



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پایان نامه کارشناسی

عنوان مقاله- یادگیری ماشین برای پذیرش مدل های

چابک

گزارش نهایی

نگارش

ستایش ثانوی

استاد راهنما

دکتر رضا صفابخش

اردیبهشت ۱۴۰۰

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پایان نامه کارشناسی

عنوان مقاله-یادگیری ماشین برای پذیرش مدل های چابک گزارش نهایی

نگارش

ستایش ثانوی

استاد راهنما

دکتر رضا صفابخش

اردیبهشت ۱۴۰۰

سپاس‌گزاری

به نام آفریدگار پاک که انسان را از خاک آفرید و به واسطه عقل بر تمام موجودات رحمان داد و آنگاه دانش را وسیله تکامل عقل قرار داد .
با تشکر از استاد محترم جناب آقای دکتر رضا صفا بخش که این جانب را در تهیه این پروژه یاری نمودند از خداوند متعال پیروزی و موفقیت روز افزون ایشان را خواستاریم.

ستیش ثانوی

اردیبهشت ۱۴۰۰

چکیده

در این مقاله به دنبال یافتن یک مدل برای پذیرش اسکران از طریق یادگیری ماشین هستیم به این منظور با معرفی بیشتر اسکران و شرایط پذیرش آن و همچنین طرح مدل هایی از یادگیری ماشین پیش میرویم .

اسکران یکی از بهترین روش های متدولوژی چابک است که به صورت غالب مورد استفاده قرار می گیرد . امروزه استفاده از اسکران در پروژه ها و صنایع به دلیل مزایایی که دارد بسیار مورد استقبال قرار می گیرد ، با این حال، با توجه به شرایط و ویژگی هایی که برای پذیرش اسکران در پروژه ها نیاز داریم ممکن است استفاده از اسکران تا حدودی دچار چالش شود . تلفیق روش های چابک و علم داده برای دیدن دستاوردهای مکرر توسعه نرم افزار و برنامه ها باعث شده تا از تکنیک های یادگیری ماشین برای پیش بینی استفاده از روش اسکران چابک است .

در این مقاله با بررسی چالش های پذیرش اسکران و همچنین مدل ها و تکنیک های یادگیری ماشین سه مدل از یادگیری ماشین برای پذیرش اسکران مورد بررسی قرار گرفته و بهترین مدل با بیشترین دقت با توجه به محاسبات انجام شده معرفی خواهد شد.

واژه های کلیدی:

متدولوژی چابک، یادگیری ماشین، پذیرش اسکران، مدل پیش بینی، مهندسی نرم افزار، مدل های یادگیری ماشین ، اسکران

صفحه	فهرست مطالب	عنوان
۱.....	مقدمه	۱ مقدمه
۲.....	۱-۱ مقدمه	۱-۱ مقدمه
۵.....	۲ متدولوژی چابک	۲ متدولوژی چابک
۶.....	۱-۲ مقایسه مدل چابک با آبشار	۱-۲ مقایسه مدل چابک با آبشار
۶.....	۱-۱-۲ مدل آبشار	۱-۱-۲ مدل آبشار
۹.....	۲-۲ اسکرام	۲-۲ اسکرام
۹.....	۱-۲-۲ ایده های کلیدی	۱-۲-۲ ایده های کلیدی
۱۰.....	۲-۲-۲ تیم اسکرام	۲-۲-۲ تیم اسکرام
۱۳.....	۳-۲-۲ اسپرینت	۳-۲-۲ اسپرینت
۱۳.....	۴-۲-۲ پذیرش	۴-۲-۲ پذیرش
۱۶.....	۳-۲ موفقیت پروژه های چابک	۳-۲ موفقیت پروژه های چابک
۱۴.....	۱-۳-۲ معیارهای تعیین نتایج پروژه	۱-۳-۲ معیارهای تعیین نتایج پروژه
۱۷.....	۲-۳-۲ خلاصه	۲-۳-۲ خلاصه
۱۸.....	۳ یادگیری ماشین	۳ یادگیری ماشین
۱۹.....	۱-۳ یادگیری ماشین با روشهای چابک	۱-۳ یادگیری ماشین با روشهای چابک
۲۴.....	۱-۱-۳ خلاصه	۱-۱-۳ خلاصه
۲۶.....	۴ مدل های یادگیری ماشین	۴ مدل های یادگیری ماشین
۲۷.....	۱-۴ بررسی مدل ها	۱-۴ بررسی مدل ها
۲۸.....	۱-۱-۴ مدل اول: مجموعه ویژگی های کامل	۱-۱-۴ مدل اول: مجموعه ویژگی های کامل
۲۹.....	۲-۱-۴ مدل دوم: مدل پذیرش لگاریتمی تبدیل شده	۲-۱-۴ مدل دوم: مدل پذیرش لگاریتمی تبدیل شده
۳۰.....	۳-۱-۴ مدل سوم: پذیرش لگاریتمی تبدیل شده با ویژگی ها حذف شده	۳-۱-۴ مدل سوم: پذیرش لگاریتمی تبدیل شده با ویژگی ها حذف شده
۳۳.....	۴-۱-۴ خلاصه	۴-۱-۴ خلاصه
۳۴.....	۵ جمع بندی و نتیجه گیری و پیشنهادات	۵ جمع بندی و نتیجه گیری و پیشنهادات
۳۶.....	منابع و مراجع	منابع و مراجع

شکل	فهرست اشکال و جداول	صفحه
۱-۲	مراحل مدل آبخاری	۷
۲-۲	فرآیند اجرای اسکرام	۹
۱-۳	رابطه بین ویژگی ها و پذیرش اسکرام	۲۱
۱-۴	رابطه بین پذیرش واقعی و پیش بینی شده برای مجموعه ویژگی های کامل	۲۸
۲-۴	باقی مانده ها ای مجموعه ویژگی های کامل	۲۹
۳-۴	همبستگی واقعی و پیش بینی شده برای تغییر شکل لگاریتمی	۲۹
۴-۴	باقی مانده ها برای پذیرش لگاریتمی تبدیل شده	۳۰
۵-۴	رابطه بین ویژگی ها و پذیرش اسکرام با ویژگی های حذف شده	۳۰
۶-۴	رابطه بین ویژگی ها و پذیرش اسکرام ا ویژگی های حذف شده	۳۱
۷-۴	مقدار واریانس و خطا طبق فرمول های گفته شده (جدول)	۳۱

فصل اول

مقدمه

۱- مقدمه

روش های توسعه نرم افزار چابک در بسیاری از صنایع اقتصاد جهان مورد استفاده قرار می گیرد. چارچوب Scrum (اسکرام) روش غالب Agile (متدولوژی چابک) است که برای توسعه ، تحویل و نگهداری محصولات پیچیده نرم افزاری استفاده می شود. متدولوژی های چابک برای طیف گسترده ای از اهداف توسعه برنامه ها استفاده می شوند و به همین دلیل بسیار محبوب هستند.

قبل از تصویب گسترده Agile ، بسیاری از سازمان ها خود را درگیر روش های سنتی "آبشار" میکردند . متدولوژی چابک به جای اینکه ماه ها یا سال ها منتظر بماند تا یک پروژه نرم افزاری از طریق طراحی ، توسعه ، آزمایش و استقرار به راه خود ادامه دهد ، روی تکرارهای کوتاه با هدف تولید سریع یک محصول قابل تحویل برای تأمین نیازهای فوری صاحب مشاغل متمرکز شد ، و سپس به طور مرتب تکرار می شود زیرا نیازها بیشتر تصفیه می شوند.

تردیدی نیست که روش های چابک برای همیشه در سازمان هایی که سرعت تغییرات همچنان در حال تسریع است ، روش توسعه و انتشار کارایی سازمان ها را تغییر داده اند. در حالی که موفقیت پروژه های نرم افزاری زمانی که از روش های Agile در مقایسه با WaterFall (مدل آبشاری) استفاده می شود، به طور قابل توجهی بهبود یافته است ، ولی بخش بزرگی از پروژه ها همچنان به چالش کشیده می شوند یا شکست می خورند.[۱]

سازمان ها با اسکرام شروع می کنند و اغلب با مشکلاتی در زمینه پذیرش روبرو می شوند که مانع سود مورد انتظار می شود. یک روش برای بهبود پذیرش اسکرام ، بررسی برخی از الگوهای ضد پذیرش اسکرام و پرداختن به آنها است. در اینجا برخی از الگوهای رایج وجود دارد.[۲]

- تیم توسعه در برابر اسکرام مقاومت می کند.
- آموزش اسکرام توسط تیم توسعه به خوبی رعایت نشده و آنها به عادت های قدیمی برمی گردند.
- تیم ها فاقد مهارت لازم برای بدست آوردن نیازهای واضح و مشخص با دامنه و رفتار مشخص و تجزیه آنها به داستان های کوچک هستند.
- تخمین نادرست است و انجام آن زمان زیادی می برد.
- تیم ها از اسکرام در جایی استفاده می کنند که قابل استفاده نباشد.
- تیم ها از یک روش غیر چابک در یک sprint (سرعت) استفاده می کنند.
- درگیر نکردن صاحبان محصولات.

- تیم‌هایی که با هم کار می‌کنند چالش‌هایی دارند.

- مدیر یا رهبر تیم اسکرام، نقش خود را در اسکرام درک نمی‌کند. [۲]

از سال ۲۰۱۵، نیاز به تلفیق روش‌های چابک و علم داده برای دیدن دستاوردهای مکرر توسعه نرم‌افزار و برنامه‌ها ذکر شده است. در این مقاله استفاده از تکنیک‌های یادگیری ماشین برای پیش‌بینی استفاده از روش اسکرام چابک بررسی خواهد شد. [۱]

هدف اصلی این مقاله استفاده از یادگیری ماشین برای ایجاد مدل‌های پیش‌بینی برای پذیرش اسکرام و شناسایی یک مدل اولیه با بالاترین دقت پیش‌بینی است. مدل‌های یادگیری ماشین با استفاده از تکنیک‌های آماری رگرسیون خطی چندگانه اجرا شده است. به طور خاص، یک مدل کامل پذیرش مجموعه ویژگی‌ها، یک مدل پذیرش لگاریتمی تبدیل شده و یک لگاریتمی تبدیل شده با مدل پذیرش ویژگی‌های حذف شده، برای صحت پیش‌بینی ارزیابی خواهد شد. مدل مجموعه کامل ویژگی‌ها شامل چهارده ویژگی:

۱. مهندسی بیش از حد (over-engineering)

۲. مزیت نسبی (relative advantage)

۳. شناخت (recognition)

۴. تجربه (experience)

۵. کار تیمی (teamwork)

۶. تخصص (specialization)

۷. تشدید تعهد (escalation of commitment)

۸. سازگاری (compatibility)

۹. مدیریت منابع (resource management)

۱۰. همکاری مشتری (customer collaboration)

۱۱. پیچیدگی (complexity)

۱۲. آموزش (training)

۱۳. مدیریت sprint (سرعت) (sprint management)

۱۴. رفتار سازمانی (organizational behaviour) شامل

مجموعه ویژگی های کامل برای پیش بینی پذیرش اسکرام است.

مشابه مدل لگاریتمی تبدیل شده ، مدل ویژگی های حذف شده نیز مورد بررسی قرار میگیرند ،
بااین تفاوت که سه ویژگی از چهارده ویژگی یعنی ویژگی های ”تجربه ، تشخیص و سازگاری” از مجموعه
ویژگی های پیش بینی پذیرش اسکرام حذف شده است.

در ادامه ی این مقاله به بررسی آماری موفقیت پروژه های Scrum پرداخته و با توجه به معیار
های تعیین نتیجه ی پروژه، موفقیت این پروژه ها مورد بررسی قرار میگیرند و نهایتاً با توجه به مدل
های یادگیری ماشین بهترین مدل برای تصویب پذیرش اسکرام معرفی خواهد شد.

فصل دوم

متدولوژی

چابک

۲ متدولوژی چابک

در توسعه نرم افزار ، شیوه های چابک شامل کشف نیازمندی ها و توسعه راه حل ها از طریق تلاش مشترک تیم های خودسازمان دهنده و عملکردی تعاملی با مشتری (های) / کاربر نهایی (کاربران نهایی) است. این برنامه از برنامه ریزی انطباقی ، توسعه تکاملی ، تحویل زودرس و بهبود مستمر حمایت میکند و پاسخ های انعطاف پذیر را برای تغییر تشویق می کند. توسط بیانیه (Manifesto) توسعه نرم افزار چابک رایج شد.

ارزش ها و اصول مورد حمایت در این بیانیه از طیف گسترده ای از چارچوب های توسعه نرم افزار ، از جمله اسکرام و کانبان (Kanban) استخراج شده و آنها را مورد حمایت قرار می دهند. در حالی که شواهد حکایتی زیادی وجود دارد مبنی بر اینکه اتخاذ روش ها و ارزشهای چابک ، چابکی متخصصان نرم افزار ، تیم ها و سازمان ها را بهبود می بخشد ، اما شواهد تجربی درهم ریخته است و یافتن آنها دشوار است. [۳]

۲-۱ مقایسه مدل چابک با آبشار

همانطور که قبلاً گفته شد موفقیت پروژه های نرم افزاری زمانی که از روش های چابک در مقایسه با روش آبشاری (Waterfall)، استفاده می شود ، به طور قابل توجهی بهبود یافته است در این بخش به توضیح بیشتر مدل آبشاری میپردازیم تا درک این مقایسه راحت تر شود.

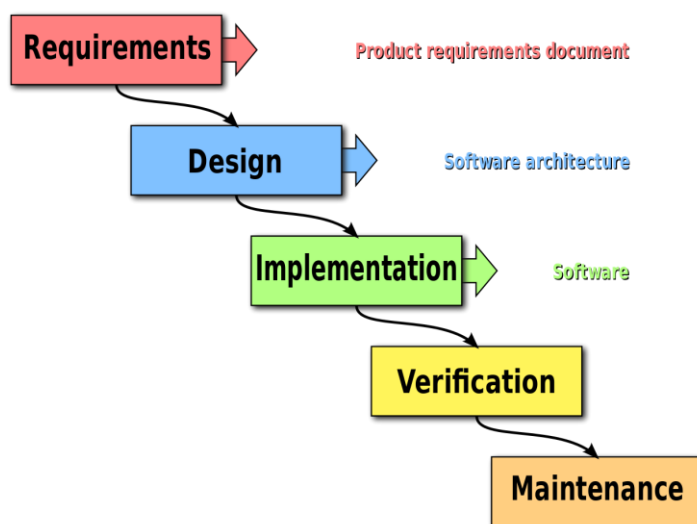
۲-۱-۱ مدل آبشار

مدل آبشار تجزیه فعالیت های پروژه به مراحل متوالی خطی است ، جایی که هر مرحله به موارد قابل تحویل مرحله قبلی بستگی دارد و مربوط به یک تخصص است. این روش برای موارد خاصی از طراحی مهندسی معمول است در توسعه نرم افزار ، این روش معمولاً در بین رویکردهای کمتر تکرار شونده و انعطاف پذیر قرار دارد ، زیرا پیشرفت در یک جهت ("رو به پایین" مانند یک آبشار) از طریق مراحل تصور، شروع ، تجزیه و تحلیل ، طراحی ، ساخت ، آزمایش ، استقرار و نگهداری جریان دارد. مدل توسعه آبشار از صنایع تولیدی و ساختمانی نشأت گرفته است. که در آن محیط های فیزیکی بسیار ساختار یافته به این معنی بودند که تغییرات طراحی خیلی زودتر از روند توسعه گرانتر می شوند وقتی برای اولین بار برای توسعه نرم افزار تصویب شد ، هیچ گزینه جایگزینی برای کارهای خلاقانه دانش بنیان وجود نداشت.

در مدل اصلی آبشار رویس (Royce)، مراحل زیر به ترتیب دنبال می شوند:

۱. نیازمندی های سیستم و نرم افزار: در سند مورد نیاز محصول ثبت شده است.

۲. تجزیه و تحلیل: در نتیجه مدل ها ، طرحواره و قوانین تجاری ایجاد می شود.
 ۳. طراحی: منجر به معماری نرم افزار می شود.
 ۴. کدگذاری: توسعه ، اثبات و ادغام نرم افزار
 ۵. آزمایش: کشف سیستم و رفع اشکال در نقص
 ۶. عملیات: نصب ، انتقال ، پشتیبانی و نگهداری سیستم های کامل
- با این وجود ، مدل های مختلف آبشار اصلاح شده (از جمله مدل نهایی رويس) می توانند تغییرات جزئی یا عمده ای در این روند داشته باشند.
- این تغییرات شامل :
- بازگشت به چرخه قبلی پس از یافتن نقص در پایین دست ، یا بازگشت به مرحله طراحی در صورت در نظر گرفتن مراحل پایین دست است. [۴]



شکل ۲-۱: مراحل مدل [۳]

آبشاری تفاوت های کلیدی مدل آبشار و چابک:

۱. آبشار یک مدل چرخه زندگی متوالی خطی است در حالی که چابک تکرار مداوم توسعه و آزمایش در فرایند توسعه نرم افزار است.
۲. روش چابک به انعطاف پذیری شناخته شده است در حالی که آبشار یک روش ساخت نرم افزار ساخت یافته است.
۳. متدولوژی چابک از یک رویکرد افزایشی پیروی می کند در حالی که آبشار یک روند طراحی پی در پی است.

۴. چابک همزمان با توسعه نرم افزار تست را انجام می دهد در حالی که در تست روش آبشار پس از مرحله "ساخت" انجام می شود.

۵. چابک امکان تغییر در نیاز به توسعه پروژه را فراهم می کند در حالی که آبشار پس از شروع توسعه پروژه هیچ محدودیتی برای تغییر نیازمندی ها ندارد. [۵]

در حالی که این روش ها تفاوت های قابل توجهی دارند ، با این حال هر روش مدیریت پروژه در نهایت یک هدف دارد:

تسهیل تکمیل پروژه ها.

به همین منظور ، هر روش از طریق ساختار و ارتباطات ، فرایندهای کاری تیم شما را مدیریت کند. اگرچه هر یک از این روش ها را به طرز متفاوتی پیاده سازی می شوند ، اما چابک ، آبشار ، کانبان و اسکرام همه این اشتراکات را دارند.

اگرچه ممکن است تفاوت بین روش ها کم باشد ، اما مطمئن باشید که وجود دارد. در حقیقت ، این جزئیات به ظاهر کوچک تفاوت زیادی در نحوه عملکرد یک روش ایجاد می کنند. با این حساب ، بیایید بررسی کنیم که کدام روش مدیریت پروژه متمایز است.

آبشار برای پروژه هایی که به صورت خطی تکمیل شده اند بهترین عملکرد را دارد و اجازه بازگشت به مرحله قبل را نمی دهد.

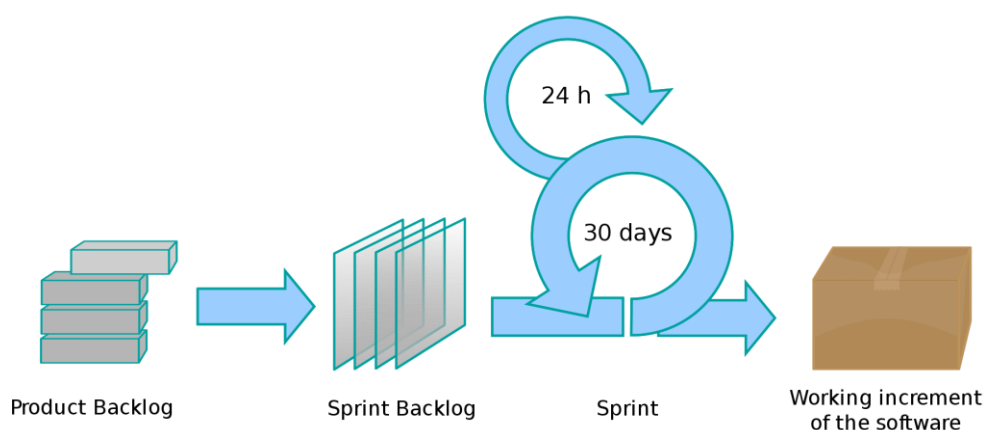
چابک بر گردش کارهای سازگار و همزمان تمرکز دارد. روشهای چابک پروژه ها را به دوره های تکراری کوچکتر تقسیم می کند.

کانبان در درجه اول به فکر بهبود روند است.

اسکرام به فکر انجام کارهای سریعتر است. [۶]

۲-۲ اسکرام

اسکرام چارچوبی است که با بهره‌گیری از ذهنیت چابک برای تولید، تحویل و حفظ محصولات پیچیده، با تأکید اولیه بر توسعه نرم افزار مورد استفاده قرار گرفته است. اگرچه در زمینه‌های دیگر از جمله تحقیق، فروش، بازاریابی و فن آوری‌های پیشرفته نیز استفاده شده است. این برنامه برای تیم‌های ده نفره یا کمتر طراحی شده است، که بیش از یک ماه و معمولاً دو هفته کار خود را به اهدافی تقسیم می‌کنند که می‌تواند طی تکرارهای زمانی مشخص، به نام اسپرینت (sprint) انجام شود. تیم اسکرام پیشرفت در جلسات روزانه (بسته به وقت) ۱۵ دقیقه یا کمتر را که "اسکرام روزانه" نامیده می‌شود، ارزیابی می‌کند. در پایان اسپرینت، تیم دو جلسه دیگر برگزار می‌کند: مرور اسپرینت که کارهایی را که برای ذینفعان برای گرفتن بازخورد انجام شده است نشان می‌دهد، و اسپرینت گذشته که تیم را قادر به انعکاس و پیشرفت می‌کند.



شکل ۲-۲: فرآیند اجرای اسکرام [۷]

۲-۲-۱ ایده‌های کلیدی

اسکرام یک چارچوب تکراری و افزایشی برای مدیریت کارهای پیچیده است. این چارچوب فرضیه هایرویکرد سنتی و متوالی برای توسعه محصول را به چالش می‌کشد و تیم‌ها را قادر می‌سازد تا با تشویق همکاری فیزیکی یا همکاری آنلاین نزدیک همه اعضای تیم، خود را سازماندهی کنند، و همچنین ارتباط چهره به چهره روزانه بین همه اعضای تیم و رشته‌های درگیر را دارد. یک اصل کلیدی اسکرام

تشخیص دوگانه (dual recognition) است که مشتریان دامنه نیازمندی های مورد نظر خود را تغییر میدهند (که اغلب نوسانات نیازمندی ها نامیده می شود) و چالش های غیرقابل پیش بینی وجود دارد - که برای آنها یک رویکرد پیش بینی یا برنامه ریزی شده مناسب نیست.

این تغییرات از منابع مختلفی ناشی می شود ، اما فهمیدن دلیل آن بی ربط و بی فایده است پس باید تغییر را به راحتی پذیرفت و برای منافع آنالیز کرد. اسکرام یک رویکرد تجربی مبتنی بر شواهد را اتخاذ می کند - قبول می کند که مسئله را نمی توان به طور کامل درک کرد یا در ابتدا تعریف کرد ، و در عوض بر روی چگونگی به حداکثر رساندن توانایی تیم برای ارائه سریع ، پاسخگویی به نیازهای جدید و سازگار شدن با تکامل تمرکز می کند. [۷]

۲-۲-۲ تیم اسکرام

واحد اساسی اسکرام تیم کوچکی از افراد است که متشکل از یک مالک محصول (product owner) ، یک استاد اسکرام (Scrum Master) و توسعه دهندگان (developers) است. این تیم خود-مدیریت ، عملکردی متقابل است و در یک زمان بر یک هدف متمرکز است: هدف محصول.

• مالک محصول صاحب محصول ، نماینده ذینفعان محصول و صدای مشتری است (یا ممکن است

خواسته های یک کمیته را نشان دهد)، مسئول ارائه نتایج خوب تجاری است. از این رو ، مالک محصول

درمورد تعداد محصول و به حداکثر رساندن ارزشی که تیم تحویل می دهد ، پاسخگو است. صاحب

محصول ، محصول را از نظر نتایج مشتری مداری تعریف می کند ، آنها را به پس زمینه محصول اضافه

می کند و براساس اهمیت و وابستگی ، آنها را اولویت بندی می کند.

یک تیم اسکرام باید فقط یک مالک محصول داشته باشد (اگرچه یک مالک محصول می تواند بیش از یک تیم را پشتیبانی کند) و به شدت توصیه می شود که این نقش را با نقش استاد اسکرام ترکیب نکنید. صاحب محصول باید بر جنبه تجاری توسعه محصول تمرکز کند و بیشتر وقت خود را صرف برقراری ارتباط با ذینفعان و تیم کند. صاحب محصول چگونگی دستیابی تیم به یک راه حل فنی را تعیین نمیکند ، بلکه به دنبال اجماع بین اعضای تیم و مهندسان (توسعه دهندگان) در تیم اسکرام است. بنابراین، یک مالک خوب محصول باید بتواند آنچه را که مورد نیاز تجارت است برقرار کند ، از آنها بپرسد که چرا به آن نیاز دارند (زیرا ممکن است راه های بهتری برای دستیابی به آن وجود داشته باشد) و پیام را به شرکت های ذینفع از جمله توسعه دهندگان ، در صورت لزوم با استفاده از زبان فنی منتقل کند. صاحب محصول از ابزارهای تجربی اسکرام برای مدیریت کارهای بسیار پیچیده در عین کنترل ریسک و دستیابی به ارزش استفاده می کند.

ارتباطات مسئولیت اصلی صاحب محصول است. توانایی انتقال اولویت ها و همدلی با اعضای تیم و ذینفعان برای هدایت توسعه محصول در مسیر صحیح حیاتی است. نقش صاحب محصول ، پل ارتباطی بین تیم و سهامداران آن را ایجاد می کند و به عنوان نماینده ای برای ذینفعان تیم و به عنوان نماینده تیم در کل جامعه ذینفعان عمل می کند. موارد زیر برخی از وظایف ارتباطی صاحب محصول با ذینفعان است:

۱. نسخه های منتشر شده را تعریف و اعلام کند.
۲. وضعیت تحویل و کالا را اعلام کند.
۳. در جلسات پیشرفت را به اشتراک بگذارد.
۴. موانع ، وابستگی ها و مفروضات را با سهامداران به اشتراک بگذارد.
۵. درباره اولویت ها ، محدوده ، بودجه و برنامه مذاکره کند.
۶. اطمینان حاصل کند که product backlog (محصول پشتیبان) محصول قابل مشاهده ، شفاف و واضح است.

همدلی یک ویژگی اصلی برای صاحب محصول است - توانایی قرار دادن خود در جای دیگری. یک مالک محصول با ذینفعان مختلف با زمینه های مختلف ، نقش های شغلی و اهداف گفتگو می کند - و باید بتواند از این دیدگاه های مختلف قدردانی کند. برای اثربخشی ، عاقلانه است که یک مالک محصول از سطح جزئیات مورد نیاز مخاطب مطلع شود. توسعه دهندگان نیاز به بازخورد و مشخصات کامل دارند تا بتوانند محصولی را تا حد انتظار تولید کنند ، در حالی که یک حامی مالی اجرایی ممکن است فقط به خلاصه پیشرفت نیاز داشته باشد. ارائه اطلاعات بیشتر از حد ضروری ممکن است باعث کاهش علاقه سهامداران و اتلاف وقت شود. یک وسیله ارتباطی مستقیم توسط دارندگان محصولات فصلی ترجیح داده می شود.[۷]

• توسعه دهندگان

اصطلاح توسعه دهندگان به هرکسی که در توسعه و پشتیبانی سیستم یا محصول نقشی داشته باشد ، گفته می شود و می تواند شامل محققان ، معماران ، طراحان ، متخصصان داده ، آمار شناسان ، تحلیلگران ، مهندسان ، برنامه نویسان و آزمایش کنندگان باشد. با این حال ، به دلیل گیجی که ممکن است برخی از افراد احساس نکنند که اصطلاح "توسعه دهنده" در مورد آنها اعمال می شود ، اغلب فقط به عنوان اعضای تیم نامیده می شوند.

تیم در حال سازماندهی خود است. در حالی که هیچ کاری نباید به تیم منتقل شود مگر از طریق صاحب محصول ، و انتظار می رود استاد اسکرام تیم را در برابر حواس پرتی محافظت کند ، اما تیم

به منظور دستیابی به حداکثر درک و بازخورد بی واسطه ، به تعامل مستقیم با مشتریان و یا سهامداران دعوت می شود.[۷]

• استاد اسکرام

اسکرام توسط یک استاد اسکرام تسهیل می شود ، که مسئول از بین بردن موانع توانایی تیم در ارائه اهداف و محصولات قابل تحویل است. استاد اسکرام یک رهبر سنتی تیم یا مدیر پروژه نیست بلکه به عنوان یک مانع بین تیم و هرگونه تأثیر حواس پرتی عمل می کند. استاد اسکرام اطمینان حاصل میکند که چارچوب اسکرام با مربیگری تیم در تئوری و مفاهیم اسکرام دنبال می شود ، اغلب جلسات اصلی را تسهیل می کند و تیم را به رشد و پیشرفت ترغیب می کند. از این نقش به عنوان یک تسهیل کننده تیم یا رهبر خدمتگذار برای تقویت این دیدگاه های دوگانه نیز یاد شده است. مسئولیت های اصلی یک استاد اسکرام شامل موارد زیر است اما محدود به آنها نیست :

- ۱ . کمک به مالک محصول در حفظ و نگهداری محصول با شکلی که اطمینان حاصل کند کار لازم به خوبی درک شده است ، بنابراین تیم می تواند به طور مداوم پیشرفت کند
۲. کمک به تیم برای تعیین تعریف انجام شده برای محصول ، با ورود سهامداران اصلی
۳. مربیگری تیم ، در اصول اسکرام ، به منظور ارائه ویژگی های با کیفیت بالا برای محصول خود
۴. آموزش ذینفعان اصلی و بقیه اعضای سازمان در مورد اصول اسکرام (و احتمالاً چابک)
۵. کمک به تیم اسکرام برای جلوگیری یا جلوگیری از موانع پیشرفت تیم ، چه داخلی و چه خارجی
۶. ارتقا-خود سازماندهی و عملکرد متقابل در تیم
۷. تسهیل رویدادهای تیمی برای اطمینان از پیشرفت منظم

استاد اسکرام به افراد و سازمان ها کمک می کند تا تفکر تجربی و ناب داشته باشند ، و امیدها را برای اطمینان و قابل پیش بینی بودن پشت سر بگذارند. یکی از راه های متفاوت بودن نقش اصلی اسکرام بایک مدیر پروژه این است که یک مدیر پروژه ممکن است مسئولیت مدیریت افراد را داشته باشد و استاد اسکرام این وظایف را ندارد. از آنجا که انتظار می رود تیم توانمند و خود سازماندهی شود ، یک استاد اسکرام جهت دهی محدودی را ارائه می دهد. اسکرام به طور رسمی نقش مدیر پروژه را تشخیص نمی دهد ، زیرا تمایلات سنتی برای کنترل و کنترل مشکلاتی را به وجود می آورد.[۷]

۲-۲-۳ اسپرینت

اسپرینت (که به عنوان تکرار یا جعبه زمان (iteration or timebox) را باز کرده و آن را دو بار اجرا کنید. سپس حالت اجرا را از نيز شناخته می شود) واحد اصلی توسعه در اسکرام است. اسپرینت یک تلاش زمانی است. به این معنی که مدت زمانی برای هر اسپرینت از قبل توافق و تعیین شده است و به طور معمول بین یک هفته تا یک ماه است و به طور میانگین دو هفته است.

هر اسپرینت با یک رویداد برنامه ریزی اسپرینت شروع می شود که در آن یک هدف اسپرینت ساخته می شود و تعداد زیادی از مسابقات اسپرینت ظاهر می شود، که شامل کارهای در نظر گرفته شده برای اسپرینت آینده است. هر اسپرینت با دو رویداد به پایان می رسد: اول بررسی اسپرینت (پیشرفتی که به ذینفعان نشان داده شده است) و دوم بررسی و جمع بندی اسپرینت گذشته sprint backlog (شناسایی دروس و پیشرفت برای اسپرینت های بعدی).

در آغاز یک اسپرینت، تیم اسکرام یک جلسه ی برنامه ریزی اسپرینت را برگزار می کنند تا:

۱. موافقت با هدف اسپرینت و شرح مختصری از آنچه تیم اسکرام پیش بینی می کنند تا پایان اسپرینت (براساس اولویت های تعیین شده توسط صاحب محصول) ارائه کنند.

۲. موارد بررسی و جمع بندی شده ی محصول را که به این هدف کمک می کند، انتخاب کنند.

۳. بحث و توافق متقابل درمورد اینکه چه مواردی در طول آن اسپرینت انجام شود، یک

sprint backlog را تشکیل دهند.

حداکثر مدت زمان برنامه ریزی اسپرینت هشت ساعت برای یک اسپرینت ۴ هفته ای است. با تشریح کار مفصل، اگر تیم معتقد باشد که آنها نمی توانند آن کار را در یک اسپرینت انجام دهند، ممکن است برخی از اقلام موجود در محصول تقسیم شده یا به محصول قبلی برگردانده شوند [۷].

۲-۲-۴ پذیرش

ترکیب کردن اسکرام با سایر روش های توسعه نرم افزار معمول است زیرا اسکرام کل چرخه عمر محصول را پوشش نمی دهد. بنابراین، سازمان ها برای ایجاد یک اجرای جامع تر، نیاز به افزودن فرایندهای اضافی دارند. به عنوان مثال، در ابتدای توسعه محصول، سازمان ها معمولاً راهنمایی فرایندها را درمورد تجارت، جمع آوری و اولویت بندی نیازها، طراحی سطح بالا و پیش بینی بودجه و برنامه اضافه می کنند. نویسندگان و جوامع مختلف افرادی که از اسکرام استفاده می کنند نیز تکنیک های

دقیق‌تری را برای نحوه استفاده یا انطباق اسکرام با مشکلات خاص یا سازمان‌ها پیشنهاد داده‌اند. بسیاری از افراد، از این تکنیک‌های روش شناختی به عنوان الگوها یاد می‌کنند.

• اسکرامبان

scrumban یک مدل تولید نرم افزار مبتنی بر اسکرام (scrum) و کانبان (kanban) است. اسکرامبان مخصوصاً برای تعمیر و نگهداری محصول با موارد کار مکرر و غیرمنتظره مانند نقص تولید یا خطاهای برنامه نویسی مناسب است. در چنین مواردی ممکن است اسپرینت‌های کوتاه مدت چارچوب اسکرام سود کمتری داشته باشند، گرچه با توجه به تیم و شرایط موجود، حوادث روزانه اسکرام و سایر اقدامات هنوز هم می‌توانند اعمال شوند. تجسم مراحل کار و محدودیت‌های کار ناتمام و نقص به صورت همزمان از مدل کانبان آشناست. با استفاده از این روش‌ها، گردش کار تیم به گونه‌ای هدایت می‌شود که حداقل زمان اتمام هر مورد کار یا خطای برنامه نویسی را ممکن می‌سازد، و از طرف دیگر اطمینان می‌دهد که هر عضو تیم به طور مداوم به کار گرفته می‌شود. برای نشان دادن هر مرحله از کار، تیم‌هایی که در همان فضا کار می‌کنند اغلب از یادداشت‌های پس از آن یا تخته سفید بزرگ استفاده می‌کنند. در مورد تیم‌های غیرمتمرکز، از نرم افزارهای صحنه پردازی مانند Jira، Assembla یا Agilo می‌توان استفاده کرد.

تفاوت عمده بین اسکرام و کانبان این است که در اسکرام کار به اسپرینت‌هایی تقسیم می‌شود که مدت زمان مشخصی را دوام می‌آورند، در حالی که در کانبان جریان کار مداوم است. این در جداول مرحله کار قابل مشاهده است، که در اسکرام پس از هر اسپرینت، تخلیه می‌شوند، در حالی که در کانبان تمام کارها در یک جدول مشخص شده‌اند. اسکرام روی تیم‌هایی با دانش چند وجهی متمرکز است، در حالی که کانبان تیم‌های تخصصی و کاربردی را امکان پذیر می‌کند.

• اسکرام اسکرام

scrum of scrums تکنیکی برای کار کردن اسکرام در مقیاس است، برای تیم‌های مختلفی که روی یک محصول کار می‌کنند، به آنها امکان می‌دهد درباره پیشرفت وابستگی متقابل خود، با تمرکز بر چگونگی هماهنگی ارائه نرم افزار، به ویژه در زمینه‌های همپوشانی و یکپارچه سازی، بحث کنند. بسته به شانس (زمان بندی) اسکرام اسکرام، اسکرام مربوطه روزانه برای هر تیم اسکرام با تعیین یک عضو به عنوان سفیر برای شرکت در اسکرام اسکرام با سفیران سایر تیم‌ها به پایان می‌رسد. بسته به زمینه، سفیران ممکن است همکار فنی یا استاد اسکرام هر تیم باشند.

به جای اینکه به سادگی یک به روزرسانی پیشرفت انجام شود، اسکرام اسکرام باید به چگونگی کار تیم‌ها برای حل، کاهش یا پذیرش هرگونه خطرات، موانع، وابستگی‌ها و مفروضات که مشخص شده‌اند، متمرکز شود. اسکرام اسکرام این خطرات را از طریق یک بلاگ برای خودش ردیابی می‌کند.

کند ، مانند تخته ی خطر که معمولاً منجر به هماهنگی بیشتر و همکاری بین تیم ها میشود . به عبارت دیگر باید شبیه یک اسکرام روزانه باشد ، و هر سفیر به چهار سوال زیر پاسخ دهد:

از زمان آخرین دیدار ما ، تیم شما چه خطرات ، موانع ، وابستگی ها یا مفروضاتی را برطرف کرده است؟ قبل از اینکه دوباره با هم ملاقات کنیم ، تیم شما چه خطرات ، موانع ، وابستگی ها یا فرضیاتی را برطرف می کند؟ آیا خطرات ، موانع ، وابستگی ها یا پیش فرض های جدیدی باعث کند شدن سرعت تیم شما یا جلوگیری از آنها می شود؟ آیا شما در معرض خطر ، مانع ، وابستگی یا فرض جدیدی هستید که مانع تیم دیگری شود؟

• اسکرام در مقیاس بزرگ

(LeSS) یک چارچوب توسعه محصول است که اسکرام را با قوانین و دستورالعمل های مقیاس گذاری گسترش می دهد بدون اینکه اهداف اصلی اش را از دست بدهد.

دو سطح برای چارچوب وجود دارد: سطح اول LeSS برای حداکثر هشت تیم طراحی شده است.

سطح دوم ، معروف به " LeSS Huge " عناصر مقیاس اضافی را برای توسعه با u معرفی می کند.[۷]

۲-۳ موفقیت پروژه های چابک

در مورد مزایا و موفقیت روش های توسعه نرم افزار چابک نسبت به روش های سنتی مانند روش آبشار توضیحات کافی وجود دارد. با این حال ، حتی وقتی سازمان ها از روش های چابک برای پروژه های توسعه نرم افزار استفاده می کنند ، کمتر از نیمی از این پروژه ها بود موفق به نظر می رسد. [۱]

۲-۳-۱ معیارهای تعیین نتایج پروژه

معیارهای مدرن گروه Standish برای تعیین نتایج پروژه به عنوان محدودیت سه گانه شناخته میشود .

این محدودیت ها عبارتند از:

۱. این پروژه در حد بودجه برنامه ریزی شده باقی مانده است. (OnBudget.)

۲. پروژه در یک برآورد زمان معقول حل شد. (OnTime.)

۳. یک نتیجه رضایت بخش - این پروژه رضایت کاربر و مشتری را فراهم آورد حتی اگر تغییراتی در محدوده اولیه ایجاد شده باشد.

بنابراین ، تعریف های نتیجه پروژه با در نظر گرفتن محدودیت سه گانه می تواند به شرح زیر خلاصه شود:

۱. موفقیت آمیز - پروژه ای که هر سه محدودیت OnTime ، OnBudget و نتیجه مطلوب را برآورده کرده است.

۲. چالش برانگیز - پروژه ای که پس از اتمام پروژه دو محدودیت از سه محدودیت را به دست آورده است ، به عنوان مثال پروژه با بودجه با نتیجه رضایت بخش تحویل داده شد اما به زمان برنامه ریزی شده تحویل نرسید.

۳. ناموفق - پروژه ای که قبل از اتمام آن لغو شده بود یا تکمیل شده بود اما مورد استفاده قرار نگرفت. گزارش CHAOS گروه Standish در سال ۲۰۱۸ اظهار داشت که ۴۲٪ از پروژه های چابک بررسی شده موفق بوده اند ، در حالی که ۵۰٪ به چالش کشیده شده اند و ۸٪ نیز به عنوان شکست گزارش شده اند .

در حالی که موفقیت ۴۲ درصدی نرخ ایده آلی نیست ، اما با این حال پیشرفت نسبت به گزارش های

قبلی CHAOS گروه Standish است. به عنوان مثال ، گزارش سال ۲۰۱۵ برای سال ۲۰۱۱ تا ۲۰۱۵ موفقیت پروژه چابک را ۳۹ % ، پروژه ها ناموفق را ۵۲ % و پروژه های به چالش کشیده شده را ۹ % اعلام کرد. هنگام ترکیب پروژه های چابک و آبشار ، نتایج پروژه موفقیت آمیز با ۲۹ % پروژه هایی که چالش داشتند ۵۲ % و پروژه های ناموفق را ۱۹ % تجربه کردند.

نظرسنجی جهانی همچنین اظهار داشت که "۹۵ درصد پاسخ دهندگان گزارش کرده اند که حداقل برخی از پروژه های چابک آنها موفق بوده اند و ۴۸ درصد گزارش داده اند که بیشتر یا همه پروژه های چابک آنها موفق بوده اند."

بنابراین ، نویسندگان بدون توجه به صنعت ، روش و اندازه پروژه ، از میزان موفقیت پایین نتایج پروژه توسعه نرم افزار آگاه هستند. ما خوش بین هستیم که موفقیت در نتیجه پروژه آینده سیر صعودی خواهد داشت ، با این حال ، ما همچنین آگاه هستیم که تسریع در فن آوری های خودمختار و همگرا می تواند مشکل را عمیق تر کند.

بنابراین ، ما فکر می کنیم که می توان از الگوریتم های یادگیری ماشین (به طور خلاصه (ی.م)) برای کمک به بهبود موفقیت نتایج پروژه استفاده کرد. به عنوان آغازی برای حل این مسئله پیچیده ، این مقاله تحقیقاتی در حال توسعه مدل های (ی.م) برای پیش بینی پذیرش روش چابک بود.

قبل از پیش بینی نتایج پروژه ، ما فکر می کنیم که پیش بینی استفاده از روش چابک در مراحل اولیه ورود روش ها در پروژه های توسعه نرم افزار می تواند به درک و نتایج آینده پروژه های چابک کمک قابل توجهی کند. به عبارت دیگر ، ما معتقدیم که با درک مسئله زودتر در مرحله تصویب ، می توان به تیم پروژه اجازه داد استراتژی هایی را اجرا کند که می تواند مسیر نتایج پروژه آینده را محوری کند. [۱]

۲-۳-۲ خلاصه

در این فصل به معرفی چارچوب چابک و انواع آن پرداختیم. با معرفی مدل سنتی آبشار مراحل آن را بررسی کردیم و گفتیم که چرا منسوخ شده و با مقایسه ی این مدل با مدل چابک به مزایای مدل چابک پی بردیم. همچنین با مقایسه ی روش های اسکرام و کانبان و چابک و آبشار به تفاوت هایشان پی بردیم. سپس به معرفی کامل اسکرام با جرییات شامل تعریف تیم اسکرام، اعضای تیم و وظایف هر یک، اسپرینت و نهایتاً پذیرش پرداختیم.

سپس با معرفی معیار هایی برای تعیین نتیجه ی پروژه به بررسی موفقیت پروژه پرداختیم.

فصل سوم

یادگیری

ماشین

۳-۱ یادگیری ماشین با روشهای چابک

از منظر مهندسی، یادگیری ماشین شامل توسعه نرم افزاری است که اصول علمی را پیاده سازی میکند. این فرآیند پیچیده را می توان در سه مرحله ساده کرد. اولین قدم، تدوین یک فرضیه در مورد یک پدیده است که شامل انتخاب مدل نیز می شود. دومین قدم، برای آزمایش فرضیه و اعتبارسنجی مدل داده ها را جمع آوری میشوند. سرانجام در گام سوم، فرضیه افزایش مداوم مدل با تکرار اصلاح می شود.

هر دو روش توسعه نرم افزار چابک و یادگیری ماشین شامل یک رویکرد تکراری برای ارائه راه حل برای چالش های پیچیده است. در واقع، مطالعات گذشته با موفقیت از (ی.م) در زمینه توسعه نرم افزار چابک استفاده کرده است. این مطالعه توانست یک مدل (ی.م) تولید کند که می تواند با استفاده از شبکه های عصبی مصنوعی برای طبقه بندی یا پیش پردازش داده ها برای خوشه بندی، با استفاده از داده های دارای برچسب دستی، به دقت پیش بینی ۸۸.۹ درصد دست یابد.[۱]

یکی دیگر از حوزه های تحقیقاتی در توسعه نرم افزار چابک که در آن مدل های (ی.م) اغلب استفاده می شود، تخمین تلاش است.

برآورد تلاش برای توسعه نرم افزار فرآیند برآورد تلاش مورد نیاز تیم توسعه دهنده نرم افزار در محیط چابک برای توسعه و نگهداری نرم افزار است. در مطالعات انجام شده از الگوریتم های (ی.م) برای پیش بینی تلاش استفاده شده است.

Satapathy و Rath از الگوریتم های (ی.م) مانند جنگل تصادفی (Random Forest) یا ((RF))، درخت تصمیم (decision tree) یا (DT) و تقویت گرادیان تصادفی (SGB) استفاده کردند تا رویکرد دستی و خسته کننده اشاره به تخمین تلاش را بهبود بخشند. نتایج نشان می دهد که DT، RF و SGB با رویکرد نقطه داستان بهبود یافته اند، با این حال، SGB بهتر از دو الگوریتم (ی.م) دیگر است.

پژوهشگران همچنین یک روش تخمین خودکار به نام "برآورد خودکار" ایجاد کردند. ساخت مدل (ی.م) مطالعه از الگوریتم های یادگیری نظارت شده استفاده می کند. این مدل (Planning Poker) یا (PP) برای برآورد تلاش استفاده شده است. PP تمام ذینفعان اصلی تیم برنامه ریزی چابک را برای برآورد تلاش لازم برای انجام یک کار درگیر می کند، که معمولاً از کارت های بازی با تخمین با استفاده از سری فیبوناچی استفاده می کند.

نتایج این مطالعه نشان می دهد که الگوریتم های یادگیری ماشین تصمیم گیری درخت J48 و الگوریتم های مدل تدارکاتی درخت LMT از PP بهتر هستند. نتایج همچنین حاکی از ترکیب PP با J48 یا LMT باعث کاهش هزینه های کل می شود که در آینده می تواند برآورد تلاش انسانی را افزایش دهد. مطالعات دیگری که روش های چابک و یادگیری ماشین را ترکیب کرده است که با موفقیت روش های چابک را در تجزیه و تحلیل داده های بزرگ گنجانیده است و با استفاده از روش یادگیری ماشین تصمیم گیری درخت J48 مدلی را برای پیش بینی نقص کد نرم افزار هنگام آزمایش خودکار

ساخته است. با دقت ۸۵ درصد، زمان مورد نیاز برای تشخیص این مشکلات بالقوه را به شدت کاهش می دهد. علاوه بر این، شلایر-اسمیت روشهای چابک را در توسعه سیستم توصیه های زمان واقعی علم داده برای مزایایی مانند چرخه های توسعه سریعتر، مکانیسم های بازخورد سریع و بهبود کار تیمی گنجانده است. قبل از اینکه بتوان مدل های پیش بینی یادگیری ماشین را ساخت و ارزیابی کرد، چند مرحله پیش پردازش باید انجام شود. اولین مرحله پیش پردازش استخراج و سنتز چالش های پذیرش اسکرام و چابک است. این مقاله در مقاله ای تحت عنوان "مدل تشخیص چالش های پذیرش اسکرام: SACDM"¹ منتشر شد که در آن یک مدل مفهومی برای آزمایش و ارزیابی تهیه شده است. چالش های پذیرش اسکرام یک مرور داستانی در مورد چالش های پذیرش چابک و اسکرام موجود در سطح جهانی و توسط پزشکاندر آفریقای جنوبی انجام شده است. چالش های سنتز شده به عنوان متغیرهای مستقل مدل استفاده شد. اولین تکرار چارچوب مفهومی² (CF) معروف به SACDM ۱۹ عامل مستقل ایجاد کرده است که برای ارزیابی پذیرش اسکرام به عنوان عامل وابسته استفاده می شود. برخی از عوامل مستقل شامل:

ساختار سازمانی، فرهنگ سازمانی، کار گروهی، تجربه، ارتباطات، همکاری، پیچیدگی، سازگاری، و مزیت نسبی چارچوب اسکرام.

این CF یک مدل سفارشی اقتباس شده از تئوری انتشار نوآوری³ (DOI) است. شرح هر یک از متغیرهای مستقل و وابسته را می توان از مقاله دسترسی آزاد "مدل تشخیص چالش های پذیرش اسکرام: SACDM"⁴ بدست آورد.

برای امکان شناسایی عواملی که به طور قابل توجهی در پذیرش چارچوب اسکرام نقش دارند، نیاز به آزمایش و ارزیابی⁵ ارائه شد که روند سه تکرار CF را که منجر به عوامل مهم می شوند توصیف میکند. در طی تکرار دوم SACDM، CF به چارچوب مفهومی پذیرش چالش های اسکرام (SACCF) تغییر نام یافت. پرسشنامه نظرسنجی آنلاین که به عنوان مقیاسی از نوع لیکرت Likert-type استفاده می شود، داده های پاسخ از ۷۸ مورد پرسشنامه را جمع آوری کرد. از مقیاس نوع لیکرت برای ثبت نتایج در گذشته از پذیرش اسکرام در داخل سازمان، تیم و به صورت جداگانه استفاده شد.

تیم متشکل از تمرین کنندگان اسکرام در سازمانهای آفریقای جنوبی بودند. طرح تحقیق به صورت پرسشنامه بررسی روایی و پیمایشی صورت گرفت. برای تجزیه و تحلیل تحقیق، مجموعه ای از ۲۰۷ پاسخ معتبر به این نظرسنجی برای انجام تجزیه و تحلیل فاکتورهای اکتشافی (EFA) Exploratory و تحلیل آلفای کرونباخ (Cronbach's alpha) استفاده شد که روایی و پایایی

¹ "Scrum adoption challenges detection model: SACDM"

² Conceptual Framework

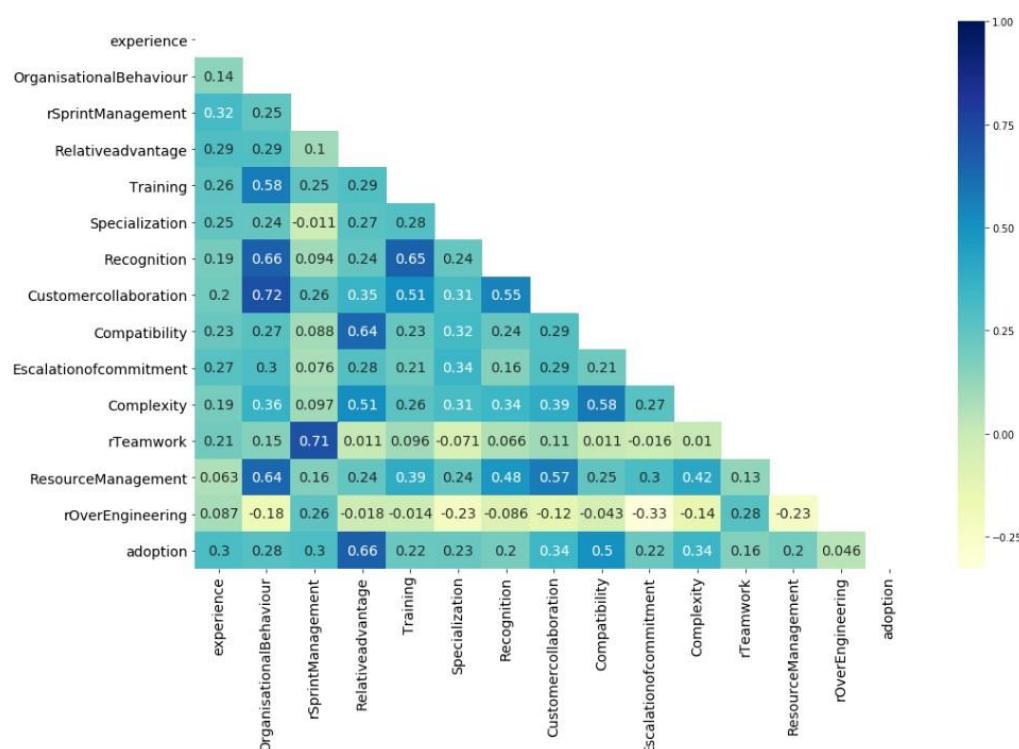
³ Diffusion of Innovation

⁴ "Scrum adoption challenges detection model: SACDM"

⁵ "Factors that contribute significantly to Scrum adoption"

پرسشنامه رابه عنوان ابزار اندازه گیری تأیید کرد. EFA همچنین نشان داد که عوامل را می توان از ۱۹ به ۱۴ متغیرمستقل کاهش داد.

شکل زیر ۱۴ فاکتور با استفاده از اسکران را به عنوان متغیر وابسته نشان می دهد. نتایج حاصل از آمار رگرسیون خطی همبستگی و چندگانه (MLR)multiple linear regression برای شناسایی عواملی که رابطه خطی معنی داری با پذیرش اسکران دارند استفاده شد. فاکتورهای قابل توجه : مدیریت اسپرینت اسکران و پیچیدگی و مزیت نسبی چارچوب اسکران بودند.



شکل ۳- ۱: رابطه بین ویژگی ها و پذیرش اسکران [۱]

با استفاده از تجزیه و تحلیل در مورد استفاده از اسکران ، چهار فرضیه از نظر آماری برای اسکران قابل توجه نشان داده شد. این فرضیه ها به شرح زیر است:

۱. مدیریت اسپرینت: بین مدیریت اسپرینت و پذیرش اسپرینت رابطه خطی (همبستگی مثبت) معناداری وجود دارد.

۲. مقاومت در برابر تغییر: بین مقاومت در برابر تغییر و پذیرش اسکرام رابطه خطی (همبستگی منفی) معنی داری وجود دارد.

۳. مزیت نسبی: بین مزیت نسبی و پذیرش اسکرام رابطه خطی (همبستگی مثبت) معناداری وجود دارد.

۴. پیچیدگی: بین پیچیدگی و پذیرش اسکرام رابطه خطی (همبستگی منفی) معناداری وجود دارد.

سه مقاله فقط شرح داده شده به ما اجازه می دهد تا در مرحله اول ، یک مدل مفهومی بسازیم و قابلیت اطمینان و اعتبار مدل را به عنوان یک CF آزمایش کنیم. در مرحله دوم ، نویسندگان بیشتر عوامل مهمی را در پذیرش اسکرام یافتند. این عوامل با استفاده از ضرایب همبستگی و MLR به صورت کمی مورد تجزیه و تحلیل قرار گرفتند. در مرحله سوم ، می توانیم فرضیه های تحقیق را آزمایش کنیم. پروژه ها با استفاده از روش های چابک بنابراین ، این مقاله تحقیقاتی به توسعه توانایی تیم ها و سازمان ها برای پیش بینی پذیرش اسکرام با استفاده از تجزیه و تحلیل پیش بینی می پردازد.

عواملی که در بالا مورد بحث قرار گرفتند ، بخشی از فرایند مهندسی ویژگی را تشکیل می دهند ، که پیش نیاز ساختمان مدل یادگیری ماشین است. برای ساخت و آزمایش مدل های یادگیری ماشین ، داده های نمونه باید بین مجموعه آموزش و مجموعه آزمایش تقسیم شود. هم برای مجموعه آموزش و هم برای مجموعه آزمون ، پذیرش اسکرام (متغیر وابسته) و ویژگی ها (متغیرهای مستقل) به عنوان استدلال به مدل اضافه می شوند. نمونه کد زیر ویژگی ها و پذیرش اسکرام را به `train_test_split` اضافه می کند.

عملکرد کتابخانه یادگیری ماشین یادگیری برای پایتون عواملی که در بالا مورد بحث قرار گرفتند ، بخشی از فرایند مهندسی ویژگی را تشکیل می دهند ، که پیش نیاز ساختمان مدل یادگیری ماشین است. برای ساخت و آزمایش مدل های یادگیری ماشین ، داده های نمونه باید بین مجموعه آموزش و مجموعه آزمایش تقسیم شود. هم برای مجموعه آموزش و هم برای مجموعه آزمون ، پذیرش اسکرام (متغیر وابسته) و ویژگی ها (متغیرهای مستقل) به عنوان استدلال به مدل اضافه می شوند. نمونه کد زیر ویژگی ها و استفاده از اسکرام را به عملکرد `train_test_split` کتابخانه یادگیری ماشین یادگیری `learnscikit` - برای پایتون اضافه می کند . درست مانند خط زیر: [۱]

```
X_train, X_test, y_train, y_test =
train_test_split ( features, adoption,
test_size=0.3, random_state=4)
```

در نمونه کد بالا ، حالت تصادفی برای آزمایش و تکرارپذیری تنظیم شده است `random_state=4`. این مجموعه داده به یک مجموعه آموزش ۵۷.۶۹ درصد و یک مجموعه

آزمون ۴۳. ۳۰ درصد تقسیم شده است. قبل از آموزش مدل ها عادی بودن داده ها و برآورده شدن همه فرضیات مهم بود. این فرضیات فرض عادی بودن باقیمانده ها ، فرض عدم همبستگی باقیمانده ها ، فرضیات خطی و همسان سازی و فرض عدم چند خطی بودن است [۱]

معیار اطلاعات بیزی^۱ معیار انتخاب مدل برای لیست محدود مدل هاست . ما از آن استفاده کردیم زیرا معیار پرکاربرد و مشهوری برای انتخاب مدل در رگرسیون خطی است. هرچه مقدار BIC کمتر باشد مدل بهتر است. [۱]

$$BIC = \ln(n) \cdot k - 2 \ln(L)$$

1. k = the data points.
2. n = the number of parameters estimated by the model.
3. L = the maximum value of the likelihood function of the model.

همانطور گفته شد k برابر نقاط داده n برابر تعداد پارامترهای تخمین زده شده توسط مدل. L معادل حداکثر مقدار تابع احتمال مدل. [۱] هستند. برای این مقاله ، از سه مدل برای آزمایش صحت پیش بینی استفاده کردند.

۱. مدل پذیرش لگاریتمی (log) تبدیل شده^۲

۲. مدل کامل پذیرش مجموعه ویژگی ها^۸

۳. لگاریتمی تبدیل شده با مدل پذیرش ویژگی های حذف شده^۹

هر یک از این سه مدل با استفاده از روش تجزیه و تحلیل آماری یادگیری ماشین MLR با استفاده از ۱۴ متغیر توضیحی برای پیش بینی تصویب اسکرام همانطور که قبلاً ذکر شد ، استفاده می کنند.

^۱ Bayesian Information Criterion (BIC)

^۲ transformed logarithmic (log) adoption model

feature set adoption model^۸ transformed log with

omitted features adoption model^۹

همانطور که در مقدمه نیز اشاره شد، مدل مجموعه کامل ویژگی ها شامل چهارده ویژگی:

۱. مهندسی بیش از حد (over - engineering)
۲. مزیت نسبی (relative advantage)
۳. شناخت (recognition)
۴. تجربه (experience)
۵. کار تیمی (teamwork)
۶. تخصص (specialization)
۷. تشدید تعهد (escalation of commitment)
۸. سازگاری (compatibility)
۹. مدیریت منابع (resource management)
۱۰. همکاری مشتری (customer collaboration)
۱۱. پیچیدگی (complexity)
۱۲. آموزش (training)
۱۳. مدیریت sprint (سرعت) (sprint management)
۱۴. رفتار سازمانی (organizational behaviour)

برای پیش بینی اسکرام است و شامل مجموعه ویژگی های کامل برای پیش بینی تصویب اسکرام است. مشابه مدل لگاریتمی تبدیل شده ، مدل ویژگی های حذف شده نیز مورد بررسی قرار میگیرند ، با این تفاوت که سه ویژگی از چهارده ویژگی یعنی ویژگی های ”تجربه ، تشخیص و سازگاری” از مجموعه ویژگی های پیش بینی پذیرش اسکرام حذف شده است [۱]

۳-۱-۱ خلاصه

در این فصل به معرفی یادگیری ماشین پرداختیم و سپس با بررسی ترکیب روش چابک با یادگیری ماشین ارتباط آن ها را بررسی کردیم و سپس به معرفی انواع مدل های یادگیری ماشین پرداختیم و آنها را با جزییات معرفی کرده و مقایسه کردیم تا بهترین مدل را بدست آوریم.

با معرفی سه مقاله به بررسی چالش های پذیرش اسکرام پرداخته و موارد تحقیق را در این مقاله ها شرح دادیم و با توجه به آن ها به ویژگی های پذیرش اسکرام دست یافتیم و حتی توانستیم با حذف سه ویژگی به ۱۴ ویژگی برسیم. بعد با معادله بیزی و مقالات به سه مدل از یادگیری ماشین برای پذیرش اسکرام پرداختیم و آن ها را بررسی کردیم.

فصل چهارم

مدل های یادگیری

ماشین

برای یادآوری خواننده ، پیش بینی پذیرش اسکرام که به عنوان پذیرش در حال انجام است ، تمرکز این مقاله است.

۴-۱ بررسی مدل ها

با توجه به سه مدل ذکر شده در فصل قبلی یعنی [۱]:

۱. مدل پذیرش لگاریتمی (\log) تبدیل شده

۲. مدل کامل پذیرش مجموعه ویژگی ها

۳. لگاریتمی تبدیل شده با مدل پذیرش ویژگی های حذف شده

و همچنین شکل ۳-۱ که همبستگی مجموعه ویژگی ها را نشان می دهد. هرچه همبستگی قوی تر باشد ، رنگ نمایش داده شده تیره تر می شود. سوالات عبارات منفی درمورد ویژگی های مدیریت اسپرینت ها ، کار تیمی و مهندسی بیش از حد رمزگذاری شدند (مشخص شده توسط پیشوند r). برخی از روابط بین ویژگی ها و اهمیت آنها در زیر بحث شده است.

۱. رابطه مثبت و معنی داری بین مزیت نسبی و پذیرش ($r=0.66, p<0.001$)

همبستگی از نظر قدرت از متوسط تا قوی بود .

۲. رابطه مثبت و معناداری بین شناخت و رفتار سازمانی ($r=0.66, p<0.001$) .

همبستگی از نظر قدرت از متوسط تا قوی بود.

۳. رابطه مثبت و معناداری بین مزیت نسبی و سازگاری ($r=0.64, p<0.001$)

همبستگی از نظر قدرت متوسط بود.

۴. رابطه مثبت و معناداری بین همکاری و آموزش مشتری ($r=0.51, p<0.001$) .

همبستگی از نظر قدرت متوسط بود.

۵. رابطه مثبت و معناداری بین مدیریت منابع و رفتار سازمانی ($r=0.64, p<0.001$)

همبستگی از نظر قدرت متوسط بود.

۶. رابطه مثبت و معناداری بین کار تیمی و مدیریت اسپرینت ($r=0.71, p<0.001$) از

نظر قدرت همبستگی قوی بود.

۷. رابطه مثبت و معناداری بین پیچیدگی و مزیت نسبی ($r=0.51, p<0.001$) همبستگی از نظر قدرت متوسط بود.

۸. رابطه مثبت و معناداری بین مدیریت منابع و آموزش ($r=0.39, p<0.001$) همبستگی از نظر قدرت از ضعیف تا متوسط بود.

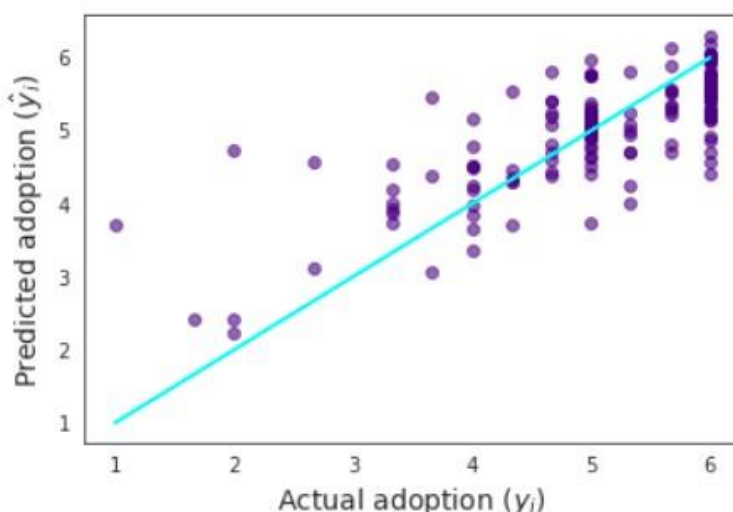
۹. رابطه مثبت و معناداری بین مدیریت منابع و شناخت ($r=0.48, p<0.001$) همبستگی از نظر قدرت متوسط بود.

۱۰. رابطه مثبت و معناداری بین سازگاری و پذیرش ($r=0.50, p<0.001$) همبستگی از نظر قدرت متوسط بود.

۴-۱-۱ مدل اول: مجموعه ویژگی های کامل

مدل اول مدل مجموعه ویژگی های کامل است که دارای همبستگی واقعی و پیش بینی شده ۰.۷۵ است.

- شکل ۴-۱ پذیرش واقعی و پیش بینی شده را نشان می دهد :



شکل ۴-۱: رابطه بین پذیرش واقعی و پیش بینی شده برای مجموعه ویژگی های کامل

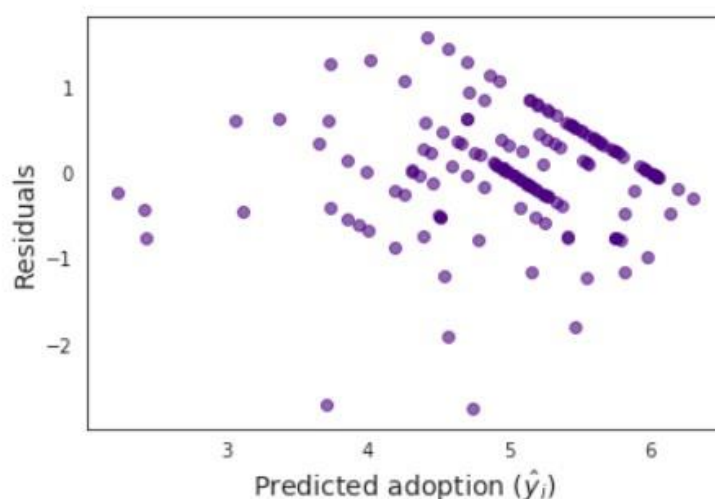
- شکل ۴-۲ باقی مانده ها را نشان می دهد:

فاصله پیش بینی ۹۵ درصد برای مدل بالا و پایین در مدل کامل پذیرش ویژگی ، به ترتیب ۰.۸۳ و ۰.۹۸ است [۱].

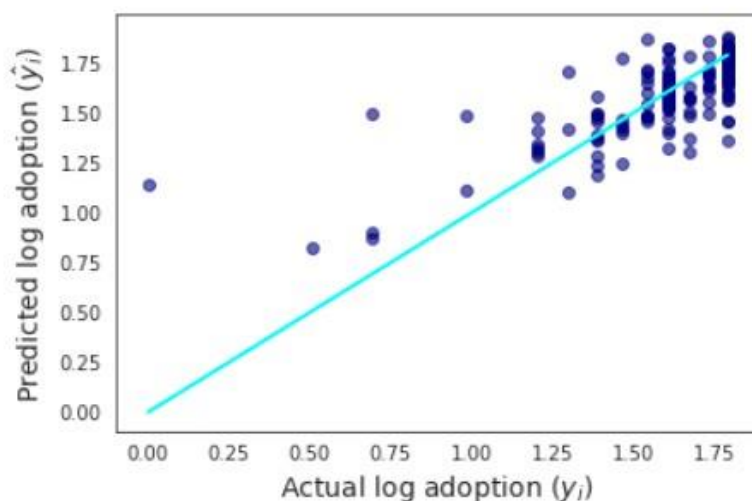
۴-۱-۲ مدل دوم: مدل پذیرش لگاریتمی تبدیل شده

برای مدل دوم، شکل ۴-۳ همبستگی واقعی و پیش بینی شده برای تغییر شکل لگاریتمی را نشان می‌دهد:

شکل ۴-۴ مقادیر باقیمانده و پیش بینی شده را نشان می‌دهد:



شکل ۴-۲: باقی مانده ها ای مجموعه ویژگی های کامل [۱]



شکل ۴-۳: همبستگی واقعی و پیش بینی شده برای تغییر شکل لگاریتمی [۱]

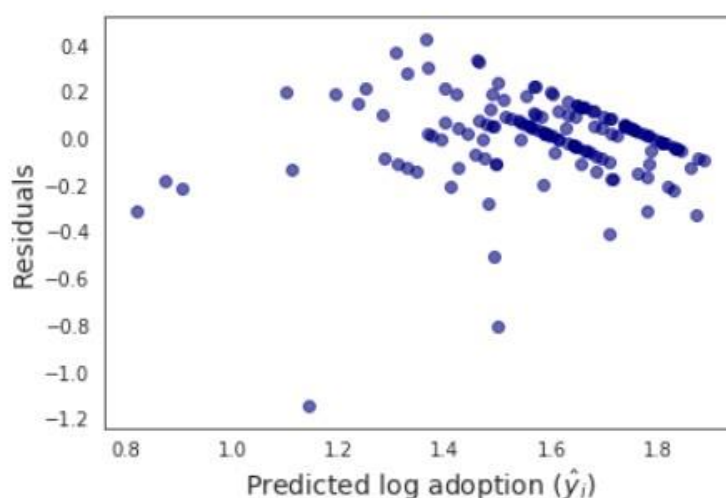
همبستگی واقعی و پیش بینی شده ورود به سیستم ۰.۷۳ است. فاصله پیش بینی ۹۵ درصد برای مدل بالا و پایین در مدل پذیرش ورود به ترتیب ۳.۸۰ و ۳.۰۱ است [۱].

۴-۱-۳ مدل سوم: پذیرش لگاریتمی تبدیل شده با ویژگی های حذف شده
مدل سوم با استفاده از لگاریتم و همانطور که قبلاً ذکر شد، با حذف سه ویژگی یعنی تجربه، شناخت و سازگاری ساده می شود.

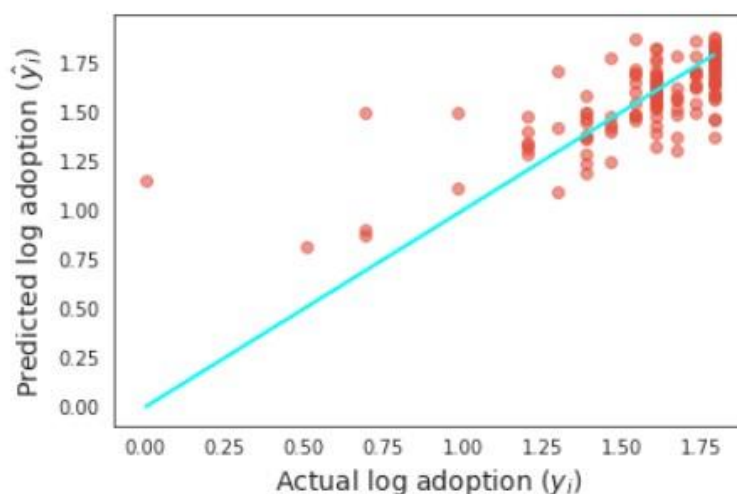
شکل ۴-۵ مدل تبدیل شده و ساده را نشان می دهد:

باقیمانده های این مدل در شکل ۴-۶ نشان داده شده است:

پذیرش واقعی و پیش بینی شده لگاریتمی با همبستگی ویژگی های حذف شده ۰.۷۳ است. فاصله



شکل ۴-۴: باقی مانده ها برای پذیرش لگاریتمی تبدیل شده [۱]



شکل ۴-۵: رابطه بین ویژگی ها و پذیرش اسکرام با ویژگی های حذف شده [۱]

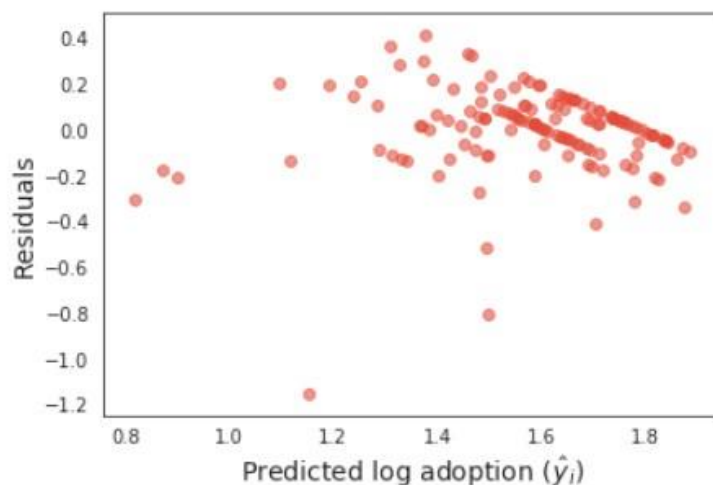
پیش بینی ۹۵ درصد برای حد بالا و پایین به ترتیب ۳.۷۹ و ۳.۰۱ در تصویب گزارش با مدل ویژگی های حذف شده است.

شکل ۴-۷ مربع R و خطای MSE را برای هر یک از سه مدل پیش بینی یادگیری ماشین نشان می‌دهد.

مربع R اندازه گیری آماری واریانس مقادیر پیش بینی شده تقسیم بر واریانس داده است . [۱]

$$R^2 = 1 - \frac{SS_{regression}}{SS_{total}} \quad (4-1)$$

۰ درصد مربع R نشان می دهد که مدل ها هیچ یک از تغییرپذیری داده ها را توضیح نمی دهند، به عبارت دیگر ، این از پیش بینی میانگین بدتر است. مقدار ۱۰۰ درصد مربع R نشان می دهد که مدل تمام تغییرپذیری داده ها را توضیح می دهد.



شکل ۴-۶: رابطه بین ویژگی ها و پذیرش اسکرام ۱ ویژگی های حذف شده [۱]

جدول ۴-۷: مقدار واریانس و خطا طبق فرمول های گفته شده [۱]

Adoption Model	R-squared	R-squared %	MSE
Transformed log model	0.527	52.7	0.039
Full feature set model	0.564	56.4	0.507
Transformed log with omitted features model	0.527	52.7	0.038

معیار MSE با در نظر گرفتن مقدار پیش بینی شده مشاهده ، خط رگرسیون را تا چه اندازه به نقاط داده نزدیک می کند محاسبه می کند و خودسری مربوط به مقدار باقیمانده مربع ها را از بین می برد .

به بیان دیگر ، معادله MSE میانگین خطای مربع پیش بینی های ما را اندازه گیری می کند که در آن خروجی واقعی است و پیش بینی مدل است . هرچه مقدار MSE کمتر باشد ، واریانس خطا کمتر است.

[۱]

$$MSE = \frac{1}{n} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (۴-۲)$$

پیش پردازش و مهندسی ویژگی داده های پاسخ به ما امکان ساخت و ارزیابی سه مدل یادگیری ماشین را می دهد. این سه مدل مجموعه جامعی از مدل های پیش بینی نبودند زیرا این رویکرد فراتر از محدوده این مقاله تحقیق بود. ما می خواستیم بررسی کنیم که آیا مدل های مختلفی که شامل تغییرشکل و مجموعه ویژگی های ساده هستند می توانند پذیرش اسکرام را با واریانس و خطا کمتر ، به عبارت دیگر ، در چه صحت پیش بینی پیش بینی کنند.

سه مدل ارزیابی شده در این مطالعه مدل پذیرش لگاریتمی ، مدل پذیرش مجموعه کامل ویژگی ها و گزارش با مدل پذیرش ویژگی های حذف شده بود. همانطور که در نتایج ذکر شد ، مقدار مربع R میزان نزدیک بودن داده ها به خط رگرسیون را اندازه گیری می کند و میانگین مربعات خطا MSE میانگین مربع خطاها را اندازه گیری می کند.

مدل پذیرش مجموعه کامل ویژگی دارای مقدار واریانس متوسط ۰.۵۶۴ است که بیش از نیمی از نمونه های مدل را توضیح می دهد. هرچه مربع R به ۱ نزدیکتر باشد ، دقت پیش بینی بیشتر است. این مدل همچنین دارای مقدار MSE ۰.۵۰۷ است که نشانگر میزان خطای بالایی است زیرا مقدار خطا به یک نزدیکتر است.

مدل پذیرش ورود به سیستم تحول در مدل مجموعه ویژگی های کامل است. این مدل دارای مقدارمربع R ۰.۵۲۷ با مقدار MSE ۰.۰۳۹ است.

بلافاصله مشخص می شود که مدل لگاریتمی مدل بهتری است دقت پیش بینی پذیرش برای به دلیل اینکه MSE به صفر نزدیکتر است در حالیکه مقدار مربع R بیشتر از ۰.۵ و کمتر از ۰.۶ است، شبیه به مدل مجموعه کامل ویژگی ها.

مدل سوم ، گزارش با مدل تصویب ویژگی های حذف شده است. این مدل با حذف سه ویژگی از چهارده ویژگی مجموعه را ساده کرد. سه ویژگی عبارتند از تجربه ($p\text{-value}=0.929$) ، تشخیص ($p\text{-value}=0.969$) و سازگاری ($p\text{-value}=0.820$) به دلیل مقادیر بالای p حذف شدند. هرچه

مقادیر p بیشتر باشد، اهمیت یک عامل برای پذیرش کمتر است. با حذف سه ویژگی، مقدار مربع R برابر با ۰.۵۲۷ و MSE برابر با ۰.۳۸۰ است. بنابراین لگاریتم تبدیل شده با مدل ویژگی های حذف شده بهترین مدل پیش بینی است حتی اگر سطح خطای حاشیه ای کمتری نسبت به مدل لگاریتمی داشته باشد. ما کاملاً مطمئن هستیم که میتوانیم با یک نمونه تصادفی بیشتر، دقت پیش بینی را بهبود ببخشیم. علاوه بر این، ما می توانیم با ایجاد مدلی با واریانس کمتر و MSE ، بهترین مدل متناسب را بهبود بخشیم [۱۰].

۴-۱-۴ خلاصه

در این فصل به صوت دقیق تر و جزئی تر به معرفی مدل ها پرداختیم و با توجه به ویژگی ها ارتباطات یببین این مدل ها به وجود آوردیم و با توجه به اعداد بدست آمده روابط را طبقه بندی کردیم. شکل هرمدل با توجه به اعداد در این بخش قابل مشاهده است. با معرفی دو رابطه ی ریاضی که مربوط به محاسبات آماری است مدلی که کمترین خطا را داشت بدست آوردیم.

فصل پنجم

جمع بندی و نتیجه گیری و

پیشنهادهات

این مقاله تحقیقی در مورد توسعه مدل‌های یادگیری ماشین برای پیش بینی صحت پذیرش اسکرام براساس یک مجموعه ویژگی مشتق شده از داده های پاسخ پرسشنامه نظرسنجی گزارش می دهد. از حجم نمونه داده های پاسخ ۲۰۷ برای آموزش و آزمایش مدل های پیش بینی استفاده شد. قبل از آموزش و آزمایش مدل ها ، تمیز کردن و پیش پردازش داده ها مورد نیاز بود. ما هر کدام از این سه مدل را با تقریباً ۷۰ درصد از مجموعه داده آموزش دادیم در حالی که ۳۰ درصد برای آزمایش مدل ها استفاده شد. سه مدل پیش بینی یک مدل کامل پذیرش مجموعه ویژگی ها ، مدل پذیرش لگاریتمی تبدیل شده و یک گزارش ثبت شده با مدل پذیرش ویژگی های حذف شده بود. مدل پذیرش با بالاترین دقت پیشبینی ، گزارش تبدیل شده با مدل ویژگی های حذف شده با مربع $R = 0.527$ و $(MSE) = 0.38$ بود. هنگام جستجوی ترکیبی از $(R) = 0.564$ و $(MSE) = 0.507$ ، مدل کامل ویژگی کمترین دقت را داشت.

پیامدهای این یافته ها ، اگرچه هنوز مقدماتی است ، اما به محققان و پزشکان این امکان را می دهد که درک بهتری از ویژگی های پیش بینی پذیرش اسکرام داشته باشند. محدودیت های این تحقیق سه برابر است. اولاً ، آموزش و تقسیم آزمایشی استفاده شده در این مقاله تحقیقاتی برای مجموعه داده قبلاً به عنوان نمایش تعصبات گزارش شده است . ثانياً ، معیار ارزیابی مدل مربع R که برای برازش مدل ها استفاده می شود ، یکی از بسیاری از معیارهای مورد استفاده برای ارزیابی مدل پیش بینی است. ثالثاً ، در این تحقیق مقدماتی از تکنیک های اعتبارسنجی مدل اضافی مانند نمونه بوت استرپ استفاده نشده است.

معیارهایی مانند حداکثر خطا ، میانگین خطای مطلق ، می توان از خطای مربع مربعی برای ارزیابی بیشتر دقت پیش بینی استفاده کرد. با استفاده از یک نمونه تصادفی بزرگتر می توان دقت پیش بینی مدل های استفاده شده در این مقاله تحقیق را بهبود بخشید. تحقیقات بیشتر می تواند یک مدل رگرسیون لجستیک با یک مجموعه داده بزرگتر برای پیش بینی نتایج پروژه چابک ، اصلاح چارچوب مفهومی و روش مورد نیاز ایجاد کند.

منابع و مراجع

- [1] R. Hanslo and M. Tanner, *Machine Learning models to predict Agile Methodology adoption*. 2020. doi: [10.15439/2020F214](https://doi.org/10.15439/2020F214).
- [2] A. Shalloway, "How to Adopt Scrum Effectively," p. 8.
- [3] "Agile software development," *Wikipedia*. May 24, 2021. Accessed: May 28, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Agile_software_development&oldid=1024859340.
- [4] "Waterfall model," *Wikipedia*. May 18, 2021. Accessed: May 28, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Waterfall_model&oldid=1023844880.
- [5] "Agile Vs Waterfall: Know the Difference Between Methodologies." <https://www.guru99.com/waterfall-vs-agile.html> (accessed May 28, 2021).
- [6] "Agile vs. Waterfall vs. Kanban vs. Scrum," Oct. 23, 2017. <https://www.lucidchart.com/blog/agile-vs-waterfall-vs-kanban-vs-scrum> (accessed May 28, 2021).
- [7] "Scrum (software development)," *Wikipedia*. May 19, 2021. Accessed: May 28, 2021. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Scrum_\(software_development\)&oldid=1023978900](https://en.wikipedia.org/w/index.php?title=Scrum_(software_development)&oldid=1023978900).