارتباط باشید.

وبسایت: cesasj.ut.ac.ir ایمیل: cesadonyayesefroyek@gmail.com

تلگرام: @cesa_pr ای‌تا: @donya_sefroyek

برای اطلاع از اخبار و فعالیت‌های انجمن و نشریه، صفحات مجازی انجمن را دنبال کنید.

تلگرام: @cesa_ut ای‌تا: @cesa_ut

اینستاگرام: cesa_ut لینکدین: [linkedin.com/company/cesa-ut](https://www.linkedin.com/company/cesa-ut)

آپارات: cesa_ut یوتیوب: cesa_ut

باعث خرسندی ماست که این شماره از نشریه را با چشمانتان بدرقه کردید. به امید پویایی و حس و حال مثبت برای جامعه تحصیلی و دانشگاه‌های کشور.

برای قدم زدن در دنیای صفر و یک با ما همراه باشید.



شب و روزتان به زیبایی **دنیای یک** **صفر یک**

صفحات مجازی ما را دنبال کنید.

@cesa_ut



cesa_ut

