«کارآموزی مهندسی یادگیری ماشین»

گزارش کارآموزی

ارائه شده به:

سرکار خانم دکتر منیره عبدوس

توسط:

محمد هاشمی

دانشکده مهندسی و علوم کامپیوتر

تابستان سال ۱۴۰۰

پیش‌گفتار

تاثیر جادویی علم شگفت‌انگیز هوش‌مصنوعی[1] زندگی ما را راحت‌تر و متمایزتر از قبل کرده است. اخیرا نیز علاقه زیادی در زمینه یادگیری ماشین[2] بوجود آمده است و افراد زیادی متوجه گستره کاربردهای جدید ایجاد شده توسط رویکرد یادگیری ماشین می‌شوند. این تکنیک که جزو زیر شاخه های علم بزرگ هوش‌مصنوعی محسوب می‌شود یک نقشه‌ی راه برای ارتباط با دستگاه ایجاد می‌کند و أن را برای پاسخ به دستورالعمل ها و دستورات ما قابل فهم می‌سازد. در این گزارش مهمترین و پرکاربردترین روش‌ها و الگوریتم‌های یادگیری ماشین مورد مطالعه و بررسی قرار می‌گیرد. همچنین درباره کاربردهای شاخص و واقعی که امروزه در دنیای واقعی در صنعت استفاده می‌شوند بحث می‌شود. بدلیل کاربردهای بسیار متنوعی که این علم در صنعت پیدا کرده‌است امروزه شرکت های نوپا[3] و همچنین شرکت‌های بزرگ و قدیمی بسیاری هستند که الگوریتم‌های یادگیری ماشین را به مرحله پیاده‌سازی و بهره‌وری رسانده‌اند. فرصت‌های شغلی‌ای نظیر دانشمندداده[4] و مهندس‌یادگیری ماشین دلایلی هستند که بنده را مجاب کرده‌اند تا یک دوره کامل مهندسی یادگیری ماشین به صورت پروژه محور را در تابستان ۱۴۰۰ در شرکت رهنما کالج بگذرانم. از صمیم قلب از تمامی منتورها و هم‌گروهی پرتلاشم سرکار خانم پرستو فلک‌افلاکی که مرا در به پایان رساندن این دوره همراهی کردند صمیمانه متشکرم.

محمد هاشمی
دانشجوی کارشناسی مهندسی کامپیوتر
شهریور ۱۴۰۰

---

[1] Artificial intelligence
[2] Machine learning
[3] Start-up
[4] Data scientist

فهرست مطالب

چکیده

در این دوره کارآموزی پایه‌ترین و اساسی‌ترین الگوریتم‌ها و روش‌های یادگیری ماشین و یادگیری عمیق مورد مطالعه قرار می‌گیرند. برای درک مفاهیم موجود این حوزه، نیاز به داشتن دانش خوبی در زمینه ریاضیات به خصوص جبرخطی و آمارواحتمالات است. این دوره شامل ۱۰ هفته می‌باشد و هر هفته تمارین و پروژه‌های مربوط به خود را دارد. در هفته آغازین، که به هفته *دست‌گرمی* نامیده شد، با یادگیری کتابخانه‌های مهم کار با داده در زبان برنامه نویسی پایتون، آمادگی خود را برای شروع دوره کسب نمودیم. همچنین با کتابخانه Flask برای API نویسی نیز آشنا شدیم و یک API ساده طراحی و پیاده‌سازی نمودیم. مهارت‌های ورژن‌سازی به کمک Git نیز آموزش دیده شد تا بتوانیم برای پروژه نهایی دوره از آن برای کد نویسی بصورت گروهی استفاده نماییم. در هفته دوم آموزش در قسمت ریاضیات، با شروع از مفاهیم جبرخطی شروع شد. این هفته تمرین عملی ای وجود نداشت و صرفا مطالب تئوری در ویدئوهای آموزشی دوره یاد گرفته‌شد. در ادامه این هفته مطالب مهم در آمارواحتمال مانند متغیرهای تصادفی، انواع توزیع‌های آماری و تخمین زدن به کمک MLE و MAP آموزش داده شد. در هفته سوم نیز مباحث مهم بهینه‌سازی به کمک الگوریتم‌های Gradient Descent و Lagrange بحث و بررسی شد. مبانی یادگیری ماشین و دو الگوریتم پایه‌ی این بخش یعنی رگرسیون و دسته‌بندی دو موضوعی بودن که در ادامه این هفته تدریس شد و همچنین تمرین‌های عملی آن نیز در زبان پایتون نوشته و تحلیل شد. همچنین مباحث مهمی اعم از نحوه مواجهه با دادگان imbalanced نیز مورد مطالعه قرار گرفت. در هفته بعد مباحث مربوط به الگوریتم‌های مهم غیر نظارتی و Mixture Model ها مورد بررسی قرار گرفت. متریک‌های مهم در تمام الگوریتم‌های نظارتی و همچنین غیرنظارتی نیز تدریس و باهم مقایسه شدند. در این هفته مبحث بسیار مهم یادگیری عمیق شروع به تدریس شد. ابتدا کاربردهای آن و دلیل روی‌آوردن به الگوریتم‌های یادگیری عمیق به جای روش‌های سنتی یادگیری ماشین در عصر حاضر مورد بررسی قرار گرفت. مباحث مربوط به شبکه‌های عصبی و نحوه یادگیری در این مدل ها به کمک روش Back propagationبه صورت دقیق مورد مطالعه قرار گرفت. یک تمرین عملی نیز در قسمت پیاده‌سازی شبکه‌های عصبی به کمک فریمورک TensorFlow و Keras نیز انجام شد. در هفته پنجم ابتدا شبکه‌های عصبی پیچشی (CNN) و فواید آن نسبت به شبکه‌های عصبی ساده مورد مطالعه قرار گرفت. معماری‌ها و ساختارهای متفاوتی این نوع شبکه عصبی و نحوه پیاده‌سازی آنها تنها به کمک TensorFlow آموزش داده شد. به همین منظور یک تمرین عملی دیگر برای پیاده‌سازی این نوع شبکه‌ها روی دیتاست ارقام MNIST انجام شد. در انتهای این هفته RNNها به عنوان یکی دیگر از پرکاربردترین شبکه‌های عصبی بررسی و دلایل استفاده از آن و کاربردهای وسیع آن مورد مطالعه قرار گرفت. در ادامه اتوانکودرهای به عنوان یک روش غیرنظارتی مورد بررسی قرار گرفت و یک تمرین عملی از آن نیز زده شد. در هفته ششم نیز، مباحث مقدماتی کلان‌داده ها از قبیل Map Reduce، Hadoop و Spark در حد آشنایی و معرفی تدریس شدند. از هفته هفتم الی دهم پروژه نهایی این دوره کارآموزی تعریف و انجام شد. شرکت رهنماکالج با همکاری شرکت سنجاق، پروژه ای با عنوان *تشخیص ناهنجاری و خزنده در وب* تعریف نمودند. دیتای این پروژه مجموعه‌ای از لاگ‌های سرور وبسایت سنجاق است که هدف پیاده سازی یک مدل غیرنظارتی برای تشخیص خزنده یا ناهنجاری بودن یا نبودن درخواست‌هایی است که به این وبسایت زده می‌شود. ما در این پروژه روش های متعددی اعم از Isolation Forest، Local Outlier Factor،

Autoencoder ها و ... را پیاده سازی و نتایج آنها را با هم مقایسه نمودیم. اتوانکودر ها بهترین نتایج را به همراه داشتند از آن برای فاز production استفاده نمودیم. این پروژه در دو فاز انجام شد که فاز اول اغلب اختصاص به تحلیل داده و EDA به منظور استخراج و مهندسی ویژگی داده شد. در آخر فاز اول نیز ارائه ای از کارهایی که صورت گرفت به تیم منتور و داوران رهنماکالج داده شد. در فاز دوم نیز مدل‌های مختلف غیر نظارتی پیاده سازی و مقایسه شدند و در آخر به کمک فریمورک‌های Flask و ReactJS یک وب‌اپلیکیشن برای تست بهتر و کاربرپسندانه‌تر پیاده‌سازی شد. همچنین ارائه دوم نیز از کارهای انجام شده در هفته دهم صورت گرفت.

به علت محدودیت ۳۰ صفحه‌ای که برای گزارش کارآموزی در نظرگرفته شده است، امکان بررسی تمامی مطالب فراگرفته شده در این دوره نیست. اما مهمترین نکات و بخصوص قسمت‌هایی که دارای تمرین‌های عملی داشته است در این گزارش قابل نوشته شده اند. تمامی کدهای این دوره در ریپازیتوری[5] گیتهاب بنده قابل مشاهده هستند. علاوه برآن، کدها و مستندات مربوط به پروژه نهایی در این ریپازیتوری[6] وجود دارند. برای درک بهتر روند پروژه و کارهایی که برای آن انجام شده به شدت پیشنهاد می‌شود تا اسلایدهای فاز اول و فاز دوم را مطالعه کنید. نمودارها و توجیحاتی که برای کارهایی که انجام شده در آن است که در این گزارش موجود نیست.

[5] https://github.com/mohammadhashemii/ML-RahnemaCollege
[6] https://github.com/mohammadhashemii/Web-Crawler-Detection

۱ مقدمه

یادگیری ماشین یکی از زیرمجموعه های هوش مصنوعی است که به سیستم‌ها این امکان را می‌دهد تا بدون برنامه‌نویسی به صورت صریح توانایی یادگیری و پیشرفت داشته باشند. تمرکز اصلی یادگیری ماشینی بر توسعه برنامه‌های رایانه‌ای است که بتوانند به داده‌ها دسترسی پیدا کنند و از آن‌ها برای یادگیری خود استفاده کنند. یادگیری ماشین نه تنها یک تکنولوژی جدید و کاربردی است بلکه تغییرات بسیار زیادی در دنیا ایجاد کرده‌است و تاثیرات عمیقی در دنیا و بخصوص حوزه اقتصاد و کسب‌وکارها داشته است. همچنین با افزایش اهمیت کلان داده‌ها[۷] و پیچیده شدن فرآیند تجزیه و تحلیل آنها، یادگیری ماشین به یک روش اصلی برای رسیدگی به مسائل مربوط به کلان‌داده‌ها تبدیل شده است. در این مواقع یادگیری ماشین می‌تواند به کارهای زیر رسیدگی کند:

- پردازش تصویر، تشخیص چهره، بینایی‌کامپیوتر و تشخیص شیء.
- پردازش زبان طبیعی، برنامه‌های تشخیص و شناسایی صدا و زبان.
- پیش‌بینی قیمت بازار و سهام بورس.
- زیست‌شناسی محاسباتی، تشخیص تومورهای سرطانی، تعیین توالی DNA و کشف دارو.

به همین دلیل یادگیری ماشینی به خاطر آوردهایی که برای سرمایه‌گذاران دارد و همچنین تحولاتی که در سایر حوزه‌ها می‌تواند ایجاد کند، به یک موضوع داغ تبدیل شده است. این مساله باعث شده است در سراسر جهان بسیاری از افراد خواستار تحصیل و کسب تجربه در رشته هوش مصنوعی و یادگیری ماشین شوند. در این دوره کارآموزی، ابتدا اساسی ترین و محوری‌ترین موضوعات در این علم مورد مطالعه و بررسی قرار می‌گیرد. به دلیل عجین شدن یادگیری ماشین با ریاضیات و آمارواحتمال، پیش از بررسی روش‌ها و الگوریتم‌های یادگیری ماشین، به ریاضیات به خصوص، جبرخطی و آمارواحتمال پیشرفته پرداخته می‌شود. در ادامه مفاهیم یادگیری ماشین و انواع روش‌ها و الگوریتم‌های یادگیری ماشین اعم از یادگیری نظارتی[۸] و بدون‌نظارت[۹] هم به‌صورت تئوری و هم عملی مورد بررسی قرار گرفته می‌شود. سپس یکی از مهمترین و به روزترین زیرشاخه ای از یادگیری ماشین به نام یادگیری عمیق[۱۰] معرفی می‌شود. به پیچیده شدن توابعی که قرار است توسط ماشین یاد گرفته شوند، بسیاری از مواقع الگوریتم های یادگیری ماشین که برای یادگیری نیازمند به ویژگی های از پیش تعریف‌شده توسط انسان هستند و همچنین پیچیده بودن توابع هدف، این الگوریتم‌ها به مرور زمان از کارایی‌شان کاسته شد. در این دوره، یادگیری عمیق و کاربرهای آن به صورت پروژه ای و پیشرفته نیز مورد مطالعه قرار گرفته است. در انتها نیز، گریزی به موضوع کلان‌داده‌ها و نحوه‌ی پیاده‌سازی سیستم های یادگیری که قرار است برای داده در مقیاس بزرگ استفاده شود، زده می‌شود. یکی دیگر از ویژگی‌هایی که این دوره را متمایز از دیگر دوره‌های کارآموزی می‌کند این است که علاوه بر مهارت‌های فنی و عملی، مهارت های نرم اعم از کارگروهی، رزومه‌نویسی و شیوه ارائه تدریس

[7] Big data
[8] Supervised learning
[9] Unsupervised learning
[10] Deep learning

می‌شود ولی در این گزارش این موارد به دلیل دور بودن از مباحث اصلی، مورد بررسی قرار نمی‌گیرد.

۱-۱ معرفی سازمان محل کارآموزی

رهنماکالج[11] یک فضـای کارآموزی حرفه‌ای اسـت که در بازه‌ی زمانی چندهفته‌ای(برای دوره مهندســی یادگیری ماشــین ۱۰ هفته) تجربه‌ی کار واقعی را در پروژه‌های واقعی به شــکل تیمی یا انفرادی رقم می‌زند. در هر دوره بهترین متخصص‌های هر رشته، تجربه‌هایشان را به شکل عملیاتی به کارآموزها انتقال می‌دهند، در انجام پروژه پایانی همراهی‌شــان می‌کنند و فوت‌وفن های کار در دنیای واقعی را آموزش می‌دهند. در این دوره‌های کارآموزی علاوه بر دانش فنی، برای کارآموزها کارگاه‌های مهارت نرم برگزار می‌شــود تا بتوانند تکنیک‌های رفتاری و نحوه‌ی تعامل تیمی را تمرین کنند. رهنما اولین دوره‌ای برگزار کرد در سال ۱۳۹۵ به عنوان دوره برنامه‌نویسی بود. از آن پس، با توجه به عملکرد خوب این شـرکت، دوره‌های دیگری اعم از مهندسـی نرم‌افزار، طراحی UI/UX، دیجیتال مارکتینگ، منابع انسانی و یادگیری ماشین را برگزار می‌کند.

۱-۲ شرح فعالیت و اهداف سازمان محل کارآموزی

هدف از دوره‌ی مهندسـی یادگیری ماشـین رهنماکالج، ایجاد کردن فرصـت تجربه واقعی برای اجرای پروژه‌های هوش مصـنوعی که در صـنعت اتفاق میافتد اسـت. برای ورود به این دوره کارآموزی، یک آزمون ورودی که شـــامل مباحث تئوری(جبرخطی و آمارواحتمال) و یک آزمون عملی[12](پروژه کوچک یادگیری ماشـین) گرفته می‌شـود. در این دوره از دوره های یادگیری‌ماشـین رهنماکالج، از بین بالای ۷۰۰ نفر شـــرکت‌کننده تنها ۳۰ نفر وارد این دوره کارآموزی شـــدند. یکی دیگر از اهداف این سازمان، آماده‌سازی کارآموزان برای ورود به مسیر شـغلی مهندسـی تحلیل داده و یادگیری ماشـین در شـرکت‌ها و سـازمان‌های معتبر اسـت. با توجه به گفته‌های مدیر رهنماکالج، ورود پیدا کردن کارآموزان به شـرکت های معتبر داخلی پس از اتمام دوره کارآموزی، از افتخارات و دلایل پیشـرفت این سـازمان اســـت. همچنین از دیگر فعالیت های این ســـازمان می‌توان به معرفی کردن کارآموزان توانمند به شـرکت‌های معتبر اشـاره کرد. با توجه به پیشـرفت و عملکرد بسـزایی که رهنماکالج در این چند سـال اخیر داشته است، شرکت های معتبری درخواست نیروی انسانی از این سازمان می‌کنند.

۱-۳ شرح کلی فعالیت‌های مرتبط سازمان با رشته تحصیلی

به طور کلی اغلب فعالیت‌های رهنماکالج حاوی جنبه‌های آموزشـی اسـت. در واقع کار اصـلی رهنماکالج، کمک به افراد بااستعداد و باانگیزه برای ورود به مسیر شغلی موردعلاقه‌شان است. به طور خاص در دوره کارأموزی یادگیری ماشـــین، کارآموزان مهارت‌های لازم برای کار کردن در مشـــاغل مربوط به هوش‌مصـنوعی را فرا می‌گیرند و طبق آمار و ارقامی که این سـازمان اعلام کرده‌اسـت، ۹۵ درصد از کارآموزان پس از اتمام دوره در یک شرکت معتبر داخلی استخدام می‌شوند.

[11] https://rahnemacollege.com/
[12] https://github.com/mohammadhashemii/Justice-In-Work

۲ هدف از کارآموزی

هدف اصلی، علاقه بسیار زیاد به مباحث یادگیری ماشین و یادگیری عمیق است! امروزه با توجه به فراگیر شدن هوش مصنوعی و به خصوص یادگیری ماشین در تمامی حوزه‌ها، کمتر کسی وجود دارد که به این حوزه علاقه پیدا نکند. بنده به طور خاص مطالعات خود را در این زمینه از اواخر ترم سوم(زمستان ۹۹) شروع کردم و شروع به انجام چند پروژه تحقیقاتی به کمک اساتید دانشگاه کردم. دلیل انتخابم از مهندسی یادگیری ماشین به عنوان دوره کارآموزی، آمادگی برای وارد شدن به صنعت و انجام پروژه‌هایی است که در زندگی واقعی‌مان در حال رخدادن می‌باشند. به طور خاص، دوره یادگیری ماشین رهنماکالج، دارای محتوایی است که تمام نیازها برای یک مهندس یادگیری ماشین یا دانشمند داده را برطرف می‌کند. مهارت‌های تخصصی که در این دوره کارآموزی توسط کارآموز کسب می‌شوند عبارتند از:

- جبر خطی: تسلط بر جبرخطی موردنیاز و تجزیه‌ها
- آمار و احتمالات در یادگیری ماشین: تسلط بر توابع احتمالاتی و محاسبه MLE و MAP
- بهینه سازی[۱۳]: شناخت توابع بهینه‌سازی و نحوه عملکرد آنها
- مبانی هوش‌مصنوعی: شناخت مسائل مرتبط با هوش‌مصنوعی و یادگیری ماشین و ارتباط آنها
- الگوریتم‌های یادگیری ماشین: بررسی الگوریتم‌های مختلف، روش حل و نحوه پیاده‌سازی
- شبکه‌های عصبی[۱۴] و یادگیری عمیق: درک شبکه‌های عصبی و عمیق، الگوریتم‌های یادگیری آنها و تسلط بر مسائل مختلف
- آشنایی با کلان‌داده‌ها: آشنایی با عملگرهای اصلی در پردازش کلان‌داده‌ها، Hadoop و Spark
- کتابخانه‌های پایتون: آشنایی با کتابخانه‌های مختلف پایتون برای کار کردن در حوزه یادگیری ماشین.

در ادامه در هر قسمت به یکی از موضوعات گفته‌شده پرداخته می‌شود. در این دوره قسمت ریاضیات برای من از اهمیت زیادی برخوردار بود. در هیچ کارآموزی شرکت دیگری، ریاضیات مورد نیاز برای یادگیری ماشین آموزش داده نمی‌شود در صورتی که پایه و اساس الگوریتم‌ها و روش های حل مساله مسائل مربوطه، نیازمند فهم عمیقی از مباحث جبر خطی و أمارواحتمال است. از دیگر اهداف من از این دوره‌ی کارآموزی این بود که چگونه مسائل واقعی ای که در صنعت وجود دارند با مباحث تئوری موجود در یادگیری ماشین ارتباط برقرار می‌کنند. پروژه نهایی این دوره کارآموزی پاسخ خوبی به این سوال بود.

۲-۱ فعالیت‌های انجام‌شده

اغلب زمان صرف شده در این دوره کارآموزی صرف یادگیری مفاهیم و پیاده سازی الگوریتم‌های یادگیری ماشین شده‌است. در ۲ الی ۳ هفته آخر این دوره، پیاده سازی پروژه نهایی و

[13] Optimization
[14] Artificial Neural Network (ANN)

اســتفاده از مفاهیم فراگرفته شــده در یک پروژه واقعی به صــورت عملی وظیفه‌ی ما کارآموزان بود. در جدول زیر به تفکیک هفته، موضــوعات مباحث آموزشــی و وظایفی که بر عهده کارآموزان بود، قابل مشاهده است:

هفته	موضوع	فعالیت‌ها و وظایف
اول	آشنایی با کتابخانه‌های کار با داده در پایتون و APIنویسی	• مرور و یادگیری کتابخانه Pandas و Scipy به همراه تمرین عملی. • یادگیری استفاده از ابزارهای مصورسازی[15] در پایتون. • یادگیری کتابخانه Flask برای API نویسی. • یادگیری و تسلط بر Git.
دوم	ریاضیات	• مشــاهده ویدئوهای مربوط به آموزش ریاضــیات مورد نیاز در یادگیری ماشین. اعم از جبرخطی و آمارواحتمال. • تاکید به تسلط به Decomposition ها در جبرخطی. • درک عمیق از مفاهیم مربوط به PCA ها. • تخمین زدن پارامتر ها توسط روش‌های MLE و MAP.
سوم	بهینه‌سازی و مقدمات یادگیری ماشین	• یادگیری مفاهیم پایه و ضروری بهینه سازی یعنی Gradient Descent و Lagrange. • آشنایی با مفاهیم اولیه یادگیری ماشین. • یادگیری اولین الگوریتم یادگیری ماشــین به نام رگرســیون[16] و پیاده‌سازی آن در پایتون. • یادگیری تئوری و مفاهیم بسته‌بندی در یادگیری ماشین. • یادگیری متریک های یادگیری ماشــین و نحوه اســتفاده از آنها برای دیتاهای imbalanced.

[15] Visualization
[16] Regression

چهارم	دسته‌بندی[17] و مقدمات یادگیری عمیق	• ادامه مبحث دسته‌بندی و پیاده سازی یک پروژه کوچک در پایتون. • یادگیری مفاهیم مربوط به Mixture Model ها. • آشنایی با مفاهیم پایه یادگیری عمیق و دلایل پیدایش آن. • شبکه‌های عصبی و یادگیری Backpropagation به عنوان روش یادگیری آنها. • پیاده‌سازی پروژه تشخیص دستخط روی دیتاست MNIST.
پنجم	CNN ها و RNN ها	• یادگیری و تمرین با Framework های یادگیری عمیق: Tensorflow. • پیاده سازی Autoencoder ها به عنوان یک مدل غیرنظارتی. • آشنایی با LSTM ها به عنوان یکی از قویترین مدل های RNN. • CNN ها و پیاده سازی به کمک کتابخانه Keras. • آشنایی با GAN ها. • آشنایی با مفاهیم مقدماتی Reinforcement Learning.
ششم	آشنایی با کلان‌داده ها	• Map Reduce • Hadoop ,Spark
هفتم	ارائه فاز اول پروژه	• تعریف شدن پروژه پایانی. • تحلیل اکتشافی داده[18] • تمیزسازی داده. • آماده‌سازی اسلاید برای ارائه فاز اول.
هشتم و نهم	توسعه پروژه نهایی	• تعریف و آموزش دادن مدل‌های غیرنظارتی پایه روی ویژگی‌های استخراج‌یافته. • مقایسه مدل‌های مختلف و ارزیابی[19].
دهم	ارائه فاز دوم پروژه نهایی	• ساخت یک API برای تست کردن مدل آموزش دیده. • آماده‌سازی اسلاید برای ارائه نهایی.

[17] Classification
[18] Exploratory Data Analysis (EDA)
[19] Evaluation

جدول ۱-۲: چشم‌اندازی از فعالیت ها و وظایف کارآموزی به تفکیک هفته

۳ شرح فعالیت‌های کارآموزی

در این قسمت، به ترتیب هفته، فعالیت‌هایی که در این دوره کارآموزی انجام شده‌است به صورت **خلاصه** شرح داده می‌شود:

۳-۱ هفته اول – آشنایی با کتابخانه‌های کار با داده در پایتون و API نویسی

در این هفته آمادگی های لازم برای ورود به مباحث آموزشی اصلی کسب شد. ابتدا با چند کتابخانه پایتون برای کار و تحلیل داده، آموزش های لازم برای تمیزسازی داده[۲۰] و تحلیل اکتشافی داده(EDA) داده شد. در انتهای این هفته، یک تمرین برنامه‌نویسی(البته اختیاری) برای پیاده سازی یک API به کمک کتابخانه Flask انجام شد. شرح جزئی و نکاتی مهمی که از مطالب این هفته استخراج شده است را در ادامه می‌بینیم:

۳-۱-۱ آشنایی با Pandas[۲۱]

در این قسمت مروری بر دستورات مهم در این کتابخانه خواهیم داشت. مطالب این قسمت از یک دوره آنلاین از وبسایت دیتاکمپ[۲۲] گرفته‌شده است.

کتابخانه Pandas بر روی دو کتابخانه دیگر پایتون به نام های Numpy[۲۳] و Matplotlib[۲۴] که خود از مهمترین کتابخانه‌های محاسبات عددی و ماتریسی و تصورسازی الگوریتم‌های یادگیری ماشین محسوب می‌شوند. در پژوهش های در حوزه علوم کامپیوتر و پروژه‌های صنعتی، اغلب، داده های جدولی[۲۵]، برای توصیف و نمایش داده به‌کار می‌روند. به همین دلیل Pandas نیز این قالب برای نمایش داده استفاده می‌کند که به آن DataFrame گفته می‌شود. در ادامه به چند مورد از پرکاربردترین و اساسی‌ترین دستورات تعریف شده برای دیتافریم ها می‌پردازیم:

دستور	توضیح و عملکرد دستور
`df.head()`	نمایش دادن چند سطر ابتدایی دیتافریم.
`df.info()`	نمایش اسامی ستون‌ها، جنس داده ذخیره شده در هر ستون، و اینکه آیا دارای

[20] Data cleaning
[21] https://pandas.pydata.org/
[22] https://www.datacamp.com/courses/data-manipulation-with-pandas
[23] https://numpy.org/
[24] https://matplotlib.org/
[25] Tabular data

	Missing value هست یا خیر.
`df.describe()`	محاسبه اساسی‌ترین روابط آماری برای دیتافریم مانند میانگین، میانه و ... .
`df.values`	مقادیر ذخیره شده در ستون‌ها که به صورت یک آرایه نامپای دوبعدی برمی‌گردد.

جدول ۱-۳: چند مورد از پرکاربردترین دستورات *Pandas*

همچنین توابع و روش‌هایی نیز برای پیاده‌سازی توابع دلخواه ما وجود دارد مانند مرتب‌سازی، ادغام کردن دیتافریم‌ها و ... که این روش‌ها در ریپازیتوری[۲۶] گیتهاب بنده قابل مشاهده هستند.

۳-۱-۲ آشنایی با Flask[27]

هدف از این قسمت آشنایی و تمرین برنامه‌نویسی برای API نویسی به زبان پایتون است.

ابتدا نحوه‌ی نصب و راه‌اندازی کتابخانه Flask در اینجا آمده‌است. یک پروژه ساده به عنوان تمرین این بخش در ریپازیتوری[۲۸] بنده نیز قابل مشاهده است.

۳-۲ هفته دوم – ریاضیات

در این قسمت به مهمترین مباحث ریاضیات در یادگیری ماشین یعنی جبرخطی و آمارواحتمال می‌پردازیم. قبل از شروع جبرخطی، خوب است که کاربردهای آنرا در یادگیری ماشین و هوش‌مصنوعی بدانیم. این کاربردها در مواردی مانند رگرسیون خطی، پردازش تصویر، Regularization، PCA[29]، SVD[30] و سایر موارد که در ادامه به آنها می‌پردازیم مشهود هستند.

۳-۲-۱ جبرخطی

جزوه‌ای از مباحث تدریس شده از این قسمت نوشته شده است که در ریپازیتوری[۳۱] بنده قابل مشاهده است. در اینجا چکیده‌ای از این مطالب را مرور می‌کنیم.

۳-۲-۱-۱ فضای برداری[۳۲]

تعریف: یک فضای برداری روی اعداد حقیق تشکیل شده است از : الف) یک مجموعه V که اعضای آن بردار نامیده می‌شوند. ب) عملگر باینری جمع[۳۳] روی بردار ها شروط زیر را ارضا کند:

---

[26] https://github.com/mohammadhashemii/ML-RahnemaCollege/tree/master/Week-0#1-pandas
[27] https://flask.palletsprojects.com/en/2.0.x/
[28] https://github.com/mohammadhashemii/ML-RahnemaCollege/tree/master/Week-0#2-flask
[29] Principal Component Analysis
[30] Singular-Value Decomposition
[31] https://github.com/mohammadhashemii/ML-RahnemaCollege/tree/master/Week-1
[32] Vector space
[33] Addition operation

۱. خاصیت جابجایی.
۲. خاصیت شرکت‌پذیری.
۳. وجود داشته باشد یک بردار صفر که برای هر بردار v $v \in V: v + 0 = v$.
۴. به ازای هر بردار $v \in V$ یک بردار یکتای $-v$ وجود داشته باشد که $v + (-v) = 0$.

ج) عملگر دیگری به نام عملگر ضـــرب اســـکالر[۳۴]، اگر $a \in \mathcal{R}$ و $v \in V$ آنگاه $a.v \in V$.
همچنین شروط زیر را ارضا کند:
۱. برای هر $v \in V$ باید $1.v = v$
۲. $(\alpha_1 . \alpha_2).v = \alpha_1.(\alpha_2.v)$
۳. $\alpha(v_1 + v_2) = \alpha.v_1 + \alpha.v_2$
۴. $(\alpha_1 + \alpha_2).v = \alpha_1.v + \alpha_2.v$

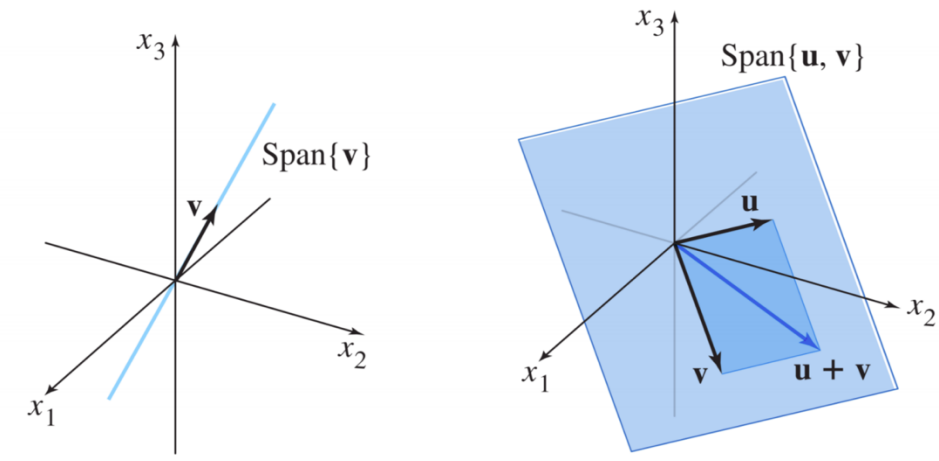
۲-۱-۲-۳ وابستگی خطی[۳۵]

تعریف ۱: بردار v را **ترکیب خطی** بردارهای $\{v_1, v_2, \dots, v_n\}$ می‌نامیم اگر:

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n \quad , \alpha_i \epsilon \mathbb{R}$$

تعریف ۲: مجموعه‌ی همه بردارهای v ملنند بالا را **پوشش خطی**[۳۶] مجموعه $\{v_1, v_2, \dots, v_n\}$ می‌نامیم و بصورت زیر نمایش می‌دهیم:

$$span(v_1, v_2, \dots, v_n) =< v_1, v_2, \dots, v_n >$$



شکل ۱-۳: مثال‌هایی از *span*

[34] Scalar multiplication
[35] Linear dependency
[36] Linear span

نکته: هر $span(v_1, v_2, \dots, v_n)$ خود یک فضـــای برداری روی $\mathbb{R}$ به حســـاب می‌آید. به این‌گونه مجموعه‌ها **زیرفضای**[۳۷] v نیز گفته‌می‌شود.

نکتـه: $span(v_1, v_2, \dots, v_n)$ کوچکترین زیرفضـــا برای v اســـت کـه شـــامـل $\{v_1, v_2, \dots, v_n\}$ است.

تعریف ***وابســتگی خطی***: مجموعه $\{v_1, v_2, \dots, v_n\} \subseteq V$ را وابســته خطی می‌نامیم اگر $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \in \mathbb{R}$ ای وجود داشته باشد که هیچ کدام صفر نباشند و :

$$0 = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$$

به عبارت دیگر حداقل یکی از بردارهای v_i بتواند بصـورت ترکیب خطی بقیه بردارها نوشـته شود.

تعریف ***اســتقلال خطی***: مجموعه $\{v_1, v_2, \dots, v_n\} \subseteq V$ را مســتقل خطی می‌نامیم اگر $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \in \mathbb{R}$ ای وجود داشته باشد که:

$$0 = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$$

آنگاه همه‌ی α_i ها صفر باشند.

تعریف ***پایه***: هر مجموعه مســتقل خطی $\{v_1, v_2, \dots, v_n\} \subseteq V$ که V را span کند، یک پایه برای فضای برداری V به حساب می‌آید.

به علت حجیم شدن متن گزارش امکان بررسی همه‌ی مطالب گفته شده در این قسمت وجود ندارد. جزوه کاملتر آن در ریپازیتوری[۳۸] گیتهاب بنده قابل مشاهده است.

۳-۲-۲ آمارواحتمال

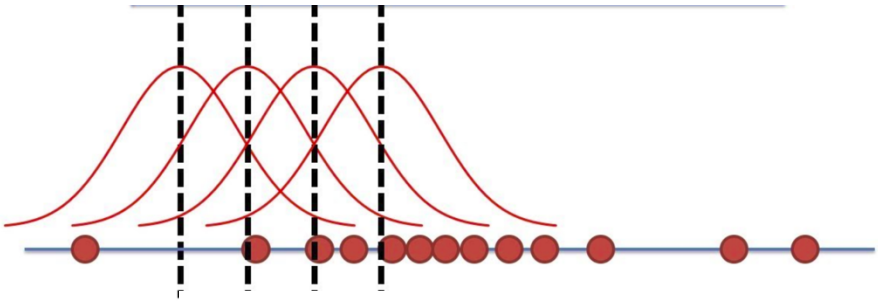
مباحث مختلفی در این قسمت در دوره کارآموزی تدریس داده شده است. اما در این گزارش به طور مختصر به یکی از مهمترین قسمت‌های آن ها کاربرد بسیار زیادی در مباحث تئوری یادگیری ماشین دارد می‌پردازیم.

۳-۲-۲-۱ Maximum Likelihood Estimation (MLE)

این روش برای تخمین زدن پارامترهای یک متغیر تصـــادفی اســـت به کمک وجود تعداد

[37] Subspace
[38] https://github.com/mohammadhashemii/ML-RahnemaCollege/tree/master/Week-1

محدودی از نمونه‌های آن متغیر تصادفی به کار می‌رود. یعنی نمونه‌هایی که داریم دارای توزیع مشابه با توزیع متغیر تصادفی است که قصد به تخمین زدن آن داریم. برای درک بهتر این روش، آنرا با یک مثال توضیح می‌دهیم. فرض کنید $D = (X_1, X_2, \dots, X_n) \sim \mathcal{N}(\mu, \sigma^2)$، تا از نمونه‌هایمان n هستند که دارای توزیع احتمال نرمال با میانگین μ و انحراف معیار σ است. همچنین مقدار σ^2 معلوم مساله است. هدف ما تخمین زدن پارامتر μ می‌باشد.



نمودار ۱-۳: حالت‌های متفاوت برای پارامتر μ

در نمودار ۱-۳ می‌توان حالات متفاوت برای اندازه μ را مشاهده کرد که به طبع برای هر μ، یک توزیع متفاوت برای تابعی که می‌خواهیم تخمین بزنیم خواهیم داشت. دایره های قرمز نشان‌دهنده نمونه‌هایی است داریم. اگر پارامترهای مساله را در یک برداری مثل بردار θ قرار دهیم آنوقت داریم:

$$p_\theta(x) \coloneqq p(x|\theta)$$

$$p_{\mu,\sigma^2}(x) \coloneqq p(x|\mu,\sigma^2)$$

که بهترین پارامتر، پارامتری است که احتمال $p(D|\theta)$ را **بیشینه** کند.

تعریف تابع likelihood:

$$\mathcal{L}(\theta|x) \coloneqq p(D|\theta) = p(x_1, x_2, \dots, x_n|\theta) = \prod^{n} p(x_i|\theta)$$

بردار θ ای که این تابع را بیشینه کند به عنوان MLE از θ شناخته می‌شود و به عنوان θ_{MLE} نشان داده می‌شود. برای اینکه محاسبات راحتتر شود معمولا از تابع بالا یک لگاریتم گرفته می‌شود:

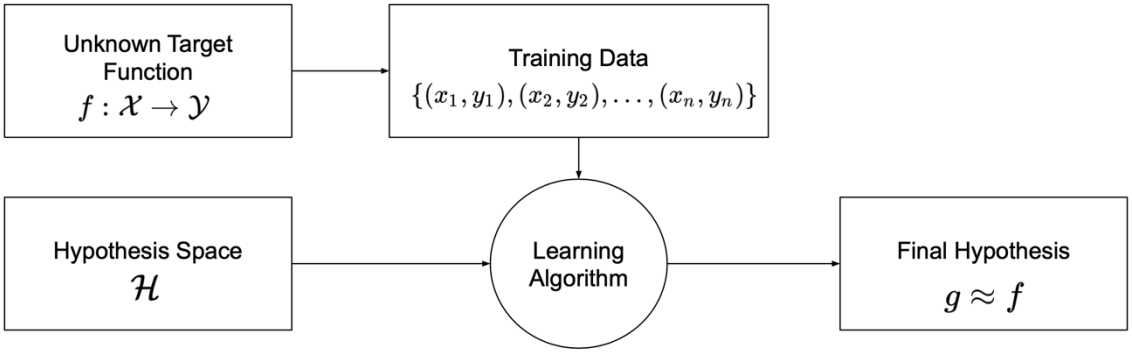
$$\log\big(\mathcal{L}(\theta|x)\big) = \sum^{n} \log\left(p(x_i|\theta)\right)$$

برای پیدا کردن بهترین پارامتر، باید از تابع بالا مشتق گرفت و برابر صفر قرار داد. مثال حل

شده آن در اسلایدهای تدریس شده در دوره کارآموزی موجود می‌باشد.

۳-۳ هفته سوم – بهینه‌سازی و مقدمات یادگیری ماشین

اگر بخواهیم تعریف کلی از یادگیری ماشین داشته باشیم باید بگوییم هدف این علم، جستجو برای بهترین تابعی است که ورودی‌های داده‌های آموزش را به خروجی‌هایشان نگاشت می‌کند همچنین روی داده آزمون نیز تعمیم پذیر باشد. چگونه به دنبال این تابع بگردیم؟ باید ابتدا یک فضای فرضیه[۳۹] که مجموعه ای از توابع است را انتخاب کنیم. یعنی به عنوان مثال مجموعه توابع خطی به فرم $f = wx + b$. برای اینکه ارزیابی‌ای روی توابع این مجموعه داشته باشیم باید یک تابع هزینه[۴۰] تعریف می‌کنیم تا مجموع خطای نگاشت هر نمونه از دیتاست را حساب کنیم. طبیعتا تابعی که کمترین هزینه(مجموع خطا) را داشته باشد، مقصود ماست.



شکل ۲-۳: فرآیند کلی در الگوریتم های یادگیری ماشین

مسائل موجود در یادگیری ماشین به دسته‌های زیر تقسیم می‌شوند.

- یادگیری با نظارت (supervised): برچسب داده‌ها موجود است.
- یادگیری غیرنظارتی (unsupervised): برچسب داده‌ها موجود نیست.
- یادگیری تقویتی (reinforcement): تعامل با محیط و یادگیری آن رفتارکردن.

البته دسته‌های دیگری نیز موجود هستند ولی پایه و اساس یادگیری ماشین برروی همین ۳ دسته می‌چرخد. به دلیل حجیم‌شدن متن گزارش امکان بررسی تمامی الگوریتم‌ها بصورت جزئی در اینجا نیست اما در ادامه به برخی از آنها و پیاده‌سازی هایشان در زبان پایتون به صورت **خلاصه** می‌پردازیم:

۳-۳-۱ رگرسیون[۴۱]

در دسته‌ی الگوریتم‌های با نظارت قرار می‌گیرد و فضای خروجی نیز پیوسته است. یکی از

[39] Hypothesis space
[40] Cost function
[41] Regression

مثال‌های معروف آن تشخیص قیمت خانه از روی ویژگی‌های آن است. در جدول پایین ویژگی‌ها و در ستون آخر قیمت هر خانه به عنوان برچسب آمده است.

X1 (size)	X2 (age)	X3 (rooms)	Y (price)
100	10	3	200
200	2	4	300
40	10	1	40
50	0	1	70
...	...	...	...

جدول ۲-۳: دیتاست مربوط به تشخیص قیمت خانه

مانند همه‌ی الگوریتم‌های یادگیری ماشین، باید یک فضای فرضیه برای حل مساله رگرسیون انتخاب شود که می‌توان آنرا به صورت $y = w^T x + w_0$ نوشت. به این معنی که اگر نمونه‌ها دارای یک ویژگی باشند، فضای فرضیه یک خط، اگر دارای دو ویژگی یک صفحه و برای بیشتر از دو ویژگی یک ابرصفحه[۴۲] است. هدف پیدا کردن بردار وزن‌های w و عرض از مبدا w_0 به عنوان مجهول های مساله است. به همین دلیل، با استفاده از روش پیمایشی[۴۳]، ابتدا بردار وزن‌ها و عرض از مبدا را مقداردهی اولیه می‌کنیم و به کمک تعریف یک تابع خطا[۴۴]، خطای تابع تخمین‌زده شده، آنرا ارزیابی و در صورت نیاز وزن‌ها را به روز رسانی می‌کنیم. تابع خطایی که معمولا برای رگرسیون در نظر گرفته می‌شود تابع مجموع مربعات خطا است که به اختصار به MSE معروف می‌باشد. نحوه‌ی به‌روزرسانی وزن‌ها به کمک بهینه‌سازی تابع خطا با روش Gradient Descent صورت می‌گیرد که بدلیل حجیم شدن متن گزارش جزئیات و محاسبات ریاضی آن در اینجا نمی‌پردازیم.

به عنوان یک تمرین عملی برای بخش رگرسیون روی دیتاست Boston برای تشخیص قیمت خانه، یک پیاده سازی رگرسیون خطی به کمک کتابخانه Sickit-learn در ریپازیتوری گیتهاب[۴۵] بنده قابل مشاهده است.

۲-۳-۳ متریک‌های ارزیابی برای رگرسیون

متریک‌های متفاوتی برای ارزیابی مدل رگرسیون وجود دارد که هرکدام مزایا و عیب‌های خود را دارد.

۱. MSE: میانگین مجموع مربعات خطا است. می‌توان گفت رایج‌ترین است اما عیبی که دارد این است که شهودی به ما از فاصله نمی‌دهد. (زیرا فرمول فاصله یک جذر هم دارد)

[42] Hyper space
[43] Iterative
[44] Loss function
[45] https://github.com/mohammadhashemii/ML-RahnemaCollege/tree/master/Week-2

۲. RMSE: جذر میانگین مجموع مربعات خطا است که این بار دقیقا تعریف فاصله اقلیدوسی است که شهود دقیقتری از مفهوم فاصله برایمان ایجاد می‌کند

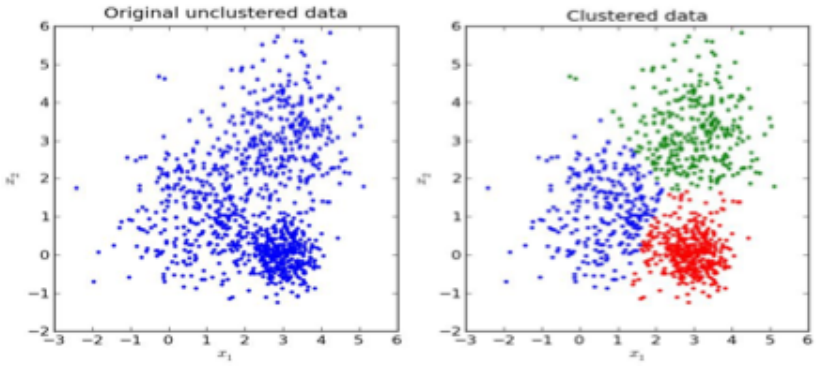
۳. MAE: میانگین قدرمطلق خطا است.

۴. MAPE: میانگین درصدی قدرمطلق خطاست. یعنی به صورت درصدی قدرمطلق اختلاف مقدار مطلوب با مقدار پیش‌بینی شده حساب می‌گردد که بدلیل درصدی بودن شهود بهتری از عملکرد مدل می‌توان گرفت.

۵. R2-score: یکی از عیوب متریک MSE این است که لزوما نمی‌توان نتیجه گرفت که اگر MSE یکبار ۱۰ شد بهتر از موقعی است که ۱۰۰ شده است. زیرا میزان بزرگی و scale داده‌ها در آن لحاظ نمی‌شود. ولی در r2-score دقیقا این موضوع لحاظ می‌شود.

۳-۳-۳ خوشه بندی[۴۶]

این نوع الگوریتم‌ها از روش‌های غیرنظارتی محسوب می‌شوند به عبارتی نیاز به وجود برچسب برای داده‌ها وجود ندارد. و وظیفه مدل این است که با توجه به توابع شباهتی که تعریف می‌شوند، شباهت بین نمونه از روی فضای ورودی را محاسبه و داده‌های را دسته‌های مختلف خوشه بندی کند.



شکل ۳-۳: قبل و بعد از اجرای الگوریتم خوشه بندی

برای خوشه‌بندی الگوریتم‌های متعددی معرفی شده‌اند که یکی از معروفترین های آنها الگوریتم K-means است. هدف این الگوریتم، خوشه بندی دیتا به k دسته است که نمونه‌های داخل یک کلاس کمترین واریانس را داشته باشند و مرکز هر دسته نسبت به دسته‌های دیگر بیشترین فاصله را داشته باشند. به همین منظور می‌توان گفت هدف این روش کمینه کردن تابع خطای زیر است. اگر نمونه i ام عضو کلاستر k باشد آنگاه $r_{ik} = 0$ و خواهیم داشت:

$$loss = \sum_{n}^{} \sum_{k}^{} r_{ik} ||x_n - \mu_k||^2$$

پیدا کردن بهترین جواب یک مساله NP-hard محسوب می‌شود برای همین یک روش بازگشتی برای این

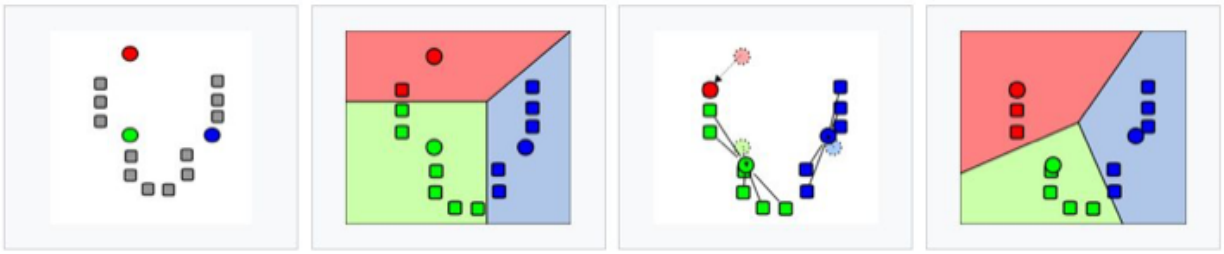
[46] Clustring

مساله پیشنهاد شده است که به شرح زیر می‌باشد:

قدم ۱: مشخص کردن k نقطه به عنوان مرکز دسته‌ها به صورت تصادفی.

قدم ۲: اختصاص دادن برچسب نمونه به برچسب نزدیک‌تری مرکز به آن نمونه.

سپس دوباره قدم ۱ را تکرار میکنیم(البته در مراحل بعدی، قدم ۱ دیگر به صورت تصادفی نخواهد بود) پس از تکرار چندین مرحله، دیگر تفاوت خاصی در مراکز دسته‌ها و برچسب نمونه‌ها صورت نمی‌گیرد و الگوریتم در آنجا متوقف می‌شود و نمونه‌ها به k دسته خوشه بندی شده‌اند.



شکل ۴-۳: مراحل الگوریتم *K-means*

۳-۴ هفته چهارم – دسته‌بندی[47] و مقدمات یادگیری عمیق

دسته‌بندی از الگوریتم‌های با نظارت یادگیری ماشین محسوب می‌شود و این‌بار خروجی مساله برخلاف مساله رگرسیون، گسسته است. یعنی قرار است در فاز پیش‌بینی، مدل با دیدن نمونه به همراه ویژگی‌هایش متعلق به کدام یک از دسته‌های موجود است. بدلیل زیاد شدن حجم متن مقاله، بررسی تمام جزییات و مفاهیم تئوری بحث دسته‌بندی در اینجا امکان پذیر نیست. اما به عنوان یک پیاده‌سازی تمرین عملی دسته بندی روی دیتاست Titanic در ریپازیتوری[48] گیتهاب بنده قابل مشاهده است.

۳-۴-۱ آشنایی با یادگیری عمیق

هدف اصلی الگوریتم‌های یادگیری عمیق، تخمین تابع به کمک **ترکیب** تابع‌های ساده‌تر و ساخت تابع‌های پیچیده‌تر(معمولا غیرخطی) است. در یادگیری ماشین می‌بایستی فضای فرضیه در ابتدای کار مشخص شود ولی در الگوریتم‌های یادگیری به کمک شبکه‌های عصبی، فضای فرضیه از روی ترکیب توابع ساده‌تر غیرخطی به صورت خودکار ساخته می‌شود. به طور مثال اگر فرض کنیم ۳ تابع غیرخطی داریم، آنگاه فضای فرضیه به صورت زیر نوشته می‌شود:

[47] Classification
[48] https://github.com/mohammadhashemii/ML-RahnemaCollege/tree/master/Week-3

$$\mathcal{F} = \{f(X) = W_3 g(W_2 g(W_1 X))\}$$

که در آن g یک تابع ساده غیرخطی مانند $ReLu$، $tanh$ و یا ... است.

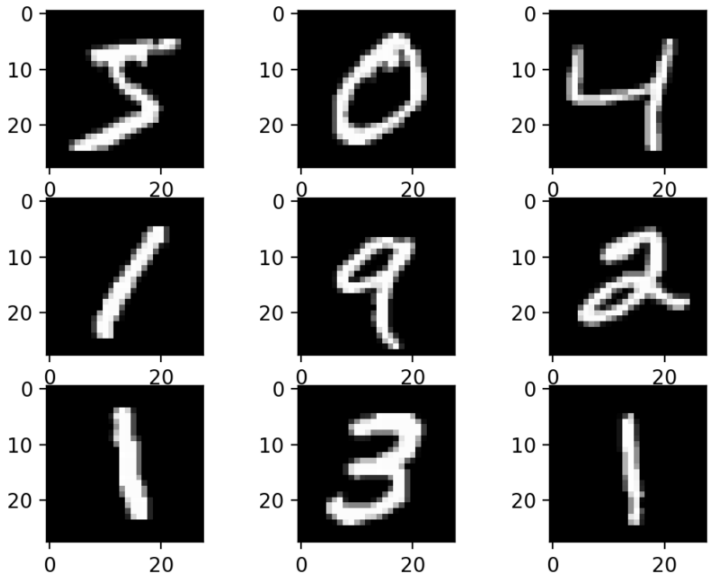
اگر بخواهیم دلایل روی آوردن به الگوریتم‌های یادگیری عمیق به‌جای روش‌های سنتی یادگیری ماشین متذکر شویم، می‌توان به موارد زیر اشاره نمود:

- عدم نیاز به *Feature selection* و همچنین *Feature extraction*: یکی از مراحلی که در اغلب تسک‌های یادگیری ماشین صورت می‌گیرد بحث استخراج ویژگی توسط انسان است. قبل از آموزش مدل‌های یادگیری، باید مجموعه‌ای از ویژگی‌ها توسط کاربر مشخص شود تا فرآیند آموزش صورت بگیرد. همچنین برخی از این ویژگی‌ها بدلایل ممکن است تاثیر خاصی رو فرآیند آموزش نداشته باشد یا حتی موجب بیش‌برازش[۴۹] شود که توسط روش‌های موجود در حوزه *feature selection* مانند محاسبه وابستگی[۵۰] ویژگی‌ها به خروجی اغلب با رابطه پیرسون، برخی از ویژگی‌ها حذف می‌شوند. این دو فرآیند در الگوریتم‌های یادگیری عمیق عملا وجود ندارند.
- هزینه‌بر بودن الگوریتم‌های یادگیری ماشین: همین بحث استخراج ویژگی در اغلب موارد فرآیند هزینه‌بر و زمانبری صورت می‌شود. زیرا باید به صورت دستی و توسط انسان تعریف شوند و همچنین در دیتاست‌های با مقیاس بزرگ گاهی مواقع این فرآیند نشدنی است.

۳-۴-۱-۱ پیاده سازی شبکه‌های عصبی

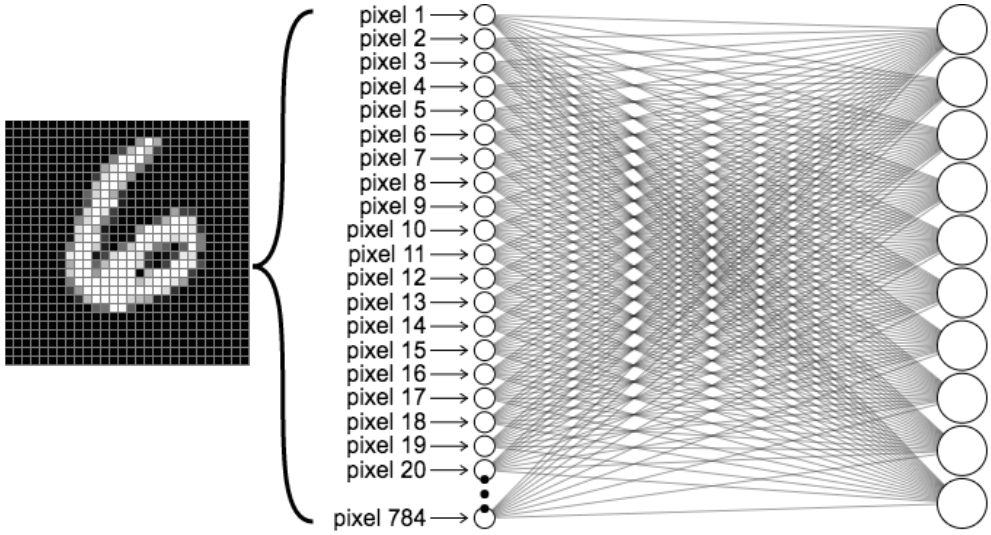
متاسفانه توضیح الگوریتم‌های شبکه‌های عصبی و نحوه‌ی آموزش آنها با روش‌هایی مثل back propagation در این گزارش با توجه به محدودیت صفحات گزارش امکان‌پذیر نیست. اما در این قسمت پیاده‌سازی یک شبکه عصبی با لایه‌های fully connected به کمک فریمورک‌های TensorFlowو Keras برای تشخیص ارقام روی عکس‌های دستخط از دیتاست معروف MNIST خواهیم داشت.

[49] Overfitting
[50] Correlation



شکل ۵-۳: دیتاست ارقام دستنویس *MNIST*

دیتاست MNIST دارای ۶۰۰۰۰ نمونه به عنوان داده آموزش و ۱۰۰۰۰ داده به عنوان داده آزمون است. سایز تصاویر ۲۸*۲۸ پیکسل است و تصاویر همانطور که در شکل ۳-۳ می‌بینید، به صورت سیاه و سفید هستند به عبارت دیگر ابعاد دیتاست آموزش به صورت $(60000, 28, 28)$



شکل ۶-۳: نحوه اتصال نورون‌ها در شبکه‌عصبی با یک لایه مخفی

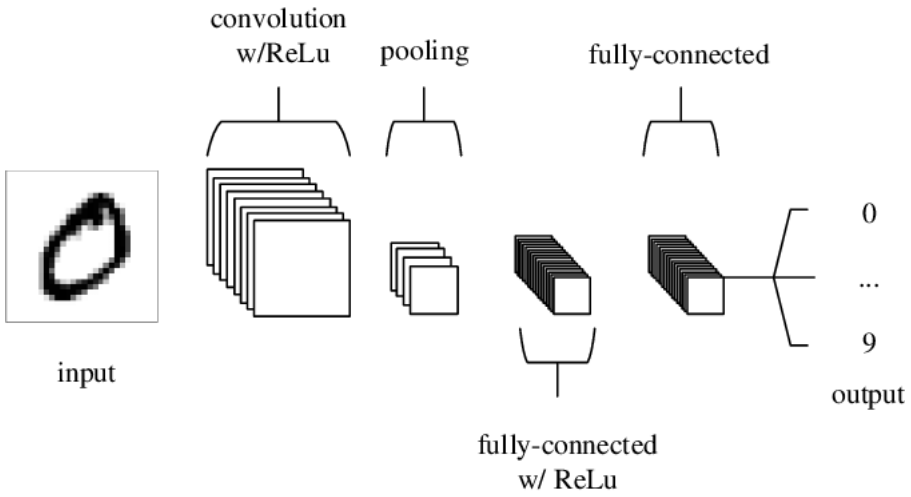
می‌باشد. مقادیر درایه‌های ماتریس عکس‌های اعداد بین ۰ تا ۲۵۵ که نشان‌دهنده شدت روشنایی برای هر نقطه می‌باشد. برای عملکرد بهتر و سریعتر همگراشدن مدل به هنگام بهینه‌سازی، عمل نرمال‌سازی صورت می‌گیرد تا همه مقادیر در بازه بسته ۰ تا ۱ قرار بگیرند. در ادامه باید معماری شبکه‌عصبی مورد نظر را مشخص کنیم. در این تمرین، صرفا از یک شبکه عصبی با یک لایه مخفی

استفاده می‌کنیم. برای این منظور ابتدا ماتریس ۲۸*۲۸ را به یک بردار ۷۸۴=۲۸*۲۸ تبدیل می‌کنیم. این ۷۸۴ تا نورون ورودی را به تمام نورون‌های لایه مخفی وصل می‌کنیم.

در آخر یک لایه خروجی با ۱۰ نورون با تابع softmax برای پیش‌بینی قرار می‌دهیم. همچنین تابع غیرخطی ای که استفاده می‌کنیم ReLu و برای جلوگیری از بیش‌برازش از dropout با احتمال ۰.۲ استفاده می‌کنیم تا در هر مرحله از آموزش ۰.۲ از نورون‌های لایه مخفی به صورت تصادفی را خاموش کند. نتایج ارزیابی و دقت مدل آموزش دیده را می‌توانید در کدی که برای این قسمت زده شده در ریپازیتوری[۵۱] گیتهاب بنده مشاهده کنید.

۳-۵ هفته پنجم – CNN ها و RNN ها

در این هفته آموزش‌های لازم در رابطه با شبکه‌های عصبی پیچشی[۵۲] و شبکه‌های عصبی



شکل ۷-۳: معماری یک شبکه عصبی پیچشی برای آموزش روی دیتاست *MNIST*

همروند[۵۳] داده شد و در انتها یک تمرین عملی از پیاده‌سازی یک CNN روی دیتاست MNIST و مقایسه نتایج آن با نتایج شبکه عصبی ساده(Fully connected) انجام شد. دقتی که در تمرین هفته قبل برای شبکه‌عصبی با لایه‌های fully connected گرفته شد حدود ۹۴ درصد روی داده آزمون بود اما برای شبکه عصبی پیچشی ۹۸ درصد. کد این تمرین نیز در ریپازیتوری[۵۴] گیتهاب بنده قابل مشاهده می‌باشد.

[51] https://github.com/mohammadhashemii/ML-RahnemaCollege/tree/master/Week-4
[52] Convolutional Neural Networks
[53] Recurrent Neural Networks
[54] https://github.com/mohammadhashemii/ML-RahnemaCollege/tree/master/Week-5

۳-۶ هفته ششم – آشنایی با کلان داده‌ها

در این هفته، مفاهیم پایه و اساسی برای کلان‌داده‌ها[55] از قبیل Hadoop ،Map reduced و spark تدریس شده‌اند. به علت نزدیک‌نبودن این مبحث به مفاهیم یادگیری ماشین در این گزارش خلاصه‌ای از آن نوشته نشده است.

۳-۷ هفته هفتم – ارائه فاز اول پروژه

از این هفته به بعد تمام تمرکز کارآموزان روی انجام پروژه پایانی این دوره بود. صورت این پروژه با همکاری شرکت داخلی سنجاق[56] که یک بازار آنلاین خدمات به‌حساب می‌آید تعریف شده است. صورت کامل این پروژه در اینجا آمده است. اما اگر بخواهیم مختصر آنرا شرح دهیم، هدف پروژه راه‌اندازی یک سیستم آفلاین برای تشخیص ناهنجاری و خزنده بر بستر وب[57] به کمک الگوریتم‌های غیرنظارتی است. دیتایی در فاز آموزش مدل در اختیار داشتیم، لاگ سرور سایت سنجاق است. این دیتا در طول مدت زمان یک‌هفته ای جمع آوری شده بود و نکته مهمی که این پروژه را چالشی می‌کرد، عدم وجود برچسب برای هر رکورد از دیتاست بود. در این فاز، پیش پردازش های لازم روی دیتای خام و فرآیند EDA بصورت متمرکز انجام شد. قبل از بررسی، پیشنهاد می‌شود فایل ارائه فاز اول را در اینجا مطالعه شود. همچنین تمامی کدها و منابع پروژه در ریپازیتوری[58] گیتهاب بنده قابل مشاهده است.

۳-۷-۱ تحلیل اکتشافی داده (EDA)

دیتاستی که در این پروژه استفاده شده متشکل از لاگ NGINX سرور وبسایت سنجاق می‌باشد. تعداد رکوردهای این دیتاست، ۱۲۶۰۰۳۳ است. نمونه ای از این دیتاست در اینجا قابل مشاهده است. ویژگی‌های هر رکورد از این دیتاست عبارتند از:

ویژگی	جنس	توضیحات
ip	string	به عنوان نمونه: 207.213.193.143
time	string	به عنوان نمونه: 2021-05-12 05:06:00+04:30
method	string	Get/Put/Post/Options/Head
status_code	integer	وضعیت یک درخواست را نشان می‌دهد

[55] Big data
[56] https://sanjagh.pro/tehran/
[57] Web crawler detection
[58] https://github.com/mohammadhashemii/Web-Crawler-Detection

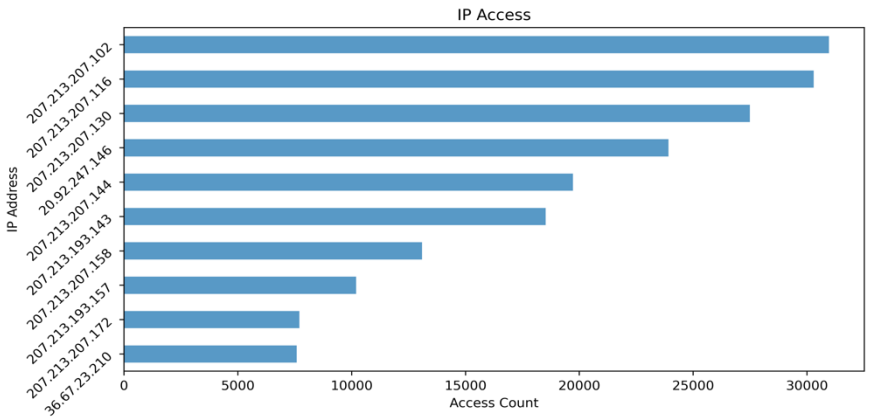
path	string	به عنوان نمونه: cdn/articles/1148001967
response_length	integer	حجم اطلاعات موجود در آن درخواست.
user_agent	string	به عنوان نمونه: Googlebot-Image/1.0
response_time	float	زمان پاسخ سرور به کلاینت

جدول ۳-۳: ویژگی‌های دیتاست استفاده شده

تحلیل اکتشافی داده یا همان EDA برای هر پروژه‌ای که با داده سروکار دارد الزامی است. EDA راهی برای مصورسازی، خلاصه‌سازی و تفسیرسازی اطلاعات مخفی و آشکار یک مجموعه دادگان است. EDA قدمی اساسی در علم داده است که با انجام آن یک دید خوبی از ویژگی های آماری یک دیتاست کسب می‌کنیم. با اتمام این مرحله می‌توان از ویژگی های بدست آمده در یک روش نظارتی یا غیرنظارتی بهره برد. قبل از بررسی این مرحله توصیه می‌شود فایل نوت‌بوک نوشته شده در ریپازیتوری[۵۹] گیتهاب بنده مطالعه شود.

ابتدا چند نمودار و شکل را رسم می‌کنیم تا شهود خوبی از دیتا کسب کنیم. به مرور می‌توان از روی همین نمودار ها برخی از ویژگی های درخواست‌های جعلی[۶۰] را فهمید.

۱. بیشترین IP های مشاهده شده:



نمودار ۲-۳: تعداد درخواست برای هر *IP*

سپس رکوردهای دیتاست را براساس بیشترین درخواست از آی‌پی ها مرتب می‌کنیم:

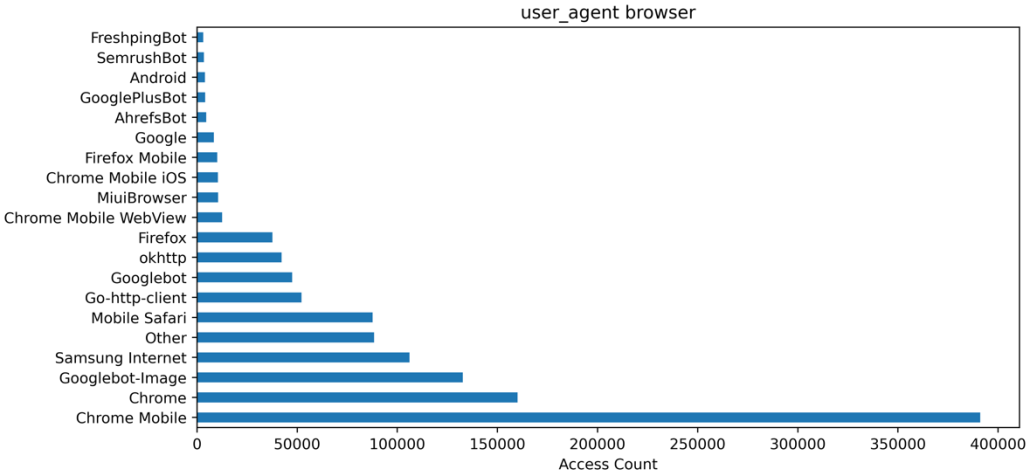
[59] https://github.com/mohammadhashemii/Web-Crawler-Detection/tree/master/notebooks
[60] Anomaly requests

	ip	time	method	status_code	path	response_length	user_agent	response_time
0	207.213.207.102	2021-05-12 06:25:56+04:30	Get	304	cdn/articles/1148001967	0	Googlebot-Image/1.0	16.0
30968	207.213.207.116	2021-05-12 07:40:46+04:30	Get	304	cdn/profiles/1074674108	0	Googlebot-Image/1.0	16.0
61258	207.213.207.130	2021-05-12 09:25:34+04:30	Get	200	amp/price/1313296747	125767	Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Bu...	28.0
88748	20.92.247.146	2021-05-12 05:08:46+04:30	Get	200	js/profession.c67de06df71c34fc126d.js	107970	sentry/21.4.1 (https://sentry.io)	16.0
112660	207.213.207.144	2021-05-12 09:02:56+04:30	Get	200	amp/price/252451961	121639	Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Bu...	28.0
132378	207.213.193.143	2021-05-12 05:06:00+04:30	Get	304	cdn/profiles/1026106239	0	Googlebot-Image/1.0	32.0
150901	207.213.207.158	2021-05-12 09:25:55+04:30	Get	304	cdn/articles/2121333045	0	Googlebot-Image/1.0	16.0
163992	207.213.193.157	2021-05-12 05:07:04+04:30	Get	200	amp/blog/article/1197238235	101087	Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Bu...	144.0
174189	207.213.207.172	2021-05-12 09:28:50+04:30	Get	304	cdn/pro_photo_gallery/1540103160	0	Googlebot-Image/1.0	28.0
181890	36.67.23.210	2021-05-12 05:06:00+04:30	Head	200	877499224	0	Go-http-client/2.0	28.0

مشاهده می‌شود که اتفاقا IP هایی که بیشترین درخواست‌ها از آنها آمده محتمل به جعلی بودن و یا برخاسته از یک خزنده باشند هست. زیرا با نگاه کردن به user_agent های آنها شاهد نمونه هایی از بات ها مانند GoogleBot و ... هستیم.

۲. بیشترین مرورگرهای دیده شده: برای اینکه بفهمیم مرورگر کاربر چه بوده است از کتابخانه پایتونی به نام user_agents[61] استفاده کردیم. با استفاده از این کتابخانه می توان مرورگر، اینکه درخواست از یک بات بوده یا نه، اینکه درخواست از یک کامپیوتر یا موبایل بوده و ... از روی user agent درخواست بدست‌آوریم.

در نمودار۲-۳ میبینیم که برخی از مرورگرهایی که یک بات هستند ترافیک زیادی برای سایت ایجاد کرده‌اند. به علت محدود بودن حجم گزارش، برای مطالعه دیگر بررسی‌ها و نمودارهایی که



نمودار ۳-۳: تعداد درخواست برای مرورگر هر *user agent*

[61] https://pypi.org/project/user-agents/

به شهود ما از دیتاست و درخواست‌های جعلی کمک می‌کند، به اسلایدهای ارائه فاز اول رجوع شود.

۳-۷-۲ مهندسی ویژگی[۶۲]

تا این قسمت شهود خوبی از ویژگی‌ها و عوامل تاثیر گذار بر جعلی بودن یک درخواست از سمت کلاینت گرفته‌ایم. با توجه به دشوار بودن تشخیص anomaly بودن یا نبودن از روی تک درخواست، تصمیم بر این گرفتیم تا دیتاست را براساس IP و User agent گروه‌بندی کنیم. با این کار گویی داریم یک نشست[۶۳] تعریف می‌کنیم. پس در این مرحله سعی بر تعریف و مهندسی‌سازی ویژگی برای هر نشست داریم. تعریف ویژگی ها از یک مقاله[۶۴] پژوهشی الهام گرفته شده است و در جدول ۳-۳ لیست کامل آنرا می‌توانید مشاهده کنید.

ویژگی برای نشست	توجیه و دلیل استفاده از این ویژگی
تعداد درخواست‌ها	زیادبودن تعداد درخواست در یک نشست موجب افزایش احتمال خزنده یا بات بودن آن کاربر می‌شود.
انحراف معیار عمق درخواست‌ها [۶۵]	کاربرهای انسان، درخواست هایی که معمولا در یک نشست می‌زنند دارای عمق path با طول های متفاوتتر نسبت به خزنده‌ها و بات‌ها است.
درصد درخواست‌های از خانواده ۴۰۰	معمولا خزنده‌ها به پاسخ‌هایی از سمت سرور با خطایی از خانواده ۴۰۰ بیشتر مواجه می‌شوند.
درصد درخواست‌های از خانواده ۳۰۰	خانواده ۳۰۰ به معنای redirect شدن به یک صفحه دیگر است. خزنده‌ها و بات‌ها معمولا بیشتر با این پاسخ روبرو می‌شوند.
درصد درخواست‌های با متد HEAD	در خزنده‌ها یا بات ها بیشتر است زیرا احتمال مواجهه با صفحات پاک‌شده یا تاریخ گذشته در این نوع از کاربران زیادتر است.
درصد درخواست‌های عکس	خزنده‌ها و بات‌ها معمولا به عکس ها درخواست نمی‌دهند.
جمع و میانگین response_length و response_time	بدلیل اینکه کاربرهای انسان از مرورگر برای دسترسی به صفحات وب استفاده می‌کنند، وقتی به یک صفحه درخواست می‌زنند، نشست مجبور به دریافت منابع و عکس‌های متفاوتی است و همین باعث زیاد شدن زمان پاسخ و حجم درخواست می‌شود.
ویژگی‌های استخراج شده از user_agent	مرورگر-سیستم‌عامل-بات بودن یا نبودن- از یک pc بودن یا نبودن.

[62] Feature engineering
[63] session
[64] https://link.springer.com/chapter/10.1007/978-3-642-21916-0_52
[65] STD of path's depth

تعداد درخواست‌های به robtots.txt	خزنده‌ها برای اینکه متوجه شوند سرور سایت امکان خزش به چه صفحاتی را داده است، به robots.txt درخواست می‌زنند. اینگونه اطلاعات در آنجا نوشته شده است.

جدول ۴-۳: ویژگی‌های مهندسی شده و استخراج شده از دیتاست برای هر تشست

۳-۸ هفته هشتم – توسعه پروژه نهایی

در این هفته از ویژگی‌های تعریف شده در قسمت قبلی استفاده می‌کنیم تا مدل‌های پایه[۶۶] را آموزش دهیم. به دلیل نداشتن برچسب برای دادگان، باید از مدل های غیرنظارتی بهره می‌بردیم. مدل های غیرنظارتی زیادی در علم یادگیری ماشین برای تشخیص داده پرت[۶۷] تا به الان شناخته شده اند. از معروفترین آنها می‌توان به موارد زیر اشاره نمود:

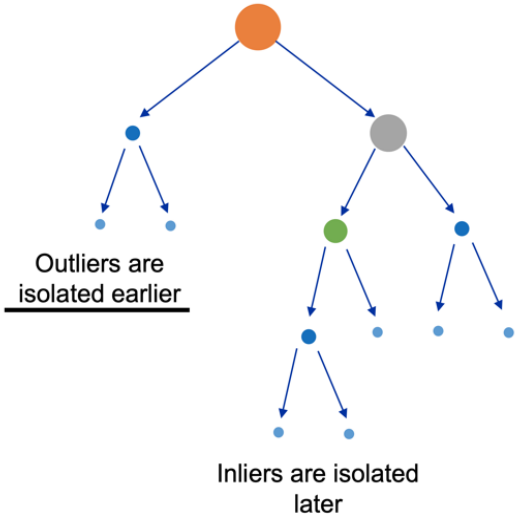
- Isolation Forest
- Local Outlier Factor
- One-class SVM
- Robust covariance

خوشبختانه کتابخانه Scikit-learn[۶۸] این الگوریتم ها را پشتیبانی و پیاده‌سازی کرده است و در این پروژه از آن استفاده کردیم. در این هفته دو مدل پیاده سازی شده است. کدها و نوت‌بوک های مربوط به این قسمت نیز در ریپازیتوری[۶۹] گیتهاب بنده قابل مشاهده هستند.

۳-۸-۱ پیاده سازی مدل Isolation Forest

این روش، تکنیکی برای تشخیص داده‌های پرت به روش غیرنظارتی است. در این روش مشاهده‌ها، با انتخاب ویژگی‌ها به صورت تصادفی و جداسازی مقدار آن به بیشترین و کمترین مقادیر ویژگی انتخابی، ایزوله می‌شوند. به دلیل خاصیت بازگشتی بودن این روش، این روش با یک ساختار درختی قابل نمایش است. بدلیل اینکه مقادیر ویژگی های داده‌های پرت به طرز قابل توجهی با بقیه دادگان تفاوت دارد، داده‌های پرت زودتر در درخت تصمیم ایزوله می‌شوند(شکل ۱-۳). به عبارتی با تنظیم کردن یک آستانه که خود یک ابرپارامتر برای تعداد جداسازی از بالا تا پایین(برگ) درخت میتوان داده‌های پرت را شناسایی کرد. پیاده‌سازی ای که در Scikit-learn برای این روش شده است، یک امتیاز بین -۱ و ۱ با استفاده از تعداد جداسازی ها به هر نمونه از دیتا می‌دهد. هرچه امتیاز به -۱ نزدیک تر باشد، به معنای پرت بودن(در اینجا خزنده یا بات بودن کاربر) می‌باشد.

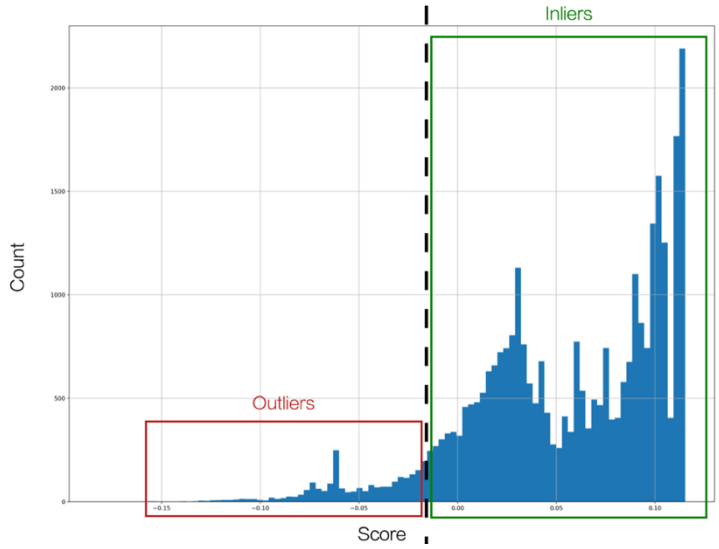
[66] Baseline
[67] Outlier detection
[68] https://scikit-learn.org/stable/modules/outlier_detection.html
[69] https://github.com/mohammadhashemii/Web-Crawler-Detection/tree/master/notebooks



شکل ۸-۳: داده‌های پرت زودتر به برگ درخت می‌رسند.

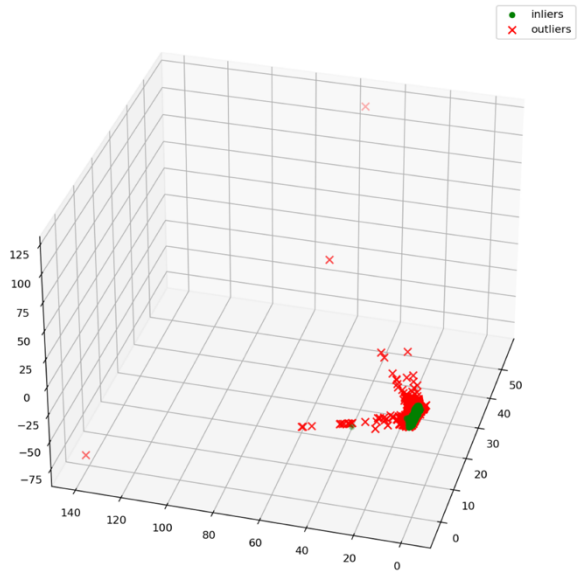
همانطور که گفته شده ابرپارامتر آستانه باید توسط ما تعیین شود و با توجه به نداشتن برچسب برای دادگان، امکان fine-tune کردن این پارامتر وجود ندارد. پس در این مرحله فعلا بصورت تجربی و آزمون‌وخطا این آستانه تعیین شد. در هفته بعد، ارزیابی دقیقتری به عمل خواهد آمد. در نمودار ۳-۳ هیستوگرام امتیازهای نمونه ها مشاهده می‌شود. در این حالت، نمونه‌های با امتیاز کمتر از ۰ به عنوان داده‌پرت در نظر گرفته‌شده‌اند.

با این آستانه گذاری از ۳۱۵۴۱ نشستی که با گروهبندی به کمک IP و User agent بدست‌آمده بود، ۶۱۵ تای آنها داده پرت تشخیص داده شدند. برای اینکه بفهمیم چقدر این مدل درست آموزش دیده، به کمک کتابخانه user_agent و ویژگی is_bot بودن آن دیدیم که ۶۰۳ تا از ۶۱۵ تایی که داده‌پرت تشخیص داده شدند، بات هستند. این به این معنا بود که نسبتا مدل به عنوان یک مدل پایه عملکرد خوبی داشته است. اما هنوز به صورت قطعی بدلیل عدم وجود برچسب برای دادضگان نمی‌توان دقت برای روش و مدل به کار گرفته شده مشخص نمود.



نمودار ۴-۳: تفکیک داده‌های پرت و غیرپرت با
آستانه گذاری بصورت تجربی

.



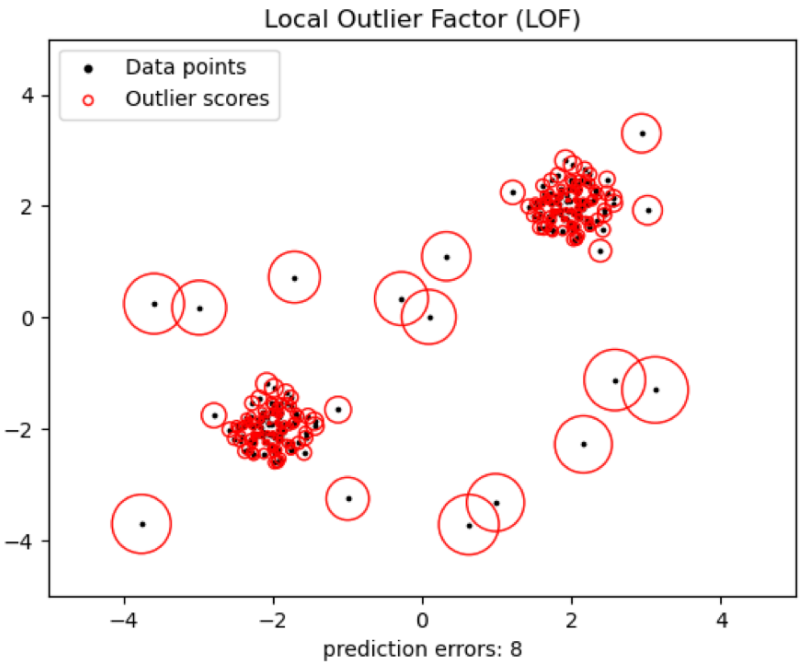
نمودار ۵-۳: مصورسازی دادگان پرت و نرمال به کمک *PCA*

۳-۸-۲ پیاده‌سازی مدل Local Outlier Factor

در روش LOF، انحراف محلی چگالی[۷۰] هر نمونه نسبت به همسایگان خود محاسبه می‌شود. پرت بودن یک داده به میزان ایزوله و تنها بودن آن نمونه نسبت به همسایگان خود سنجیده می‌شود. همچنین چگالی محلی به کمک متریک فاصله‌ای که در روش KNN محاسبه می‌شود صورت

[70] Local deviation of density

می‌گیرد. در نمودار ۵-۳ نیز مشاهده می‌کنید که هرچه یک نمونه همسایگان کمتری داشته باشد،



نمودار ۶-۳: نحوه عملکرد الگوریتم *LOF*

احتمال داده پرت بودن آن نیز بیشتر است.

اما این روش نتایج خوبی برای دیتاستی که ما داشتیم به همراه نداشت زیرا حتی محتمل‌ترین نمونه هایی که به عنوان داده پرت درنظر گرفته بود گاها داده نرمال بودند. به همین دلیل از این روش در ادامه استفاده‌ای نشد.

۳-۹ هفته نهم – توسعه پروژه نهایی (ادامه)

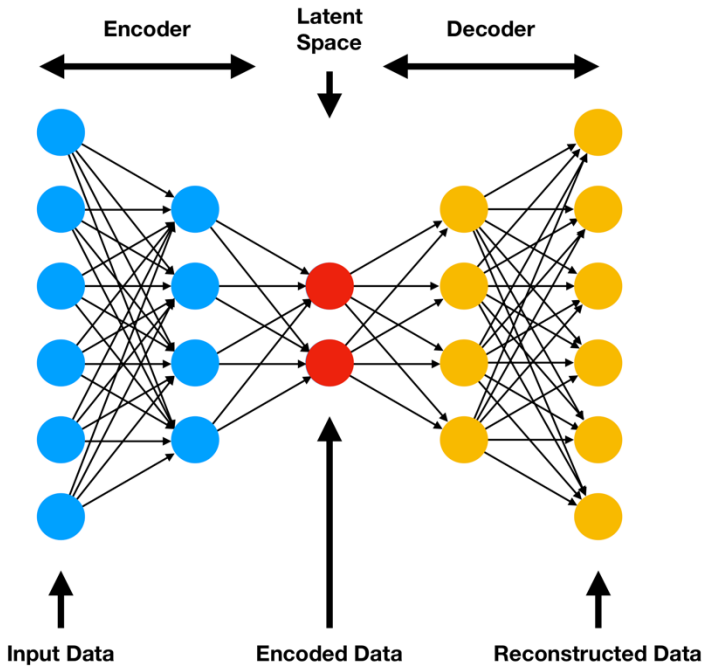
در این هفته، روش ها و الگوریتم های دیگری نظیر اتوانکودرها[۷۱] و الگوریتم های مبتنی بر پردازش زبان‌های طبیعی[۷۲] پیاده‌سازی و بررسی شدند. همچنین ارزیابی بهتر و نسبتا دقیقتری روی خروجی مدل‌ها انجام شد تا از صحت درستی عملکرد الگوریتم‌ها مطمئن شویم. کدها و توضیحات این هفته نیز در ریپازیتوری[۷۳] گیتهاب بنده قابل مشاهده است.

۳-۹-۱ پیاده‌سازی الگوریتم Auto-encoder

اتوانکودرها یک نوع خاصی از شبکه‌های عصبی هستند که مقادیر نورون‌های ورودی را در خروجی کپی می‌کنند. در فرآیند آموزش این نوع شبکه، دیگر نیازی به برچسب داده‌ها وجود ندارد برای همین می‌توان اتوانکودر ها را جزو الگوریتم‌های غیر نظارتی دانست. به طور معمول لایه‌های

[71] Auto-encoder
[72] Natural Language Processing (NLP)
[73] https://github.com/mohammadhashemii/Web-Crawler-Detection/tree/master/notebooks

مخفی در این نوع شبکه‌ها، دارای تعداد نورون کمتری نسبت به نورون‌های ورودی و خروجی دارند. به همین دلیل، لایه‌های مخفی اطلاعات ضـــروری را در خود ذخیره می‌کنند و از نویز ها صـــرف نظر می‌کنند. در فرآیند آموزش این نوع شـــبکه دو قســـمت مهم encode و decode وجود دارد. در مرحله‌ای encoding، مقادیر ورودی را فشرده‌سازی می‌کند و به فضای برداری لایه مخفی می‌برد. در مرحلـه



شکل ۹-۳: قسمت‌های اصلی *auto-encoder* ها

decoding، اطلاعات فشرده‌شده، بازسازی[۷۴] می‌شوند. در شکل ۲-۳ نیز این مراحل مصور شده است.

اتوانکودرها کاربردهای وسیعی در زمینه پردازش تصویر و بینایی ماشین دارند همچنین نتایج درخشـــانی در زمینه تشـــخیص ناهنجاری نیز داشـــته‌اند. در فرآیند decoding، هنگامی که عمل بازســازی صــورت می‌گیرد می‌توان خطای یکســان نبودن داده خروجی با داده ورودی اولیه را حســاب کرد. به عبارت دیگر، لایه مخفی ســـعی بر یادگیری یک embedding از داده‌های ورودی اســـت و می‌خواهد داده‌های ورودی را تنها با داشـــتن ویژگی های موجود در لایه مخفی بازســازی کند. به طبع اگر داده پرتی که اختلاف زیادی از نظر مقادیر ویژگی‌ها با داده‌های دیگر وجود داشـــته باشـــد، خطای بازسازی بسیار زیاد و متفاوت است. پس با این رویکرد می‌توان از اتوانکودر ها برای تشخیص ناهنجاری نیز اســتفاده نمود. در فرآیند آموزش از بهینه‌ســاز[۷۵] Adam و تابع خطای MSE برای محاســبه خطا اســتفاده کردیم. همچنین از ReLu به عنوان تابع غیرخطی‌ســاز اســتفاده کرده‌ایم. دیتاســت با توزیع

[74] Reconstruction
[75] Optimizer

۸۰-۲۰ به دو داده آموزش و تست تقسیم شد و در جدول مقادیر خطا پس از آموزش برای شبکه‌های با معماری‌هایی متفاوت قابل مشاهده است.

تعداد نورون‌ها	خطای داده آموزش	خطای داده تست
[15, 7, 15]	0.42	0.48
[15, 3, 15]	**0.28**	**0.39**
[15, 7, 3, 7, 15]	0.29	0.43
[15, 7, 7, 7, 15]	0.31	0.42

جدول ۵-۳: مقادیر خطای *auto-encoder* با معماری‌های گوناگون

پس از آموزش مدل، برای فاز پیش‌بینی ناهنجاری‌بودن یا نبودن، مشابه الگوریتم‌های قبلی، باید یک حد آستانه برای خطای MSE نیز تعریف کرد. با توجه به دانش قبلی و بررسی بات‌بودن یا نبودن نشست‌های داخل دیتافریم براساس user agent شان، حدود ۵ درصد از دیتاست به واضح بات بودند، بنابراین آستانه خطای MSE به طوری انتخاب شد که ۵ درصد از نمونه ها به عنوان ناهنجاری تشخیص داده شوند.

۳-۹-۲ ارزیابی مدل نهایی

اتوانکودر با توجه به اینکه عملکرد بهتری روی دیتاست داشت به عنوان مدل نهایی انتخاب شد و تصمیم بر این شد تا این مدل به فاز production برود. قبل از آن با اینکه برچسب دیتاها وجود نداشت، ارزیابی نسبتا دقیقی توانستیم ارائه دهیم.

۳-۹-۲-۱ وجود برچسب برای بعضی از نمونه‌ها

شرکت سنجاق، برای اینکه از عملکرد مدل‌های یادگیری ماشین کارآموزان مطلع شود، از روی عمد با استفاده از دو خزنده به وبسایت سنجاق درخواست زده بود و تیم فنی سنجاق توقع داشت مدل های یادگیری ما توانایی تشخیص این دو خزنده را داشته باشند که خوشبختانه مدل اتوانکودر به خوبی این دو نشست را جعلی یا همان خزنده پیش‌بینی کرد.(توضیحات کامل در اسلایدها)

۳-۹-۲-۲ انتظارات ما از مدل

در هفته هفتم، در فاز EDA چندین ویژگی آماری تعریف و مهندسی شدند. به طور مثال یکی از آنها تعداد درخواست‌های نشست بود و ما توقع داشتیم که برای نشست‌های جعلی، تعداد درخواست ها به طور قابل توجهی بیشتر باشد. پس از پیش‌بینی مدل اتوانکودر، میانگین تعداد درخواست‌ها در نشست‌های نرمال و همچنین در نشست های جعلی را مقایسه نمودیم و دقیقا همان چیزی که انتظار می‌رفت رخ داد. در جدول زیر این مقایسه‌ها قابل مشاهده هستند که فقط دو تا از ویژگی‌های تعریف شده نتیجه مطلوبی از خود نشان ندادند:

Average of	# of requests	Path length STD	Percentage of 4xx	Percentage of 3xx	Percentage of HEAD reques	consecutive repeated requests	robots.txt requests	Percentage of image requests
Outliers	231	0.43	3.21%	9.33%	0.34%	0.81	0.08	9.74%
Inliers	25	0.39	0.68%	26%	0.00003	0.62	0.0	28.16%

جدول ۶-۳: میانگین مقادیر ویژگی های به برای هر ویژگی برای هر نشست و مقایسه آنها

۳-۹-۲-۳ گزارش دسته‌بندی[۷۶]

برای اینکه ارزیابی دیگری داشته باشیم، ۱۵۰ تا از مطمئن‌ترین پیش‌بینی های مدل را به صورت دستی برچسب زدیم و ارزیابی‌های کلاسیک مدل‌های نظارتی را برای آنها اعمال نمودیم:

Accuracy	90%
Precision	85.71%
Recall	100%
F1-score	92.30%

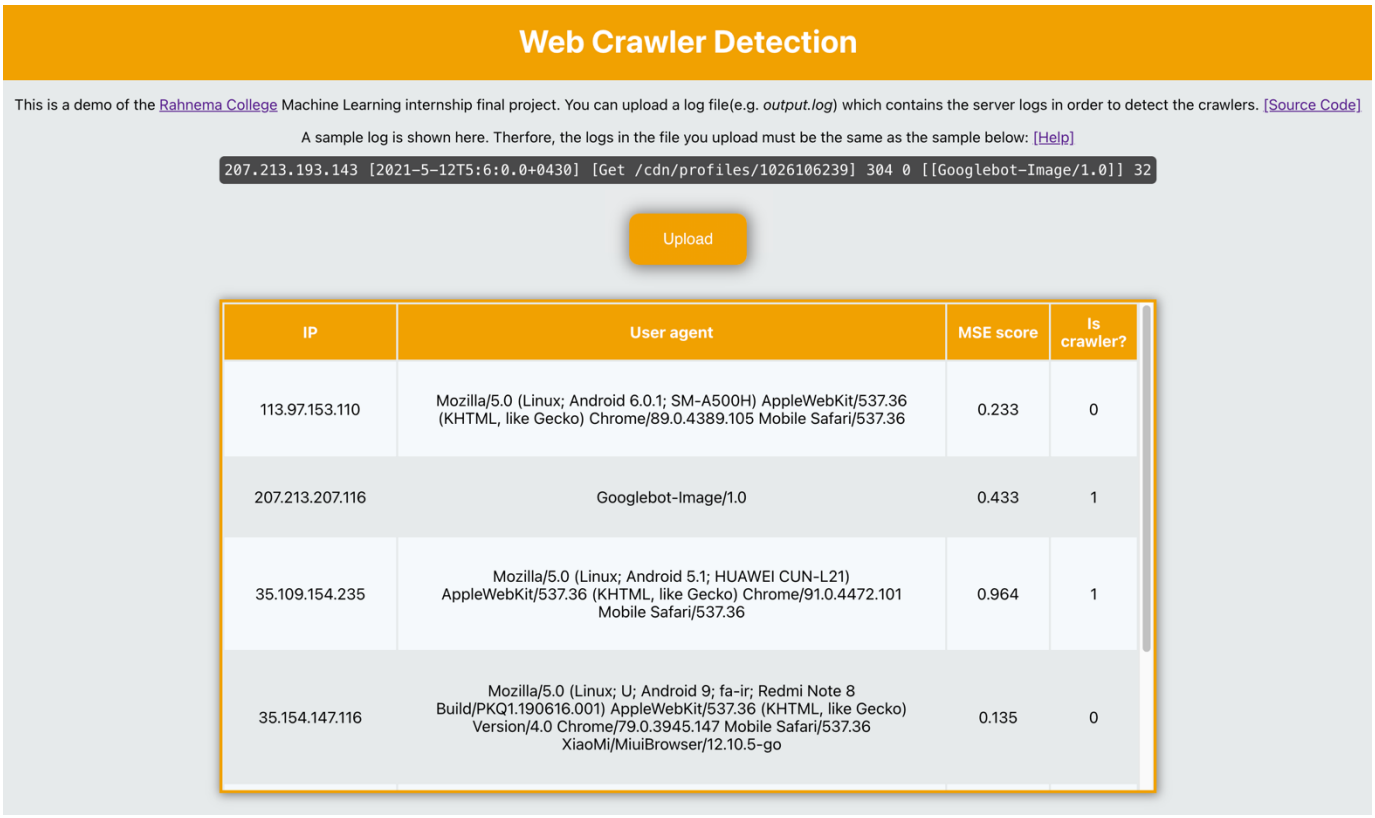
جدول ۷-۳: گزارش دسته‌بندی برای مطمئن‌ترین پیش‌بینی‌ها

۳-۱۰ هفته دهم – ارائه فاز دوم پروژه نهایی

در هفته آخر مدل نهایی به فاز production رسید. با استفاده از تکنولوژی Flask برای

[76] Classification report

بک‌اند و ReactJs برای فرانت‌اند، یک صـفحه‌ی وب طراحی کردیم تا کاربر بتواند با آپلود کردن یک فایل *output.log* از لاگ سـرور خود (مثال)، ناهنجاری‌ها(خزنده‌ها و بات‌ها) را به کمک مدل هوش‌مصنوعی شناسایی کند. در آخر اسلایدهای ارائه فاز نهایی تهیه و تنظیم شد. کدهای مربوط به API در ریپازیتوری[۷۷] گیتهاب بنده نیز قابل مشاهده است. همچنین برای دسترسی به صفحه دموی طراحی از این لینک استفاده کنید. (البته ممکن است گاهی اوقات سرور بالا نباشد و آنوقت برای اجرا گرفتن دمو روی سیستم محلی خود، طبق توضیحات اینجا عمل کنید.)



شکل ۱۰-۳: تصویری از محیط وب‌اپلیکیشن طراحی شده

[77] https://github.com/mohammadhashemii/Web-Crawler-Detection/tree/master/App

۴ خلاصه

هفته اول تا شـــشـــم این دوره کارآموزی اغلب به آموزش و فراگرفتن مطالب مهم در یادگیری ماشـــین ســـپری شـــد. مباحث مهم جبرخطی و آماراحتمال در ریاضـــیات از اســـاســـی ترین مقدمات و قسمت‌هایی است که هر مهندس یادگیری ماشین یا دانشمند داده نیاز دارد تا بتواند بهترین مدل‌ها و الگوریتم‌ها را برای یک تســـک هوش مصـــنوعی انتخاب و پیاده‌ســـازی کند. اینکه یک مهندس توانایی تفکیک بین الگوریتم‌های یادگیری ماشـین متعددی که وجود دارند قائل شـود داشـته باشـد، از شـروط قبولی در تمام آزمون‌های اســـتخدامی برای مشـــاغلی از قبیل دانشـــمند داده، تحلیلگر داده و مهندس یادگیری ماشـــین می‌باشـــد. بنابراین می‌بایســـتی با تمامی الگوریتم‌های این زمینه به اندازه کافی اخت گرفت که تمرین‌های عملی و تدریس‌هایی که در این دوره کارآموزش صـورت گرفت، یک قدم ما را به این هدف نزدیکتر کرد. همچنین تسلط به الگوریتم‌های یادگیری عمیق با توجه به حجیم بودن دیتای روزمره و پیچیده بودن توابعی که قرار اســـت کار تخمین زدن را برای ما انجام بدهند لازم و ضـــروری است.

در هفته‌های هفتم الی دهم، بهترین و معروفترین الگوریتم‌های غیر نظارتی که در ســـال‌های اخیر نتایج بسزایی در تمامی زمینه‌ها از خود نشان دادند مورد مطالعه و پژوهش قرار گرفتند. توانستیم یک ســـیســـتم که قابلیت تشـــخیص جعلی بودن یا نبودن درخواســـت هایی که توســـط کاربران به یک وبســـایت زده می‌شـود را با دقت خوبی داشـته باشـد، پیاده سـازی کنیم. در این پروژه از اتوانکودرها به عنوان یک شبکه‌عصبی با یک رویکرد غیرنظارتی در عین حال دقت بالا، استفاده نمودیم.

نظرات کارآموز جهت بهبود کارآموزی:

شـــرکت رهنماکالج به قطع یکی از بهترین مراکز برای آمادگی کارآموز برای ورود به صـــنعت هوش‌مصـنوعی در ایران اسـت. تمامی اسـاتید و منتورهای این دوره، از مجرب‌ترین ها در شـرکت‌های داخلی معتبر می‌یاشـند. تنها نکته ای که می‌توانم پیشـنهاد کنم به عنوان کسـی که این دوره کارآموزی را گذرانده این اســـت مباحث تئوری ریاضـــیات یعنی جبرخطی و آمارواحتمال این دوره را به شـــدت جدی گرفته، زیرا کمتر جایی این مباحث به صورت حرفه‌ای تدریس می‌شود.