پایان‌نامه دوره کارشناسی

مهندسی کامپیوتر- گرایش نرم افزار

عنوان پروژه:

طراحی و پیاده سازی سیستم تشخیص مدل های یادگیری جهت پیش بینی نمره دانشجویان

دانشجو:

پانیذ 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 اســـتفاده می شـــود، حوزه درســـی و علوم شـــناختی در مورد مدل یادگیری و پیش بینی نمرات دانشـجویان و دانش آموزان اسـت. پیش‌بینی مدل یادگیری و نمرات، یک موضـوعی اسـت که تحقیقاتی در مورد آن انجام شده است. محققان به‌دنبال آن هستند که ویژگی هایی که بر یادگیری تاثیر دارند را کشف و با اســـتفاده از آن‌ها مدل یادگیری و نمرات هر دانشـــجو یا دانش‌آموز را پیش‌بینی کنند. امروز با توجه به پیشرفت علم و فناوری و در دسترس بودن اینترنت برای بیشتر مردم، آموزش مجازی یا LMS خیلی مطرح شـده اسـت. محققان سـعی می کنند با توجه به فعالیت های هر دانشـجو و هر دانش‌آموز در سـامانه های یادگیری مجازی مدل یادگیری و نمرات این دانش‌آموز را پیش‌بینی کنند.

کلمات کلیدی

مدل یادگیری، نمرات دانشجویان، پیش‌بینی، سامانه یادگیری مجازی، هوش مصنوعی

فهرست مطالب

فهرست جدول‌ها

فهرست شکل‌ها

فصل ۱

مقدمه

۱-۱ مقدمه

پژوهش در مورد بهبود یادگیری [1]دانش‌آموزان و دانشـــجویان از موضـــوعاتی 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 هسـتند و سـپس تولید محتوا برای دانشـجو و دانش‌آموز با توجه به مدل یادگیری اوست[2]. این تحقیقات سعی کرده با کسب اطلاعاتی همچون ارزیابی ها و فعالیت دانشجویان و دانش‌آموزان مدل های یادگیری را پیش بینی کند؛ همچنین با اسـتفاده از این اطلاعات سـعی کردند که نمره دانشـجویان را پیش بینی کنند. این تحقیق ها با پرسـشـنامه [2]های سـاده شـروع شـده و امروز با اسـتفاده از روش های یادگیری ماشین و داده کاوی و اطلاعات سامانه های یادگیری مجازی (LMS) ، مدل یادگیری هر دانشجو و نمره او را پیش‌بینی می‌کنند[1].

یکی از این تحقیقات در Open University (دانشگاهی در بریتانیا) [1] با بیش از ۳۲ هزار دانشجو انجام شـــد. این پژوهش ســـعی کرده بود با توجه به فعالیت ها و ارزیابی هایی که دانشـــجویان در ســـامانه

[1] Improve learning

[2] questionnaire

یادگیری مجازی انجام داده اند ، و روش های یادگیری ماشـــین [1]و داده کاوی [2]مدل یادگیری و موفقیت دانشجویان را پیش بینی کند[1].

پیش‌بینی مدل یادگیری و میزان موفقیت دانشـــجویان به بهبود کیفیت آموزش، مدیریت بهتر کلاس‌های آنلاین، بهبود میزان موفقیت دانشـــجویان[3]، بهره‌وری بیشـــتر از منابع آموزشـــی و بهبود تجربه دانشـجویان کمک می‌کند. با پیش‌بینی مدل یادگیری دانشـجو 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 تصـویری تهیه کرد. با این کار یادگیری دانشـجو یا دانش‌آموز بهتر می‌شـود و می تواند نتایج بهتری کسب کند[12].

در این پژوهش ما سـعی کردیم با اسـتفاده از دیتاسـت پژوهش OULAD و با اسـتفاده از الگوریتم های مختلف یادگیری ماشین پیش بینی میزان موفقیت دانشجویان و دسته بندی 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 پیش بینی مدل یادگیری دانشـــجویان باتوجه به میزان فعالیت هر دانشجو در سامانه مجازی آنها را با استفاده از روش k-means به ۵ گروه تقسیم کردیم.

[1] machine learning

[2] Data analysis

[3] The success rate of students

فصل ۲

پیشینه

۲-۱ مقدمه

مدل‌های یادگیری مختلفی برای یادگیری آموزشی در مدارس و دانشگاه ها وجود دارد و هر فردی مدل یادگیری مخصوص به خود را دارد. همین علت باعث شده که محقیقن و پژوهش‌گران با انجام پژوهش‌های مختلف در صدد پیش بینی مدل یادگیری دانشجویان و پیش‌بینی موفقیت دانشجویان هستند. این تحقیقات از سال ۱۹۶۰ با پرسش‌نامه‌های دستی شروع شده و 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، [2] در حالی که این عوامل برای محیط های 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، جامعه شناسی، فیزیولوژی و روانشناسی مقایسه می کند. پس از آن، هشت ویژگی برای مشخص کردن یادگیرندگان آنلاین طراحی شده است، و بررسی رفتارهای یادگیری آنلاین مرتبط با این ویژگی ها انجام و تجزیه و تحلیل می شود. نتایج

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

۲-۲-۱ عاطفی[1]

مقوله عواطف حول محور این اســـت که یادگیرندگان آنلاین تا چه اندازه یادگیرندگان خودراهبر هســـتند. بر اســـاس 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 نامحبوب نســـبت به همتایان بی‌انگیزه خود کلیک کنند. در مقابل، یادگیرندگان غیرفعال به سادگی مطالب و ارزیابی های لازم را به پایان می رسانند

۲-۲-۲ اجتماعی[2]

یادگیرندگان آنلاین همچنین در نحوه واکنش آنها به تعامل و ارتباط با همســـالان متفاوت هســـتند. برخی بحث را دوســـت ندارند و ترجیح می دهند خودشـــان مطالعه کنند. دیگران در حمایت از کار گروهی رشد می کنند

۲-۲-۳ فیزیولوژی [3]

ویژگی های بصری و کلامی به مدل فلدر-سیلورمن اشاره دارد. زبان آموزان آنلاین اطلاعات را از منابع مختلف دریافت می کنند: دیداری (مانند مناظر، تصاویر، نمودارها و نمادها) و شنیداری (مانند صداها و

---

1 Emotional

2 social

3 Physiology

کلمات). یادگیرندگان دیداری از نظر بصری حساس تر هستند و درک بهتری از مطالب ارائه شده به صورت دیداری دارند، در حالی که زبان آموزان شنوایی با گوش دادن یا خواندن مطالب، اطلاعات را با عملکرد بهتر به دست می آورند. سایر فراگیران با نحوه ارائه مطالب سازگار می شوند.

۲-۲-۴ روانشناسی[1]

مقوله روانشناسی به راهبردهایی اشاره دارد که دانش آموزان برای درک اطلاعات از آنها استفاده می کنند. ویژگی‌های حسی و شهودی که به نشانگر نوع مایرز-بریگز (MBTI) اشاره می‌کند، منعکس‌کننده چیزی است که فراگیران توجه خود را روی آن متمرکز می‌کنند. 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، ویژگی‌های متوالی و کلی را از مدل فلدر-سیلورمن معرفی می‌کنیم زیرا ترتیب ارائه مواد بر کارایی یادگیری تأثیر می‌گذارد. 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.

[1] Psychology

۲-۳ پیش بینی [1]مدل های یادگیری و تحقیقات انجام شده در مورد آن

تحقیقات زیادی برای پیش‌بینی مدل یادگیری انجام شده است. از سال ۱۹۶۰ اولین تحقیق رسمی شروع شد و تاکنون این تحقیقات ادامه دارد. برای انجام این پژوهش‌ها از هوش مصنوعی هم کمک گرفته شده است که با استفاده از این علم بتوان مدل‌های یادگیری دانشجویان و دانش‌آموزان را بهتر و دقیق تر پیش‌بینی 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.

۲-۳-۱ The Educational Testing of the Future

این تحقیق در سال ۱۹۶۰ توسط Russell Ackoff و Fred Emery در مدارس بریتانیا انجام شد[7]. 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.

[1] Forecast

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[8]. 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 استانداردی 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 مدیریت سامانه های مجازی یادگیری (LMS[1]) بدست آمده بود، سعی شده است تا با پیش‌بینی موفقیت یا شکست دانشجویان، راهکارهایی برای بهبود عملکرد و پیشرفت دانشجویان ارائه شود[9].

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.

[1] Learning Management System

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 پروژه‌ی OULAD[1] انجام شده است که به پیش‌بینی موفقیت و شکست دانشجویان در دوره‌های آموزشی این دانشگاه پرداخته‌اند[1].

این پروژه، داده‌های جمع‌آوری‌شده 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[2] 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.

---

1 Open University Learning Analytics Dataset

2 Neural Networks

۲-۴ الگوریتم های هوش مصنوعی و پیش‌بینی مدل های 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[4]. الگوریتم های زیادی از هوش مصنوعی در این امر استفاده می‌شود که می‌خواهیم به ۴ مورد از پراستفاده ترین این الگوریتم‌ها اشاره کنیم:

- شبکه‌های عصبی [1]: 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 [2]: 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[3]: 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[4]: XGBoost یکی از محبوب‌ترین الگوریتم‌های یادگیری ماشینی برای پیش‌بینی و مدل‌سازی است. XGBoost مخفف "eXtreme Gradient Boosting" است و در واقع

---

1 Neural Networks

2 decision tree

3 Bayesian Classifiers

4 XGBoost

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 الگوریتم های 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. پژوهش هایی برای این امر از سال ۱۹۶۰ تا کنون انجام شده است و همگی این پژوهش ها سعی می‌کنند دقیق تر مدل یادگیری را مشخص کنند. هر چه تحقیق ها جدید تر شدند، ویژگی‌های بیشتری از فعالیت های دانشجویان (مخصوصا در سامانه های یادگیری مجازی) و نمراتی که از ارزیابی ها گرفتند، بررسی می‌شوند که هم مدل یادگیری مرتبط با دانشجو به اون پیشنهاد دهیم و هم میزان موفقیت این دانشجو را مشخص کند. الگوریتم های هوش مصنوعی مختلفی برای انجام این کار استفاده می‌شود که ما در پی یافتن دقیق ترین و بهینه ترین 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

دانشـجویان ثبت گردیده. این فعالیت‌ها را به انواع مختلفی تقسـیم کرده اند که دانشـجو با کلیک بر روی هر رویدادی، یکی از فعالیت‌ها برای وی در نظر گرفته می‌شـود. اطلاعات جمع آوری شـده در این تحقیق در ۷ فایل وجود دارد. در ادامه هر جدول را به طور خلاصه با داده هایی که دارد معرفی می‌کنیم. [1]

۳-۲-۱ فایل studentInfo

این فایل شـامل اطلاعات دانشـجویان در دوره‌های آموزشـی مختلف اسـت، به شـکل یک جدول با ستون‌های مختلف. برخی از ستون‌های موجود در این فایل عبارتند از:

جدول۳-۱ 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 فعالیت‌های موجود در آن است. برای اتصـال این فایل و دانشـجویان به یکدیگر ما از فایل studentVLE اسـتفاده می‌کنیم که این دو فایل به

وسـیله id_site و id_student به یکدیگر وصـل شـده اند. یعنی در فایل studentVLE متوجه می‌شـویم 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 ۳-۲. اطلاعات موجود در جدول VLE

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

۳-۲-۳ فایل Assessments و studentAssessment

فایل assessments شامل تمام ارزیابی ها انجام شده است. این ارزیابی ها در سه قالب انجام شده است:

- ارزیابی هایی که توسط استاد انجام شده است [1]
- ارزیابی هایی که توسط سیستم انجام شده است[2]
- امتحان ها [3]

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 می‌توانیم 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_presention	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

[1] TMA
[2] CMA
[3] EXAM

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
Score	نمره نهایی

۳-۲-۴ 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 شـده اسـت. هر دوره یک code_presentation دارد که ابتدا سـال را بیان می‌کند و کاراکتر بعدی ماه را مشخص می‌کند. برای مثال 2013J به معنی این هست که دوره از اکتبر [1](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 دانشـجویان فایلی به نام studentRegistration ایجاد شـده اسـت که اطلاعات این فایل در جدول ۳-۷ ذکر شده است.

جدول ۳-۶. اطلاعات موجود در جدول Courses

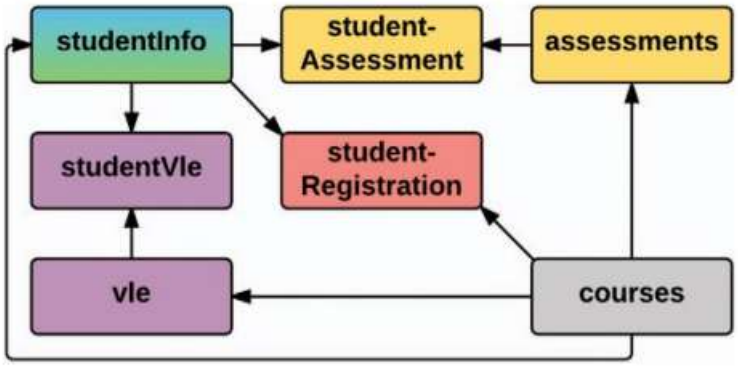
عنوان	توضیح
Code_module	کد دوره
Code_presention	کد تاریخ برگزاری
module_presentation_length	مدت دوره به تعداد روز

[1] October

جدول ۳-۷. اطلاعات موجود در جدول studentRegistration

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

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



شکل ۳-۱. ارتباط بین اطلاعات و فایل های تحقیق OULAD

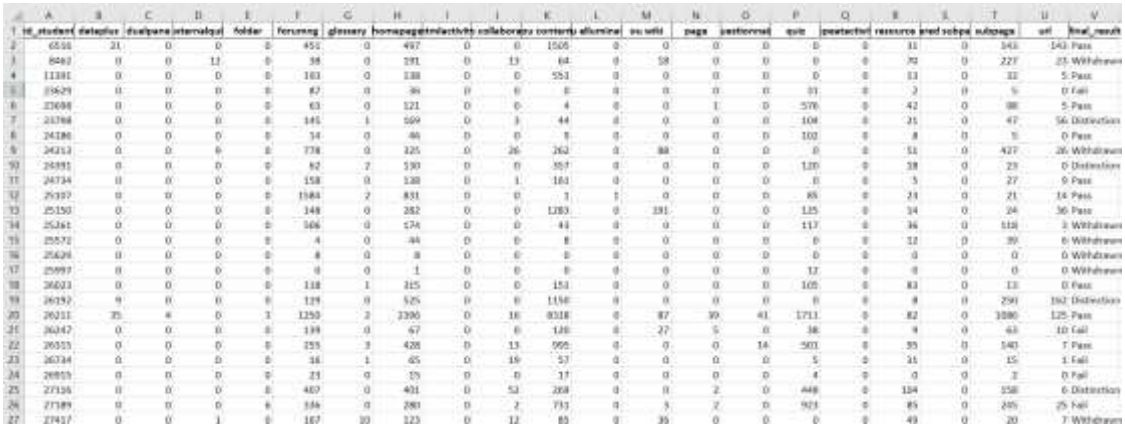
۳-۳ فایل data و توضیحات مربوط به آن

ما در این پژوهش سعی کردیم با استفاده از فعالیت‌های دانشجویان در سامانه VLE مدل یادگیری آن‌ها را تقسیم بندی 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 فعالیت ها را در فایل StudentVle قرار می‌دهیم. این کار

را با کمک کد شـــکل ۳-۲ انجام شـــد. بعد از این کار ما با اســـتفاده از id_student که در فایل studentVle وجود دارد برای هر نوع فعالیت، تمام کلیک های دانشـــجو 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 ۳-۲ . کد اضافه کردن نوع فعالیت در فایل StudentVleTest



شکل ۳-۳ . نمایی 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 خلنه عدد صـــفر قرار می‌دهیم. همچنین فایل دیتا شـــامل ۲۶۰۷۵ داده هســـت 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 خوشه 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 آن‌ها را خوشه بندی کنیم. ما از روش K-means برای خوشه بندی استفاده کردیم[2]. در این قسمت قصد داریم ابتدا روش K-means را توضیح دهیم و سپس روش اعمال این روش بر روی داده‌های خودمان را شرح دهیم.

۳-۴-۱ خوشه بندی [1]داده‌ها به روش K-means

خوشه‌بندی K-means یک روش کوانتیزه‌سازی [2]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[13].در خوشه‌بندی K-means ما داده ها را براساس فاصله اقلیدسی [3]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[14]:

- 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

[1] Clustering

[2] Quantization

[3] Euclidean distance

[1]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:

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

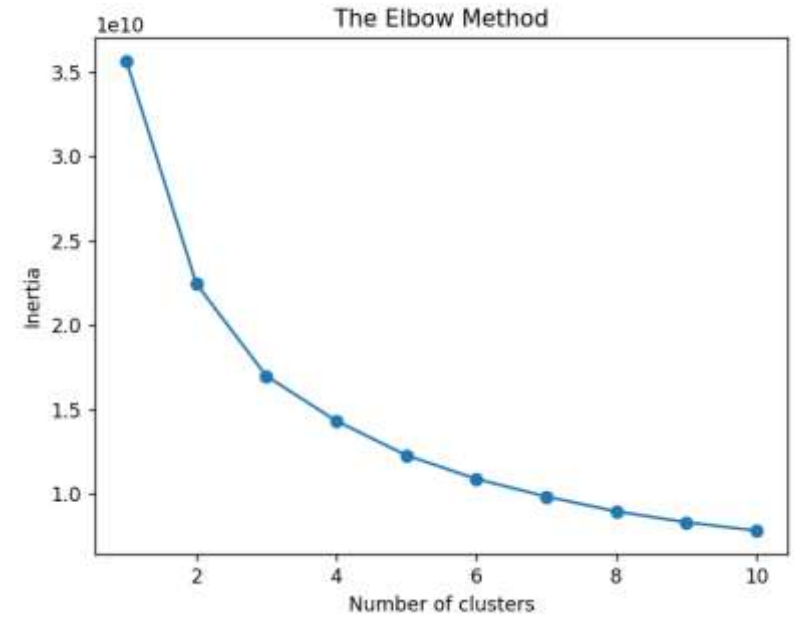
۳-۴-۲ روش انجام خوشه بندی بر روی داده‌ها

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.

[1] The center of clusters

```python
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 خطا ها را برای تعداد خوشه‌های مختلف با استفاده از این کد حساب شده

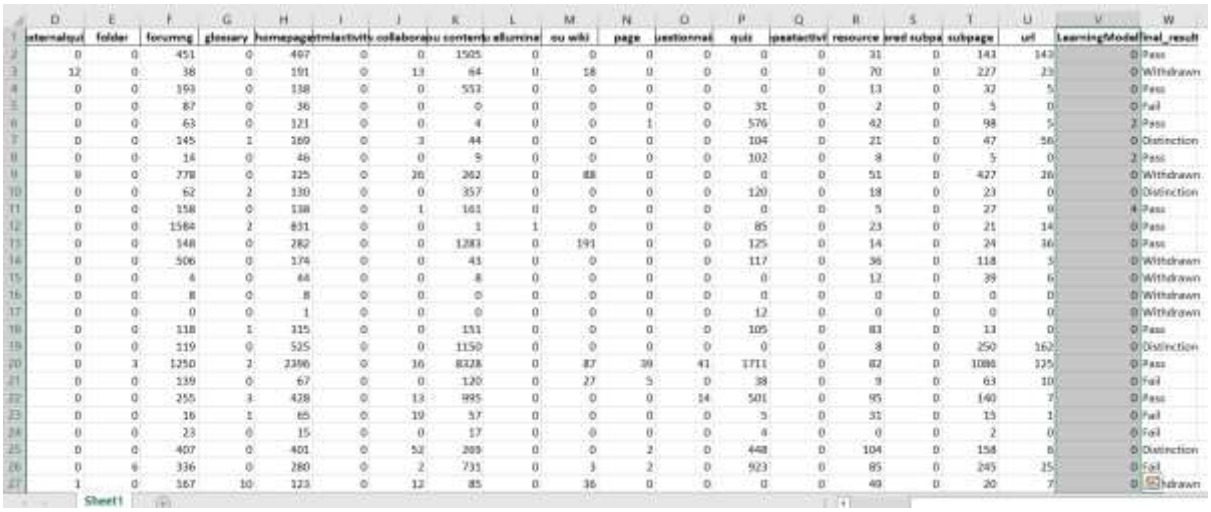


شکل ۳-۵ . نمودار L-bow که باتوجه به نمودار شکستگی در نقطه k=5 است.

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 خوشه بندی داده ها با تعداد خوشه ۵.



	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	sternalqui	folder	forumng	glossary	homepage	htmlactivity	collaborat	ou content	subpage	ou wiki	page	questionnai	quiz	resactivit	resource	srwd subpa	subpage	url	LearningModel	final_result
2	0	0	451	0	467	0	0	1505	0	0	0	0	0	0	31	0	143	143	0	Pass
3	12	0	38	0	191	0	13	64	0	18	0	0	0	0	70	0	227	23	0	Withdrawn
4	0	0	193	0	138	0	0	553	0	0	0	0	0	0	13	0	32	5	0	Pass
5	0	0	87	0	36	0	0	0	0	0	0	0	31	0	2	0	5	0	0	Fail
6	0	0	63	0	121	0	0	4	0	0	1	0	576	0	42	0	98	5	2	Pass
7	0	0	145	1	169	0	3	44	0	0	0	0	104	0	21	0	47	56	0	Distinction
8	0	0	14	0	46	0	0	9	0	0	0	0	102	0	8	0	5	0	2	Pass
9	9	0	778	0	325	0	26	262	0	88	0	0	0	0	51	0	427	26	0	Withdrawn
10	0	0	62	2	130	0	0	357	0	0	0	0	120	0	18	0	23	0	0	Distinction
11	0	0	158	0	138	0	1	161	0	0	0	0	0	0	5	0	27	0	4	Pass
12	0	0	1584	2	831	0	0	1	1	0	0	0	85	0	23	0	21	14	0	Pass
13	0	0	148	0	282	0	0	1283	0	191	0	0	125	0	14	0	24	36	0	Pass
14	0	0	506	0	174	0	0	43	0	0	0	0	117	0	36	0	118	5	0	Withdrawn
15	0	0	6	0	64	0	0	8	0	0	0	0	0	0	12	0	39	6	0	Withdrawn
16	0	0	8	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Withdrawn
17	0	0	0	0	1	0	0	0	0	0	0	0	12	0	0	0	0	0	0	Withdrawn
18	0	0	118	1	115	0	0	151	0	0	0	0	105	0	83	0	13	0	0	Pass
19	0	0	119	0	525	0	0	1150	0	0	0	0	0	0	8	0	250	162	0	Distinction
20	0	3	1250	2	2396	0	16	8328	0	87	39	41	1711	0	82	0	1086	125	0	Pass
21	0	0	139	0	67	0	0	120	0	27	5	0	38	0	9	0	63	10	0	Fail
22	0	0	255	3	428	0	13	995	0	0	0	14	501	0	95	0	140	7	0	Pass
23	0	0	16	1	85	0	19	57	0	0	0	0	3	0	31	0	15	1	0	Fail
24	0	0	23	0	15	0	0	17	0	0	0	0	4	0	0	0	2	0	0	Fail
25	0	0	407	0	401	0	52	269	0	0	2	0	448	0	104	0	158	6	0	Distinction
26	0	9	336	0	280	0	2	731	0	3	2	0	923	0	85	0	245	25	0	Fail
27	1	0	167	10	123	0	12	85	0	36	0	0	0	0	49	0	20	7	0	Withdrawn

Sheet1

شکل ۳-۷. نتیجه اجرای کد شکل ۳-۶ که در فایل data مدل‌های یادگیری خوشه بندی شدند و برچسب ۰ الی ۴ به داده ها تعلق گرفته است.

۳-۵ دسته بندی داده های دانشجویان برای پیش‌بینی میزان موفقیت

از مهمترین مفاهیم مورد کاربرد ما در این پژوهش مفهوم طبقه‌بندی [۱]در یادگیری ماشین [۲]است. در واقع طبقه بندی 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 می باشد (Fail , Pass , Withdrawn , Distinction)

به طور کلی، در یک مسئله طبقه‌بندی، هدف این است که سیستم یادگیری ماشین بتواند دسته های 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[۳]
- الگوریتم بیز ساده[۴]
- الگوریتم جنگل تصادفی[۵]
- الگوریتم ماشین بردار پشتیبان[۶]

---

1 Classification

2 machine learning

3 Decision tree algorithm

4 Naïve Bayesian

5 Random Forest

6 SVM

- الگوریتم شبکه‌های عصبی[1]
- الگوریتم‌های رده‌بندی بایاس-واریانس[2]
- الگوریتم گرادیان افزایشی بسیار قوی[3]

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

- الگوریتم بیز ساده
- الگوریتم جنگل تصادفی
- الگوریتم گرادیان افزایشی بسیار قوی

۳-۵-۱ 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. [4]

---

1 Neural network algorithm

2 Bias-Variance

3 XGboost

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.

۳- ۵ -۲ الگوریتم جنگل تصادفی[1]

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[2] 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 [3]ساخته می‌شود. به‌طور 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 به‌دست می‌آید.

---

1 Random Forest

2 Decision Tree

3 Classification and Regression Trees

با اسـتفاده از روش جنگل تصـادفی، مشـکل برازش بیش‌ازحد[1] که در الگوریتم درخت تصـمیم‌گیری وجود دارد، کاهش می‌یابد. این روش با تولید تعداد زیادی از درخت‌های تصـمیم‌گیری با اسـتفاده از داده‌های آموزشـی، از انجام برازش بیش‌ازحد خودداری می‌کند و همچنین، با ترکیب خروجی‌های مختلف درخت‌های تصـمیم‌گیری، دقت پیش‌بینی را افزایش می‌دهد .با توجه به اینکه الگوریتم جنگل تصـادفی یک الگوریتم یادگیری ماشینی پیشرفته است، در ادامه به بیان بیشتر جزئیات مراحل آن می پردازیم:

۱. سـاخت درخت‌های تصـمیم‌گیری: در ابتدا، چندین درخت تصـمیم‌گیری سـاخته می‌شـود که هر کدام از آن‌ها به صورت مستقل از داده‌های آموزشی ساخته می‌شوند.

۲. انتخاب نمونه‌های تصـادفی[2]: برای سـاخت هر درخت تصـمیم‌گیری، برخی از نمونه‌های داده‌های آموزشـی به صـورت تصـادفی انتخاب می‌شـوند. این کار باعث می‌شـود که درخت‌های تصـمیم‌گیری مسـتقل از هم باشند.

۳ .ساخت درخت‌های تصمیم‌گیری با روش دسته‌بندیCART : 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

[1] Overfitting

[2] Random samples

هر درخت برای هر نمونه محاسبه می‌شود. در نهایت، خروجی نهایی برای هر نمونه به دست می‌آید که از طریق تصمیم‌گیری روی خروجی‌های درخت‌ها به‌دست می‌آید.

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

XGBoost [1]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:

- دقت [2]بالا XGBoost :نشان داده است که در بسیاری از مجموعه داده‌های بنچ‌مارک، عملکرد برتری دارد.
- قابلیت مقیاس‌پذیری[3] XGBoost : می‌تواند با مجموعه داده‌های بزرگ با میلیون‌ها نمونه و هزاران ویژگی کار کند.

---

1 eXtreme Gradient Boosting

2 Accuracy

3 Scalability

- کارآیی[1] XGBoost :برای بهینه‌سازی بسیار بهینه شده است و مدل‌ها را با سرعتی بسیار بالاتر از دیگر پیاده‌سازی‌های Gradient Boosting آموزش می‌دهد.

- پایداری XGBoost :قادر به کار با مقادیر گمراه‌کننده، پرت و سایر نواحی داده‌ای نامنظم است.

- انعطاف‌پذیری[2] 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)

۳-۶-۱ جمع آوری داده

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 قبلا توضیح داده شد از مجموعه داده بنچمارک OULAD[3] برای استخراج داده ها استفاده شده است.

---

[1] Efficiency

[2] Flexibility

[3] Benchmark

۳-۶-۲ پیش پردازش 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 کردن کتابخانه ها و همینطور داده ی مورد نظر پیش پردازش [1]روی آن صورت می گیر[4].

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
import warnings
warnings.filterwarnings('ignore')
```

شکل ۳-۸. کد افزودن کتابخانه های لازم در این پروژه در زبان پایتون

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

# I consider that we have 100% of data
```

شکل ۳-۹. کد افزودن فایل اکسل لازم در این پروژه در زبان پایتون.

[1] Preprocessing

```
print("============ Exploratory data analysis ============")
print("# view dimensions of dataset")

print(df.shape)

print("# preview the dataset")

print(df.head())

print("#Rename column names: but we don't need")

print("# view summary of dataset")

print(df.info())
```

شکل ۳-۱۰. 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("#if we have missing values ")

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("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 ۳-۱۱ کد بررسی ویژگی‌های عددی و غیر عددی در زبان پایتون

۳-۶-۳ تقسیم داده‌ها به دو مجموعه آموزشی[1] و آزمایشی[2]

برای ارزیابی عملکرد مدل، داده‌ها به دو مجموعه تقسیم می‌شوند. مجموعه آموزشی برای آموزش مدل استفاده می‌شود و مجموعه آزمایشی برای ارزیابی عملکرد مدل استفاده می‌شود. مراحل این فرایند به ترتیب در تکه کد های زیر نمایش داده شده است[4].

[1] Train

[2] Test

```
# declare feature vector and target
df = df.drop(['id_student'], axis=1)
df = df.drop(['LearningModel'])
X=df.drop(['final_result'] , axis=1)
y=df['final_result']
```

شکل ۳-۱۲. جدا کردن داده های ورودی و حذف برچسب‌های اضافی از فایل data در پایتون

```
# 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)   # I changed 0.3 to 0.2
```

شکل ۳-۱۳ جدا کردن داده های ورودی و برچسب های آموزشی و آزمونی به ۸۰ به ۲۰ درصد در پایتون

train_test_split در scikit-learn یک تابع بسیار مفید برای تقسیم داده‌ها به دو بخش آموزش(train) و آزمون(test) است. این تابع از بسته model_selection در scikit-learn برای تقسیم داده‌های یک مجموعه داده به دو بخش استفاده می‌شود. در شکل ۳-۱۳ تابع train_test_split با گرفتن داده‌های ورودی x و برچسب‌ها y و درصد تقسیم آن‌ها بین داده‌های آموزش و آزمون، داده‌ها را به صورت تصادفی به دو بخش تقسیم می‌کند.

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

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

شکل ۳-۱۴. بررسی ابعاد داده های ورودی آموزشی و آزمونی در پایتون

۳-۶-۴ طراحی مدل

در این مرحله، مدل ماشین برای حل مسئله طراحی می‌شود. این شامل انتخاب نوع مدل (مانند بیز ساده. شبکه‌های عصبی، درخت تصمیم و...) و تعیین پارامترهای مدل (مانند تعداد لایه‌ها[1]، تعداد نورون‌ها[2] و سایر پارامترهای مشخص کننده مدل) است. در این قسمت ما تصمیم گرفتیم از مدل بیز ساده (naïve bayes) نام برده استفاده کنیم.

۳-۶-۵ 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[4]. 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 نرمال‌سازی[3] یا استاندارد سازی[4] 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 استانداردسازی، مقیاس‌بندی 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

[1] Layers

[2] Neurons

[3] Normalization

[4] Standardization

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

RobustScalerیک روش استانداردسازی است که در برابر داده‌های پرت و نویزدار مقاوم است .

معمولا در استانداردسازی داده‌ها، میانگین [1]و واریانس [2]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 [3]و نویزدار [4]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 ۲۵ و ۷۵ درصدی) محاسبه می‌شود. سپس، با استفاده از فرمول زیر، داده‌ها به مقیاس استاندارد نگاشت می‌شوند:

$$x_{scaled} = \frac{(x - median)}{IQR}$$

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.

[1] mean

[2] Variance

[3] Outlier data

[4] noisy

```
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())
```

شکل ۳-۱۵. مقیاس بندی ویژگی ها با RobustScaler در پایتون

تا اینجا ما داده ها را آماده کردیم. حال می‌خواهیم که سه الگوریتمی که گفته شد را در این قسمت قرار دهیم. کد این سه الگوریتم در شکل های ۳-۱۶، ۳-۱۷ و ۳-۱۸ زده شده است.

```
# 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))
```

شکل۳-۱۶. مدل آموزشی naïve bayes در پایتون

```
from sklearn.ensemble import RandomForestClassifier

#random forest
rfc = RandomForestClassifier(n_estimators=2000)
print("# fit the model")
print(rfc.fit(X_train, y_train))
```

شکل۳-۱۷. مدل آموزشی random forest در پایتون

```
import xgboost as xgb
xgb = XGBClassifier()

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

شکل۳-۱۸. مدل آموزشی XGBoost 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)
```

شکل ۳-۱۹. پیش بینی برچسب های داده های آزمونی در پایتون

۳-۷ ارزیابی عملکرد مدل

پس از اتمام آموزش مدل، عملکرد آن با استفاده از داده‌های آزمایشی ارزیابی می‌شود[4]. 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[1]، صحت[2]، بازخوانی [3]و امتیاز F1-score استفاده می‌شود. معمولاً ماتریس درهم‌ریختگی[4] نیز برای ارزیابی دقیق‌تر عملکرد مدل استفاده می‌شود. که در ادامه توضیحات بیشتری داده خواهد شد.

۳-۷-۱ معیار ارزیابی دقت

این معیار 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 پیش بینی را برای داده های آزمون و آموزش نمایش می دهد[4].

[1] Accuracy

[2] precision

[3] recall

[4] Confusion matrix

```
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(y_pred_train)

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

شکل ۳-۲۰. نمایش دقت پیش بینی برای داده های آزمون و آموزش در پایتون

خروجی کد شکل ۳-۲۰ را در شکل ۳-۲۱ می‌توانید ببینید.

```
GaussianNB()
#predct the result
['Distinction' 'Fail' 'Withdrawn' ... 'Fail' 'Wi
================ check accuracy score ==========
Model accuracy score: 0.3143
#Compare the train-set and test-set accuracy
['Distinction' 'Fail' 'Fail' ... 'Distinction' '
Training-set accuracy score: 0.3192
Check for overfitting and underfitting
Training set score: 0.3192
```

شکل ۳-۲۱. خروجی کد شکل ۳-۲۰.

علاوه بر این ها در یادگیری ماشین، ما معمولاً با دو مفهوم اصلی overfitting و کم‌شدن underfitting روبرو هستیم. این دو مفهوم نشان دهنده این هستند که مدل به چه میزان با داده‌های آموزش سازگار است. زیاده روی در یاددهی به موقعیتی گفته می‌شود که مدل به گونه‌ای بر روی داده‌های آموزش بسیار دقیق است که تقریباً به خاطر بسیاری از جزئیات ناهمخوانی در داده‌های جدید به خوبی عمل

نمی‌کند. برای مثال، اگر یک مدل به صـــورت بســـیار دقیقی بر روی داده‌های آموزش عمل کند، اما بر روی داده‌های تسـت نتایج ناامید کننده‌ای داشـته باشـد، به این مفهوم گفته می‌شـود که مدل اضـافه‌شـده اسـت. 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 صـورت بسـیار سـاده‌ای طراحی شـود، به گونه‌ای که بر روی داده‌های آموزش هم به درسـتی کار نکند در نتیجه بر روی داده‌های تسـت نیز نتایج بدی تولید می‌کند، به این مفهوم گفته می‌شـود که مدل کم‌شده است.

```
print("Check for overfitting and underfitting")

print('Training set score: {:.4f}'.format(gnb.score(X_train, y_train)))

print('Test set score: {:.4f}'.format(gnb.score(X_test, y_test)))
```

شکل ۳-۲۲. چک کردن overfitting و underfitting

خروجی کد شکل ۳-۲۲:

```
Check for overfitting and underfitting
Training set score: 0.3192
Test set score: 0.3143
```

شکل ۳-۲۳. خروجی کد شکل ۲۲-۳.

با توجه به شکل ۳-۲۳ خروجی overfitting و underfitting نداریم.

۳-۷-۲ معیار دقت صفر[1]

علاوه بر این معیارها، معیار دقت صفر 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 این‌که ببینیم مدل ما به چه میزان بهتر از پیش‌بینی اکثریت عمل می‌کند، مورد استفاده قرار گیرد.

در تکه کد پایتون شکل ۳-۲۴ دقت صفر را محاسبه می کنیم.

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

print(y_test.describe())

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

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

شکل ۳-۲۴. محاسبه دقت صفر.

[1] Null Accuracy

```
Compare model accuracy with null accuracy
count     5215
unique       4
top       Pass
freq      2169
Name: final_result, dtype: object
Null accuracy score: 0.4159
```

شکل ۳-۲۵. خروجی کد شکل ۳-۲۴

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

۳-۷-۳ ماتریس اغتشاش[1]

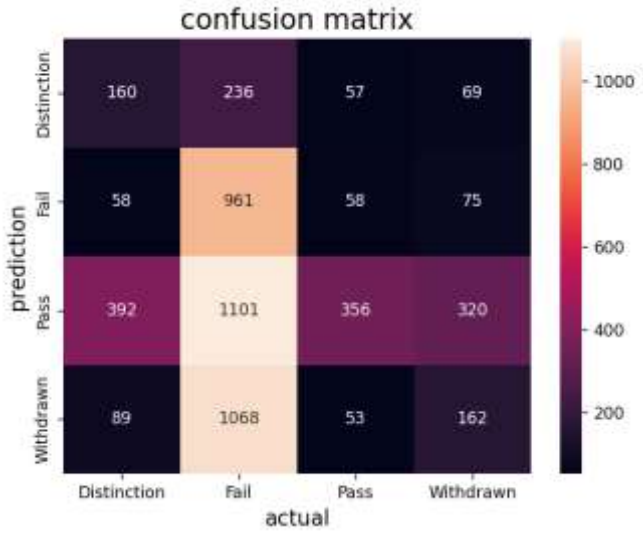
ماتریس اغتشاش در این مسئله دسته‌بندی که چهار نتیجه ممکن دارد (distinction fail,pass, withdrawn)، در یک جدول ۴ × ۴ نمایش داده می شود. ردیف‌های جدول به کلاس‌های واقعی اشاره دارند و ستون‌ها به کلاس‌های پیش‌بینی شده مربوط می‌شوند. در این ماتریس اغتشاش، قطر اصلی (از بالا سمت چپ تا پایین سمت راست) تعداد پیش‌بینی‌های صحیح را برای هر کلاس نشان می‌دهد، در حالی که عناصر خارج از قطر، میزان اشتباهات پیش‌بینی را نشان می‌دهند.

تکه کد پایتون زیر ماتریس اغتشاش این پیش بینی را به دست آورده و نشان می دهد:

[1] Confusion matrix

```
print("============ Confusion matrix ==============")

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

from sklearn.metrics import confusion_matrix

labels = ['Withdrawn' , 'Fail' , 'Pass' , 'Distinction']
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()
```

شکل ۳-۲۶. نمایش confusion matrix در پایتون



شکل ۳-۲۷. خروجی کد شکل ۳-۲۶. سطر ها نشان دهنده پیش‌بینی برچسب و ستون ها نشان دهنده برچسب واقعی است.

۳-۷-۴ احتمالات چند کلاسه

محاسبه احتمالات چند کلاسه[1] جزو معیارهای ارزیابی پیش‌بینی مدل در ماشین لرنینگ است. این معیار به عنوان یکی از روش‌های ارزیابی کیفیت مدل برای دسته‌بندی چند کلاسه استفاده می‌شوند. با محاسبه احتمالات چند کلاسه، می‌توانیم ببینیم که مدل به چه اندازه موفق بوده است تا دسته بندی صحیح را برای داده های ورودی ارایه دهد. نمودار هیستوگرام[2] می‌تولند به عنوان یک ابزار مفید برای بررسی کیفیت مدل در پیش‌بینی چند کلاسه استفاده شود. با توجه به شکل و توزیع احتمالات چند کلاسه روی نمودار هیستوگرام، می‌توانیم بررسی کنیم که آیا مدل به درستی دسته‌بندی‌های خود را انجام داده است یا خیر. بررسی کیفیت مدل با استفاده از نمودار هیستوگرام به اینصورت است:

- بررسی توزیع احتمالات: با بررسی توزیع احتمالات چند کلاسه روی نمودار هیستوگرام، می‌توانیم بررسی کنیم که آیا مدل به درستی دسته‌بندی‌های خود را انجام داده است یا خیر. برای مثال، اگر توزیع احتمالات برای هر کلاس بسیار مشابه باشد و همه کلاس‌ها به یک شکل بر روی نمودار هیستوگرام قرار داشته باشند، این نشان دهنده یک مدل خوب است که به درستی دسته‌بندی می‌کند. اما اگر توزیع احتمالات بسیار متفاوت باشد و برخی کلاس‌ها به شدت بیش بار شده باشند، این نشان دهنده وجود مشکل در دسته‌بندی مدل است.
- بررسی تفاوت بین احتمالات: با بررسی تفاوت بین احتمالات چند کلاسه روی نمودار هیستوگرام، می‌توانیم بررسی کنیم که آیا مدل به درستی دسته‌بندی‌های خود را انجام داده است یا خیر. برای مثال، اگر بین احتمالات کلاس‌ها تفاوت زیادی وجود داشته باشد، این نشان دهنده وجود مشکل در دسته‌بندی مدل است.

---

1 multi classes probabilities

2 histogram

- بررسی انحراف از مقدار مورد انتظار: با بررسی انحراف از مقدار مورد انتظار برای هر کلاس، می‌توانیم بررسی کنیم که آیا مدل به درستی دسته‌بندی‌های خود را انجام داده است یا خیر. برای مثال، اگر احتمالات مدل برای یک کلاس به طور قابل توجهی از مقدار مورد انتظار برای آن کلاس کمتر باشد، این نشان دهنده وجود مشکل در دسته‌بندی مدل است.

با توجه به موارد فوق، می‌توان نمودار هیستوگرام رابرای بررسی کیفیت مدل در پیش‌بینی چند کلاسه به کار برد. تکه کد شکل ۳-۲۸ احتمال پیش بینی ۴ برچسب fail , pass , distinction ,withdrawn را ابتدا برای ۱۰ نمونه از داده ی آزمون و سپس جداگانه برای هر برچسب نشان می دهد.

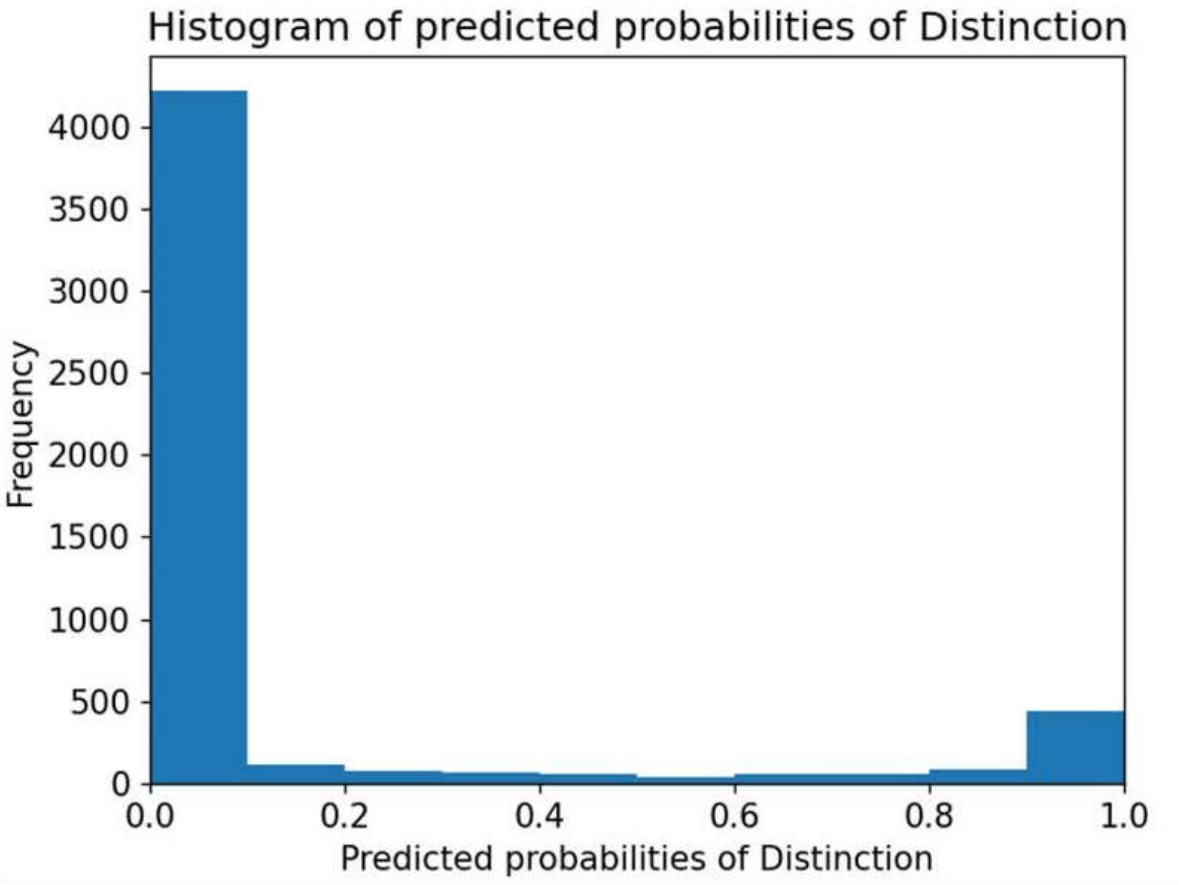
```
print("******************** Calculate class probabilities ******************")

print("# print the first 10 predicted probabilities of classes 'Distinction' 'Fail' 'Pass' 'Withdrawn' ")

y_pred_prob = gnb.predict_proba(X_test)[0:10]

print(y_pred_prob)

print("# store the probabilities in dataframe")
#==============?===============
y_pred_prob_df = pd.DataFrame(data=y_pred_prob, columns=gnb.classes_)

print(y_pred_prob_df)

print("# print the first 10 predicted probabilities for class 'Distinction' ")

print(gnb.predict_proba(X_test)[0:10, 0])

print("# print the first 10 predicted probabilities for class 'Fail'")

print(gnb.predict_proba(X_test)[0:10, 1])

print("# print the first 10 predicted probabilities for class 'Pass' ")

print(gnb.predict_proba(X_test)[0:10, 2])

print("# print the first 10 predicted probabilities for class ' Withdrawn ' ")

print(gnb.predict_proba(X_test)[0:10, 3])
```

شکل ۳-۲۸. نمایش احتمالات پیش‌بینی ۱۰ نمونه.

تکه کد شکل ۳-۲۹، نمودار هیستوگرام برای احتمالات پیش‌بینی کلاس Distinction را نمایش می‌دهد که خروجی آن در شکل ۳-۳۰ نمایش داده شده است.

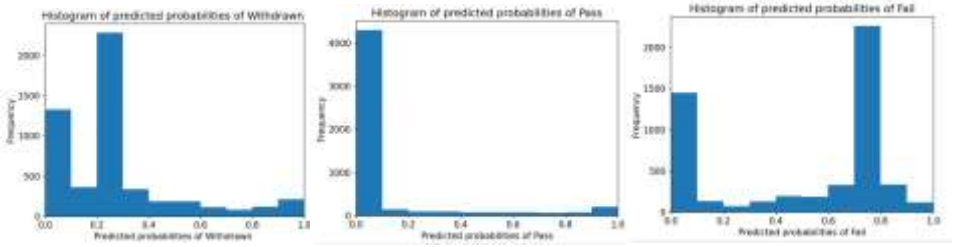
```
print("plot histogram for class Distinction ")

y_pred0 = gnb.predict_proba(X_test)[:, 0]

print(y_pred0)

# plot histogram of predicted probabilities


# adjust the font size
plt.rcParams['font.size'] = 12

# plot histogram with 10 bins
plt.hist(y_pred0, bins = 10)
# set the title of predicted probabilities
plt.title('Histogram of predicted probabilities of Distinction')
# set the x-axis limit
plt.xlim(0,1)
# set the title
plt.xlabel('Predicted probabilities of Distinction')
plt.ylabel('Frequency')
plt.show()
```

شکل ۳ -۲۹ کد نمایش نمودار هیستوگرام برای احتمالات پیش بینی کلاس Distinction



شکل ۳ -۳۰ نمایش گرافیکی نمودار هیستوگرام Distinction

در شکل ۳-۳۱ بقیه نمودار هیستوگرام را برای بقیه برچسب ها رسم شده. همانطور که در شکل ۳-۳۱ می‌بینیم تفاوت نمودار های هیستوگرام زیاد است و این نشان از دسته بندی و پیشبینی نامطلوب دارد.



شکل ۳-۳۱. به ترتیب از راست به چپ نمودار هیستوگرام برچسب های Fail و Pass و Withdrawn.

۳-۷-۵ صحت پیش بینی مثبت ها(Precision)

در یادگیری ماشین، Precision (صحت پیش‌بینی مثبت‌ها) یکی از معیارهای ارزیابی مدل است که برای ارزیابی دقت پیش‌بینی مثبت‌ها (True Positive) استفاده می‌شود. Precision نسبت تعداد مثبت‌هایی است که به درستی توسط مدل پیش‌بینی شده‌اند به تعداد کل پیش‌بینی‌های مثبت از طریق مدل. به طور ریاضی:

$$precision = \frac{TP}{TP + FP}$$

که TP تعداد مثبت‌هایی است که به درستی توسط مدل پیش‌بینی شده‌اند و FP تعداد منفی‌هایی است که به اشتباه توسط مدل به عنوان مثبت پیش‌بینی شده‌اند.

۳-۷-۶ معیار Recall

در ماشین لرنینگ، recall به معنای تعداد نمونه‌های واقعی یک کلاس که به درستی تشخیص داده شده‌اند، به تعداد کل نمونه‌های واقعی آن کلاس است. به عبارت دیگر، recall نسبت تعداد نمونه‌هایی است که به درستی به یک کلاس تعلق دارند به تعداد کل نمونه‌های واقعی آن کلاس. فرمول ریاضی آن به شکل زیر است:

$$Recall = \frac{TP}{TP + FN}$$

۳-۷-۷ معیار F1 score

معیار f1 score نیز همانند معیار recall یکی از معیارهای ارزیابی عملکرد مدل‌های یادگیری ماشین در مسائل دسته‌بندی است. معیار f1 score یک ترکیب از دو معیار دیگر به نام precision و recall است.

فرمول محاسبه f1 score به صورت زیر است:

$$\text{f1 score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

معیار f1 score در واقع یک معیار ترکیبی است که هم precision و هم recall را در نظر می‌گیرد. معمولاً در مسائلی که هدف آن‌ها دسته‌بندی نمونه‌هایی با تعداد برابری از دو دسته است، f1 score به عنوان یکی از معیارهای ارزیابی مدل‌های یادگیری ماشین به کار می‌رود. این معیار نشان می‌دهد که مدل چه میزان از نمونه‌های مثبت را به درستی شناسایی کرده است و به همین دلیل در بسیاری از مسائل دسته‌بندی مورد استفاده قرار می‌گیرد.

تکه کد شکل ۳-۳۲ اطلاعات خوبی به دست ما می‌دهد که در ادامه به آن پرداخته ایم:

```
print("================= racall === precision === fp tp rate============")

from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

شکل ۳-۳۲. نمایش گزارشی از معیار های ارزیابی recall, precision, f1-score

```
================= Classification metrices ===============
              precision    recall  f1-score   support

  Distinction       0.23      0.31      0.26       522
        Fail        0.29      0.83      0.43      1152
        Pass        0.68      0.16      0.26      2169
   Withdrawn        0.26      0.12      0.16      1372

    accuracy                           0.31      5215
   macro avg        0.36      0.36      0.28      5215
weighted avg        0.44      0.31      0.27      5215
```

شکل ۳-۳۳. خروجی کد شکل ۳-۳۲.

در این قسمت از کد ابتدا TP و TN و FP و FN را برای ۴ کلاس موجود با توجه به جدول ماترس ابهام (در اینجا cm) به دست می آوریم:

```python
print("================= Classification metrices ===============")

from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))

#======Distinction======

TPD = cm[0][0]
TND = cm[1][1] + cm[1][2] + cm[1][3] + cm[2][1] + cm[2][2] + cm[3][3] + cm[3][1] + cm[3][2] + cm[3][3]
FPD = cm[0][1] + cm[0][2] + cm[0][3]
FND = cm[1][0] + cm[2][0] + cm[3][0]


#======Fail=======

TPF = cm[1][1]
TNF = cm[0][0] + cm[0][2] + cm[0][3] + cm[2][0] + cm[2][2] + cm[2][3] + cm[3][0] + cm[3][2] + cm[3][3]
FPF = cm[1][1] + cm[1][2] + cm[1][3]
FNF = cm[0][1] + cm[2][1] + cm[3][1]

#======Pass=======

TPP = cm[2][2]
TNP = cm[0][0] + cm[0][1] + cm[0][3] + cm[1][0] + cm[1][1] + cm[1][3] + cm[3][0] + cm[3][1] + cm[3][3]
FPP = cm[2][0] + cm[2][1] + cm[2][3]
FNP = cm[0][2] + cm[1][2] + cm[3][2]

#======Withdrawn=======

TPW = cm[3][3]
TNW = cm[0][0] + cm[0][1] + cm[0][2] + cm[1][0] + cm[1][1] + cm[1][2] + cm[2][0] + cm[2][1] + cm[2][2]
FPW = cm[3][0] + cm[3][1] + cm[3][2]
FNW = cm[0][3] + cm[1][3] + cm[2][3]
```

شکل ۳-۳۴. نحوه محاسبه TP و TN و FP و FN

```
#===============Fail=============


print("# print Fail precision score")

precisionF = TPF / float(TPF + FPF)

print('Fail Precision : {0:0.4f}'.format(precisionF))

recallF = TPF / float(TPF + FNF)

print('Fail Recall or Sensitivity : {0:0.4f}'.format(recallF))

fscoreF=2*(precisionF * recallF)/(precisionF + recallF )

print('Fail f1 score: {0:0.4f}'.format(fscoreF))

#==============Pass=============

print("# print Pass precision score")

precisionP = TPP / float(TPP + FPP)


print('Pass Precision : {0:0.4f}'.format(precisionP))

recallP = TPP / float(TPP + FNP)

print('Pass Recall or Sensitivity : {0:0.4f}'.format(recallP))

fscoreP=2*(precisionP * recallP)/(precisionP + recallP )

print('Pass f1 score: {0:0.4f}'.format(fscoreP))
```

شکل ۳-۳۵. محاسبه و نمایش ۳ معیار ارزیابی recall , precision , f1 score

۳-۸ بهبود نتیجه نهایی

تا کنون ما نحوه پیاده سازی یک الگوریتم هوش مصنوعی برای پیش‌بینی موارد خواسته شد در داده های خود را دیدیم. نتیجه‌ای که الگوریتم naïve bayes برای معیار accuracy داد حدود ۰.۳۴ بود که با توجه به معیار null accuracy که ۰.۴۱ بود، بسیار پایین و کم دقت است. با توجه به این موضوع ما ملزم به این

بودیم که دقت پیاده سازی خودمان را بالا ببریم. در ادامه می‌خواهیم در مورد بالا بردن سطح دقت در این مسئله صحبت کنیم.

۳-۸-۱ روش‌های بالا بردن دقت در مسائل

برای بالا بردن سطح دقت در یک مسئله که به با الگوریتم‌های هوش مصنوعی انجام می‌شوند؛ روش‌های مختلفی وجود دارد. ما به چند مورد از این روش‌های می‌پردازیم:

۱. **تنظیم پارامترهای مدل**: در کد هر الگوریتم پارامتر های مختلفی داریم. مثلا یکی از پارامترها تعداد تکرار الگوریتم هست. می‌توانیم با تغییر تعداد تکرار هر الگوریتم، میزان سطح دقت را تغییر دهیم.

۲. **افزایش حجم داده آموزشی**: با افزایش حجم داده آموزشی، می‌توانید دقت مدل خود را افزایش دهید. این کار می‌تواند باعث بهبود قابل توجهی در دقت مدل شما شود. همچنین، با افزایش حجم داده‌ها، می‌توانید از روش‌هایی مانند cross-validation برای ارزیابی دقیق‌تر مدل استفاده کنید.

۳. **استفاده از روش‌های feature engineering**: با استفاده از روش‌هایی مانند انتخاب ویژگی‌ها (feature selection) و استخراج ویژگی‌های جدید (feature extraction)، می‌توانید دقت مدل خود را بهبود دهید. به‌طور مثال، اگر برخی از ویژگی‌های موجود در داده‌های شما بی‌اهمیت هستند، می‌توانید آن‌ها را حذف کنید. همچنین، با استفاده از روش‌هایی مانند PCA و t-SNE می‌توانید ویژگی‌های جدیدی از داده‌های خود استخراج کنید و از آن‌ها برای آموزش مدل استفاده کنید.

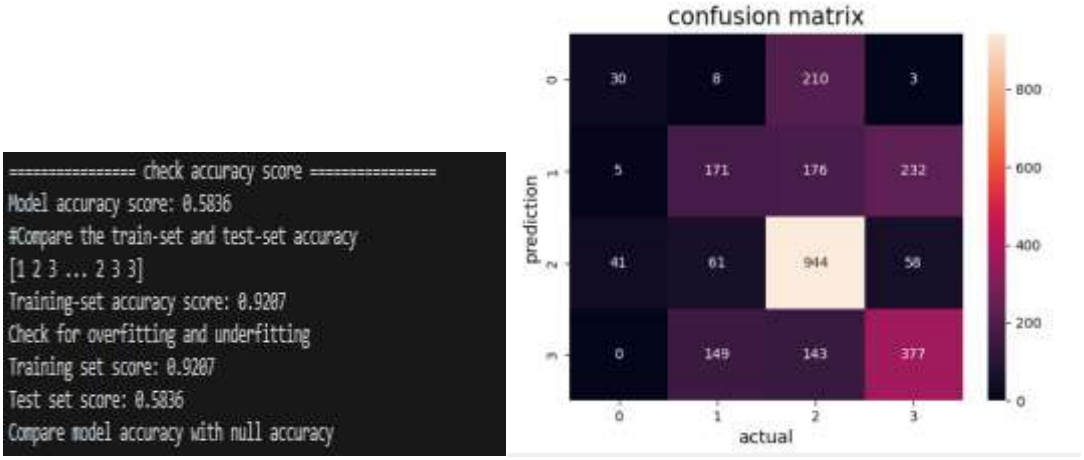
۴. **استفاده از مدل‌های مختلف:** یکی از راه‌های افزایش دقت، تغییر الگوریتم برای حل آن مسئله است. برای مثال اگر ما برای حل این مسئله از روش naïve bayes استفاده می‌کردیم، می‌توانیم با تغییر این الگوریتم به الگوریتم های دیگر برای این کار دقت حل مسئله خود را بالا ببریم. Random Forest، Neural Network و SVM استفاده کنید. این مدل‌ها هر کدام دارای ویژگی‌ها و پارامترهای خاص خود هستند که می‌تواند بر دقت آن‌ها تأثیرگذار باشد.

۵. **انجام پیش‌پردازش داده‌ها:** با انجام پیش‌پردازش داده‌ها می‌توانید دقت مدل خود را بیشتر کنید. به‌طور مثال، با استفاده از روش‌هایی مانند نرمال‌سازی داده‌ها، تبدیل داده‌ها به فضای برداری و حذف داده‌های پرت می‌توانید دقت مدل خود را بهبود ببخشید.

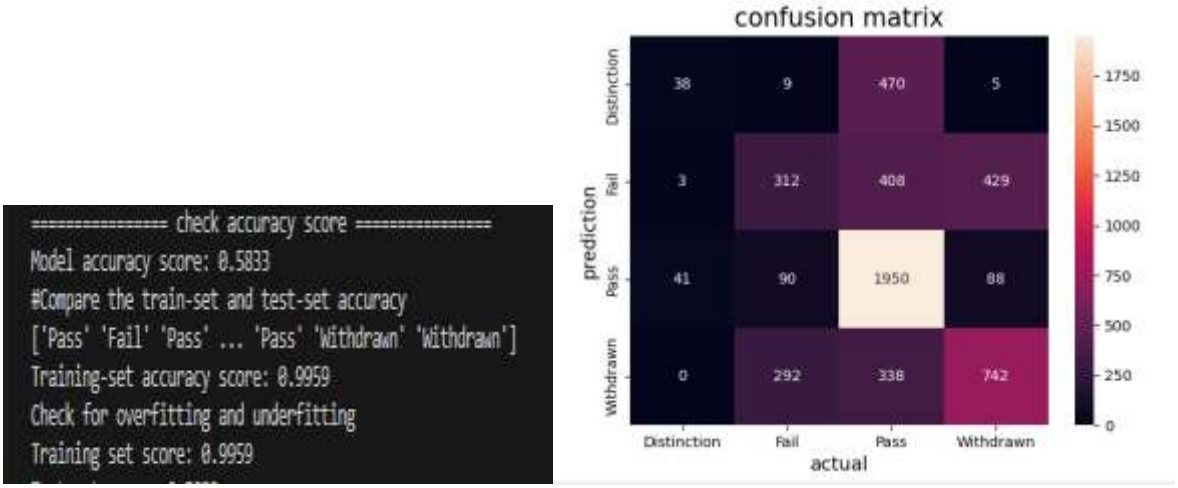
ما برای بالا بردن دقت مسئله خود، از ۲ مدل یادگیری دیگر استفاده کردیم. همچنین پیش‌پردازش بر روی داده ها انجام دادیم. تعداد داده‌های آموزشی خود را بیشتر کردیم و در نهایت پارامترهای الگوریتم را تغییر دادیم.

۳-۸-۲ تغییر مدل یادگیری

همان‌طور که در قسمت ۵-۶-۳ ذکر شد، ما از سه مدل یادگیری استفاده کردیم. دقت مدل naïve bayes خیلی پایین بود برای مسئله ما ولی مدل‌های random forest و XGboost دقت بهتری نسبت به مدل naïve bayes دارند. دقت این دو مدل در حدود ۰.۵۵ الی ۰.۵۷ است.



شکل ۳-۳۶. خروجی الگوریتم XGboost که دقت آن ۰.۵۸ است.



شکل ۳-۳۷. خروجی الگوریتم random forest که دقت آن ۰.۵۸ است.

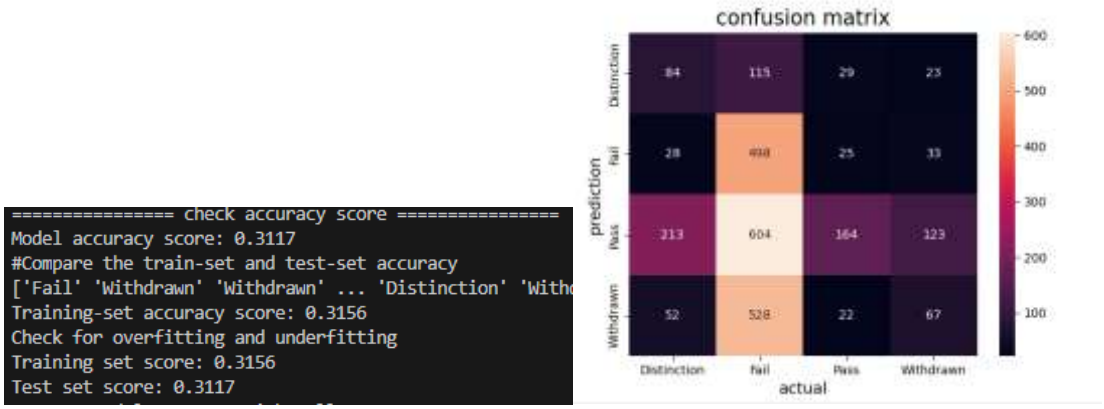
۳-۸-۳ افزایش حجم داده آموزشی

یکی دیگر از راه‌های افزایش دقت در مسائل، افزایش حجم داده آموزشی است. با توجه به اینکه دیتاست ما از پژوهش OULAD گرفته شده، پس امکان افزایش داده ها نبود. بهترین راهی که امکان استفاده از آن بود در هنگام تقسیم داده به دو دسته آموزشی و آزمایشی، میزان بیشتری از داده‌ها را به داده‌های آموزشی بدهیم و مقدار کمتری به داده‌های آزمایشی بدهیم. برای این کار کد قسمت ۳-۶-۳ را مطابق شکل ۳-۳۷ تغییر می‌دهیم.

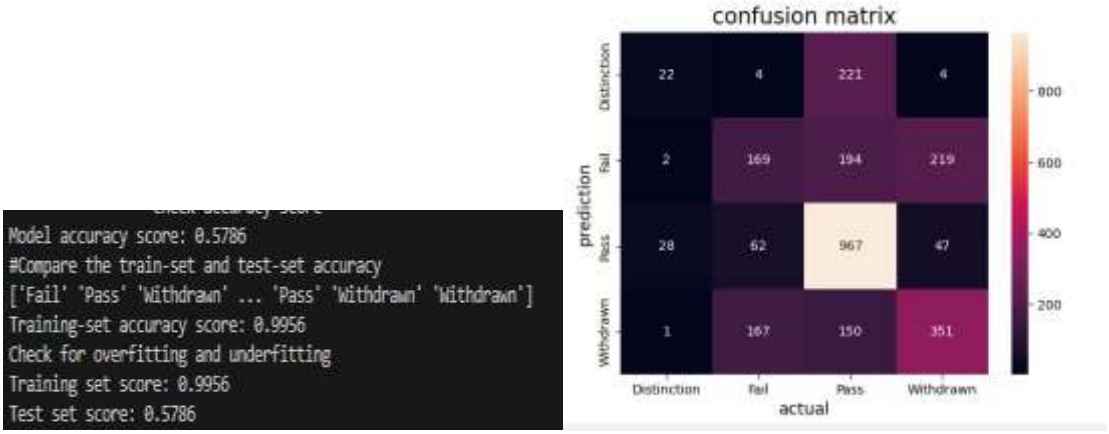
```
# split X and y into training and testing sets

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state = 0)   # I changed

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

شکل ۳-۳۸. ۰.۱ داده را به داده‌های آزمایشی می‌دهیم و ۰.۹ را به داده‌های آموزشی می‌دهیم.

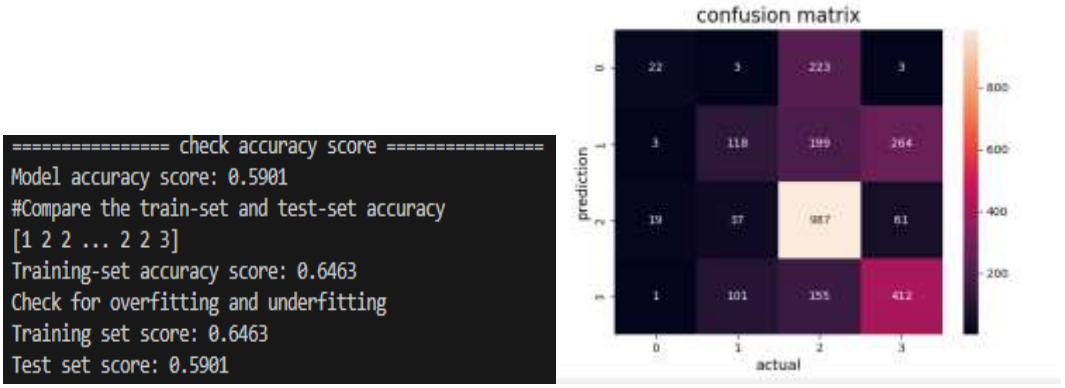
بعد این نتیجه اجرای سه الگوریتم را در شکل‌های ۳-۳۹ و ۳-۴۰ و ۳-۴۱ مشاهده می‌کنیم.



شکل ۳-۳۹. خروجی الگوریتم naïve bayes . دقت ۰.۳۱۲ است.



شکل ۳-۴۰. خروجی الگوریتم random forest . دقت ۰.۵۷۹ است.



شکل ۳-۴۱. خروجی الگوریتم XGboost . دقت ۰.۵۹ است.

۳-۸-۴ انجام پیش پردازش بر روی داده‌ها

یکی از عوامل تاثیر گذار روی دقت مدل ما، حذف داده های نویز و دور افتاده است. ما در اینجا قصد داشتیم با بررسی میانگین داده ها در هر ستون، آنهایی که خیلی اختلاف دارند را حذف می‌کنیم. کد شکل ۳-۴۱ پیش پردازش ما را نشان می‌دهد.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['final_result'] = le.fit_transform(df['final_result'])
# I consider that we have 100% of data

df = df.drop('id_student', axis=1) # حذف ستون شناسه دانشجو از دیتافریم

df = df.drop('LearningModel', axis=1) # حذف ستون شناسه دانشجو از دیتافریم

df = df[(df < (df.mean() + 10 * df.std())) & (df > (df.mean() - 10 * df.std()))].dropna() # اعمال پیش پردازش
```

شکل ۳-۴۲. کد پیش پردازش بر روی داده برای حذف داده های دور افتاده. میتوان مقدار عدد ۱۰ را با توجه به داده ها تغییر داد. در ادامه پیش پردازش های انجام شده با عدد ۳ انجام شده‌اند.

در جدول ۳-۸ میزان دقت برای هر مدل با اعمال پیش پردازش مشاهده می‌کنیم.

جدول ۳-۸. میزان دقت با اعمال پیش پردازش برای هر مدل یادگیری

نام مدل یادگیری	میزان دقت مدل یادگیری
naïve bayes	۰.۳۵
XGboost	۰.۵۸
Random forest	۰.۵۹

۳-۸-۵ تنظیم پارامترهای مدل

یکی دیگر از اقداماتی که می‌توان برای بالا بردن دقت مدل یادگیری انجام داد، تنظیم پارامترهای مدل یادگیری است. پارامترهایی که یک مدل یادگیری می‌تواند داشته باشد:

پارامترهای مدل یادگیری می‌توانند بسته به نوع مدل، الگوریتم یادگیری و هدف مسئله متفاوت باشند. در ادامه، به عنوان یک مثال، پارامترهایی که برای برخی از مدل‌های یادگیری مورد استفاده قرار می‌گیرند را بررسی می‌کنیم:

- learning_rate: نرخ یادگیری، یعنی مقداری که در هر بار به‌روزرسانی وزن‌ها و بایاس‌ها به آن‌ها افزوده می‌شود.
- n_estimators: تعداد دفعات اجرای الگوریتم.
- num_epochs: تعداد دوره‌های آموزش، یعنی تعداد بارهایی که داده‌ها به مدل ورودی داده می‌شوند.
- batch_size: اندازه دسته‌بندی، یعنی تعداد داده‌هایی که در هر مرحله به مدل ورودی داده می‌شوند.
- num_layers: تعداد لایه‌های شبکه عصبی.
- num_units: تعداد واحدهای هر لایه در شبکه عصبی.
- dropout_rate: نرخ انسدادی، یعنی مقداری که در هر بار آموزش، برای تعدادی از واحدها به صورت تصادفی برابر با صفر قرار داده می‌شوند تا از برازش زیاد به داده‌های آموزش جلوگیری شود.
- activation_function: تابع فعال‌سازی، یعنی تابعی که بر روی خروجی لایه‌های شبکه عصبی اعمال می‌شود.
- loss_function: تابع هزینه، یعنی تابعی که برای محاسبه خطا در پیش‌بینی از داده‌های تست و استفاده در فرآیند به‌روزرسانی وزن‌ها و بایاس‌ها استفاده می‌شود.
- Optimizer: بهینه‌ساز، یعنی الگوریتمی که برای به‌روزرسانی وزن‌ها و بایاس‌ها استفاده می‌شود.
- metrics: معیارها، یعنی مجموعه‌ای از معیارهایی که برای ارزیابی کیفیت پیش‌بینی‌های مدل استفاده می‌شود.

لازم به ذکر است که این پارامترها ممکن است بسته به نوع مدل و پیاده‌سازی آن متفاوت باشند و همچنین پارامترهایی مانند تعداد فیلترها و اندازه‌ی هسته در شبکه‌های کانولوشنالی و یا ضرایب رگولاریزه در برخی مدل‌ها نیز وجود دارند.

در این پژوهش ما این مقادیر را برای الگوریتم‌های XGboost و الگوریتم random forest مشـــخص کردیم. برای الگوریتم XGboost مطابق شـکل ۳-۴۲ نرخ یادگیری را ۰.۱ دادیم و تعداد تکرار الگوریتم را روی ۱۰۰ گذاشـــتیم. اگر این مقدار را بیشـــتر قرار دهیم دچار over fit می‌شـــویم. همچنین برای الگوریتم random forest تعداد تکرار الگوریتم را مطابق شـکل ۳-۴۳ روی ۲۰۰۰ قرار دادیم. با توجه به این اقدامات و اقدامات قبلی دقت این دو الگوریتم مطابق جدول ۳-۹ می‌شود.

```
#================

from xgboost import XGBClassifier
xgb = XGBClassifier(learning_rate=0.1 , n_estimators=100)
```

شکل ۳-۴۳.

```
from sklearn.ensemble import RandomForestClassifier

#random forest
rfc = RandomForestClassifier(n_estimators=2000)
```

شکل ۳-۴۴.

جدول ۳-۹. میزان دقت با تنظیم پارامترهای مدل یادگیری

نام مدل یادگیری	میزان دقت مدل یادگیری
XGboost	۰.۵۸۵
Random forest	۰.۵۹

همان طور که در جدول ۳-۹ مشـــاهده می‌کنید، تغییر چندانی در اعداد به وجود نیامد. تنظیماتی که برای مدل‌های یادگیری انجام دادیم، در بهترین عملکرد، این مقادیر را می‌دهند. با تغییر این مقادیر ممکن اســـت باعث کم شدن دقت مدل یادگیری هم شویم.

فصل ۴

نتیجه‌گیری

۴-۱ مقدمه

این پژوهش در راستای کمک به دانشجویان و دانش‌آموزان در راه بهبود یادگیری انجام شد. در این پروژه ما با بررسی عوامل مختلف بر روی یادگیری دانشجویان و دانش‌آموزان، با توجه به تحقیقات خودمان تصمیم گرفتیم که با توجه به فعالیت‌های دانشجویان در سامانه‌ی یادگیری مجازی، مدل‌های یادگیری آن‌ها را خوشه‌بندی کنیم و با سپس با توجه به این فعالیت‌ها میزان موفقیت دانشجویان را پیش بینی کنیم. در این بخش قصد داریم در مورد نتایجی که در انجام این تحقیق بدست آوردیم را توضیح دهیم.

۴-۲ دلایل بررسی مدل یادگیری

یکی از سوالات در ابتدای پژوهش ما این بود که چرا با وجود تحقیقات بسیار زیاد در این حوزه، و بدست آمدن نتایج آن‌ها، مجدد این تحقیقات در مورد مدل یادگیری دانشجویان ادامه دارد؟ برای پاسخ به این سوال تاریخچه این پژوهش‌ها را بررسی کردیم. از سال ۱۹۶۰ برای اینکه میزان توانایی دانشجویان را بررسی کنند یک پرسشنامه شامل ۶۰ سوال تهیه شد که از دانش‌آموزان خواسته شد این پرسشنامه‌ها را پر کنند. در آن پژوهش، دانش‌آموزان را از نظر سطح مهارتی به ۴ دسته تقسیم کردند. بعد این تحقیقات دیگری هم انجام شد. با بررسی سیر این تحقیقات متوجه شدم که بعد از ارزیابی توانایی دانشجویان، تحقیقات به سمت میزان موفقیت دانشجویان پیش رفت. بعد از آن برای بررسی عوامل دانشجویان غیر موفق، یکی از دلایلی که برای بررسی انتخاب شد، مدل یادگیری دانشجویان بود. یعنی هر چه به جلوتر آمدیم، موضوعات تحقیق جزئی تر می‌شد. همچنین با ظهور سامانه های یادگیری مجازی تحقیق در این حوزه بیشتر شد و با استفاده از مدل

های هوش مصنوعی سعی کردند مواردی که می‌خواهند را از بررسی عوامل مختلف یک دانشجو در سامانه، پیش‌بینی کنند.

۳-۴ مدل‌های یادگیری و معیارهای ارزیابی آن‌ها

یکی دیگر از مواردی که در این پژوهش انجام دادیم، بررسی مدل‌های مختلف یادگیری برای بررسی مسئله خودمان بود. ما سه مدل یادگیری naïve bayes و random forest و XGboost را برای انجام تحقیق خود انتخاب کردیم. در ابتدا این تصور را داشتیم که الگوریتم XGboost دقت بسیار بالاتری باید داشته باشد ولی در هنگام اجرا متوجه شدیم که دقت این الگوریتم در این مسئله خیلی بالاتر از الگوریتم random forest نیست.اما لازم است توجه داشته باشیم random forest میزان overfitting بالایی دارد که آن را برای استفاده بر روی داده های جدید غیر قابل اعتماد می کند برای همین استفاده از مدل آموزش داده شده با XGboost توصیه می شود. علاوه بر این با توجه به این که میزان دقت صفر مسئله ما ۰.۴۱ بود دریافتیم که الگوریتم naïve bayes با دقت ۰.۳۵ در بهترین حالت اصلا مناسب حل این مسئله نیست. دو الگوریتم دیگر با دقت حدود ۰.۵۸ به نسبت خیلی بهتر بودند اما در حالت کلی به این نتیجه رسیدیم برای بهبود این مدل یا باید پیش پردازش خیلی دقیق تری انجام دهیم یا باید تعداد داده‌های خودمان را زیاد کنیم. بهترین راه حل هم به نظر، افزایش تعداد داده ها می‌باشد.

مراجع

[1] AbuJbara, A., et al. (2018). OULAD: Predictive modeling of academic success using learning analytics in higher education. IEEE Transactions on Learning Technologies, 11(2), 168-178..

[2] Jin, Y., Liu, Y., Li, Y., & Wang, Z. (2017). Enhanced learning resource recommendation based on online learning style model. IEEE Transactions on Learning Technologies, 10(2), 233-242.

[3] Analytics Vidhya. (2019). Model Validation for Classification. Retrieved from

[4] Prashant Gupta. (2019). Naive Bayes Classifier in Python. Kaggle. Retrieved from

[5] Matplotlib. (2021). Matplotlib.pyplot.plot — Matplotlib 3.7.1 documentation. Retrieved

[6] Analytics Vidhya. (2021, June 17). Confusion Matrix for Multi-Class Classification. Retrieved

[7] Ackoff, R. L., & Emery, F. E. (1960). The educational testing of the future. Oxford Review of Education, 2(3), 181-198.

[8] National Center for Education Statistics. (n.d.). National Assessment of Educational Progress (NAEP). Retrieved from

[9] Alamri, A., Alshehri, M., Alfarraj, O., Alshahrani, M., & Alshammari, N. (2017). Predicting student success using learning analytics. International Journal of Emerging Technologies in Learning, 12(12), 124-130.

[10] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[11] Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5), 603-619.

[12] Fleming, N. D. (2011). VARK: A guide to learning styles. Retrieved

[13] "K-means clustering." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. 28 May 2023. Web. 11 Jun. 2023.

[14] Matsheka, I. (2021). K-means Clustering Algorithm: Applications, Evaluation Methods, and Drawbacks. Towards Data Science. Retrieved

Abstract

Artificial intelligence is one of the most important sciences that is integrated with our lives today. One of the areas where this science is used is the curriculum and cognitive sciences regarding the learning model and the prediction of students' grades. Learning model prediction and grades is a topic that has been researched. Researchers seek to discover the characteristics that affect learning and use them to predict the learning model and grades of each student. Today, due to the advancement of science and technology and the availability of the Internet for most people, virtual education or LMS is very popular. Researchers try to predict the learning model and grades of each student according to the activities of each student in virtual learning systems.

Keywords: Learning model, student grades, prediction, virtual learning system, artificial intelligence

Shahid Rajaee Teacher Training University
Faculty of Computer Engineering
Department of software
B. Sc. Thesis

Title:

Design and Implementation of a Learning Model Detection System for Predicting Students' Grades

Supervisor:
Dr. Sahar Kianian

By:
Paniz Taheri
Sajad Rahmani

Spring 2023