



گزارش پروژه‌ی داده‌کاوی

HR DataSet

الهه رحمتی

95103923

استاد راهنما: دکتر خدمتی

دستیار آموزشی: آقای یزدانی



تابستان 1398



فهرست

3.....	مقدمه
4.....	مرور ادبیات
6.....	تشریح داده‌ها
6.....	توضیح ویژگی‌ها و نمودارهای توزیع مقادیر هر یک از آن‌ها
12.....	پیش پردازش داده‌ها
13.....	Missing values
14.....	Remove duplicates
14.....	Outlier and noisy data detection
15.....	Data transformation
15.....	Data reduction
15.....	شناسایی و استخراج داده‌های مورد نظر جهت داده‌کاوی
15.....	Imbalanced Data و SMOTE
16.....	PCA
16.....	داده‌کاوی و شناسایی الگوهای پنهان در داده
17.....	مدل‌های ناپارامتری
17.....	KNN
17.....	Decision Tree
18.....	Random Forest
18.....	مدل‌های پارامتری
18.....	LDA و QDA
18.....	Logistic Regression
19.....	ارزیابی الگوهای شناسایی شده و تعیین الگوهای مطلوب
19.....	مدل KNN
19.....	مدل Decision Tree
20.....	مدل Random Forest
21.....	مدل LDA
21.....	مدل QDA
22.....	مدل Logistic Regression
22.....	مدل Naive Bayesian



- 22.....ارائه نهایی الگوها و دانش کسب شده
- 23.....نتیجه گیری
- 24.....ضمیمه
- 25.....منابع و مراجع

امروزه با توجه به گسترش روزافزون تولید داده در سطوح سازمان‌ها، بررسی و تحلیل روی این داده‌های حجیم از جمله دغدغه‌های اصلی مدیران و سران سازمان‌ها است. هرچند همچنان تا بهینه‌سازی و استفاده‌ی مناسب از این داده‌ها فاصله زیادی داریم و مدیران ارشد زیادی نیستند که بر این مباحث مسلط باشند و از تحلیل این داده‌ها، در راستای اداره سازمان خود استفاده کنند، اما آنچه در بسیاری از کشورهای دنیا و سازمان‌های مطرح شاهد هستیم این است که این گسترش روزافزون، توجه بسیاری از این مدیران را به خود جلب کرده است و تکنیک‌های داده‌کاوی و تحلیل داده‌ها را با بهره‌گیری از افراد ماهر در این زمینه، در سطوح مختلف سازمانی خود پیاده‌سازی می‌کنند و در تلاش هستند تا با نتایج آنها، مدیریت سازمان خود را جهت‌دار کنند و به نوعی بحث هوش تجاری و داده‌کاوی را در تصمیم‌گیری‌های خود دخیل نمایند.

یکی از این سطوح سازمانی که به تازگی نگاه‌ها را به خود معطوف کرده است و قابلیت به‌کارگیری تکنیک‌های داده‌کاوی در سازمان را دارد بخش منابع انسانی یا همان HR است. منابع انسانی یکی از مهم‌ترین بخش‌های هر سازمان است که نیازمند برنامه‌ریزی فراوان است، اما اگر اندازه سازمان به لحاظ تعداد نیروی انسانی زیاد باشد، کار برنامه‌ریزی و تصمیم‌گیری درباره‌ی هر گروه از افراد بسیار سخت خواهد بود. الگوهای مختلفی که ممکن است در میان کارمندان ما وجود داشته باشد و خبری از آن نداشته باشیم، می‌تواند نقش بسیار حیاتی را برای بقای سازمان ما ایفا کند. تحلیل و استفاده از روش‌های داده‌کاوی برای سازمان‌ها چالش بزرگی است، بیشتر به این خاطر که سازمان‌ها داده‌های خود را به درستی و درواقع قابل استفاده ذخیره نکرده‌اند، یا با هم سازگار نیستند و انواع مشکلات مختلف دیگری که داده‌ها برای استفاده دارند که متأسفانه بیشتر سازمان‌های ایرانی درگیر چنین مشکلاتی هستند و در بسیاری از موارد حتی شاهد آن هستیم که داده‌ای وجود ندارد که بخواهیم کار تحلیل و داده‌کاوی روی آن انجام دهیم.



نکته مثبت داده‌کاوی روی داده‌های منابع انسانی، این است که اصولاً چون شرکت‌ها بر اطلاعات نیروهای خود اشراف دارند و حداقل‌هایی در این زمینه رعایت می‌شود، می‌توانیم به این تکیه کنیم که احتمالاً پایگاه داده‌هایی راجع به منابع انسانی سازمان‌های متوسط و بزرگ وجود دارد و چون نظام پیاده‌سازی آنها مشابه است، احتمالاً نحوه جمع‌آوری و ثبت آنها مشابه یکدیگر صورت گرفته است و می‌توان بدون دردسر زیادی از آنها استفاده کرد. بدین ترتیب داده‌های منابع انسانی یک سازمان، می‌تواند نقطه خوبی برای به کارگیری روش‌های داده‌کاوی و استخراج الگوهای پنهان میان این داده‌ها باشد.

مرور ادبیات

با توجه به این مطلب که تعداد پژوهش‌های صورت گرفته در زمینه تصمیم‌گیری مبتنی بر داده‌کاوی در حوزه‌های مختلف مدیریت منابع انسانی محدود است، لذا در این زمینه، تحقیقاتی که به مرور آنها پرداخته باشند، انگشت‌شمار می‌باشد. با این وجود برخی محققان به مطالعه مروری در این زمینه پرداخته‌اند که در راس همه آنها باید به پژوهش صورت گرفته توسط استرومیر و پیازا در سال 2013 اشاره نمود که یک مطالعه مروری ارزشمند در این زمینه به شمار می‌رود. این پژوهش به صورت سیستماتیک به مرور تحقیقات در زمینه داده‌کاوی منابع انسانی پرداخته تا زمینه‌ساز شکل‌گیری مطالعات آینده گردد. چنین پژوهش‌هایی زمینه‌ساز آغاز پژوهش‌های بعدی در حوزه داده‌کاوی بر منابع انسانی بودند که فارغ از پژوهش ذکر شده در بالا، چندین پژوهش و مقاله داخلی نیز صورت گرفته است که به برخی از آنها اشاره خواهیم کرد :

مقاله چارچوب به کارگیری رویکرد داده‌کاوی در حوزه مدیریت منابع انسانی

این مقاله که توسط نسترن حاجی حیدری، دانشیار دانشکده مدیریت دانشگاه تهران و دو دانشجوی کارشناسی ارشد نوشته شده است، در زمینه مدیریت منابع انسانی به بررسی رویکرد منابع انسانی پرداخته است. در این مقاله بررسی شده است که داده‌های منابع انسانی به شیوه‌های اثربخش مورد تحلیل قرار نمی‌گیرند و با اتکا به روش‌های داده‌کاوی می‌توان به تصمیم‌گیری پیرامون مسائل مختلف پرداخت. هدف این پژوهش بررسی تحلیلی تحقیقاتی بوده است که از تکنیک‌های مختلف داده‌کاوی برای تجزیه و تحلیل مسائل مرتبط با مدیریت منابع انسانی بهره برده‌اند؛ تا در نتیجه بتوان چارچوبی راهبردی برای به کارگیری روش‌های داده‌کاوی در حوزه‌های مدیریت منابع انسانی ارائه نمود. برای این منظور، 89 تحقیق مستقل و ارزشمند از منابع داخلی و خارجی استخراج و مرور شده است. در نتیجه، ابتدا حوزه‌های مختلف مدیریت منابع انسانی که در این تحقیقات مورد توجه داده‌کاوان قرار داشته است، مشخص گردیده که از آن جمله می‌توان به موضوعات استخدام و گزینش، آموزش و توسعه، غیبت و ترک خدمت، مدیریت عملکرد و ... اشاره نمود. سپس با عنایت به متدولوژی CRISP-DM به تشریح مراحل مختلف تصمیم‌گیری مبتنی بر داده‌کاوی در مدیریت منابع انسانی پرداخته‌اند و در نهایت چارچوبی مناسب برای مطالعه در این موضوع به دست آمد که راهنمایی کلان برای مدیران منابع انسانی است تا هوشمندانه‌تر از منابع اطلاعاتی درون سازمانی خود مبتنی بر اهداف استفاده نمایند. برای پژوهشگران نیز چارچوب مذکور تصویری منسجم از مطالعات پیشین را بازنمایی می‌کند که می‌تواند در تحقیقات آتی در عمل بررسی و صحت‌گذاری شود.

این مقاله توسط عطیه مهاجر شجاعی دانشجوی کارشناسی ارشد دانشگاه آزاد قزوین نوشته شده است و با محوریت استفاده از تکنیک های داده کاوی در مدیریت منابع انسانی تنظیم شده است. افزایش تعدادی از نشریات مرتبط با داده کاوی در موضوع مدیریت منابع انسانی، حاکی از حضور موفق تحقیقات جدید در این حوزه بوده و مدیریت منابع انسانی به صورت قابل توجهی، دامنه ی جدیدی از تحقیقات داده کاوی را تشکیل می دهد که تحت سلطه توسط کارهای فناوری گرا، می باشد. در این مقاله، مروری بر روش های داده کاوی در مدیریت منابع انسانی به صورت نظام مند برای کشف پیشرفت های اخیر پرداخته و نشان دهنده ی حوزه هایی برای کارهای آینده می باشد. با این حال، نیازهای حوزه محور خاص، مانند: ارزیابی موفقیت حوزه یا مطابق با استاندارد های قانونی در مقاله ی جاری، در نظر گرفته نشده است.

مهاجر شجاعی، عطیه، ۱۳۹۴، مروری بر روش های داده کاوی در مدیریت منابع انسانی، نخستین کنفرانس بین المللی فناوری اطلاعات، تهران، مرکز همایش های توسعه ایران.

بکارگیری داده کاوی در مدیریت منابع انسانی سازمان های فاوا (فناوری اطلاعات و ارتباطات)

این مقاله توسط سامان سیادتی دانشجوی دکتری مهندسی صنایع گروه فناوری اطلاعات، نوشته شده است و پیرامون استفاده از داده کاوی در فناوری اطلاعات و ارتباطات است. سرعت بالای تغییر و تحول در فناوری ها و زیرساخت های مورد استفاده در پروژه های فناورانه، استفاده از کارآمدترین و به روزترین متدلوژی ها را جهت حفظ و ارتقای کارایی منابع انسانی بکار گرفته شده، امری اجتناب ناپذیر نموده است. در این میان سازمان های فناوری اطلاعات و ارتباطات (فاوا) نسبت به سایر سازمان ها با چالش ها و مسائل ویژه ای مانند: کاهش ارتباطات رودررو، تغییرات سریع محیط کار و همکاران، پراکندگی جغرافیایی در پروژه های غیر متمرکز و مسائلی از این دست مواجه اند که این عوامل می توانند بر روی الگوی مدیریت منابع انسانی این نوع سازمان ها تاثیر گذار باشند. معمولا کاربران پس از طرح فرضیه ای بر اساس گزارشات مشاهده شده به اثبات یا رد آن می پردازند ، در حالی که امروزه با گسترش سیستم های پایگاه داده و حجم بالای اطلاعات ذخیره شده در این سیستم ها به روش هایی نیاز داریم که به اصطلاح به کشف دانش بپردازند ، یعنی روش هایی که با کمترین دخالت کاربر و به صورت خودکار الگوها و رابطه های منطقی را بیان نمایند. “ما در این تحقیق با استفاده از الگوریتم های داده کاوی ، به بررسی شرایط و انتخاب بهترین استراتژی در حوزه مدیریت منابع انسانی در یکی از بزرگترین سازمان های فناوری اطلاعات و ارتباطات کشور پرداخته ایم.”

سیادتی، سامان؛ محمدجعفر تارخ؛ مهدی سیدهاشمی و هومن شاه کوهی، ۱۳۹۳، بکارگیری داده کاوی در مدیریت منابع انسانی سازمان های فاوا (فناوری اطلاعات و ارتباطات، کنفرانس بین المللی توسعه و تعالی کسب و کار، تهران، موسسه مدیران ایده پرداز پایتخت ویرا

طراحی مدل انتخاب نیروی انسانی با رویکرد داده کاوی

این مقاله توسط آذر عادل و گروه مدیریت دانشگاه تربیت مدرس نوشته شده است و درباره طراحی مدل نیروی انسانی با رویکرد داده کاوی طرح شده است. موفقیت یا شکست سازمان، ارتباط مستقیمی با چگونگی جذب و نگهداری منابع انسانی آن دارد. اغلب در رابطه با برگزاری آزمون های ورودی و فرآیند جذب کارکنان، داده ها و اطلاعات فراوانی در سازمان ها وجود دارد که بدون استفاده می مانند. داده کاوی، به عنوان راه حل برای چنین مسائلی است. در این پژوهش که از حیث هدف، کاربردی و از

جنبه ماهیت از نوع پژوهش‌های همبستگی و همخوانی محسوب می‌شود، سعی شده است که با استفاده از تکنیک‌های داده‌کاوی، قواعد و روابط بین نمرات آزمون‌های ورودی و سایر متغیرهای شخصی و شغلی (که قبل از ورود هر کس به سازمان مشخص می‌شود) و وضعیت کارکنان با عملکرد شغلی و وضعیت ارتقا آنان شناسایی شود. در نتیجه با مطالعه و بررسی پایگاه‌های داده آزمون و منابع انسانی یک بانک تجاری برای 2 سال متوالی (1383 و 1384)، شاخص‌های نیروی انسانی که بر عملکرد یا ارتقا موثر بودند، شناسایی شدند. تکنیک داده‌کاوی مورد استفاده در این پژوهش، درخت تصمیم‌گیری است و استخراج قواعد نیز با استفاده از الگوریتم‌های QUEST، CHAID، C5.0 و CART انجام شده است. در نهایت ضمن ارائه مدلی جهت انتخاب متغیرهای تاثیرگذار، متغیر هدف و الگوریتم‌های مناسب؛ از بین قواعد به دست آمده، قواعد غیربدیهی مشخص و علت وجود این قواعد با کمک خبرگان تبیین شده است. از جمله نتایج، حذف متغیر ارزیابی عملکرد به عنوان متغیر هدف در روند این پژوهش است که ناشی از عدم دقت تکمیل فرم‌های ارزیابی عملکرد در فرآیند ارزیابی بانک بوده است. هم‌چنین در این پژوهش مشخص شده است از مجموع 26 متغیر بررسی شده، پنج متغیر: «نمره کل آزمون»، «امتیاز مصاحبه»، «مقطع تحصیلی»، «تجربه حرفه‌ای» و «استان محل خدمت» بر ارتقای داوطلبان تاثیرگذار بوده‌اند. این نتایج منجر به دانشی شده است که امکان کاربردی نمودن آن‌ها وجود خواهد داشت.

تشریح داده‌ها

داده‌های این پروژه، داده‌های بخش منابع انسانی یک شرکت ایرانی است که 10 ویژگی دارد. اطلاعات اولیه داده‌ها به شرح زیر است. در ادامه نمودارهایی را برای درک بهتر پراکندگی و چگونگی توزیع داده‌ها رسم می‌کنیم.

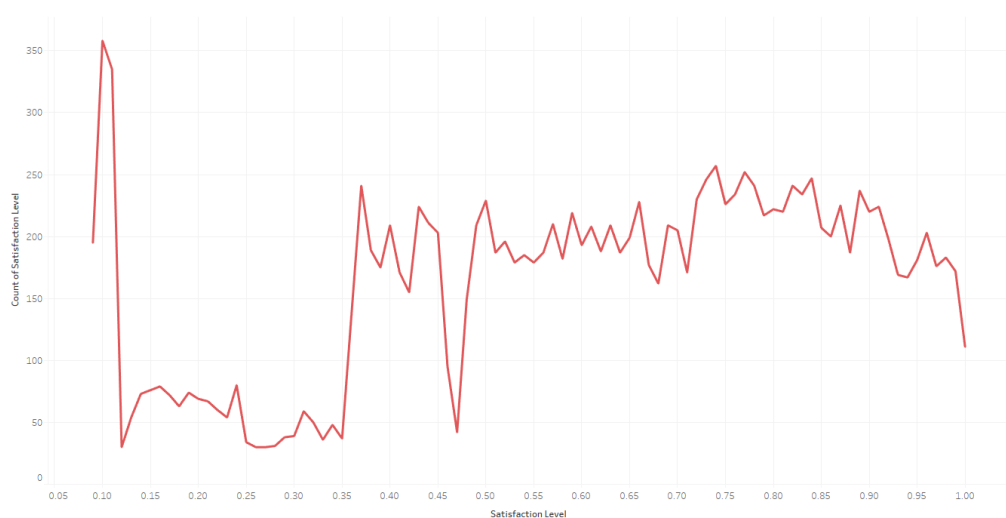
Name	Type	Missing	Min/Least	Max/Most	Average
<i>satisfaction_level</i>	Real	0	0.09	1	0.623
<i>last_evaluation</i>	Real	0	0.36	1	0.718
<i>number project</i>	integer	0	2	7	3.803
<i>average montly_hours</i>	integer	0	96	310	200.648
<i>time_spend_company</i>	integer	0	2	6	3.256
<i>Work accident</i>	Binary	0	0	1	0.15
<i>left</i>	Binary	0	0	1	0.2
<i>promotion_last_5years</i>	Binary	0	0	1	0.01
<i>sales</i>	Categorical	0	Management	Sales	-
<i>salary</i>	Categorical	0	High	Low	-

توضیح ویژگی‌ها و نمودارهای توزیع مقادیر هر یک از آن‌ها

■ satisfaction_level

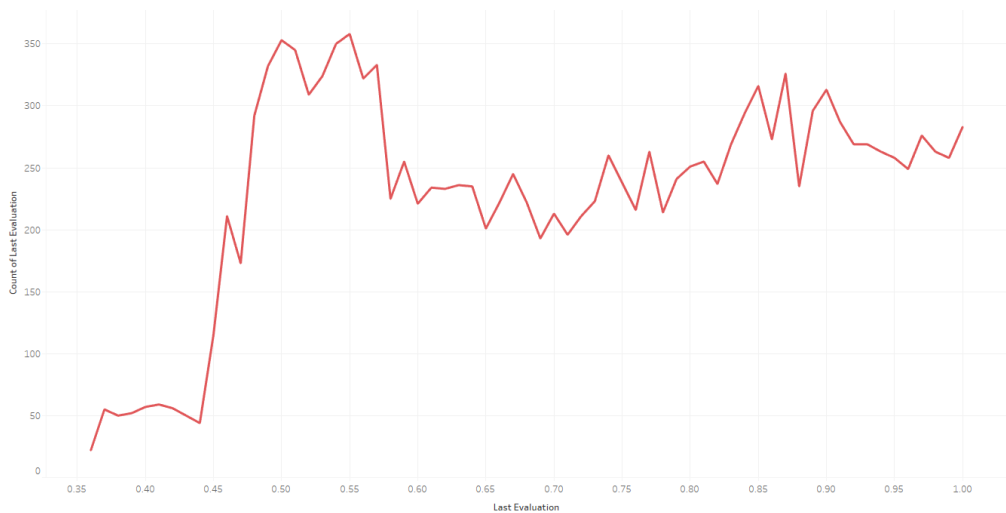


نشان‌دهنده سطح رضایت افراد از کار و وضعیت موجود در فضای کسب‌وکار است.



last_evaluation ■

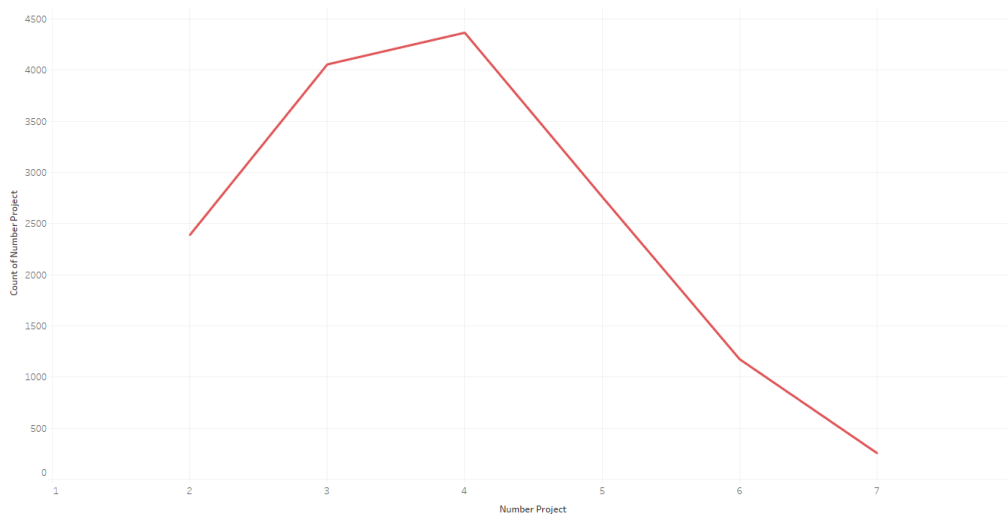
آخرین ارزیابی عملکرد شرکت از افراد را نشان می‌دهد.



number_project ■

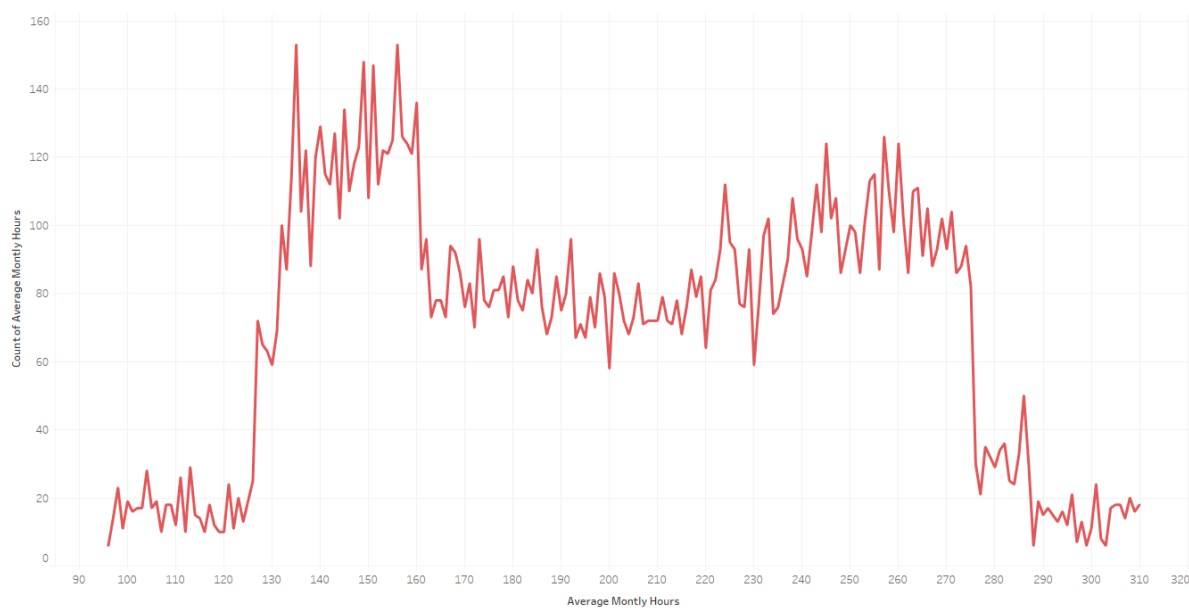


تعداد پروژه‌هایی که هر فرد در طول دوره کاری خود انجام داده‌است را نشان می‌دهد.



average_monthly_hours ■

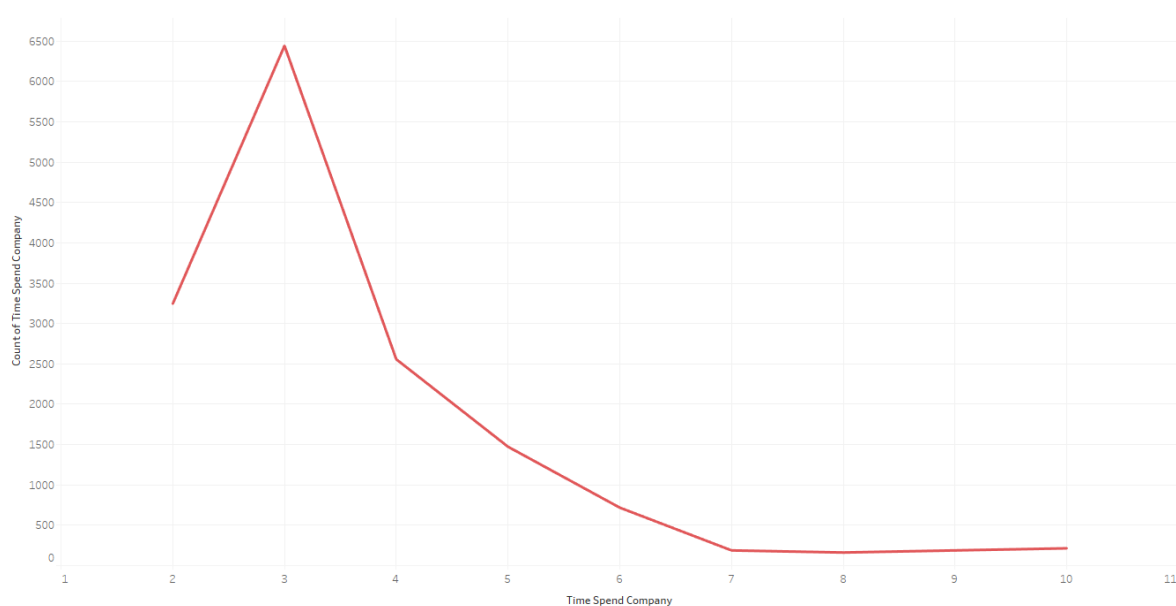
میانگین ساعتهای که فرد در یک ماه کار می‌کند را نشان می‌دهد.



time_spend_company ■

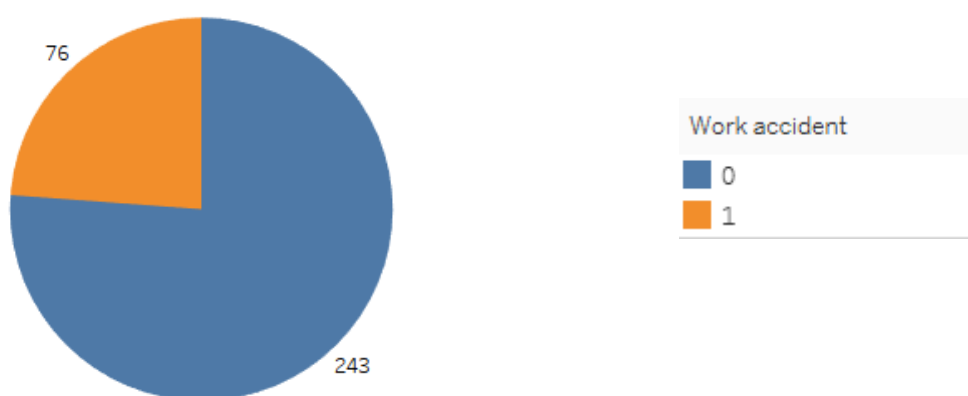


مدت زمانی که فرد در شرکت بوده است را بر حسب سال نشان می دهد.



Work_accident

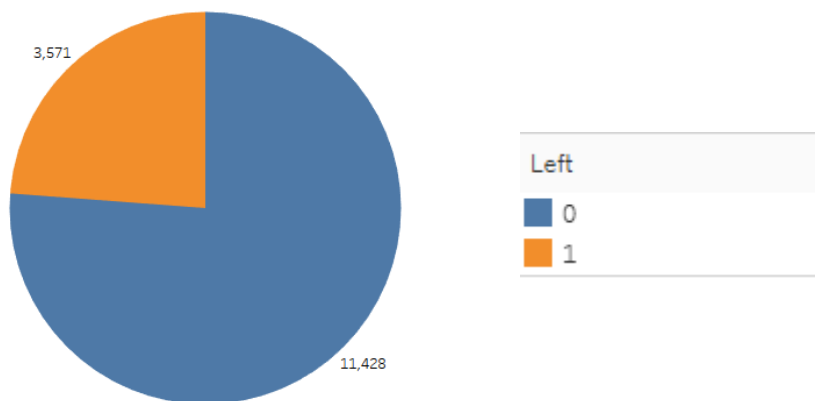
یک ویژگی باینری است که بیان می کند فرد در مدت زمان حضورش در شرکت، دچار حادثه شغلی شده است یا خیر.



Left

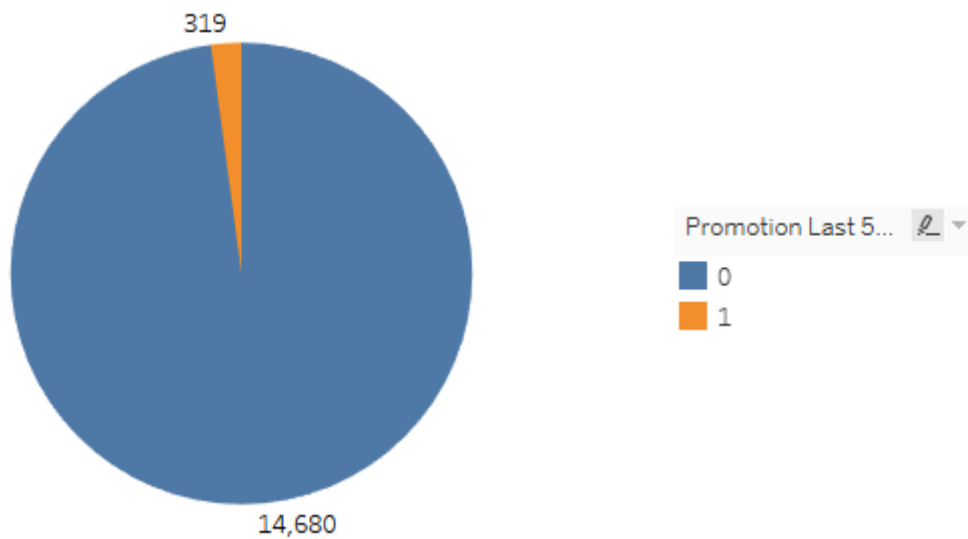


یک ویژگی باینری است که بیان می کند فرد از مجموعه خارج شده است یا خیر.



promotion_last_5years ▪

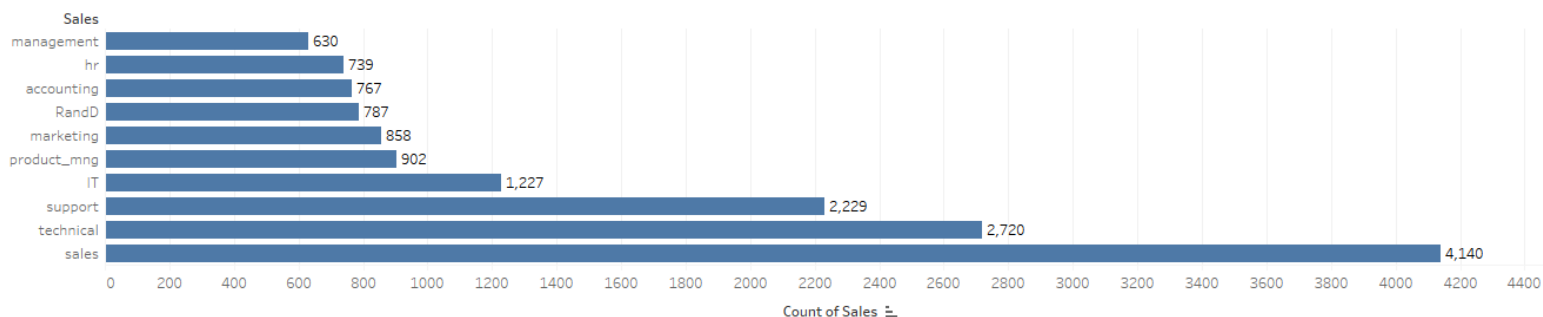
یک ویژگی باینری است که بیان می کند فرد در طی 5 سال گذشته ترفیع شغلی گرفته است یا خیر.



Sales ▪

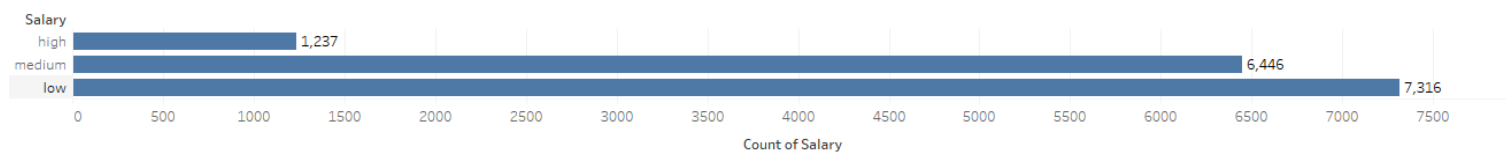


این ویژگی دپارتمانی که فرد در آن کار می کند را مشخص می کند.

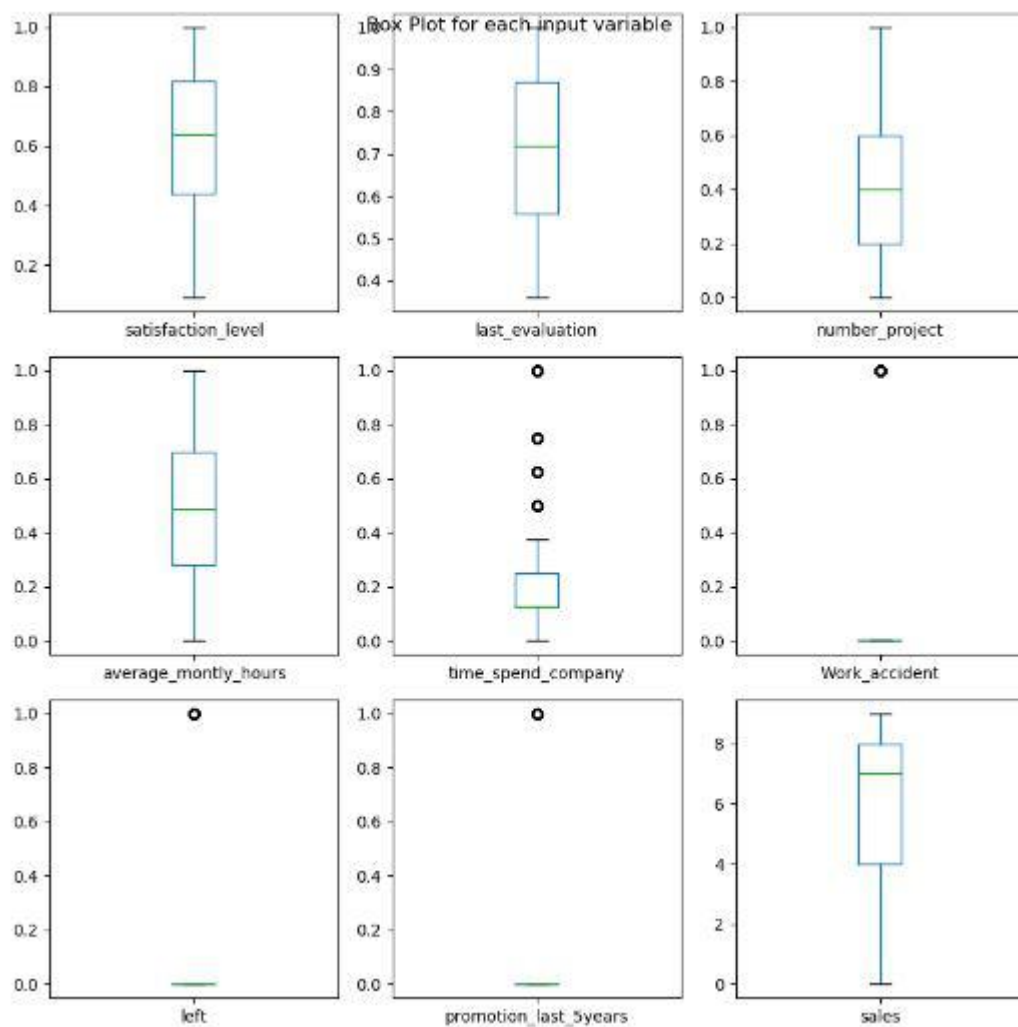


Salary ■

یک ویژگی Categorical با سه مقدار High، Low و Medium است و طبقه بندی حقوق افراد را نشان می دهد.



شکل زیر Boxplot هر یک از متغیرها را نمایش می دهد.



هدف از انجام این پروژه پیش‌بینی کلاس Salary با توجه به سایر ویژگی‌ها است. در واقع این ویژگی Class Label برای داده‌های ما می‌باشد و می‌خواهیم روی داده‌های موجود Classification انجام دهیم.

پیش پردازش داده‌ها

اصلی ترین، زمان برترین و مهم ترین گام در پروژه های داده کاوی مرحله پیش پردازش داده ها است. با دیدی که نسبت به داده داریم و بایستی پیدا کنیم، باید تصمیم بگیریم که چه اقداماتی نیاز است روی داده ی ما انجام شود تا به داده ی تمیز برسیم. در این راستا ابتدا اقدامات زیر را برای بررسی جزئی تر داده ها و روابطشان با یکدیگر انجام می دهیم.

در اولین گام، در راستای پیدا کردن درک بهتری از داده ها، تمامی Correlation های میان ویژگی ها را محاسبه و بررسی می کنیم.

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales	salary
satisfaction_level	1.00000	0.10502	-0.14297	-0.02005	-0.10087	0.05870	-0.38837	0.02561	0.00315	0.01175
last_evaluation	0.10502	1.00000	0.34933	0.33974	0.13159	-0.00710	0.00657	-0.00868	0.00777	0.01396
number_project	-0.14297	0.34933	1.00000	0.41721	0.19679	-0.00474	0.02379	-0.00606	0.00927	0.00967
average_monthly_hours	-0.02005	0.33974	0.41721	1.00000	0.12775	-0.01014	0.07129	-0.00354	0.00391	0.00708
time_spend_company	-0.10087	0.13159	0.19679	0.12775	1.00000	0.00212	0.14482	0.06743	-0.01801	-0.00309
Work_accident	0.05870	-0.00710	-0.00474	-0.01014	0.00212	1.00000	-0.15462	0.03925	0.00343	-0.00251
left	-0.38837	0.00657	0.02379	0.07129	0.14482	-0.15462	1.00000	-0.06179	0.03211	-0.00129
promotion_last_5years	0.02561	-0.00868	-0.00606	-0.00354	0.06743	0.03925	-0.06179	1.00000	-0.02734	-0.00132
sales	0.00315	0.00777	0.00927	0.00391	-0.01801	0.00343	0.03211	-0.02734	1.00000	0.00068
salary	0.01175	0.01396	0.00967	0.00708	-0.00309	-0.00251	-0.00129	-0.00132	0.00068	1.00000

همانطور که در تصویر می بینیم، با اینکه ما دنبال این هستیم که Salary را با استفاده از سایر ویژگی ها پیش بینی کنیم، اما همبستگی میان این ویژگی و سایرین، بسیار پایین است و این کار پیش بینی را سخت می کند. علاوه بر آن می تواند نشان دهنده این باشد که احتمالاً داده های ما کیفیت پایینی دارند. همچنین به نظر می رسد با توجه به این مقادیر، احتمالاً حجم زیادی از این داده ها، تولید شده اند و واقعی بودن آنها زیر سوال است.

در گام بعدی، در صدد آن هستیم تا با استفاده از بررسی VIF، تعیین کنیم که آیا ویژگی های دیگر به جز Salary را می توان با سایر ویژگی ها نتیجه گرفت یا این ویژگی ها از یکدیگر مستقل هستند (به عبارتی دیگر multicollinearity بین predictorها باید مورد بررسی قرار گیرد. با اجرای کد زیر و دریافت خروجی های آن، به بررسی این موضوع پرداختیم.

```
X_VIF = df.iloc[:,0:9].assign(const=1)
df_VIF=pd.Series([variance_inflation_factor(X_VIF.values, i) for i in range(X_VIF.shape[1])],index=X_VIF.columns)
#VIF check has to be done to control whether there's multicollinearity or not
```

نتایج به دست آمده به ما نشان می دهد که مقادیر VIF ها به طور نسبی زیاد نیست و همگی آنها اعداد کمی را به خود اختصاص داده اند و این مقادیر به معنی این است که ویژگی ها از یکدیگر مستقل هستند و در واقع نمی توان از روی ترکیبی از آنها دیگری را ساخت. پس داده ها به این لحاظ قابل استفاده و استناد هستند.

```
satisfaction_level    1.247190
last_evaluation       1.242476
number_project        1.365128
average_monthly_hours 1.286077
time_spend_company    1.077413
Work_accident         1.026154
left                  1.240613
promotion_last_5years 1.011847
sales                  1.002726
```

در ادامه با دیدی دقیق تر به پیش پردازش داده ها می پردازیم:

Missing values

خوشبختانه داده‌های موجود هیچ مقدار گم شده‌ای ندارند و از این رو لازم نیست برای این کار اقدامی انجام دهیم.

```
>>> df.isna().sum()
satisfaction_level    0
last_evaluation        0
number_project         0
average_monthly_hours  0
time_spend_company    0
Work_accident          0
left                   0
promotion_last_5years  0
sales                  0
salary                 0
dtype: int64
```

Remove duplicates

در داده‌های موجود 3008 مورد تکراری وجود دارد که لازم است آنها را حذف کنیم. نکته‌ای که در مورد این دیتاست وجود دارد و با خود دستیار محترم درس هم مطرح شد، این بود که با بررسی‌هایی که تا به اینجا کردیم و آنچه بعدتر در این گزارش به آن اشاره خواهیم کرد، این داده‌ها تولید شده‌اند و درواقع Valid نیستند. این مورد کار مدلسازی و پیش‌بینی را برای رسیدن به درجات بالای دقت سخت می‌کند. استفاده از Remove Duplicates در مدلسازی، سبب خواهد شد که 3008 عدد از رکوردهای ما حذف شود. این مقدار با وجود اینکه زیاد است اما برای ساختن مدل دقیق و کارا بهتر است حذف نشود تا مدل، داده‌ها را بهتر بشناسد، در صورت حذف آنها در نهایت دقت بسیار پایینی برای تمامی مدل‌های ساخته شده خواهیم داشت. لذا این بخش در کد وجود دارد اما کامنت شده است و ما در فرایند پروژه، ترجیح دادیم که داده‌های Duplicate را از آنجا که به ما بینش بهتری از داده‌ها می‌دادند، نگه داریم. کد زیر داده‌های Duplicate را در صورت اجرا حذف خواهد کرد.

```
#df.drop_duplicates(keep='last', inplace=True)
#Remove duplicates
```

Outlier and noisy data detection

در این بخش سعی داریم تا داده‌های پرت را در وهله اول شناسایی و در وهله دوم حذف کنیم. شناسایی و حذف داده‌های پرت یکی از مهم‌ترین گام‌های پاکسازی داده‌هاست. برای پیاده‌سازی این مرحله، ما از الگوریتم DBScan برای شناسایی داده‌های

```
model = DBSCAN(eps=0.5, min_samples=5).fit(df)
outliers_df = pd.DataFrame(df)
outliers_df[model.labels_ == -1]
df = outliers_df[model.labels_ != -1]
#because of the datatype it's better to use DBSCAN method to remove the outliers
```

پرت استفاده کردیم، سپس آیدی‌هایی که تحت عنوان داده پرت شناسایی شده بودند را از دیتاست حذف کردیم. پارامترهای الگوریتم DBScan به کار گرفته شده نیز در تصویر مشخص است.

Data transformation

از آنجایی که بیشتر ویژگی‌ها مقادیری بین 0 و 1 داشتند یا Categorical محسوب می‌شدند، اما ویژگی Average Monthly Hours مقادیر تقریباً بزرگی بین 100 تا 300 داشت، تصمیم گرفتیم تا برای رعایت حدود بین همه‌ی ویژگی‌ها و کم کردن تاثیر این ویژگی روی سایرین، این ویژگی را با روش MinMax نرمال کنیم. نرمال‌سازی این داده‌ها برای گرفتن خروجی مطلوب از مدل‌هایی که جلوتر در گزارش بررسی می‌کنیم حیاتی است.

Data reduction

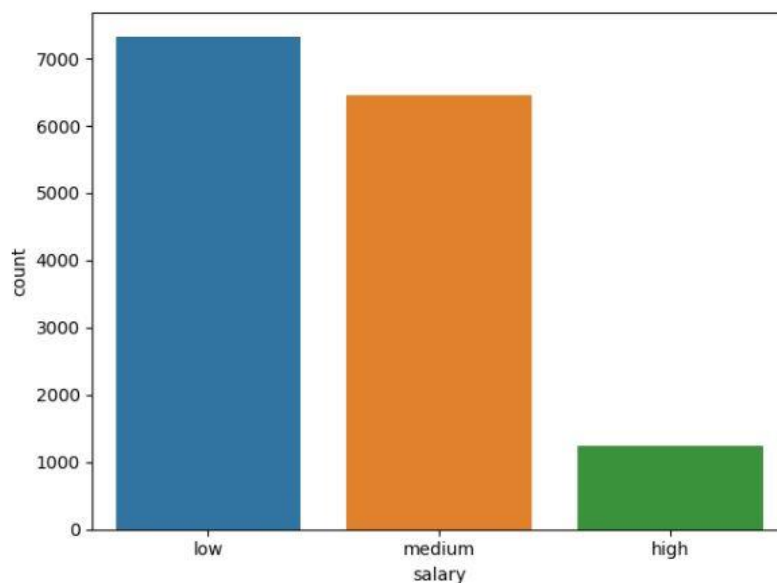
برای این بخش از روش PCA استفاده شد که در ادامه درباره‌ی آن به طور مفصل توضیح خواهیم داد. چرا که برای انتخاب تعداد ابعاد آن در هر مدل با توجه به Cross Validation مقادیر بهینه‌ی متفاوتی به دست آمد.

شناسایی و استخراج داده‌های مورد نظر جهت داده‌کاوی

در این مرحله از انجام پروژه داده‌کاوی، ما پس از انجام اقدامات لازم و متناسب با داده‌ها، به داده‌های پاک‌سازی شده دست پیدا کرده‌ایم. اکنون نیاز است تا بررسی کنیم که کدام ویژگی‌ها برای ما بیشترین اهمیت را دارند و تعیین کنیم که از چه داده‌هایی به چه شکل می‌خواهیم استفاده کنیم تا داده‌های موردنظرمان برای داده‌کاوی را استخراج کنیم.

Imbalanced Data و SMOTE

با تشریح و شناختی که از داده‌هایمان پیدا کردیم، متوجه شدیم که ویژگی Salary که در واقع همان کلاس Label ما بود، توزیع ناهمگونی دارد. به این صورت که تعداد رکوردهایی که کلاس Low داشتند 7316 بود، 6446 رکورد از کلاس Medium بود و تنها 1237 مورد از رکوردها کلاس High داشتند. این توزیع دیتا، نشان می‌دهد که ما با داده‌های ناهمگون یا Imbalanced Data سروکار داریم و بایستی برای بالا بردن دقت مدل‌هایمان به نوعی با این مسئله مقابله کنیم. اقدامی که ما تصمیم گرفتیم پیاده‌سازی کنیم، استفاده از تابع SMOTE و تولید داده در راستای ایجاد توازن میان داده‌های کلاس Label است. این تابع که بر اساس بردارهای ماشین پشتیبان یا همان SVM کار می‌کند، اساس آماری دارد و توازن میان توزیع داده‌ها را ایجاد می‌کند. بدین ترتیب، می‌توانیم داده‌های همگون یا Balanced Data برای مراحل بعدی داشته باشیم. دیتای جدید ما 21768 رکورد خواهد داشت که می‌بینیم چیزی حدود 6000 رکورد به دیتاست اضافه شده است.



توزیع داده‌ها قبل از اضافه کردن

PCA

از دیگر استراتژی‌های به کار رفته در این گام، استفاده از PCA برای کاهش ابعاد دیتاست می‌باشد. از آنجا که تعداد ویژگی‌های دیتاست ما کم است (تنها 10 Attribute داریم)، اینکه به کلی ستونی از ویژگی‌ها را کنار بگذاریم استراتژی مناسبی نخواهد بود. از سوی دیگر در ماتریس همبستگی‌ها دیدیم که Attribute‌ها در همبستگی به کلاس Label چندان تفاوت چشم‌گیری ندارند. لذا استفاده از PCA برای Subset Selection اقدام مناسبی در راستای تعیین داده‌های موردنظرمان به نظر می‌رسد. پس از اعمال PCA، ما علناً دیگر داده‌های قبلی را به فرمت پیشین نداریم و با ستون‌های جدیدی روبرو هستیم که مقادیر متفاوتی دارند (تعداد همان 10 تا است) و در ضمن هیچ تفسیری از آنها نمی‌توان داشت. در بخش مدل‌سازی و اعمال مدل‌ها، بیشتر خواهیم دید که بهینه‌سازی استفاده از این PCA چگونه صورت می‌گیرد و اینکه ما در نهایت چه ستون‌هایی از این PCA ها را برای ساختن و اعمال مدل‌ها استفاده می‌کنیم. این بهینه‌سازی از طریق امتحان کردن جایگشت‌های مختلف ستون‌های PCA (با استفاده از Cross-Validation) و تعیین دقت آنها به ازای مدل‌ها صورت می‌گیرد.

داده‌کاوی و شناسایی الگوهای پنهان در داده

همان طور که گفته شد هدف از داده‌کاوی در این پروژه پیش‌بینی Salary است. برای این منظور از تکنیک‌های Classification استفاده می‌کنیم. به عبارتی با روش‌های مختلف موجود می‌خواهیم ویژگی‌هایی که در پیش‌بینی وضعیت حقوق به ما کمک می‌کند را موردبررسی قرار دهیم. به همین جهت Salary را label قرار دادیم و به بررسی تاثیر این ویژگی‌ها در راستای پیش‌بینی این ویژگی می‌پردازیم. در نهایت نیز با استفاده از الگوهای استخراج شده و با دقت‌ترین آنها می‌توانیم در راستای بهبود عملکرد بخش حقوق و دستمزد این شرکت اقداماتی انجام دهیم. برای انجام این کار از انواع روش‌های پارامتری

و ناپارامتری (در مجموع 7 روش) برای پیش‌بینی این ویژگی استفاده کردیم، که با توجه به دیدی که از داده‌ها و توزیع آنها به دست آوردیم احتمال می‌دهیم مدل‌های ناپارامتری عملکرد بهتری داشته باشند. روش‌های استفاده شده به قرار زیر است:

روش‌های ناپارامتری

Decision tree ➤

Random Forest ➤

KNN ➤

روش‌های پارامتری

LDA ➤

QDA ➤

Naïve Bayesian ➤

Logistic regression ➤

در هر یک از این مدل‌ها سعی کردیم تا با استفاده از Cross Validation و Optimize Parameter که در قالب Forهای تودرتو نوشته شده است، بر اساس معیار دقت (Accuracy) مقادیر بهینه‌ی پارامترهای هر یک از مدل‌ها را به دست بیاوریم. در واقع برای هر یک از مدل‌ها مقادیر پارامترهایی که بیشینه دقت را به ازای آن مدل می‌دهند به دست آوردیم. برای این کار لیست‌هایی را تعریف کردیم که در طول Cross Validation مقادیر Accuracy و پارامتر را ذخیره می‌کنند و در نهایت با مرتب کردن این مقادیر، مقدار بیشینه را به دست می‌آوریم. یکی از پارامترهایی که در همه‌ی مدل‌ها وجود دارد انتخاب بهینه‌ی تعداد ویژگی‌ها از بردار تولید شده توسط روش PCA است. این بردار تولیدی با توجه به داده‌های ما در ابتدا 10 مقدار دارد و انتخاب اینکه کدام زیر مجموعه از این 10 تا ویژگی جدید را انتخاب کنیم، در نتیجه نهایی بسیار تاثیرگذار است. برای همین منظور در تمامی مدل‌ها برآنیم تا این مقدار را که در کد با متغیر c نشان داده شده است، بهینه نماییم. در ادامه در رابطه با هر یک از روش‌ها و پارامترها کمی توضیح می‌دهیم.

مدل‌های ناپارامتری

KNN

با کلیت این مدل در درس آشنا شدیم و می‌دانیم که مدل KNN با انتخاب K تا از نزدیک‌ترین داده‌ها نسبت به خود، سعی می‌کند تا پیش‌بینی درستی از داده‌ی جدید به ما بدهد. این عمل در واقع با استفاده از توابع مختلفی که فاصله دو داده را برای ما محاسبه می‌کنند اتفاق می‌افتد و ما می‌توانیم از انواع روابط استفاده کنیم. در این مدل شاخص روی "minkowski" تنظیم شده و توان آن برابر 2 است که فاصله را بر حسب فاصله‌ی اقلیدسی به دست می‌دهد. سایر پارامترها روی مقادیر پیش فرض خود قرار گرفته‌اند. پارامتری که ما قصد داریم آن را بهینه کنیم، تعداد همسایه‌ها (nneighbor) است که مقدار آن را از 1 تا 5 متغیر گذاشتیم و به کمک حلقه‌ها و محاسبه دقت به ازای هر یک از مقادیر، مقدار بهینه‌ی آن را برابر 1 به دست آوردیم. مورد دیگر انتخاب تعداد بهینه‌ی ویژگی‌های بردار تولیدی توسط PCA است. در واقع دومین پارامتر درون حلقه‌ها (c) تعداد ویژگی‌های انتخابی از بردار تولید شده توسط PCA را می‌دهد که مقدار بهینه آن 9 (از 10 تا) به دست آمده است.

Decision Tree

درخت تصمیم یکی از بهترین روش‌ها برای طبقه‌بندی داده‌ها است که 3 روش برای پیدا کردن پارامترهای شکست آن وجود دارد. در این مدل ما بر اساس Gini Index که Impurity را ملاک قرار می‌دهد عمل کردیم و شاخه‌ها را ایجاد کردیم. سایر پارامترها روی تنظیمات پیش فرض خود قرار دارند. پارامتری که قصد بهینه‌سازی آن را داریم، تعداد سطوح درخت (k) است که از 1 تا 20 متغیر می‌باشد. در انتها در این مدل به ازای 9 ویژگی از بردار تولید شده توسط PCA (c) و تعداد سطوح 19 به حالت بهینه برای دقت مدل می‌رسیم.

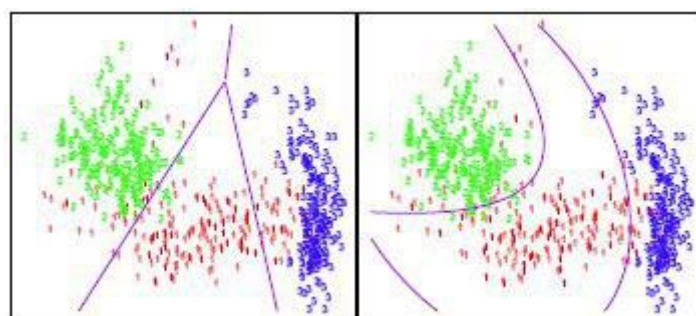
Random Forest

این مدل بر اساس پارامتر مشخص شده توسط ما، درخت تصمیم‌هایی را می‌سازد و با توجه به Test لیبلی که بیشتر نتیجه‌گیری می‌شود را به عنوان پاسخ باز می‌گرداند. اگرچه که در این مدل می‌توانستیم تعداد درخت‌های ساخته شده یعنی nest در کد را بهینه کنیم اما به علت بالا رفتن حجم محاسبات و طولانی شدن زمان آن را ثابت و برابر 1000، در نظر گرفتیم. طبق یک بار عملیاتی که برای بهینه‌سازی این پارامتر انجام دادیم این عملیات چیزی حدود 15 ساعت یا بیشتر زمان لازم دارد. سایر پارامترها نیز روی حالت پیش فرض خود قرار گرفتند و صرفاً تعداد ویژگی‌های انتخابی از بردار تولیدی توسط PCA (c) را بهینه کردیم که مقدار آن برابر 9 شد.

مدل‌های پارامتری

LDA و QDA

این دو روش که در واقع روش‌های پارامتری هستند و اساس کار آن‌ها بر این فرض است که تمامی ویژگی‌ها توزیع نرمال دارند، یکی به صورت خطی و دیگری به صورت غیر خطی به طبقه بندی داده‌ها می‌پردازند. در این روش‌ها نیز تنها پارامتری که به ازای آن بهینه‌سازی انجام دادیم، همان تعداد ویژگی‌های انتخابی از بردار تولیدی توسط PCA (c) است که مقدار بهینه آن در روش LDA برابر 9 و در روش QDA برابر 8 شد.



Logistic Regression

این روش بر اساس رگرسیون خطی و روابط آن، داده‌ها را طبقه بندی می‌کند. برای این روش پارامتر Solver را برابر Newton-cg که الگوریتم را برای بهینه‌سازی مشخص می‌کند و multi_class را multinominal قرار دادیم که loss را بر اساس کل توزیع داده‌ها مینیمم می‌کند و مهمتر اینکه این پارامتر به ازای ویژگی‌های باینری نیز درست کار می‌کند. سایر پارامترها را روی تنظیمات پیش فرض گذاشته و تغییر ندادیم. در این روش نیز بهینه‌سازی را روی c انجام دادیم و مقدار بهینه آن برابر 7 به دست آمد.

در انتهای هر یک از این مدل‌ها، بر اساس بیشینه دقت به دست آمده و پارامترهای نظیر آن، مدل را روی داده‌های Test امتحان کرده و دقت آن را ارزیابی کردیم. در ادامه به بررسی این موارد می‌پردازیم.

ارزیابی الگوهای شناسایی شده و تعیین الگوهای مطلوب

پس از آشنایی و بررسی روش‌های ذکر شده در بالا که توضیح داده شد، در این بخش به ارزیابی و مقایسه این روش‌ها می‌پردازیم و همچنین بر اساس معیار Accuracy، دقت مدل‌ها را ارزیابی کرده و بهترین الگو را بر اساس آن انتخاب می‌کنیم. برای بررسی میزان دقت هر کدام از مدل‌ها با استفاده از حلقه‌های تودرتو Cross Validation انجام دادیم. برای این کار داده‌ها را به دو بخش Train و Test تقسیم بندی می‌کنیم که این نسبت را 80 به 20 در نظر گرفتیم. بعد از ساخت مدل به کمک داده‌های Train، آن را روی داده‌های Test اعمال کرده و دقت هر مدل را می‌سنجیم. تصاویر دقت‌ها و Confusion matrix ها برای هر مدل در زیر آورده شده‌است. هر Confusion Matrix، یک ماتریس 3 در 3 است که به ترتیب وضعیت Label، Low و Medium و High را نشان می‌دهد و در بررسی مدل‌ها با آن سروکار خواهیم داشت.

مدل KNN

در تصویر روبه‌رو می‌بینیم که دقت کلی این مدل در مواجهه با داده‌های Test، 74.12% بوده است. همچنین Confusion Matrix آن نیز به ما نشان می‌دهد که وضعیت پیش‌بینی مدل ما و وضعیت حقیقی کلاس Label به چه صورت بوده است.

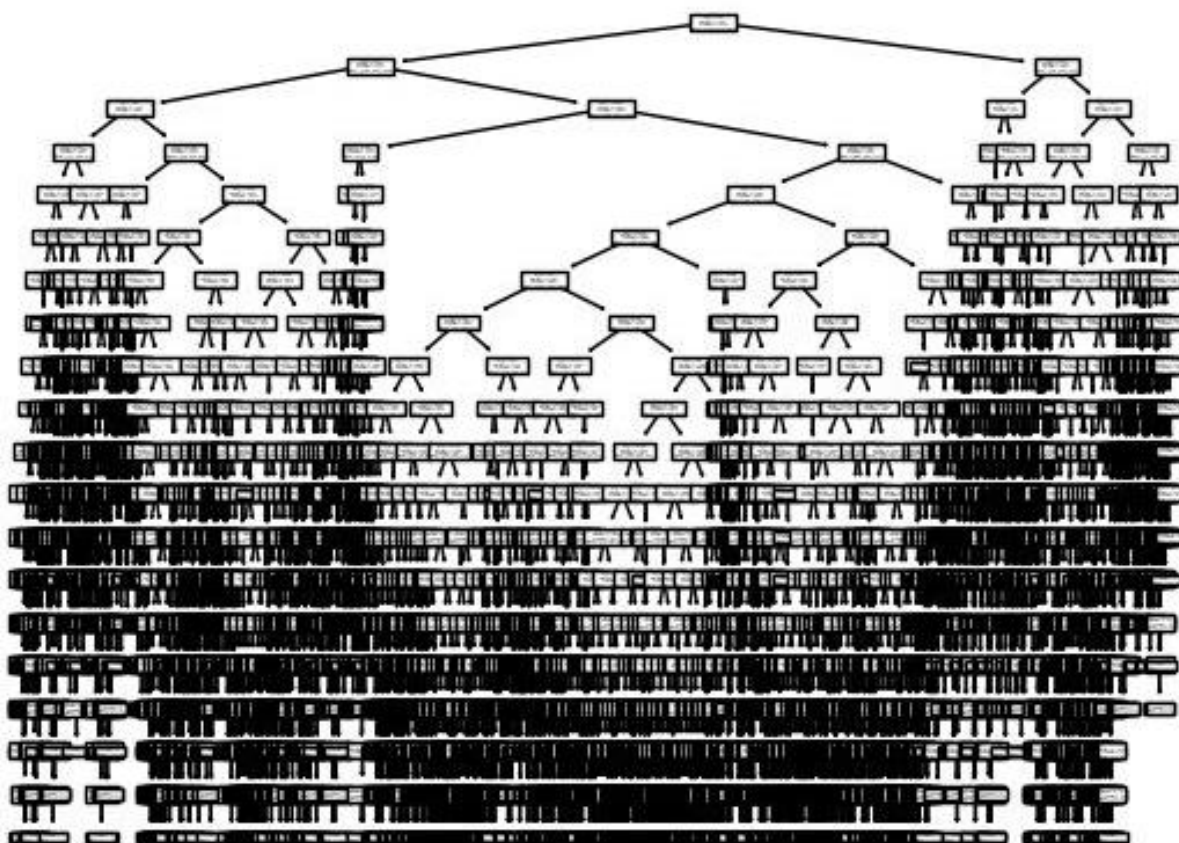
```
Knn-PCA Train Accuracy: 0.9994778370851706
Knn-PCA Prediction Accuracy: 0.7412993039443155
Knn-PCA Model for Testing Data
[[1344   31    24]
 [ 157  909  363]
 [ 129  411  942]]
```

مدل Decision Tree

دقت کلی این مدل در مواجهه با داده‌های Test، 65.03% بوده است. همچنین Confusion Matrix آن در تصویر زیر مشخص بوده و وضعیت پیش‌بینی مدل ما و وضعیت حقیقی کلاس Label را نشان می‌دهد.

```
Decision Tree-PCA Train Accuracy: 0.8926084938500812
Decision Tree-PCA Prediction Accuracy: 0.6503480278422273
Decision Tree-PCA Model for Testing Data
[[1153  101  145]
 [ 225  826  378]
 [ 247  411  824]]
```

تصویر زیر خروجی درخت تصمیم را نشان می‌دهد که به علت زیاد بودن سطوح و شاخه‌ها و مهمتر قابل فهم نبودن مقادیر(PCA) مقادیر را به کلی عوض می‌کند) جدید صرفاً نمایی شماتیک از آن را گذاشتیم که البته در کد نیز این خروجی قابل مشاهده است.



مدل Random Forest

این مدل در مواجهه با داده‌های Test، دقتی معادل 73.13٪ دارد. همچنین Confusion Matrix آن نیز به ما نشان می‌دهد که وضعیت پیش‌بینی مدل ما و وضعیت حقیقی کلاس Label به چه صورت بوده است.

```
Random Forrest Train Accuracy: 0.9993618008818751
Random Forrest Prediction Accuracy: 0.731322505800464
Random Forrest Model for Testing Data
[[1403  54  49]
 [ 139 856 409]
 [ 131 376 893]]
```

مدل LDA

در تصویر می‌بینیم که دقت کلی این مدل در مواجهه با داده‌های Test، 42.50٪ بوده است. همچنین Confusion Matrix آن نیز به ما نشان می‌دهد که وضعیت پیش‌بینی مدل ما و وضعیت حقیقی کلاس Label به چه صورت بوده است. همانطور که مشاهده می‌کنیم با افت شدید دقت در این مدل نسبت به مدل‌های قبلی همراه هستیم که پس از بررسی هر 4 مدل پارامتری، دلیل این مورد را ذکر خواهیم کرد.

```
LDA-PCA Train Accuracy: 0.420863309352518
LDA-PCA Prediction Accuracy: 0.42505800464037125
LDA-PCA Model for Testing Data
[[1174  116  109]
 [ 795  490  144]
 [ 998  316  168]]
```

مدل QDA

در تصویر می‌بینیم که دقت کلی این مدل در مواجهه با دیتای Test، 38.58٪ بوده است. همچنین Confusion Matrix آن نیز به ما نشان می‌دهد که وضعیت پیش‌بینی مدل ما و وضعیت حقیقی کلاس Label به چه صورت بوده است. این مدل نیز با افت شدید دقت همراه است.

```
QDA-PCA Train Accuracy: 0.3883731724297981
QDA-PCA Prediction Accuracy: 0.38584686774941995
QDA-PCA Model for Testing Data
[[ 581  818    0]
 [ 348 1078    3]
 [ 482  996    4]]
```


مدل Logistic Regression

در تصویر می‌بینیم که دقت کلی این مدل در مواجهه با دیتای Test، 41.78% بوده است. همچنین Confusion Matrix آن نیز به ما نشان می‌دهد که وضعیت پیش‌بینی مدل ما و وضعیت حقیقی کلاس Label به چه صورت بوده است.

```
Logistic Regression Train Accuracy: 0.418832675794848
Logistic Regression Accuracy: 0.4178654292343387
Logistic Regression Model for Testing Data
[[1221  170    8]
 [ 860  561    8]
 [1072  391   19]]
```

این مدل نیز با افت شدید دقت همراه است. تا اینجا 3 مدل پارامتری را بررسی کردیم و هر سه با این روند نزولی در دقت همراه بوده‌اند که البته دلیل آن تا حدودی واضح است و در ادامه خواهیم گفت.

مدل Naive Bayesian

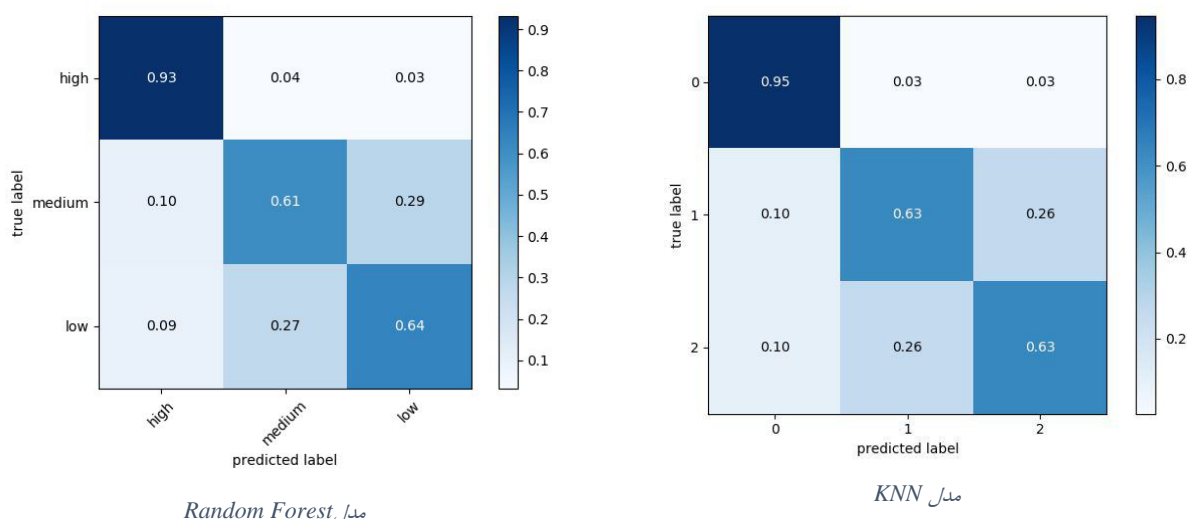
در تصویر می‌بینیم که دقت کلی این مدل در مواجهه با دیتای Test، 41.60% بوده است. همچنین Confusion Matrix آن نیز به ما نشان می‌دهد که وضعیت پیش‌بینی مدل ما و وضعیت حقیقی کلاس Label به چه صورت بوده است. این مدل هم مانند مدل‌های قبلی در دسته‌ی پارامتری افت دقت داشته است.

```
Naive Bayes Train Accuracy: 0.41395915525644
Naive Bayes Accuracy: 0.4160092807424594
Naive Bayes Model for Testing Data
[[1132  222   45]
 [ 784  594   51]
 [ 972  443   67]]
```

همانطور که ملاحظه کردیم، به طور کلی در بین تمامی مدل‌های ایجاد شده، مدل‌های پارامتری دقتی بسیار پایین‌تر از مدل‌های ناپارامتری به ما دادند. دلیل این امر، این است که اصولاً این مدل‌ها بر اساس اینکه داده‌ها توزیع نرمال داشته باشد پایه‌ریزی شده‌اند، اما داده‌های ما این ویژگی را ندارند و بنابراین دقت پایینی را به ما می‌دهند.

ارائه نهایی الگوها و دانش کسب شده

آنچه از بخش قبلی دیدیم و تحلیلی که توانستیم از آن بدست بیاوریم، ما را به این درک رساند که استفاده از الگوهای پارامتری برای این دیتاست، دقتی پایین‌تر از الگوهای ناپارامتری به همراه دارد. در کنار این قضیه، دو مدل KNN و Random Forest بهترین مدل‌های ما در راستای پیش‌بینی ویژگی Salary بودند، که بالاترین دقت‌ها را به ما می‌دادند. KNN با توجه به ماهیت مدل خود و به دست آمدن مقدار بهینه $K = 1$ علی‌رغم بهینه‌سازی که روی این پارامتر انجام دادیم، همچنان نشان از زیرسوال بودن اعتبار داده‌های ما دارد. در ماتریس‌های زیر اختلاف و چگونگی عملکرد دو روش KNN و Random Forest را می‌بینیم.



تصویر سمت راست ماتریس مدل KNN و تصویر سمت چپ مدل Random Forest را به ما نشان می‌دهد. همانطور که می‌بینیم، Random Forest در تعیین لیبل Low بهتر توانسته عمل کند و مدل KNN در تعیین لیبل High بهتر توانسته عمل کند. دقت این دو مدل بسیار نزدیک است و تنها حدود 1٪ با هم اختلاف دارند. KNN دقت 74.12٪ دارد و Random Forest دقت 73.13٪. با اینکه شاید این یک درصد معیار خیلی تعیین‌کننده‌ای برای به کارگیری مدل‌ها به جهت پیش‌بینی داده جدید نباشد، اما در عمل، می‌توانیم ترکیب هردوی این مدل‌ها را استفاده کنیم و از کاربرد جفتشان بهره ببریم. با این استدلال که اگر هر دو برابر بودند که پیش‌بینی ما از Salary همین مقدار است. از آنجا که KNN در تعیین لیبل High دقیق‌تر است و Random Forest در تعیین لیبل Low، پس وزن بیشتری به پیش‌بینی این دو مدل از این دو لیبل در عمل خواهیم داد.

در نهایت اما، برای انتخاب یک مدل و بهترین مدل، از آنجا که تعداد داده‌های با لیبل High برای ما در ابتدا بسیار پایین بود و نیاز داشتیم تا بتوانیم پیش‌بینی دقیق‌تری از لیبل High داشته باشیم، می‌توانیم مدل KNN را به دلیل دقت بالاتر در تعیین لیبل High به عنوان مرجع قرار دهیم و اگر بخواهیم تنها از یک مدل استفاده کنیم هم به این دلیل و هم به دلیل دقت بالاتر از آن استفاده کنیم.

نتیجه گیری

برای پروژه‌های داده‌کاوی آنچه در نهایت به عنوان نتیجه از آن یاد می‌شود، اصولاً درک بسیار بهتر دیتاست نسبت به درک اولیه ما از آن، بهترین مدلی که می‌تواند ما را در پیش‌بینی داده جدید یاری کند، و هر آنچه بتوان از خروجی‌هایی که در مسیر بدست آوردن داده‌ها فرا گرفتیم استخراج کرد، می‌باشد.


عدم همبستگی ویژگی‌ها به هم تا حد بالایی به ما نشان می‌دهد که داده‌های ما لزوماً روی هم تاثیر بسیار زیادی نداشته‌اند و برخلاف تصور اولیه که فکر می‌کردیم شاید درآمدهای دپارتمان‌های مختلف با یکدیگر اختلاف زیادی داشته باشند و دپارتمان‌های مشابه، درآمد مشابهی داشته باشند، اینگونه نبود.

استفاده از انواع مدل‌ها برای رسیدن به درک خوبی از داده‌ها، لازم و ضروری بود. همانطور که دیدیم مدل‌های پارامتری تصویر درستی از ویژگی‌ها و داده‌های ما به دست نمی‌آوردند و این مدل‌های ناپارامتری بودند که دقت بالایی به ما می‌دادند.

علی‌رغم اینکه ما شاهد تعداد بالایی داده Duplicate بودیم، اما اجرای گام پیش‌پردازش و حذف این داده‌ها منجر به پایین آمدن دقت مدل تا حد زیادی می‌شد. این امر سبب شد که داده‌های Duplicate را حذف نکنیم، که اصولاً کاری برخلاف حالت عادی انجام پیش‌پردازش است اما دیدیم که این حالت نتیجه بسیار بهتری را به دنبال خواهد داشت.

به طور کلی، هر سازمانی که داشته باشیم، تعداد افرادی که درآمد زیاد دارند کم خواهد بود، اما دقت ما در تعیین و شناسایی این افراد بسیار حیاتی و مهم خواهد بود، زیرا در صورتی که آنها را به درستی شناسایی نکنیم، ممکن است دچار مشکلات فراوانی در صورت‌های مالی و تعیین بودجه‌بندی شویم. مدل‌های ناپارامتری که در این پروژه از آنها استفاده کردیم، تا حد بسیار خوبی (95%) می‌توانند این افراد را شناسایی کنند، در صورتی که دیگر مدل‌ها، یا بدون نرمال‌سازی و گام‌های پیش‌پردازش، تقریباً افراد با درآمد بالا را نمی‌توانستند پیش‌بینی کنند که برای شرکت می‌توانست مشکل‌زا باشد.

از آنجا که دقت بهترین مدل‌های ما چیزی در حدود 73٪ است، شاید نتوان به طور کامل و انحصاری به نتایج آنها به عنوان نتایج قطعی و درآمد پیش‌بینی شده‌ی فرد اتکا کرد، اما استفاده از ترکیبی از آنها می‌تواند تا حد خوبی ما را در تعیین درآمد افراد یاری کند.



برای انجام این پروژه از زبان برنامه‌نویسی پایتون تحت نرم‌افزار و بستر Pycharm استفاده شده‌است. با توجه به اینکه پایتون یک زبان Open Source است کاربردهای متنوع و گوناگونی برای آن وجود دارد که در این پروژه از کتابخانه‌های آماری و مرتبط با داده‌کاوی به همراه کتابخانه‌های گرافیکی آن استفاده شده‌است. مهمترین کتابخانه‌های اضافه شده numpy و pandas هستند. با توجه به حجم بالای sklearn برای اجرای کد تنها بخش‌های خاصی از این کتابخانه به کد اضافه شده‌است. برای انجام کارهای گرافیکی نیز از کتابخانه‌های Seaborn، mlxtend و matplotlib بهره برده‌ایم. جهت اجرای آزمون VIF نیز ملزم به استفاده از statsmodels شدیم. آخرین کتابخانه‌ی مورد استفاده نیز imblearn است که جهت برطرف کردن مشکل imbalanced از آن استفاده شده‌است.

کد این پروژه در قالب فایل پایتون به طور کامل کامنت گذاری شده‌است. فایل کد در فرمت‌های پایتون و txt پیوست شده است.

منابع و مراجع

<http://rasbt.github.io>

<http://scikit-learn.org>

<https://pandas.pydata.org/>

<https://www.numpy.org/>

<https://scikit-learn.org/stable/modules/generated/>

اسلایدهای آموزشی دکتر یاسر زره‌ساز

اسلادهای آموزشی دکتر مجید خدمتی