

دستورالعمل و قالب حاضر به منظور یکسان سازی ارائه پایان نامه ها تهیه گردیده است. لطفا از این قالب و نرم افزار Word استفاده نمایید. به منظور صرفه جوئی در وقت و اجرای کامل تر دستورالعمل حاضر، توصیه می گردد که به جای ساختن یک فایل جدید، با ویرایش فایل متن حاضر پایان نامه خود را آماده نمایید.

در متن پایان نامه :

- قلم فارسی : B Nazanin، اندازه ۱۴، حالت عادی
- قلم لاتین (Latin): Times New Roman، اندازه ۱۲، حالت عادی
- فاصله سطرها ۱.۵
- تنظیمات پاراگراف: Right to Left و Justified
- برای ایجاد فهرست شکل ها، در زبانه REFERENCES به قسمت Insert Table of Figures رجوع شود.
- صفحه عنوان فارسی و انگلیسی تطابق کامل داشته باشد.
- پایان نامه باید با صفحه ی عنوان فارسی شروع شود.
- کلیه روابط در متن شماره به ترتیب شماره گذاری شود.



پایان نامه دوره کارشناسی

مهندسی کامپیوتر - گرایش نرم افزار

عنوان پروژه:

طراحی و پیاده سازی سیستم تشخیص مدل های یادگیری

دانشجویان

طراحی و پیاده سازی سیستم پیش بینی نمره دانشجویان

دانشجو:

پانید طاهری

سجاد رحمانی

استاد راهنما:

دکتر کیانیان

خرداد ۱۴۰۲

بسم الله الرحمن الرحيم

چکیده

هوش مصنوعی یکی از علوم مهمی است که امروزه با زندگی ما آمیخته شده است. یکی از حوزه هایی که از این علم استفاده می شود، حوزه درسی و علوم شناختی در مورد مدل یادگیری و پیش بینی نمرات دانشجویان و دانش آموزان است. پیش بینی مدل یادگیری و نمرات، یک موضوعی است که تحقیقاتی در مورد آن انجام شده است. محققان به دنبال آن هستند که ویژگی هایی که بر یادگیری تاثیر دارند را کشف و با استفاده از آن ها مدل یادگیری و نمرات هر دانشجو یا دانش آموز را پیش بینی کنند. امروز با توجه به پیشرفت علم و فناوری و در دسترس بودن اینترنت برای بیشتر مردم، آموزش مجازی یا LMS خیلی مطرح

شده است. محققان سعی می کنند با توجه به فعالیت های هر دانشجو و هر دانش آموز در سامانه های یادگیری مجازی مدل یادگیری و نمرات این دانش آموز را پیش بینی کنند.

کلمات کلیدی

مدل یادگیری، نمرات دانشجویان، پیش بینی، سامانه یادگیری مجازی، هوش مصنوعی

فهرست مطالب (این فهرست جهت نمونه آورده شده، فهرست خودتان را با مشورت با استاد راهنما در این قسمت قرار دهید)

1	فصل ۱
1	مقدمه
1	۱-۱ مقدمه
3	فصل ۲
3	پیشینه
3	۱-۲ مقدمه
3	۲-۲ مدل‌های یادگیری مجازی
4	۲-۲-۱ عاطفی
4	۲-۲-۲ اجتماعی
4	۲-۲-۳ فیزیولوژی
5	۲-۲-۴ روانشناسی
5	۲-۳ پیش‌بینی مدل‌های یادگیری و تحقیقات انجام شده در مورد آن
6	۱-۳-۲ The Educational Testing of the Future
6	۲-۳-۲ National Assessment of Educational Progress (NAEP)
7	۲-۳-۳ Predicting Student Success Using Learning Analytics
7	۲-۳-۴ Open University Learning Analytics dataset (OULAD)
8	۲-۴ الگوریتم‌های هوش مصنوعی و پیش‌بینی مدل‌های یادگیری
11	فصل 3

11	روش
11	1-3 مقدمه
42	فصل 4 نتیجه گیری
42	1-4 مقدمه
43	مراجع
45	واژه نامه انگلیسی به فارسی
46	پیوست ها

فهرست جدول‌ها

جدول 1-2: نتایج حاصل از سنتز دو مدار 46

فهرست شکل‌ها

شکل 1-2: ورودی و خروجی های DHT22.....**Error! Bookmark not defined.**

فصل ۱

مقدمه

۱-۱ مقدمه

پژوهش در مورد بهبود یادگیری دانش‌آموزان و دانشجویان از موضوعاتی بوده که در مورد آن تحقیقات زیادی انجام شده و نتایج خوبی هم کسب شده. برای این امر محققان و پژوهشگران درصدد تقسیم بندی مدل های یادگیری هستند و سپس تولید محتوا برای دانشجو و دانش آموز با توجه به مدل یادگیری اوست. این تحقیقات سعی کرده با کسب اطلاعاتی همچون ارزیابی ها و فعالیت دانشجویان و دانش آموزان مدل های یادگیری را پیش بینی کند؛ همچنین با استفاده از این اطلاعات سعی کردند که نمره دانشجویان را پیش بینی کنند. این تحقیق ها با پرسشنامه های ساده شروع شده و امروز با استفاده از روش های یادگیری ماشین و داده کاوی و اطلاعات سامانه های یادگیری مجازی (LMS) ، مدل یادگیری هر دانشجو و نمره او را پیش بینی می کنند.

یکی از این تحقیقات در Open University (دانشگاهی در بریتانیا) با بیش از ۳۲ هزار دانشجو انجام شد. این پژوهش سعی کرده بود با توجه به فعالیت ها و ارزیابی هایی که دانشجویان در سامانه یادگیری مجازی انجام داده اند ، و روش های یادگیری ماشین و داده کاوی مدل یادگیری و موفقیت دانشجویان را پیش بینی کند.

پیش‌بینی مدل یادگیری و میزان موفقیت دانشجویان به بهبود کیفیت آموزش، مدیریت بهتر کلاس‌های آنلاین، بهبود نرخ موفقیت دانشجویان، بهره‌وری بیشتر از منابع آموزشی و بهبود تجربه دانشجویان کمک می‌کند. با پیش‌بینی مدل یادگیری دانشجو می‌توانیم محتوایی را در اختیار او قرار دهیم که به یادگیری بهتر وی ختم شود. برای مثال اگر کسی مدل یادگیری‌اش دیداری هست، برای یادگیری بهتر وی می‌توان محتوای تصویری تهیه کرد. با این کار یادگیری دانشجو یا دانش‌آموز بهتر می‌شود و می‌تواند نتایج بهتری کسب کند.

در این پژوهش ما سعی کردیم با استفاده از دیتاست پژوهش OULAD و با استفاده از الگوریتم‌های مختلف یادگیری ماشین پیش‌بینی میزان موفقیت دانشجویان و دسته‌بندی مدل یادگیری دانشجویان را بدست آوریم. در این پژوهش قصد داشتیم میزان دقت هر الگوریتم را بسنجیم تا بهترین الگوریتم را برای این کار انتخاب کنیم. علاوه بر این برای پیش‌بینی مدل یادگیری دانشجویان با توجه به میزان فعالیت هر دانشجو در سامانه مجازی آنها را با استفاده از روش k -means به ۵ گروه تقسیم کردیم.

فصل ۲

پیشینه

۲-۱ مقدمه

مدل‌های یادگیری مختلفی برای یادگیری آموزشی در مدارس و دانشگاه‌ها وجود دارد و هر فردی مدل یادگیری مخصوص به خود را دارد. همین علت باعث شده که محققان و پژوهشگران با انجام پژوهش‌های مختلف در صدد پیش‌بینی مدل یادگیری دانشجویان و پیش‌بینی موفقیت دانشجویان هستند. این تحقیقات از سال ۱۹۶۰ با پرسش‌نامه‌های دستی شروع شده و با ثبت فعالیت‌های هر دانشجو در سامانه‌های LMS ادامه دارد.

۲-۲ مدل‌های یادگیری مجازی

یادگیری سنتی و یادگیری آنلاین از بسیاری جهات متفاوت است. به عنوان مثال، دانش آموزان در یک محیط کلاس درس ممکن است ترجیحاتی در مورد صدا، نور و دما داشته باشند، در حالی که این عوامل برای محیط‌های یادگیری آنلاین مناسب نیستند زیرا عنصر اصلی محیط یک صفحه وب تعاملی است. بنابراین، یک مدل سبک یادگیری آنلاین از هشت بعد در کار قبلی ما پیشنهاد شده است. این اثر یادگیری سنتی و آنلاین را در چهار دسته عاطفه، جامعه‌شناسی، فیزیولوژی و روانشناسی مقایسه می‌کند. پس از آن، هشت ویژگی برای مشخص کردن یادگیرندگان آنلاین طراحی شده است، و بررسی رفتارهای یادگیری آنلاین مرتبط با این ویژگی‌ها انجام و تجزیه و تحلیل می‌شود. نتایج

نشان می‌دهد که مدل جدید سبک یادگیری آنلاین ما، زبان‌آموزان آنلاین را متمایز می‌کند و به درک رفتار آنها کمک می‌کند.

۱-۲-۲ عاطفی

مقوله عواطف حول محور این است که یادگیرندگان آنلاین تا چه اندازه یادگیرندگان خودراهربر هستند. بر اساس مدل Entwistle، یادگیرندگان آنلاین با انگیزه خود را تا پایان دوره نظارت می‌کنند و سرعت می‌گیرند، بنابراین ممکن است سوابق تعاملی بیشتری با سیستم‌های آموزش الکترونیکی داشته باشند و تمایل دارند که بر منابع یادگیری نامحسوب نسبت به همتایان بی‌انگیزه خود کلیک کنند. در مقابل، یادگیرندگان غیرفعال به سادگی مطالب و ارزیابی‌های لازم را به پایان می‌رسانند

۲-۲-۲ اجتماعی

یادگیرندگان آنلاین همچنین در نحوه واکنش آنها به تعامل و ارتباط با همسالان متفاوت هستند. برخی بحث را دوست ندارند و ترجیح می‌دهند خودشان مطالعه کنند. دیگران در حمایت از کار گروهی رشد می‌کنند

۳-۲-۲ فیزیولوژی

ویژگی‌های بصری و کلامی به مدل فلدر-سیلورمن اشاره دارد. زبان‌آموزان آنلاین اطلاعات را از منابع مختلف دریافت می‌کنند: دیداری (مانند مناظر، تصاویر، نمودارها و نمادها) و شنیداری (مانند صداها و کلمات). یادگیرندگان دیداری از نظر بصری حساس‌تر هستند و درک بهتری از مطالب ارائه شده به صورت دیداری دارند، در حالی که زبان‌آموزان شنوایی با گوش دادن یا خواندن مطالب، اطلاعات را با عملکرد بهتر به دست می‌آورند. سایر فراگیران با نحوه ارائه مطالب سازگار می‌شوند.

مقوله روانشناسی به راهبردهایی اشاره دارد که دانش آموزان برای درک اطلاعات از آنها استفاده می کنند. ویژگی های حسی و شهودی که به نشانگر نوع مایرز-بریگز (MBTI) اشاره می کند، منعکس کننده چیزی است که فراگیران توجه خود را روی آن متمرکز می کنند. یادگیرندگان حسی مطلب دقیق را بر اساس حقایق ترجیح می دهند، در حالی که یادگیرندگان شهودی مفاهیم، معانی و تداعی ها را ترجیح می دهند. علاوه بر این، ویژگی های متوالی و کلی را از مدل فلدر-سیلورمن معرفی می کنیم زیرا ترتیب ارائه مواد بر کارایی یادگیری تأثیر می گذارد. برخی از آنها به طور متوالی در یک پیشرفت منطقی منظم یاد می گیرند، و برخی دیگر با جهش های شهودی یاد می گیرند تا در نهایت بفهمند. در اکثر مدل های سنتی، ویژگی ها متقابلاً منحصر به فرد هستند. به عنوان مثال، در مدل فلدر-سیلورمن، یک یادگیرنده نمی تواند همزمان متوالی و سراسری باشد. برعکس، مدل ما آن ویژگی ها را با استفاده از بردار هشت بعدی برای مشخص کردن زبان آموزان ترکیب می کند. به عنوان مثال، اگر یادگیرنده نمرات بالایی در هر دو ویژگی متوالی و کلی کسب کند، می توان گفت که یادگیرنده در انتخاب راهبردهای یادگیری بسیار انعطاف پذیر است.

۳-۲ پیش بینی مدل های یادگیری و تحقیقات انجام شده در مورد آن

تحقیقات زیادی برای پیش بینی مدل یادگیری انجام شده است. از سال ۱۹۶۰ اولین تحقیق رسمی شروع شد و تاکنون این تحقیقات ادامه دارد. برای انجام این پژوهش ها از هوش مصنوعی هم کمک گرفته شده است که با استفاده از این علم بتوان مدل های یادگیری دانشجویان و دانش آموزان را بهتر و دقیق تر پیش بینی کرد. در ادامه چند نمونه از تحقیقات انجام شده را بیان کردیم. هدف از بیان این تحقیقات سیر تغییرات و پیشرفت و افزایش دقت در انجام این پژوهش ها بود.

۱-۳-۲ The Educational Testing of the Future

این تحقیق در سال ۱۹۶۰ توسط Russell Ackoff و Fred Emery در مدارس بریتانیا انجام شد. این تحقیق به دنبال آن بود که با برگزاری یک آزمون شامل ۶۰ سوال، دانش آموزان را به ۴ دسته تقسیم کند:

۱. دانش آموزانی که به خوبی در موضوعات مختلف عملکرد خوبی داشتند.
۲. دانش آموزانی که در موضوعات خاصی عملکرد بالایی داشتند، اما در موضوعات دیگر بهترین عملکرد را نداشتند.

۳. دانش آموزانی که در موضوعات مختلف عملکرد متوسطی داشتند.

۴. دانش آموزانی که در بیشتر موضوعات عملکرد ضعیفی داشتند.

این دسته‌بندی، به دانش آموزان و معلمان کمک می‌کرد تا نقاط قوت و ضعف دانش آموزان را شناسایی کنند و برای هر دانش آموز، برنامه‌ی آموزشی مناسبی را طراحی کنند. این تحقیق، به عنوان یک پایه برای تحقیقات بعدی در زمینه مدل‌سازی و پیش‌بینی موفقیت دانش آموزان شناخته شده است.

۲-۳-۲ National Assessment of Educational Progress (NAEP)

National Assessment of Educational Progress یا NAEP یک برنامه ملی آزمون است که به منظور ارزیابی سطح آموزش و پیشرفت دانش آموزان در ایالات متحده آمریکا طراحی شده است. این برنامه تحت نظارت سازمان National Center for Education Statistics (NCES) قرار دارد.

NAEP برای ارزیابی دانش آموزان در سطوح مختلف آموزشی، از طریق اجرای آزمون‌های استاندارد استفاده می‌کند. این آزمون‌ها شامل سوالات چندگزینه‌ای و پاسخ کوتاه هستند و به منظور ارزیابی مهارت‌های خواندن، نوشتن، ریاضی و علوم اجرا می‌شوند.

۲-۳-۳ Predicting Student Success Using Learning Analytics

پژوهش "Predicting Student Success Using Learning Analytics" یکی از مطرح‌ترین تحقیقات در حوزه یادگیری و تحلیل داده‌های آموزشی است. در این پژوهش، با استفاده از تحلیل داده‌هایی که از لرنینگ مدیریت سیستم‌های (LMS) بدست آمده بود، سعی شده است تا با پیش‌بینی موفقیت یا شکست دانشجویان، راهکارهایی برای بهبود عملکرد و پیشرفت دانشجویان ارائه شود.

در این پژوهش، از الگوریتم‌های یادگیری ماشینی و تحلیل داده‌های آموزشی برای پیش‌بینی موفقیت یا شکست دانشجویان استفاده شده است. با استفاده از این الگوریتم‌ها، تحلیل‌هایی بر روی داده‌هایی از دانشجویان انجام شده و با پیش‌بینی موفقیت یا شکست آن‌ها، راهکارهایی برای بهبود عملکرد و پیشرفت دانشجویان ارائه شده است.

هدف این پژوهش ارائه راهکارهایی برای بهبود کارایی و پیشرفت دانشجویان بوده است. به عنوان مثال، با تحلیل داده‌های آموزشی، می‌توان مشخص کرد که فرآیند یادگیری دانشجویان در کدام مرحله مشکل دارد و با ارائه راهکارهایی متناسب با آن مشکل، بهبود عملکرد و پیشرفت دانشجویان را تسریع کرد. از دیگر کاربردهای این پژوهش می‌توان به بهبود فرآیند تدریس و طراحی درس‌ها، بهبود روش‌های ارزیابی دانشجویان و همچنین ارائه بازخورد دقیق‌تر به دانشجویان اشاره کرد.

۲-۳-۴ Open University Learning Analytics dataset (OULAD)

تحقیقات متعددی در دانشگاه باز Open University UK و به ویژه در پروژه OULAD (Open University Learning Analytics Dataset) انجام شده است که به پیش‌بینی موفقیت و شکست دانشجویان در دوره‌های آموزشی این دانشگاه پرداخته‌اند.

این پروژه، داده‌های جمع‌آوری شده از دانشجویانی که در دوره‌های آموزشی Open University UK شرکت کرده‌اند را شامل می‌شود و از آنها برای پیش‌بینی موفقیت دانشجویان استفاده شده است. در

این پروژه از روش‌های یادگیری ماشین، شبکه‌های عصبی و الگوریتم‌های داده‌کاوی برای پیش‌بینی موفقیت دانشجویان استفاده شده است.

در این پروژه، از داده‌های مختلفی مانند اطلاعات شخصی دانشجویان، تعداد بار ورود به سامانه، تعداد بازدید از دروس، نمرات در آزمون‌ها و اطلاعات زمانی استفاده شده است. برای هر دانشجو، یک پروفایل آموزشی تهیه شده و با استفاده از این پروفایل، میزان موفقیت دانشجو در دوره‌ی آموزشی شان پیش‌بینی شده است.

این تحقیقات نشان داده‌اند که با استفاده از داده‌های جمع‌آوری‌شده از دانشجویان، می‌توان به صورت دقیق‌تری پیش‌بینی موفقیت دانشجویان در دوره‌های آموزشی شان را انجام داد و در نتیجه، بهبود کیفیت آموزش و یادگیری را به دنبال داشت.

۴-۲ الگوریتم‌های هوش مصنوعی و پیش‌بینی مدل‌های یادگیری

امروزه، مطرح‌ترین علمی که با زندگی ما آمیخته شده است «هوش مصنوعی» است. از صنعت گرفته تا پزشکی، همگی به نوعی از هوش مصنوعی استفاده می‌کنند. از این علم در پیش‌بینی مدل‌های یادگیری هم استفاده می‌شود. الگوریتم‌های زیادی از هوش مصنوعی در این امر استفاده می‌شود که می‌خواهیم به ۴ مورد از پراستفاده‌ترین این الگوریتم‌ها اشاره کنیم:

- شبکه‌های عصبی (Neural Networks): این الگوریتم به طور گسترده‌ای در مسائل پیش‌بینی مورد استفاده قرار می‌گیرد. این شبکه‌ها معمولاً شامل لایه‌های مختلفی از نورون‌ها هستند که با هم متصل شده‌اند و قادر به یادگیری الگوهای پیچیده هستند.

- درخت تصمیم (Decision Trees): این الگوریتم به صورت درختی عمل می‌کند و برای پیش‌بینی بر اساس شرایط مختلف استفاده می‌شود. درخت تصمیم شامل گره‌ها و شاخه‌هایی است که با توجه به شرایط مختلف، به یکی از دو شاخه مختلف تعلق می‌گیرند.
 - کلاسی فایرهای بیزی (Bayesian Classifiers): این الگوریتم برای پیش‌بینی و تشخیص الگوهای پیچیده با استفاده از روش‌های احتمالاتی استفاده می‌شود. در این الگوریتم، احتمال پیشین و احتمال شرطی برای پیش‌بینی و مدل‌سازی استفاده می‌شود.
 - الگوریتم ایکس جی بوست (XGBoost): یکی از محبوب‌ترین الگوریتم‌های یادگیری ماشینی برای پیش‌بینی و مدل‌سازی است. XGBoost مخفف "eXtreme Gradient Boosting" است و در واقع یک الگوریتم ترکیبی از چندین درخت تصمیم (decision tree) است که با استفاده از روش Gradient Boosting کار می‌کند.
- ما در این پژوهش از الگوریتم random forest (یکی از الگوریتم‌های درخت تصمیم)، naïve bayes و XGboost استفاده کردیم. در هر قسمت معیارهای ارزیابی را بررسی کردیم که بهترین الگوریتم را برای این کار پیدا کنیم.

نتیجه‌گیری

پیش‌بینی مدل یادگیری یکی از عواملی است که با شناخت بهتر و دقیق تر آن می‌توان به دانش‌آموزان و دانشجویان برای یادگیری بهتر، کمک کرد. پژوهش‌هایی برای این امر از سال ۱۹۶۰ تا کنون انجام شده است و همگی این پژوهش‌ها سعی می‌کنند دقیق تر مدل یادگیری را مشخص کنند. هر چه تحقیق‌ها جدید تر شدند، ویژگی‌های بیشتری از فعالیت‌های دانشجویان (مخصوصاً در سامانه‌های یادگیری مجازی) و نمراتی که از ارزیابی‌ها گرفتند، بررسی می‌شوند که هم مدل یادگیری مرتبط با دانشجو به اون پیشنهاد دهیم و هم میزان موفقیت این دانشجو را مشخص کند. الگوریتم‌های هوش مصنوعی مختلفی برای انجام این کار استفاده می‌شود که ما در پی یافتن دقیق‌ترین و بهینه‌ترین آن‌ها برای این کار هستیم.

فصل ۳

روش انجام کار

۳-۱ مقدمه

در این بخش قصد داریم با بررسی فعالیت‌های هر دانشجو در تحقیق OULAD، مدل یادگیری دانشجو را دسته بندی و همچنین میزان موفقیت وی را پیش‌بینی کنیم. برای این امر ابتدا اطلاعات این تحقیق را شرح داده و سپس الگوریتم‌هایی که برای پیش‌بینی میزان موفقیت دانشجو انجام دادیم را معرفی می‌کنیم و در نهایت هر کدام را ارزیابی کردیم.

۳-۲ تحقیق OULAD و Data Set آن

تحقیق OULAD برای پیش‌بینی میزان موفقیت دانشجو و تحلیل و تجزیه رفتار دانشجویان انجام شده است. این تحقیق بر روی ۳۲۵۹۳ نفر در ۲۲ دوره انجام شد. ۱۰۶۵۵۲۸۰ فعالیت در سامانه VLE از دانشجویان ثبت گردیده. این فعالیت‌ها را به انواع مختلفی تقسیم کرده اند که دانشجو با کلیک بر روی هر رویدادی، یکی از فعالیت‌ها برای وی در نظر گرفته می‌شود. اطلاعات جمع آوری شده در این تحقیق در ۸ فایل وجود دارد. در ادامه هر جدول را به طور خلاصه با داده هایی که دارد معرفی می‌کنیم.

۳-۲-۱ فایل studetnInfo

این فایل شامل اطلاعات دانشجویان در دوره‌های آموزشی مختلف است، به شکل یک جدول با ستون‌های مختلف. برخی از ستون‌های موجود در این فایل عبارتند از:

جدول ۱-۳. اطلاعات موجود در جدول studentInfo

عنوان	توضیح
Id_student	شناسه دانشجو
Gender	جنسیت دانشجو
Region	منطقه محل سکونت دانشجو
Highest_education	سطح تحصیلات
Age_band	بازه سنی دانشجو
Num_of_prev_attempts	تعداد بار های شرکت در یک دوره
Studied_credits	تعداد واحد هایی که دانشجو در این درس گذرانده
Disability	آیا دارای معلولیت است یا خیر؟

۳-۲-۲ فایل VLE و StudetnVLE

فایل VLE شامل اطلاعات سامانه یادگیری مجازی و تقسیم بندی فعالیت های موجود در آن است. برای اتصال این فایل و دانشجویان به یکدیگر ما از فایل studentVLE استفاده می کنیم که این دو فایل به وسیله id_site و id_student به یکدیگر وصل شده اند. یعنی در فایل studentVLE متوجه می شویم هر دانشجو در چه روز چه تعدادی یک فعالیت را در سامانه یادگیری مجازی انجام داده است. فایل studentVLE و VLE شامل اطلاعات زیر هستند:

جدول ۲-۳. اطلاعات موجود در جدول VLE

عنوان	توضیح
Id_site	شناسه سایت که منحصر به فرد است
Code_module	کد دوره
Code_presentation	کد تاریخ برگزاری
Activity_type	نوع فعالیت در سامانه
Week_from	هفته شروع برای فعالیت
Week_to	تا هفته چندم این فعالیت بوده

جدول ۳-۳. اطلاعات موجود در جدول studentVLE

عنوان	توضیح
Id_site	شناسه سایت که منحصر به فرد است
Code_module	کد دوره
Code_presentation	کد تاریخ برگزاری
Date	زمان انجام فعالیت
Sum_click	تعداد انجام فعالیت

۳-۲-۳ فایل Assessments و studentAssessment

فایل assessments شامل تمام ارزیابی‌ها انجام شده است. این ارزیابی‌ها در سه قالب انجام شده

است:

- ارزیابی‌هایی که توسط استاد انجام شده است (TMA)

- ارزیابی هایی که توسط سیستم انجام شده است (CMA)

- امتحان ها (EXAM)

بقیه اطلاعات این فایل در جدول ۳-۴ اشاره شده است. برای اتصال فایل ارزیابی ها و دانشجویان بهم ما از فایل studentAssessment استفاده کردیم که در این فایل با استفاده از id_assessment و id_student می توانیم نمره هر دانشجو در هر ارزیابی را بدست بیاوریم. اطلاعات این فایل هم در جدول ۳-۵ وجود دارد.

جدول ۳-۴. اطلاعات موجود در جدول Assessments

عنوان	توضیح
Id_assessment	شناسه ارزیابی
Code_module	کد دوره
Code_presentation	کد تاریخ برگزاری
Assessment_type	نوع ارزیابی که سه حالت دارد
Date	تعداد روز هایی که از شروع دوره گذشته
Weight	میزان تاثیر ارزیابی در نمره پایانی

جدول ۳-۵. اطلاعات موجود در جدول studentAssessment

عنوان	توضیح
Id_assessment	شناسه ارزیابی
Id_student	شناسه دانشجو
Is_banked	آیا نمره ذخیره شده یا خیر
Date_submitted	تعداد روز هایی که طول کشیده تا تایید شود

نمره نهایی	Score
------------	-------

۴-۲-۳ فایل Courses و studentRegistration

در فایل Courses ما تمام دوره هایی که انجام شده را داریم. همان طور که ذکر شد، این تحقیق در ۲۲ دوره انجام شده است. هر دوره یک code_presentation دارد که ابتدا سال را بیان می کند و کاراکتر بعدی ماه را مشخص می کند. برای مثال 2013J به معنی این هست که دوره از اکتبر (ماه دهم) سال ۲۰۱۳ شروع شده است. اطلاعات این فایل در جدول ۳-۶ قرار داده شده است. همچنین برای ارتباط بین این فایل و فایل دانشجویان فایلی به نام studentRegistration ایجاد شده است که اطلاعات این فایل در جدول ۳-۷ ذکر شده است.

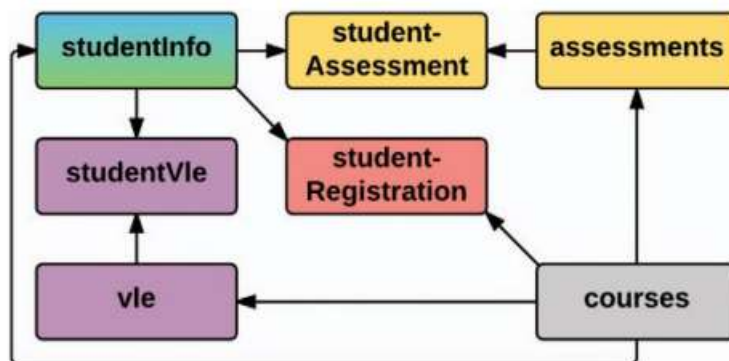
جدول ۳-۶. اطلاعات موجود در جدول Courses

عنوان	توضیح
Code_module	کد دوره
Code_presentation	کد تاریخ برگزاری
module_presentation_length	مدت دوره به تعداد روز

جدول ۳-۷. اطلاعات موجود در جدول studentRegistration

عنوان	توضیح
Code_module	کد دوره
Code_presentation	کد تاریخ برگزاری
Id_student	شناسه دانشجو
Date_registration	تاریخ ثبت نام دانشجو
Date_unregistration	تاریخ لغو ثبت نام دانشجو

می‌توان تمام فایل‌ها و اطلاعات این پژوهش را به صورت شکل ۳-۱ نشان داد.



شکل ۳-۱. ارتباط بین اطلاعات و فایل‌های تحقیق OULAD

۳-۳ فایل data و توضیحات مربوط به آن

ما در این پژوهش سعی کردیم با استفاده از فعالیت‌های دانشجویان در سامانه VLE مدل یادگیری آن‌ها را تقسیم‌بندی کنیم و سپس با توجه به اطلاعات آن میزان موفقیت یک دانشجو را پیش‌بینی کنیم. برای این امر ما با استفاده از فایل VLE تمام نوع فعالیت‌ها را در فایل StudentVle قرار می‌دهیم. این کار را با کمک کد شکل ۳-۲ انجام شد. بعد از این کار ما با استفاده از id_student که در فایل studentVle وجود دارد برای هر نوع فعالیت، تمام کلیک‌های دانشجو را با هم جمع کردیم و در فایل دیتا ما اطلاعات را به صورت شکل ۳-۳ داریم.

```
weight_studentVle = []
for i in range(0, rows_studentVle):
    weight_studentVle.append([studentVle.iloc[i, 3]])

X_studentVle = np.array(weight_studentVle, dtype=np.str_)
print(X_studentVle)

for i in range(0, rows_studentVle):
    for j in range(0, rows_vle):
        if(X_studentVle[i][0] == X_vle[j][0]):
            studentVle.at[i, "activity_type"] = X_vle[j][1]
            break

studentVle.to_excel(r'C:\Users\sajad\Desktop\FinalProject\studentVleTest.xlsx', index=False)
```

شکل ۲-۳. کد اضافه کردن نوع فعالیت در فایل StudentVleTest.

#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
1	id_student	dateplus	dualeans	itemakul	folder	forumng	glossary	homepage	actives	collaborat	contenu	illumine	ou wiki	page	questionnal	quiz	reactive	resource	red sube	subpage	url	trial_result	
2	6516	21	0	0	0	451	0	497	0	0	1305	0	0	0	0	0	0	31	0	145	143	Pass	
3	8462	0	0	12	0	38	0	191	0	13	64	0	18	0	0	0	0	70	0	227	23	Withdrawn	
4	11391	0	0	0	0	293	0	138	0	0	333	0	0	0	0	0	0	13	0	32	5	Pass	
5	23629	0	0	0	0	87	0	36	0	0	0	0	0	0	0	31	0	2	0	5	0	Fail	
6	23696	0	0	0	0	83	0	121	0	0	4	0	0	1	0	576	0	42	0	98	5	Pass	
7	23798	0	0	0	0	145	1	169	0	3	44	0	0	0	0	104	0	21	0	47	56	Distinction	
8	24186	0	0	0	0	14	0	46	0	0	9	0	0	0	0	102	0	8	0	5	0	Pass	
9	24213	0	0	9	0	778	0	325	0	28	262	0	88	0	0	0	0	31	0	427	38	Withdrawn	
10	24391	0	0	0	0	82	2	130	0	0	337	0	0	0	0	120	0	18	0	23	0	Distinction	
11	24734	0	0	0	0	158	0	138	0	1	161	0	0	0	0	0	0	5	0	17	9	Pass	
12	25107	0	0	0	0	1584	2	831	0	0	1	1	0	0	0	85	0	23	0	11	14	Pass	
13	25150	0	0	0	0	148	0	282	0	0	1283	0	191	0	0	125	0	54	0	14	36	Pass	
14	25261	0	0	0	0	306	0	174	0	0	43	0	0	0	0	117	0	39	0	118	5	Withdrawn	
15	25572	0	0	0	0	4	0	44	0	0	8	0	0	0	0	0	0	12	0	39	6	Withdrawn	
16	25629	0	0	0	0	8	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Withdrawn
17	25997	0	0	0	0	0	0	1	0	0	0	0	0	0	0	12	0	0	0	0	0	0	Withdrawn
18	26023	0	0	0	0	118	1	315	0	0	151	0	0	0	0	105	0	83	0	13	0	Pass	
19	26192	9	0	0	0	119	0	525	0	0	1190	0	0	0	0	0	0	8	0	250	162	Distinction	
20	26211	35	4	0	5	1250	2	2396	0	18	8328	0	87	39	41	1711	0	82	0	1086	125	Pass	
21	26247	0	0	0	0	139	0	67	0	0	120	0	27	5	0	38	0	9	0	63	10	Fail	
22	26315	0	0	0	0	255	3	428	0	13	995	0	0	0	0	14	561	0	85	0	140	7	Pass
23	26734	0	0	0	0	36	1	65	0	19	57	0	0	0	0	5	0	31	0	15	1	Fail	
24	26915	0	0	0	0	23	0	15	0	0	17	0	0	0	0	4	0	0	0	2	0	Fail	
25	27116	0	0	0	0	407	0	401	0	52	889	0	0	3	0	448	0	104	0	158	6	Distinction	
26	27189	0	0	0	0	6	336	0	280	0	731	0	3	2	0	1023	0	85	0	245	25	Fail	
27	27417	0	0	1	0	167	10	123	0	12	85	0	36	0	0	0	0	48	0	20	7	Withdrawn	

شکل ۳-۳. نمایشی از فایل data که با استفاده از این فایل قرار است مدل یادگیری را دسته بندی و میزان موفقیت دانشجو را

پیش بینی کنیم.

برای اینکه هیچ خانه ای خالی نماند، اگر دانشجویی در یک نوع فعالیت خاص هیچ رکوردی ثبت نکرده باشد

در داخل آن خانه عدد صفر قرار می دهیم. همچنین فایل دیتا شامل ۲۶۰۷۵ داده هست در حالی که

۳۲۵۹۳ دانشجو داریم. علت این تفاوت به دو دلیل زیر است:

- در فایل studentInfo ما تعداد id_student تکراری داشتیم که حدود ۳۸۰۰ تا بودند.
- حدود ۲۸۰۰ id_student ، هیچ تراکنش و فعالیت ثبت شده ای داخل فایل studentVle نداشتند.

۳-۴ دسته بندی مدل یادگیری دانشجویان

یکی از اهداف ما در این پژوهش این بود که با توجه به فعالیت های دانشجویان در سامانه VLE

مدل یادگیری آن ها را خوشه ی کنیم. ما از روش K-means برای خوشه بندی استفاده کردیم. در این

قسمت قصد داریم ابتدا روش K-means را توضیح دهیم و سپس روش اعمال این روش بر روی داده های

خودمان را شرح دهیم.

۳-۳-۱ خوشه بندی داده‌ها به روش K-means

خوشه‌بندی k-means یک روش کوانتیزه‌سازی برداری است که در اصل از پردازش سیگنال است و هدف آن تقسیم n مشاهده به k خوشه است که در آن هر مشاهده متعلق به خوشه‌ای با نزدیک‌ترین میانگین (مراکز خوشه یا مرکز خوشه) است که به عنوان نمونه اولیه از خوشه این منجر به پارتیشن بندی فضای داده به سلول های Voronoi می شود. در خوشه‌بندی K-means ما داده ها را براساس فاصله اقلیدسی از مرکز خوشه، خوشه‌بندی می‌کنیم. یعنی برای اینکه بدانیم یک داده را در یکی از خوشه‌ها قرار دهیم، فاصله آن داده را تا مرکز هر خوشه حساب می‌کنیم و سپس خوشه‌ای را انتخاب می‌کنیم که داده ما تا مرکز آن خوشه کمترین فاصله را داشته باشد. برای انجام این الگوریتم مراحل زیر را انجام می‌دهیم:

- تعداد خوشه های K را مشخص کنید.
- ابتدا با به هم زدن مجموعه داده ها و سپس انتخاب تصادفی K نقاط داده برای مرکز خوشه‌ها بدون جایگزینی، مرکز خوشه‌ها را راه اندازی کنید.
- به تکرار ادامه دهید تا زمانی که تغییری در مرکز خوشه‌ها ایجاد نشود. یعنی تخصیص نقاط داده به خوشه ها تغییر نمی کند.

برای محاسبه فاصله هر داده از مرکز خوشه، از فرمول اقلیدسی استفاده می‌کنیم:

$$d(x_i, c_j) = \sqrt{\sum_{k=1}^n (x_{ik} - c_{jk})^2}$$

۳-۳-۲ روش انجام خوشه بندی بر روی داده‌ها

همان‌طور که ذکر شده بود ما تعداد تراکنش‌های یک دانشجو را در دوره‌های مختلف بدست آوردیم و در فایل data قرار دادیم. با توجه به این اطلاعات می‌خواهیم مدل یادگیری دانشجویان را خوشه‌بندی کنیم. اما یک سوال را در ابتدا باید برای خودمان پاسخ دهیم: اینکه تعداد خوشه‌های ما چندتاست؟ برای

تعیین تعداد خوشه‌ها ما از روش L-bow استفاده کردیم. برای این کار ما با استفاده از کد شکل ۳-۴، نمودار را رسم کردیم و با توجه به نمودار (شکل ۳-۵) بهترین مقدار برای تعداد خوشه‌ها عدد ۵ بود.

```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt

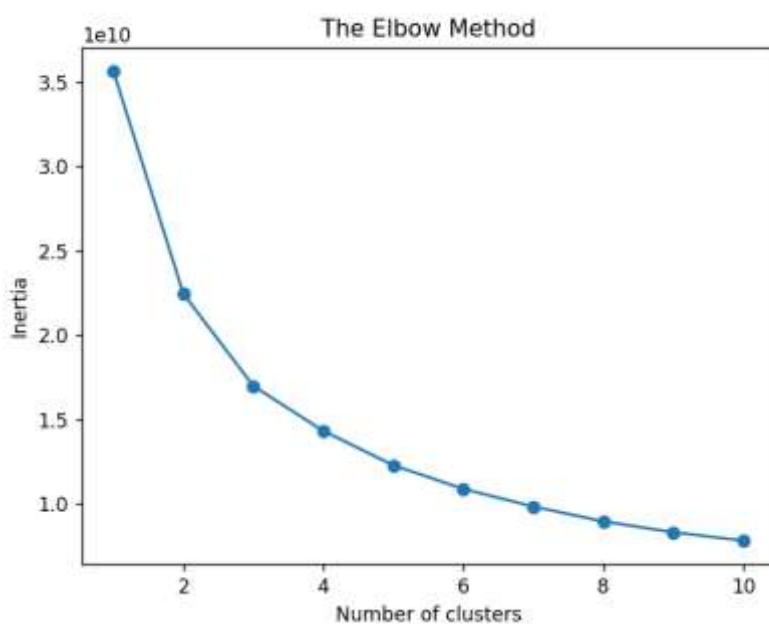
df=pd.read_excel(r'C:\Users\sajad\Desktop\FinalProject\data.xlsx',engine='openpyxl')

df=df.drop(['final_result'] , axis=1)
df=df.drop(['id_student'] , axis=1)

# مقدار تابع هدف برای تعداد مختلفی از خوشه‌ها
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X)
    inertia.append(kmeans.inertia_)

plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.title('The Elbow Method')
plt.show()
```

شکل ۳-۴. مقدار خطاها را برای تعداد خوشه‌های مختلف با استفاده از این کد حساب شده.



شکل ۳-۵. نمودار L-bow که با توجه به نمودار شکستگی در نقطه $k=5$ است.

با توجه به اینکه برای ما مشخص شد که تعداد خوشه‌ها چندتاست پس با توجه به این مورد حال کد مربوط به خوشه‌بندی را به صورت شکل ۳-۶ می‌زنیم.

```
kmeans = KMeans(n_clusters=5, n_init=26075)

kmeans.fit(X)

data_c=kmeans.labels_

for i in range(0,len(data_c)):
    data.at[i , "LearningModel"]=data_c[i]

data.to_excel(r'C:\Users\sajad\Desktop\FinalProject\data.xlsx', index=False)
```

شکل ۳-۶. کد خوشه‌بندی داده‌ها با تعداد خوشه ۵.

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W				
	external	folder	forum	glossary	homepage	mail	activity	collaborate	center	efl	en	wiki	page	journal	quiz	interactive	resource	red	subp	subpage	url	LearningModel	final_result	
1	0	0	451	0	407	0	0	1505	0	0	0	0	0	0	0	31	0	141	143	0	Pass	0		
2	12	0	38	0	191	0	13	84	0	18	0	0	0	0	0	70	0	227	23	0	Withdraw	0		
3	0	0	193	0	138	0	0	532	0	0	0	0	0	0	0	13	0	32	5	0	Pass	0		
4	0	0	87	0	36	0	0	0	0	0	0	0	0	0	31	0	2	0	5	0	Fail	0		
5	0	0	63	0	121	0	0	4	0	0	1	0	0	0	576	0	42	0	98	5	2	Pass	0	
6	0	0	145	1	269	0	3	44	0	0	0	0	0	0	104	0	21	0	47	56	0	Distinction	0	
7	0	0	18	0	46	0	0	9	0	0	0	0	0	0	102	0	8	0	5	0	2	Pass	0	
8	0	0	778	0	225	0	20	262	0	88	0	0	0	0	0	0	51	0	427	26	0	Withdraw	0	
9	0	0	62	2	130	0	0	357	0	0	0	0	0	0	120	0	18	0	23	0	0	Distinction	0	
10	0	0	158	0	138	0	1	161	0	0	0	0	0	0	0	0	5	0	27	0	0	4	Pass	0
11	0	0	1564	2	431	0	0	1	1	0	0	0	0	0	85	0	23	0	21	14	0	Pass	0	
12	0	0	148	0	282	0	0	1283	0	191	0	0	0	0	125	0	14	0	34	16	0	Pass	0	
13	0	0	506	0	174	0	0	43	0	0	0	0	0	0	117	0	36	0	118	5	0	Withdraw	0	
14	0	0	8	0	44	0	0	8	0	0	0	0	0	0	0	0	12	0	39	6	0	Withdraw	0	
15	0	0	8	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Withdraw	0	
16	0	0	0	0	1	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	Withdraw	0	
17	0	0	118	1	115	0	0	131	0	0	0	0	0	0	105	0	83	0	13	0	0	Pass	0	
18	0	0	119	0	525	0	0	1150	0	0	0	0	0	0	0	0	8	0	250	162	0	Distinction	0	
19	0	3	1250	2	2396	0	16	8328	0	87	39	41	1711	0	82	0	1086	125	0	0	0	Pass	0	
20	0	0	139	0	67	0	0	120	0	27	5	0	38	0	0	0	9	0	63	10	0	Fail	0	
21	0	0	255	3	428	0	13	995	0	0	0	0	14	501	0	95	0	140	7	0	Pass	0		
22	0	0	16	1	65	0	19	37	0	0	0	0	0	0	0	0	31	0	15	1	0	Fail	0	
23	0	0	23	0	15	0	0	17	0	0	0	0	0	0	0	0	0	0	2	0	0	Fail	0	
24	0	0	407	0	401	0	52	209	0	0	2	0	448	0	104	0	158	0	0	0	0	Distinction	0	
25	0	6	336	0	280	0	2	731	0	3	2	0	923	0	85	0	245	25	0	0	0	Fail	0	
26	1	0	167	10	123	0	12	85	0	36	0	0	0	0	0	0	49	0	20	7	0	Withdraw	0	

شکل ۳-۷. نتیجه اجرای کد شکل ۳-۶ که در فایل data مدل‌های یادگیری خوشه‌بندی شدند و برچسب 0 الی 4 به داده‌ها تعلق گرفته است.

۵-۳ دسته‌بندی داده‌های دانشجویان برای پیش‌بینی نتیجه نهایی

از مهمترین مفاهیم مورد کاربرد ما در این پژوهش مفهوم طبقه‌بندی (classification) در یادگیری ماشین (machine learning) است. در واقع طبقه‌بندی یکی از روش‌های یادگیری ماشین است که در آن، یک سیستم یادگیری ماشین به دنبال دسته‌ای از نمونه‌ها می‌گردد و سپس هر نمونه را به یکی از دسته‌های

موجود در داده‌های آموزشی نسبت می‌دهد. در اینجا نتیجه نهایی (Final_Result) دانشجویان ما دارای ۴ دسته می‌باشد (Fail , Pass , Withdrawn , Distinction)

به طور کلی، در یک مسئله طبقه‌بندی، هدف این است که سیستم یادگیری ماشین بتواند دسته‌های موجود را بهتر تمیز دهد. در اینجا هدف تشخیص نتیجه نهایی برای هر دانشجو است. برای این کار، در ابتدا داده‌های آموزشی به سیستم یادگیری ماشین ارائه می‌شود. سپس سیستم یادگیری ماشین بر اساس ویژگی‌هایی که از داده‌های آموزشی استخراج می‌کند، یک مدل آموزش داده می‌شود. این مدل به عنوان یک تابعی عمل می‌کند که برای هر نمونه جدیدی که به سیستم وارد می‌شود، مقداری را به عنوان خروجی تولید می‌کند که نشان دهنده دسته‌ای است که نمونه جدید به آن تعلق دارد. که در همین مقاله به نحوه ی آموزش مدل آن با برخی از الگوریتم ها و استفاده از در زبان پایتون خواهیم پرداخت.

برای طبقه بندی الگوریتم های متنوعی وجود دارد :

- الگوریتم درخت تصمیم‌گیری (Decision Tree)

- الگوریتم بیز ساده (Naïve Bayes)

- الگوریتم جنگل تصادفی (Random Forest)

- الگوریتم ماشین بردار پشتیبان (SVM)

- الگوریتم شبکه‌های عصبی

- الگوریتم‌های رده‌بندی بایاس-واریانس (Bias-Variance)

- الگوریتم گرادیان افزایشی بسیار قوی (XGBoost)

در اینجا می‌خواهیم به سه الگوریتمی که در این پژوهش استفاده کردیم بیشتر بپردازیم و مورد بررسی قرار دهیم:

- الگوریتم بیز ساده (Naïve Bayes)

- الگوریتم جنگل تصادفی (Random Forest)

- الگوریتم گرادیان افزایشی بسیار قوی (XGBoost)

۱-۵-۳ الگوریتم بیز ساده (Naïve Bayes)

این الگوریتم بر اساس قانون بیز که یک اصل مهم در آمار و احتمالات محاسباتی است، عمل می‌کند. اساس این قانون این است که با دانستن وقوع یک رویداد، احتمال وقوع یک رویداد دیگر محاسبه می‌شود. در الگوریتم بیز ساده، برای دسته‌بندی داده‌ها، از احتمالات شرطی که بر اساس داده‌های ورودی محاسبه می‌شوند، استفاده می‌شود. به‌طور ساده، الگوریتم بیز ساده داده را به دسته‌ای که احتمال بیشتری دارد، تخصیص می‌دهد. برای محاسبه احتمالات شرطی در الگوریتم بیز ساده، از فرضیات ساده‌ای استفاده می‌شود که به‌عنوان فرض ساده شناخته می‌شود. این فرضیه این است که ویژگی‌های ورودی به‌طور مستقل از هم هستند و احتمال وقوع هر ویژگی به‌صورت جداگانه محاسبه می‌شود و سپس احتمال شرطی دسته‌ها برای هر ویژگی با استفاده از قانون بیز محاسبه می‌شود.

قانون بیز:

$$P(A|B) = P(B|A) \times \frac{P(A)}{P(B)}$$

که در آن $P(A|B)$ نشان‌دهنده احتمال وقوع حادثه A با توجه به رخ دادن B است.
 $P(B|A)$ نشان‌دهنده احتمال وقوع B در صورت رخ دادن حادثه A است.
 $P(A)$ نشان‌دهنده فرضیه اولیه در مورد احتمال وقوع حادثه A است.
 $P(B)$ نشان‌دهنده احتمال وقوع B است.

۲-۵-۳ الگوریتم جنگل تصادفی (Random Forest)

الگوریتم جنگل تصادفی یا Random Forest Algorithm یک الگوریتم یادگیری ماشینی است که برای مسائل کلاس بندی و رگرسیون استفاده می‌شود. این الگوریتم از ترکیب چند درخت تصمیم‌گیری

(Decision Tree) تشکیل شده است و به دلیل قابلیت یادگیری در داده‌های بزرگ و پایداری در برابر انواع داده‌های نویزی به عنوان یکی از الگوریتم‌های پرکاربرد در حوزه یادگیری ماشینی شناخته می‌شود. در الگوریتم جنگل تصادفی، ابتدا تعدادی درخت تصمیم‌گیری با استفاده از روش دسته‌بندی CART (Classification and Regression Trees) ساخته می‌شود. به طور کلی، هر درخت تصمیم‌گیری برای پیش‌بینی خروجی، مجموعه‌ای از قوانین در قالب یک درخت گرافیکی را ارائه می‌دهد. سپس، برای هر نمونه ورودی، تمامی درخت‌های تصمیم‌گیری روی داده‌های آموزشی آموزش داده شده، اعمال می‌شوند و خروجی هر درخت برای هر نمونه محاسبه می‌شود. در نهایت، خروجی نهایی برای هر نمونه به دست می‌آید که از طریق تصمیم‌گیری روی خروجی‌های درخت‌ها به دست می‌آید.

با استفاده از روش جنگل تصادفی، مشکل برازش بیش‌از حد (Overfitting) که در الگوریتم درخت تصمیم‌گیری وجود دارد، کاهش می‌یابد. این روش با تولید تعداد زیادی از درخت‌های تصمیم‌گیری با استفاده از داده‌های آموزشی، از انجام برازش بیش‌از حد خودداری می‌کند و همچنین، با ترکیب خروجی‌های مختلف درخت‌های تصمیم‌گیری، دقت پیش‌بینی را افزایش می‌دهد. با توجه به اینکه الگوریتم جنگل تصادفی یک الگوریتم یادگیری ماشینی پیشرفته است، در ادامه به بیان بیشتر جزئیات مراحل آن می‌پردازیم:

۱. ساخت درخت‌های تصمیم‌گیری: در ابتدا، چندین درخت تصمیم‌گیری ساخته می‌شود که هر کدام از آن‌ها به صورت مستقل از داده‌های آموزشی ساخته می‌شوند.

۲. انتخاب نمونه‌های تصادفی: برای ساخت هر درخت تصمیم‌گیری، برخی از نمونه‌های داده‌های آموزشی به صورت تصادفی انتخاب می‌شوند. این کار باعث می‌شود که درخت‌های تصمیم‌گیری مستقل از هم باشند.

۳. ساخت درخت‌های تصمیم‌گیری با روش دسته‌بندی CART : هر درخت تصمیم‌گیری با استفاده

از روش دسته‌بندی CART ساخته می‌شود. در این روش، برای ساخت هر درخت، به صورت تصادفی یک ویژگی و یک مقدار آن انتخاب می‌شود و سپس برای تقسیم داده‌های آموزشی، به دو دسته براساس آن ویژگی تقسیم می‌شوند. این پروسه تا رسیدن به برگ‌های درخت تصمیم‌گیری ادامه می‌یابد. هر برگ درخت تصمیم‌گیری یک تصمیم نهایی برای پیش‌بینی خروجی ارائه می‌دهد.

۴. پیش‌بینی خروجی با استفاده از درخت‌های تصمیم‌گیری: برای پیش‌بینی خروجی برای هر نمونه ورودی، تمامی درخت‌های تصمیم‌گیری روی داده‌های آموزشی آموزش داده شده، اعمال می‌شوند و خروجی هر درخت برای هر نمونه محاسبه می‌شود. در نهایت، خروجی نهایی برای هر نمونه به دست می‌آید که از طریق تصمیم‌گیری روی خروجی‌های درخت‌ها به دست می‌آید.

۵-۳-۳ الگوریتم گرادیان افزایشی بسیار قوی (XGBoost)

XGBoost (eXtreme Gradient Boosting) یک الگوریتم پر قدرت و کارآمد برای حل مسائل طبقه‌بندی و رگرسیون است که بر پایه الگوریتم Gradient Boosting ایجاد شده است. XGBoost به طور

خاص برای پردازش مجموعه داده‌های بزرگ با ویژگی‌های با ابعاد بالا طراحی شده است. الگوریتم Gradient Boosting با اضافه کردن یادگیرنده‌های ضعیف (معمولاً درخت‌های تصمیم) به مدل کار می‌کند، به طوری که هر یادگیرنده جدید با جبران خطاهای باقیمانده از یادگیرنده قبلی سازگار می‌شود. در XGBoost، یک عبارت انحرافی اضافه می‌شود به تابع هدف برای جلوگیری از بیش‌برازش. علاوه بر این، XGBoost از تکنیک نمونه‌برداری مبتنی بر گرادیان برای انتخاب زیرمجموعه‌ای از ویژگی‌های اطلاعاتی برای هر درخت استفاده می‌کند، که هزینه محاسباتی را کاهش می‌دهد و دقت مدل را افزایش می‌دهد.

XGBoost دارای چندین مزیت نسبت به سایر الگوریتم‌های یادگیری ماشین است، از جمله:

1. دقت بالا XGBoost: نشان داده است که در بسیاری از مجموعه داده‌های بنچ‌مارک، عملکرد برتری دارد.
 2. قابلیت مقیاس‌پذیری XGBoost: می‌تواند با مجموعه داده‌های بزرگ با میلیون‌ها نمونه و هزاران ویژگی کار کند.
 3. کارایی XGBoost: برای بهینه‌سازی بسیار بهینه شده است و مدل‌ها را با سرعتی بسیار بالاتر از دیگر پیاده‌سازی‌های Gradient Boosting آموزش می‌دهد.
 4. پایداری XGBoost: قادر به کار با مقادیر گمراه‌کننده، پرت و سایر نواحی داده‌ای نامنظم است.
 5. انعطاف‌پذیری XGBoost: می‌تواند برای حل مسائل رگرسیون و طبقه‌بندی استفاده شود و می‌تواند با انواع مختلفی از داده‌ها از جمله متن، تصاویر و ویژگی‌های دسته‌ای کار کند.
- XGBoost در حوزه‌های مختلفی مانند مالی، بهداشت، و پردازش زبان طبیعی به گسترش رسیده است. برخی از کاربردهای رایج XGBoost شامل تشخیص تقلب، مدل‌سازی ریسک اعتباری، تشخیص سرطان، و تحلیل احساسات هستند.
- به طور کلی XGBoost یک الگوریتم یادگیری ماشین قدرتمند و انعطاف‌پذیر است که می‌تواند به راه‌حل‌های دقیق و کارآمد برای یک طیف گسترده‌ای از مسائل پیش‌بینی‌ای منجر شود.

۳-۶ مراحل آموزش یک مدل (model training)

۱-۶-۳ جمع‌آوری داده‌ها:

اولین مرحله در آموزش یک مدل ماشین، جمع‌آوری داده‌های مربوط به مسئله است که قرار است مدل به آنها آموزش داده شود. این داده‌ها می‌توانند از منابع مختلفی مانند پایگاه داده‌ها، فایل‌های متنی، تصاویر و ویدئوها جمع‌آوری شوند. که در این پژوهش همانطور که قبلاً توضیح داده شد از مجموعه داده بنچمارک OULAD برای استخراج داده‌ها استفاده شده است.

۲-۶-۳ پیش‌پردازش داده‌ها:

در این مرحله، داده‌های جمع‌آوری شده برای آموزش مدل به شکلی مناسب تبدیل می‌شوند. این شامل انجام کارهایی مانند پاکسازی داده‌ها، حذف داده‌های ناقص و تبدیل داده‌های متنی به بردارهای عددی است که مدل بتواند با آنها کار کند.

در تکه کد های زیر بعد از وارد(import) کردن کتابخانه ها و همینطور داده ی مورد نظر پیش پردازش(analysis) روی آن صورت می گیرد.

```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

شکل ۱-۶-۳ کد افزودن کتابخانه های لازم در این پروژه در زبان پایتون

```
#import dataset

df=pd.read_excel(r'C:\Users\sajad\Desktop\FinalProject\data.xlsx',engine='openpyxl')
```

شکل ۳-۶-۲-۲ کد افزودن فایل اکسل لازم در این پروژه در زبان پایتون

```
print("view dimensions of dataset")  
  
print(df.shape)  
  
print(" preview the dataset")  
  
print(df.head())  
  
print(" view summary of dataset")  
  
print(df.info())
```

شکل ۳-۶-۲-۳ نمایش اطلاعات کلی جدول داده ها در زبان پایتون

```

print("# find categorical variables")

categorical = [var for var in df.columns if df[var].dtype=='O']

print('There are {} categorical variables\n'.format(len(categorical)))

print('The categorical variables are :\n\n', categorical)

print("# view the categorical variables")

print(df[categorical].head())

print("# check missing values in categorical variabl")

print(df[categorical].isnull().sum())

print("# view frequency counts of values in categorical variables")

for var in categorical:

    print(df[var].value_counts())

print("# view frequency distribution of categorical variables")

for var in categorical:

    print(df[var].value_counts()/float(len(df)))

print("# check for cardinality in categorical variables")

for var in categorical:

    print(var, ' contains ', len(df[var].unique()), ' labels')

```

شکل ۴-۲-۳ کد بررسی پیژگی های غیر عددی در زبان پایتون

```

print("Explore Numerical Variables")

print("find numerical variables")

numerical = [var for var in df.columns if df[var].dtype!='O']

print('There are {} numerical variables\n'.format(len(numerical)))

print('The numerical variables are :', numerical)

print("view the numerical variables")

print(df[numerical].head())

print("=====")

```

شکل ۳-۶-۲-۵ کد بررسی ویژگی های عددی در زبان پایتون

۳-۶-۳ تقسیم داده ها به دو مجموعه آموزشی و آزمایشی (train , test):

برای ارزیابی عملکرد مدل، داده ها به دو مجموعه تقسیم می شوند. مجموعه آموزشی برای آموزش مدل استفاده می شود و مجموعه آزمایشی برای ارزیابی عملکرد مدل استفاده می شود. مراحل این فرایند به ترتیب در تکه کد های زیر نمایش داده شده است.

```

# declare feature vector and target

X=df.drop(['final_result'], axis=1)

y=df['final_result']

```

۳-۶-۳-۱ جدا کردن داده های ورودی و برچسب final_result در پایتون

```
#split data into separate training and test set

# split X and y into training and testing sets

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random state = 0)
```

۳-۶-۳-۲ جدا کردن داده های ورودی و برچسب های آموزشی و آزمونی به ۸۰ به ۲۰ درصد در پایتون

train_test_split در بسته sklearn یک تابع بسیار مفید برای تقسیم داده ها به دو بخش

آموزش (train) و آزمون (test) است. این تابع از بسته model_selection در sklearn برای

تقسیم داده های یک مجموعه داده به دو بخش استفاده می شود.

در شکل ۳-۶-۳-۲ تابع train_test_split با گرفتن داده های ورودی X و برچسب ها y و درصد تقسیم

آن ها بین داده های آموزش و آزمون، داده ها را به صورت تصادفی به دو بخش تقسیم می کند.

```
print("# check the shape of X_train")
print(X_train.shape)

print("# check the shape of X_test")
print(X_test.shape)
```

۳-۶-۳-۳ بررسی ابعاد داده های ورودی آموزشی و آزمونی در پایتون

۳-۶-۴. طراحی مدل:

در این مرحله، مدل ماشین برای حل مسئله طراحی می شود. این شامل انتخاب نوع مدل (مانند بیز ساده.

شبکه های عصبی، درخت تصمیم و...) و تعیین پارامترهای مدل (مانند تعداد لایه ها، تعداد نوروها و سایر

پارامترهای مشخص کننده مدل) است. در این قسمت ما تصمیم گرفتیم از مدل بیز ساده (naïve bayes) نام برده استفاده کنیم.

۵-۶-۳. آموزش مدل:

در این مرحله، مدل با استفاده از داده‌های آموزشی آموزش داده می‌شود. این شامل اعمال الگوریتم یادگیری به داده‌های آموزشی و بهبود عملکرد مدل با انجام چرخه‌های آموزش است.

✓ نکته ای بسیار مهم که قبل از آموزش مدل باید در نظر داشت مقیاس بندی ویژگی

ها (Feature scaling) می باشد.

مقیاس بندی ویژگی ها در پیش‌بینی مهم است، این شامل بسیاری از الگوریتم‌های یادگیری ماشین است که برای پیش‌بینی استفاده می‌شوند. به طور کلی، مقیاس بندی ویژگی ها به فرایند نرمال سازی یا استاندارد سازی مقادیر ویژگی‌های ورودی اشاره دارد تا آن‌ها در یک مقیاس مشابه قرار بگیرند. این می‌تواند باعث بهبود عملکرد مدل یادگیری ماشین شود با اطمینان از اینکه ویژگی‌ها به طور مساوی مهم هستند و یکدیگر را به خود نمی‌کشند.

تکنیک‌های مختلفی برای مقیاس بندی ویژگی ها وجود دارند که شامل استاندارد سازی، مقیاس بندی min-max و RobustScaler, Robust scaling می‌شوند. به طور کلی، پیشنهاد می‌شود که قبل از آموزش مدل یادگیری ماشین، به‌ویژه در الگوریتم‌هایی که حساس به مقیاس ویژگی‌های ورودی هستند، مانند K-Nearest Neighbors ، Support Vector Machines و Neural Networks ، مقیاس بندی ویژگی ها انجام شود.

در اینجا ما از روش RobustScaler برای استاندارد سازی داده ها استفاده کرده ایم.

"RobustScaler" یک روش استانداردسازی است که در برابر داده‌های پرت و نویزدار مقاوم است .

معمولا در استانداردسازی داده‌ها، میانگین و واریانس داده‌ها در نظر گرفته می‌شوند و داده‌ها با توجه به این میانگین و واریانس، به یک مقیاس استاندارد نگاشت می‌شوند. با این حال، این روش در برابر داده‌های پرت و نویزدار ضعیف عمل می‌کند و ممکن است خروجی پرتی در برگرداند.

در "RobustScaler" ، به جای استفاده از میانگین و واریانس، از مقادیر میانه و رنج بین کوچکی داده‌ها استفاده می‌شود. به طور دقیق‌تر، در این روش، ابتدا برای هر ویژگی (ستون) از داده‌ها، مقدار میانه و رنج بین کوچکی داده‌ها (با کمک مقدار 25 و 75 درصدی) محاسبه می‌شود. سپس، با استفاده از فرمول زیر، داده‌ها به مقیاس استاندارد نگاشت می‌شوند:

$$x_scaled = (x - median) / IQR \quad \checkmark$$

که در آن، x به داده مورد نظر اشاره دارد، $median$ میانه داده‌های ویژگی مربوطه، و IQR برابر با رنج بین کوچکی داده‌های ویژگی است.

استفاده از "RobustScaler" به دلیل مقاومت بیشتر این روش نسبت به روش‌های سنتی استانداردسازی، به ویژه در صورت داشتن داده‌های پرت و نویزدار، توصیه می‌شود.

```

print("feature scaling")
cols = X_train.columns
from sklearn.preprocessing import RobustScaler

scaler = RobustScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)
X_train = pd.DataFrame(X_train, columns=[cols])
X_test = pd.DataFrame(X_test, columns=[cols])

print(X_train.head())

```

۱-۵-۳ مقیاس بندی ویژگی ها با RobustScaler در پایتون

```

print("#model training")

# train a Gaussian Naive Bayes classifier on the training set
from sklearn.naive_bayes import GaussianNB

# instantiate the model
gnb = GaussianNB()

print("# fit the model")
print(gnb.fit(X_train, y_train))

```

۲-۵-۳ آموزش مدل naïve bayes در پایتون

حال مدل ما آماده ی پیش بینی final_result دانشجویان (fail,pass, distinction, withdrawn) است. در تکه کد زیر برای داده های آزمون (X_test) پیش بینی صورت گرفته است.

```

print("#predct the result")
#=====

y_pred = gnb.predict(X_test)

print(y_pred)

```

۳-۵-۳ پیش بینی برچسب های داده های آزمونی در پایتون

۳-۶-۶. ارزیابی عملکرد مدل (evaluating the performance of machine learning models):

پس از اتمام آموزش مدل، عملکرد آن با استفاده از داده‌های آزمایشی ارزیابی می‌شود. این شامل محاسبه معیارهای ارزیابی مانند دقت، صحت و سایر معیارهای مشخص کننده عملکرد مدل است.

معیارهای مورد استفاده برای ارزیابی عملکرد مدل ممکن است بسته به نوع مسئله و نوع داده‌ها متفاوت باشد. برای مثال، در مسئله این پژوهش که پیش‌بینی دسته‌بندی است، معیارهایی مانند دقت (accuracy)، صحت (precision)، بازخوانی (recall) و امتیاز F1-score استفاده می‌شود.

معمولاً ماتریس درهم‌ریختگی (confusion matrix) نیز برای ارزیابی دقیق‌تر عملکرد مدل استفاده می‌شود. که در ادامه توضیحات بیشتری داده خواهد شد.

۳-۶-۶-۱. معیار ارزیابی دقت (accuracy):

نشان دهنده تعداد نمونه‌هایی است که به درستی توسط مدل دسته‌بندی شده‌اند. به عبارت دیگر، دقت برابر است با تعداد نمونه‌هایی که به درستی تشخیص داده شده‌اند تقسیم بر تعداد کل نمونه‌ها.

با استفاده از تابع `accuracy_score` از ماژول `sklearn.metrics` می‌توان دقت را محاسبه کرد.

تکه کد پایتون زیر دقت (accuracy) پیش‌بینی را برای داده‌های آزمون و آموزش نمایش می‌دهد.

```

print("===== check accuracy score =====")

from sklearn.metrics import accuracy_score

print('Model accuracy score: {0:0.4f}'.format(accuracy_score(y_test,
y_pred)))

print("Compare the train-set and test-set accuracy")

y_pred_train = gnb.predict(X_train)

print('Training-set accuracy score: {0:0.4f}'.format(accuracy_score(y_train,
y_pred_train)))

```

۱-۱-۶-۳ نمایش دقت پیش بینی برای داده های آزمون و آموزش در پایتون

خروجی کد شکل ۱-۱-۶-۳:

Model accuracy score: 0.3143

#Compare the train-set and test-set accuracy

['Distinction' 'Fail' 'Fail' ... 'Distinction' 'Withdrawn' 'Fail']

Training-set accuracy score: 0.3212

علاوه بر این ها در یادگیری ماشین، ما معمولاً با دو مفهوم اصلی "اضافه شدن" (overfitting) و "کم شدن" (underfitting) روبرو هستیم. این دو مفهوم نشان دهنده این هستند که مدل به چه میزان با داده های آموزش سازگار است.

اضافه شدن (overfitting) به موقعیتی گفته می شود که مدل به گونه ای بر روی داده های آموزش بسیار دقیق است که تقریباً به خاطر بسیاری از جزئیات ناهمخوانی در داده های جدید به خوبی عمل نمی کند. برای مثال، اگر یک مدل به صورت بسیار دقیقی بر روی داده های آموزش عمل کند، اما بر روی داده های تست نتایج ناامید کننده ای داشته باشد، به این مفهوم گفته می شود که مدل اضافه شده است.

کم‌شدن (underfitting) به موقعیتی گفته می‌شود که مدل به گونه‌ای نسبتاً خیلی ساده است که با داده‌های آموزش به درستی کار نمی‌کند و در نتیجه نتایج بدی را در همه داده‌ها (آموزش و تست) تولید می‌کند. برای مثال، اگر یک مدل به صورت بسیار ساده‌ای طراحی شود، به گونه‌ای که بر روی داده‌های آموزش هم به درستی کار نکند در نتیجه بر روی داده‌های تست نیز نتایج بدی تولید می‌کند، به این مفهوم گفته می‌شود که مدل کم‌شده است.

```
print("Check for overfitting and underfitting")  
  
# print the scores on training and test set  
  
print('Training set score: {:.4f}'.format(gnb.score(X_train, y_train)))  
print('Test set score: {:.4f}'.format(gnb.score(X_test, y_test)))
```

۲-۱-۶-۳ چک کردن overfitting و underfitting

خروجی کد شکل ۲-۱-۶-۳:

Check for overfitting and underfitting

Training set score: 0.3212

Test set score: 0.3143

با توجه به این خروجی overfitting و underfitting نداریم.

۲-۱-۶-۳ معیار دقت صفر (Null Accuracy) :

علاوه بر این معیارها، معیار دقت صفر (Null Accuracy) نیز برای ارزیابی مدل‌های دسته‌بندی مورد استفاده قرار می‌گیرد. دقت صفر، نسبت تعداد نمونه‌هایی است که با تعلیم مدل، برچسب غیرفعال (برچسب اکثریت کلاس) دریافت می‌کنند به کل تعداد نمونه‌ها. به عنوان مثال، اگر مدل دسته‌بندی با دو کلاس A و B داشته باشیم و تعداد نمونه‌های کلاس A ۴۰۰ و تعداد نمونه‌های کلاس B ۱۰۰ باشد، دقت صفر برابر با ۸۰٪ خواهد بود. به عبارت دیگر، اگر مدلی با دقت کمتر از ۸۰٪ داشته باشیم، می‌توانیم بگوییم که مدل بهتر از یک مدل که همیشه برچسب اکثریت کلاس اکثریت را پیش‌بینی می‌کند، نیست. معیار دقت صفر در برخی موارد می‌تواند مفید باشد، به عنوان مثال در مواردی که کلاس‌ها نامتوازن باشند و تعداد نمونه‌های یک کلاس بسیار بیشتر از نمونه‌های دیگر باشد. در این موارد، مدلی که همیشه برچسب اکثریت را پیش‌بینی کند، می‌تواند دقت بالایی داشته باشد، اما به علت پیش‌بینی نامتوازن، نمی‌تواند به عنوان یک مدل خوب شناخته شود. در این شرایط، معیار دقت صفر می‌تواند به عنوان یک معیار مقایسه‌ای مفید برای این که ببینیم مدل ما به چه میزان بهتر از پیش‌بینی اکثریت عمل می‌کند، مورد استفاده قرار گیرد.

در تکه کد پایتون زیر دقت صفر را محاسبه می‌کنیم:

```
print("Compare model accuracy with null accuracy")

print(y_test.describe())

# check null accuracy score

null_accuracy = (y_test.describe()[3]/y_test.describe()[0])

print('Null accuracy score: {0:0.4f}'.format(null_accuracy))
```

۱-۲-۶-۳ چک کردن null accuracy در پایتون

Compare model accuracy with null accuracy

count 5215

unique 4

top Pass

freq 2169

Name: final_result, dtype: object

Null accuracy score: 0.4159

با مقایسه دقت صفر و دقت (accuracy) متوجه می شویم دقت این پیش بینی پایین تر از دقت صفر بوده و مطلوب نیست.

۳-۶-۶-۳ ماتریس ابهام (confusion matrix) :

مانریس ابهام در این مسئله دسته بندی که چهار نتیجه ممکن دارد (fail, pass, distinction, withdrawn)، در یک جدول ۴x۴ نمایش داده می شود. ردیف های جدول به کلاس های واقعی اشاره دارند و ستون ها به کلاس های پیش بینی شده مربوط می شوند. در این ماتریس ابهام، قطر اصلی (از بالا سمت چپ تا پایین سمت راست) تعداد پیش بینی های صحیح را برای هر کلاس نشان می دهد، در حالی که عناصر خارج از قطر، میزان اشتباهات پیش بینی را نشان می دهند.

چهار مقداری که در ماتریس ابهام نمایش داده می‌شوند عبارتند از:

مثبت واقعی (TP): True Positive تعداد نمونه‌هایی که به درستی به کلاس \hat{A} تشخیص داده شده‌اند.

۲. مثبت کاذب (FP): False Positive تعداد نمونه‌هایی که اشتباهاً به کلاس \hat{A} تشخیص داده شده‌اند، در حالی که در واقع به یکی از سه کلاس دیگر تعلق دارند.

۳. منفی کاذب (FN): False Negative تعداد نمونه‌هایی که به اشتباه به یکی از سه کلاس دیگر تشخیص داده شده‌اند، در حالی که در واقع به کلاس \hat{A} تعلق دارند.

۴. منفی واقعی (TN): True Negative تعداد نمونه‌هایی که به درستی به هیچ یک از چهار کلاس تشخیص داده نشده‌اند.

تکه کد پایتون زیر ماتریس ابهام این پیش‌بینی را به دست آورده و نشان می‌دهد:

```
print("==== Confusion matrix =====")
print("# Print the Confusion Matrix and slice it into four pieces")

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred, labels=gnb.classes_)

print('Confusion matrix\n\n', cm)

print("# visualize confusion matrix with seaborn heatmap")

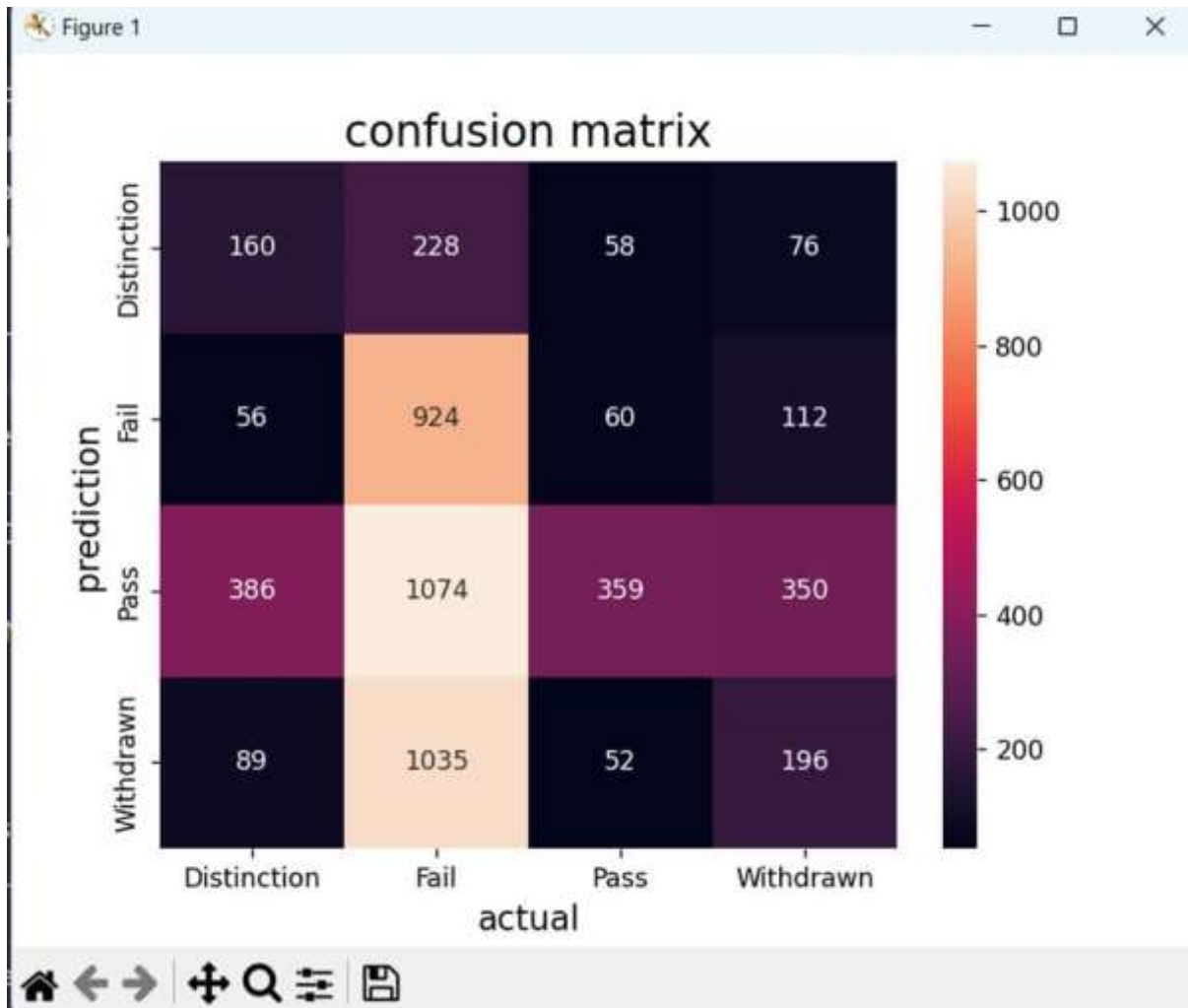
cm_matrix = pd.DataFrame(data=cm)

sns.heatmap(cm_matrix, annot=True, fmt='d', xticklabels=gnb.classes_,
            yticklabels=gnb.classes_)

plt.ylabel('prediction', fontsize=13)
plt.xlabel('actual', fontsize=13)
plt.title('confusion matrix', fontsize=17)
plt.show()
```

۳-۶-۳-۱ نمایش confusion matrix در پایتون

خروجی کد شکل ۳-۶-۳-۱:



۳-۶-۳-۱ confusion matrix ۴*۴. سطر ها نشان دهنده پیش‌بینی برچسب و ستون ها نشان دهنده برچسب واقعی است.

۳-۶-۷. بهبود عملکرد مدل:

در صورتی که عملکرد مدل نسبت به ارزیابی قابل قبول نبود، می‌توان با تغییر پارامترهای مدل، بهبود عملکرد آن را ایجاد کرد. این شامل تغییر پارامترهایی مانند تعداد لایه‌ها، تعداد نرون‌ها، نوع الگوریتم یادگیری و سایر پارامترهای مشخص کننده مدل است.

پس از مشاهده ی پایین بودن دقت در مراحل بعد با تغییر نوع الگوریتم یادگیری از بیز ساده به جنگل تصادفی و بعد هم به الگوریتم گرادیان افزایشی بسیار قوی (XGBoost) شاهد بهبود عملکرد مدل بودیم.

۸-۶-۳. استفاده از مدل:

پس از آموزش و بهبود عملکرد مدل، آن را می توان برای حل مسئله مورد نظر استفاده کرد. این شامل پیش بینی خروجی مدل برای داده های جدید است.

✓ همچنین، در هر مرحله از آموزش مدل، نیاز است که بهبود عملکرد و اعتبارسنجی مدل بررسی

شود و در صورت نیاز، پارامترها و روش های آموزش مدل تغییر داده شود.

فصل 4

نتیجه گیری

4-1 مقدمه

مراجع

در این قسمت، تمامی مراجع استفاده شده در متن فهرست می‌شوند. تمام مراجع مندرج در این بخش، باید حداقل یک‌بار در متن اصلی استفاده شده باشند. تمام مراجع لازم است با فرمت یکسان مانند مثال زیر نمایش داده شوند.

[1] Smith J., Jones A., Andersen P., Jameson G., “Here is the title of this particular journal paper,” *Journal Name*, 31(2), 105–112, 2015.

شیوه‌ی ارائه مراجع فارسی و انگلیسی

1. نام خانوادگی، نام (مولفان و مترجمان)؛ *عنوان/اصلی کتاب*؛ عنوان فرعی کتاب (جزئیات عنوان کتاب در صورت وجود داخل پرانتز)، نام سایر افراد دخیل در تالیف یا ترجمه، ناشر، محل انتشار، شماره جلد، شماره ویرایش، سال انتشار به عدد.

2. منهای، محمدباقر؛ *هوش محاسباتی* (جلد اول: مبانی شبکه‌های عصبی)، انتشارات دانشگاه صنعتی امیرکبیر، تهران، ویرایش اول، 1379.

3. نام خانوادگی، نام؛ نام خانوادگی و نام مولف دوم؛ مولف سوم؛ “عنوان مقاله به صورت عادی و داخل گیومه”، نام کامل مجله به صورت *ایتالیک*، شماره دوره یا جلد، شماره مجله، شماره صفحات، سال انتشار.

4. نام خانوادگی، نام مجری؛ *عنوان طرح پژوهشی به صورت ایتالیک*، شماره ثبت، نام کامل محل انجام و سفارش دهنده، سال انجام طرح.

5. نام شرکت/نام فرد؛ عنوان صفحه؛ *آدرس/اینترنتی*.

6. Book authors’ names; *Book Title in Italic* (and the title components, if any), Edition number, Publisher, Date of publish.

7. Van de Vegte, J.; *Feedback Control Systems*, 2nd Edition, Prentice Hall, 1990

8. Authors’ names separated by comma-dots; “The paper title in Regular Times New

Roman 12pt”, *Paper Address in Italic*, Publishing Place, paper page, Year of Publish.

9. Safonov, M.; “Stability margins of diagonally perturbed multivariable feedback systems”, *IEEE Proceedings*, Part D, p. p. 251-256, Nov. 1982.
10. Company Name/ Person Name; Page Title; *Internet Address*.

واژه نامه انگلیسی به فارسی

در شرایطی که اصطلاحات مورد استفاده نو یا در ادبیات فنی، ناآشنا باشد، تدوین واژه‌نامه برای پایان‌نامه/رساله توصیه می‌شود. لازم به ذکر است تهیه این واژه‌نامه اختیاری بوده و ضرورت تهیه آن را استاد راهنما تعیین می‌نماید.

تهیه متن واژه‌نامه به شرح زیر توصیه می‌شود:

A

Academic

دانشگاهی

Analytical

تحلیلی

پیوست‌ها

در صورت لازم، این قسمت ذکر میشود و در غیر این صورت آن را حذف کنید. این قسمت مطالبی که مرتبط با متن پایان نامه بوده اما ذکر آنها در متن اصلی موجب خستگی خواننده یا سردرگمی او شود، گنجانده می شود.

پیوست الف: عنوان پیوست

جدول ۱-۲ به صورت نمونه پیشنهاد شده است.

جدول 0-1: نتایج حاصل از سنتز دو مدار

n	Delay (ns)	Area (μm^2)	Power (mW)
N=8	1.90	1716.00	556.20
N=16	2.70	3641.00	1200.01

Abstract

In this part, the abstract of the thesis must be presented, including supposes, aims and results.

In this part, the abstract of the thesis must be presented, including supposes, aims and results.

In this part, the abstract of the thesis must be presented, including supposes, aims and results.

In this part, the abstract of the thesis must be presented, including supposes, aims and results.

Keywords: ..., ..., ..., ..., ...



Shahid Rajaee Teacher Training University

Faculty of Computer Engineering

Department of.....

B. Sc. Thesis

Title:

.....

Supervisor:

Dr.

By:

.....

Winter 2020