خلاصه مقاله MIT

تخمین غیرنظارتی امنیت رانندگی رانندگان

مقدمه:

اطلاعات به طور کامل هست و قابل استراج است ، با کمک این اطلاعات به سادگی میتوان ، جاده ها را امن تر ، محیط زیست را بهتر کرد، میتوان به سادگی این تکنولوژی را در اختیار کارخانه ها و شرکت هایی مانند بیمه قرار داد . ولی در ابتدا باید چند مسئله و چالش را برطرف نمود.این مشکلات شامل 1) محاسبه دقت تخمین امنیت راننده 2) خودرو های متفاوت 3) سبک های رانندگی متفاوت 4) نظارت دقیق تر بر رفتار راننده تا سیاستهای شرکت های بیمه به صورت شخصی تر امکان پذیر باشد.

قدم اول برای این کار ، شرکت ماکروسافت با شرکت  [Pointer Telocation](https://www.pointer.com/) قرارداد همکاری بسته که اطلاعات را برای این شرکت تهیه کند و گزارشی از رفتار رانندگان را به مدیران ناوگان های حمل نقل ارائه کند.

پوینتر تلوکیشن ، یک شرکت توسعه دستگاه های مخابراتی برای خودرو هاست که روی مدریت ناوگان تمرکز دارد( MRM (Mobile Resource Management)). که علاوه بر سرعت و موقیعیت توانایی انتشار رخ داد های متفاوتی که به رفتار راننده مربوط میشوند از جمله ترمز هارش ،چرخش هارش ، سرعتگیری هارش

را اعلام میکند

هدف این پروژه تخمین یک امتیاز برای میزان امنیت رانندگی فرد است. همچنین علاوه بر آن میتوان به معیار های دیگری همچون ، بهنیه بودن رانندگی(echo از نظر مصرف سوخت) ، تشخیص هویت راننده ،تشخیص سرقت خودرو اشاره کرد

برای این کار(هدف اصلی) ما یک شیوه ی یادگیری غیرنظارتی(unsupervised) را برای مقایسه و دسته بندی افراد روی یک جامعه نمونه انجام میدهیم.

چالش ها و اهداف:

دستگاه های این شرکت pointer رخ داد ها را تشخیص میدهند و رخ داد ها را ارسال میکنند. (تشخیص رخ داد در کاربر است ، حالت edge compiuting چون اگر بخواد تو کلود باشه حجم داده ارسالی زیاد میشه . همینجوریش هم حجم اطلاعات جا به جا شده زیاد است.{من با دنبال کردن اهداف آموزشی فعلن این کارو نمیکنم چون دیتابیسی که دارم دیتا خامه ، تو آینده اصلاح خواهم کرد}

میگه جایی که باید این اطلاعات رو هندل کنه باید خیلی قوی باشه ، ماکروسافت میاد کمک این شرکته تا یه محیط ابری برای تخمین اولیه رفتار راننده ها رو دراختیار این شرکته قرار میده.

این شرکته به یک مدل ماشین لرنینگ آن سوپروایز قابل توسعه (چون تو آینده دیتا بیشتری بهش میرسه) نیاز داره

راهکار مقایسه راننده با جامعه نمونه ، نیازمند یه مکانیزم نمره دهی برای مقایسه است. که بتونه "ریسک رانندگی" رو اندازه بگیره و اونو نسبت به جامعه ای که مد نظر قرار داره مقایسه کنه.

همه ی رانندگان توسط یه سری پارامتر دسته بندی میشن(مثل ؛ نوع ماشین ،کشور ، دستگاهی متصل شده) و هر شخص توسط افراد همگن خود مقایسه میشود. برای مثال برای جلوگیری از در نظر نگرفتن پارامتر های محیطی یک راننده اتوبوس در ایتالیا با یک راننده اتبوس در آلمان مقایسه نخواهد شد. همچنین باید از بزرگی جامعه هدف اطمینان داشته باشیم تا مقایسه منصفانه انجام شود.

راهکار:

این آنالیز را با شروع از یک دیتاست با 242 راننده شروع کردیم. که پس مدتی این تعداد به 500 هزار رسید

بعد از تصمیم گیری برای نوع مدل ماشین لرنینگ ،این مدل را به گونه ای توسعه دادیم که همه رانندگان را در برگیرد. از آن جایی که عدد دقیقی برای محاسبه دقت تخمین امنیت وجود ندارد ، تنها دو راهکار برای این ارزیابی ، دقت تخمین امینیت وجود دارد:

راهکار اول

به صورت دستی برچسب گزاری شود (رانندگی های پر خطر مشخص شوند) که مسئله به صورت سوپروایز یا سِمی سوپروایز حل شود.

مزایا : ساده شدن اندازه گیری و توسعه

معایب: پر هزینه و زمان بر ، ایمنی تعریف شده ، جامع نیست (ذهنی و سابجکتیو میشه)

راهکار دوم

یادگیری آن سوپروایز / شیوه های قدیمی احتمالاتی

مزایا: نیاز به انسان ندارد

معایب: هیچ پارامتر قوی ای ، برای براورد تخمین وجود ندارد

در این پروژه تیم تصمیم بر ، راهکار دوم گرفت.

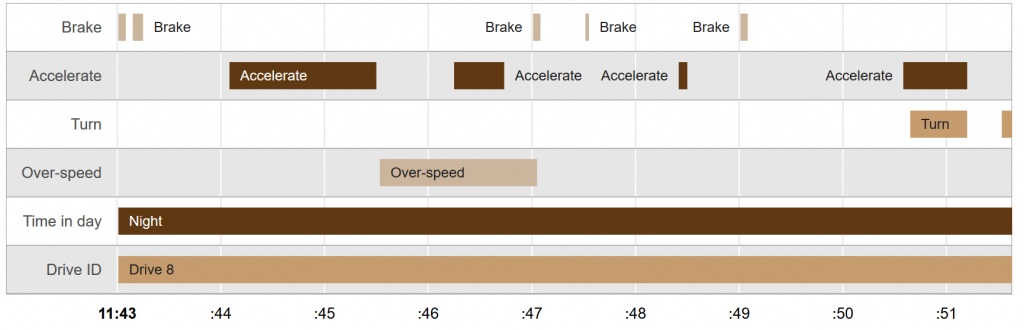
ولی برای راه کار اول ، برای اطلاعات بیشتر ، مارو ارجاع میکنه به یه مقاله دیگه با عنوان

Categorizing Driver Risk with Machine Learning

{برای پایان نامه میتونیم روی اون هم بریم ولی هدف اون نیست : <https://devblogs.microsoft.com/cse/2016/01/07/categorizing-driver-risk-with-machine-learning/>}

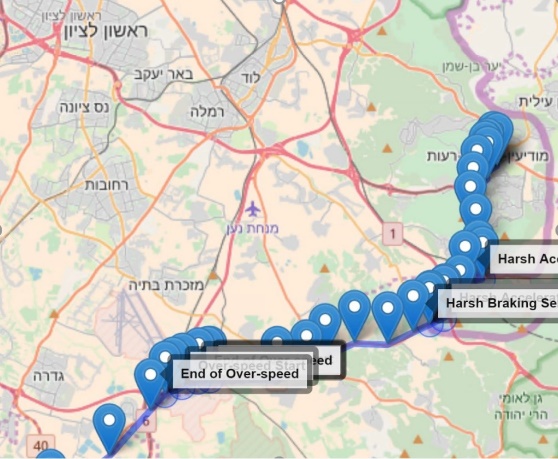
قبل از شروع راهکار غیرنظارتی ، چنتا چیز رو به نمایش میزاریم که تخمین بزنیم چه چیزایی قراره که امتیاز رانندگان رو مشخص کنند. انگیزه اصلی از نمایش اطلاعات فهم دقیق این است که الگو های هارش کِی و چگونه ایجاد میشوند.

اولین نمایش به شکل تایم لاین است :



نمایش تایم لاین ، اجازه درک بهتر از رخ داد های کوتاه مدت و تعداد این رویداد ها را به آنالیزگر میدهد.

علاوه بر این ، تایم اجازه گنجاندن رویداد ها در هم را فراهم میکند ، برای مثال زمان رانندگی(روز ، شب و..) یا آب و هوای هنگام رانندگی(بارانی ، ابری و...) این مفاهیم آنالیزگر یا لیبل­زن را کمک میکند که آیا این رفتار هارش تقصیر راننده است یا خیر.

دومین شیوه نمایش نمایش جغرافیایی است به نحوی که اتفاقی که افتاده رو نقشه مشخص کنیم: نمایش نقشه ای برای تحلیل موقعیت های مکانی است که رانندگان بیشتر در آن نواحی رفتار های هارش دارند.

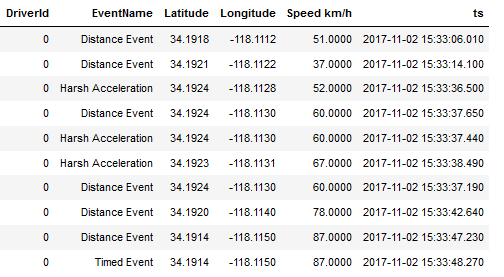
<https://github.com/Microsoft/eventsVis> کدش اینجاس

آنالیز اکتشافی داده ها

مرحله بعدی برای آنالیز اطلاعات بررسی دقیق اطلاعات رانندگان این است که بررسی کنیم که

1)شامل چیست ؟ 2)چه چیزهایی گم شده است؟ 3)آیا خطایی وجود دارد یا خیر؟

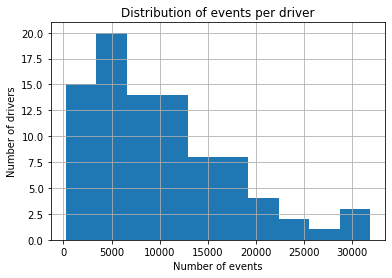
همچنین با اطلاعات موجود باید سوال های بیزینسی پاسخ داده شود. خام اطلاعات اینه:



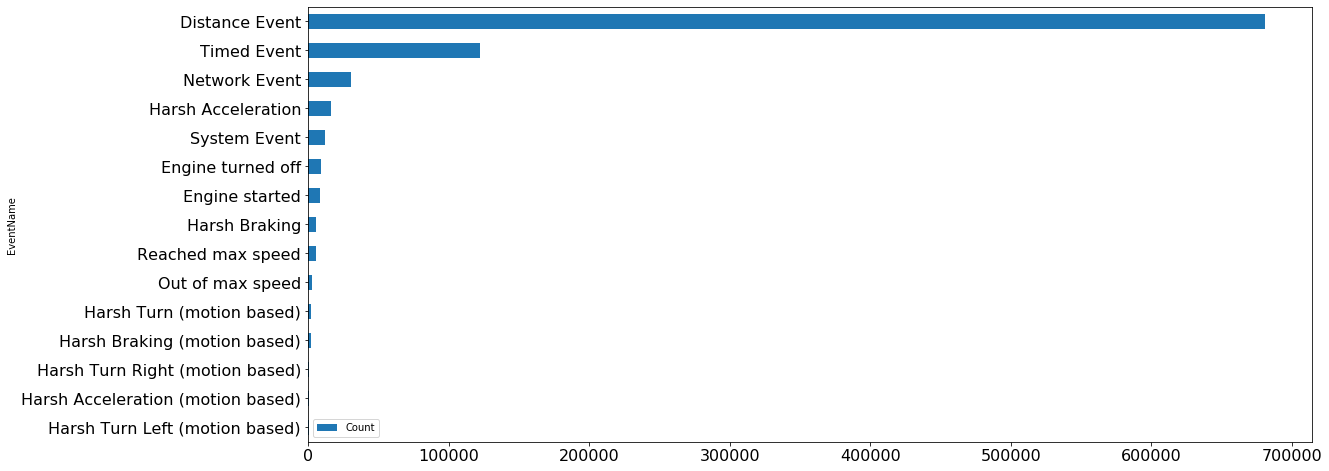
این دیتاست حاوی اطلاعات متفاوتی است ولی به طور کلی میتوان به دو نوع تقسیم نمود ، اطلاعات رفتاری و اطلاعات عملکردی ، که در ستون EventName قرار دارند که به آن اطلاعات gps و سرعت و تایم استمپ نیز اضافه شده و ستون driverId راننده رامشخص میکند.

این رویداد ها در دستگاه های شرکت Pointer و توسط سنسور های آنها تولید شده ، که در این مقاله اشاره ای به نحوه تولید این سیگنال ها ندارد. ولی به طور کلی هر کدام از این رویداد ها توسطی سنسوری اندازه گیری شده به نحوی که اطلاعات سنسور به آستانه خاصی رسیده یا از آن عبور کرده باشد.

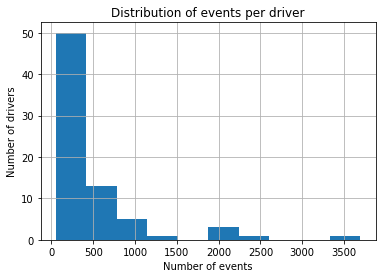
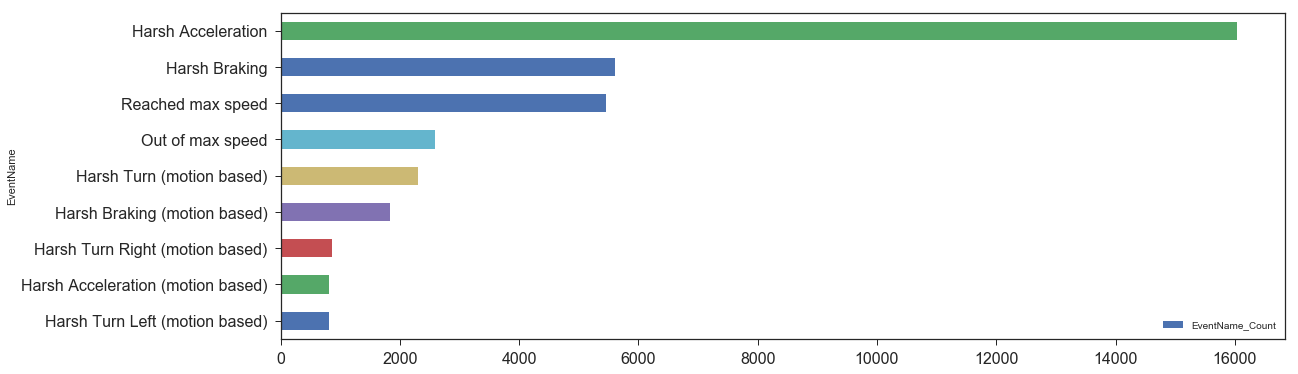
نمودار توزیع تعداد رویداد ها به تعداد رانندگان:



میانگین تعداد رویداد ها برای هر راننده 7435 است البته برای تخمین نوع رانندگی راننده ، ما تنها به تعداد رویداد زیاد نیاز نداریم بلکه به تنوع و نوع رخ دادهای موجود نیاز داریم. تنوع رویداد های موجود را متوان در زیردید



همانطور که مشاهده میشود دو نوع رویداد موجود است ؛ یکی رویداد های مربوط به سیستم و دیگری رویداد های مربوط به رفتار راننده چون رویداد های (distance, timed, network and system) ارتباطی با رفتار راننده ندارند ، ما آن ها را برای تخمین رفتار راننده استفاده نمیکنیم و تنها برای محاسبه مسافت طی شده و estimate over-speeding استفاده میکنیم. بعد از تمیز کردن اطلاعات ، مجدد نمودار تعدادرخ دادها و توزیع رانندگان را بررسی میکنیم:



نمودار فوق چند حقیقت را آشکار میکند:

1)تعداد رویداد های ‘Reached max speed’ با تعداد رویداد های ‘Out of max speed’ برابر نیست که نشان دهنده خطا است ! ما تصمیم گرفتیم چون این رویداد ها از مقایسه سرعت خودرو با یک عدد ثابت (120 کیلومتر بر ساعت) به وجود آمده اند نه محدودیت سرعت واقعی پس میتوانند در نظر گرفته نشوند.در مقابل از یک api که سرعت مجاز در نقاط مختلف جاده را بررسی میکند استفاده کردیم که سرعت مجاز واقعی را بدست آوریم همچنین این api کمک میکند که خطای ناشی از gps که ممکن است خودرو را در جاده نشان ندهد جبران کرده و موقعیت را دقیق میکند. (bing map api که برای پایتون هم lib داره و مشکل تحریمی برای ایران نداره فقط توی ایران خودش گفته ضعیف عمل میکنه چون دیتا زیاد نداشته ولی از اون جایی که دیتاستایی که من دارم هیچ کدومش ایران نیست ، ازش استفاده میکنم)

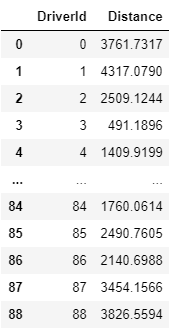
2)ما دو نوع رویداد در دیتاست مشاهده مکنیم که نمایشگر رخ داد های یکسانی هستند،

Harsh Acceleration و Harsh Acceleration (motion based) که از دو نوع سنسور متفاوت خارج میشوند (احتمالا یکی از ecu خودرو و گذر از دور موتوری خاص و دیگری از طریق اکسلورومتر)

3) تفاوت میان harsh left و harshright برای ما اهمیت ندارد ، پس تنها از harsh turn استفاده خواهیم نمود.

از آنجایی که مسافت رانندگی رانندگان یکسان نیست ، پس برای یک مقایسه منصفانه ابتدا مسافتی هر کدام طی کرده اند را از طریق Haversine formula بدست می­آوریم و بر تعداد رویداد های آنها تقسیم میکنیم تا اطلاعات نرمالیزه شوند: همچنین تعداد رانندگانی که تعداد رخ دادهای ارسالی آنها از یک حد مینیمم(50 رویداد) باشد را حذف میکنیم. چراکه outlier حساب شده و میزان مسافت طی شده آنها بسیار کم است و محاسبات را از حالت جنرال خارج میکند.

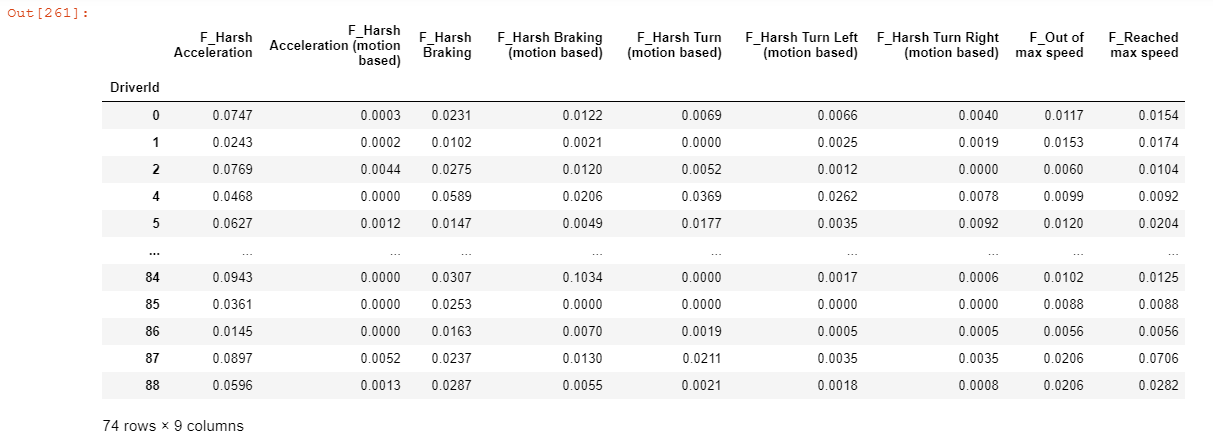
Haversine formula: یک رابطه برای مشخص کردن فاصله ی دایروروی بین دو طول و عرض جغرافیایی است که از قانون law of haversines خارج شده است (وارد جزئیات نمیشم چون مباحث ژئومتریک عه)



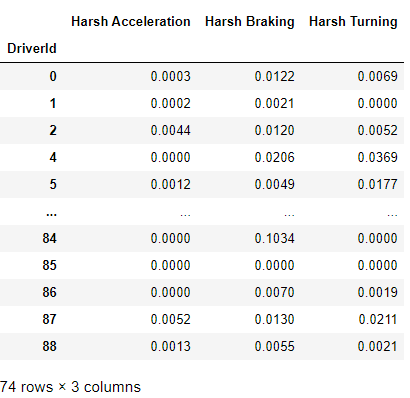
مسافت طی شده توسط هر راننده توسط هر راننده محاسبه شد:

ساختن فیچر ها (مهندسی ویژگی ها)

ما یک مجموعه از ویژگی های نرمالیزه شده برای هر راننده تعریف میکنیم. به نحوی که تعداد کل رخ داد های هر راننده را بر مسافت طی شده توسط وی تقسیم کرده و به عنوان بردار ویژگی او ،در نظر میگیریم.

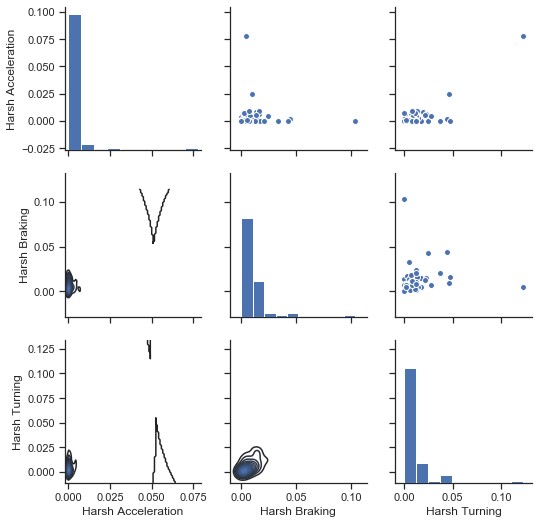


خروجی برای هر راننده به شکل فوق درامد {در انجام محاسبات به دلیل کوچک بودن اعداد بهتر است لگاریتم بگیریم ولی الان دارم مقاله رو بازنویسی میکنم ، برای اینکه عدد هام یکی دربیاد دست نمیزنم}

در ادامه میگه از اونجایی که دو نوع متفاوت از رویداد ها ( motion based و عادی) داریم یا باید میکس کنیم یا تکی استفاده کنیم ، میگه که تکی استفاده میکنیم، همچنین از ویژگی over speeding فعلا میگذریم . جدول زیر رو به عنوان بردار ویژگی در نظر میگیریم که بتونیم تحلیلش کنیم:

حالا یه نگاه بندازیم ببنیم چی داریم وابستگی اینا ، اوت لایر چیزی مونده ؟

نمودار زیر توزیع اطلاعات مربوط به رفتار راننده را در اختیار ما قرار میدهد:



در مثلث بالای نمودار اسکترپلات اطلاعات بر حسب یکدیگر و در قطر اصلی هیستوگرام و توزیع اطلاعات را مشاهده و در مثلث پایینی KDE (Kernel Density Estimation) را مشاهده میکنیم(kde تابع توزیع احتمالی است که احتمال وقوع در آن مشخص میشود نقاطی که تراکم بیشتری دارد احتمال رخ داد بیشتر است)

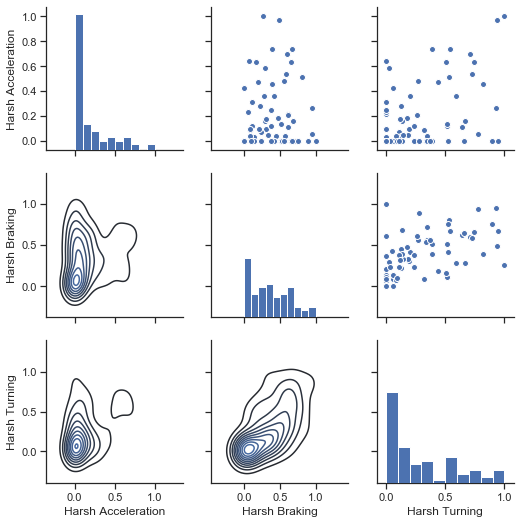
همه ی ویژگی ها با جزئیات دقیق آماده شدند ولی از اسکتر پلات مشخص است که ویژگی ها به سمت راست کشیده شده اند و داده های پرت وجود دارد پس باید با آن ها به صورت جداگانه حذف کنیم

همانطور که گفته شد ، تعدادی داده های پرت وجود دارند که برای انجام محاسبات آماری خطا ایجاد میکنند ، پس برای هندل کردن آنها ، ابتدا از یک تبدیل استفاده که آن ها را به توزیع نرمال شبیه کند(boxcoxtrsnform){این تبدیل رو گزاشتم تو ضمیمه (ته داستان)} سپس با قاعده ی μ+k\*σ داده های پرت را حذف میکنیم (آن هایی که از میانگین دور مشوند را با پارامتر k حذف مکنیم)

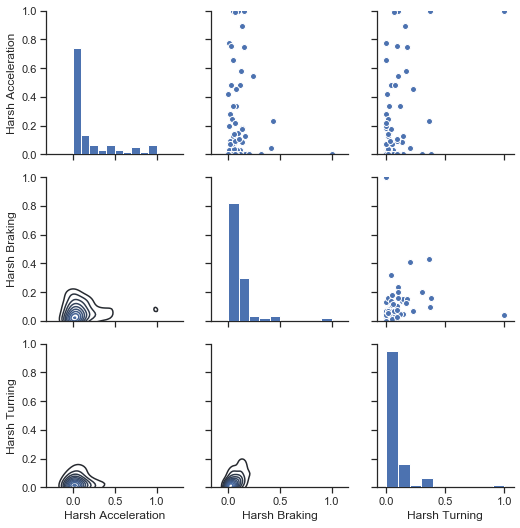
راهکار دوم ، این است که داده هایی که از حدی بزرگ تر میشوند را حذف کنیم ولی این راهکار سخت تر از قبلی است چرا که پیدا کردن آستانه بزرگ کوچک بودن زیاد ساده نیست. راهکار سوم استفاده از روش های فیت کردن توزیع چند متغییره است. برای مثال توزیع نرمال چند متغییره(اینو باید کدشو ببنیم) .

راهکار اول :

بعد از تبدیل box cox توزیع فوق به شکل زیر در می آید همچنین به دلیل کوچک بودن مقادیر فاصله ی بین آن ها رو کمی زیاد کنیم تا نمودار قابل درک تر باشد برای اینکار همه اطلاعات را به min/max-min تقسیم کرده تا فواصل بهتر مشاهده شود



همانطور که قابل مشاهده است نمودار های توزیع هنوز به حالت متقارن نیست ولی نسبت به به داده های اولیه توزیع متقارن تری دارد. حال به حذف دادهایی که از یک میزان مشخص از انحراف معیار دورتر هستند میپردازیم(آن هایی که انحراف معیار بیشتر از 2.5 دارند را حذف میکنیم و مجددا نرمالیزه میکنیم):



حالا دیگه دیتا تمیز شد 😊

مدل کردن اطلاعات:

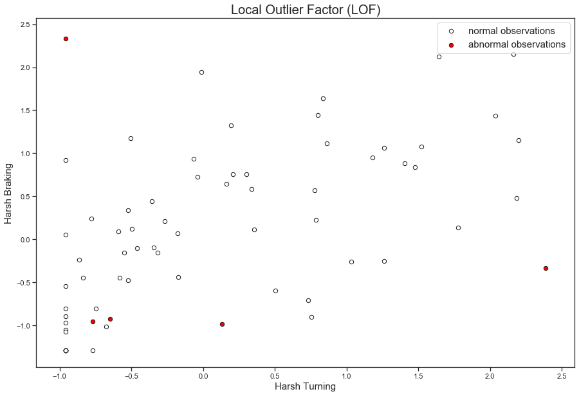
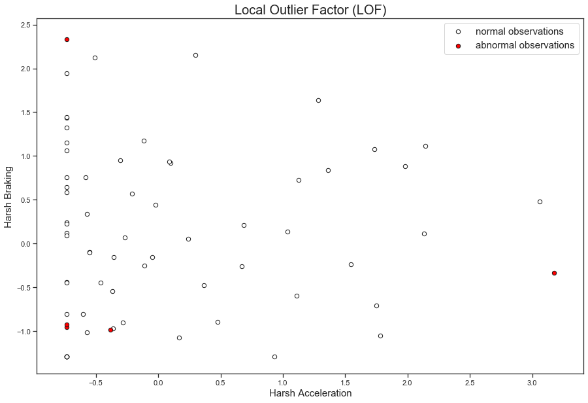
فرضیه اصلی

فرض اولیه ما برای انالیز اطلاعات این بود که هر چقدر اتفاقات هارش بیشتری رخ دهد ، آن راننده رفتار بدتری دارد ، و راننده ای که اتفاقات هارشی مثل ، سرعتگیری زیاد ، ترمز شدید ، سرعت غیرمجاز ، و.. اینا نداشته باشد ، رفتار آن راننده خوب است.

راهکار های مدل کردن که در این مقاله ارائه شد

1. مسئله مانند یک anomaly detection فرض کرد یا مسئله را یک کلاستر کردن بین راننده خوب و بد در نظر گرفت
2. یافتن یک پارامتر چند متغیره که فرض ما را در چند بعد در بر بگیرد و به ما امکان می دهد رانندگی ناامن را با استفاده از احتمالات و یک توزیع چند متغیره ، تخمین بزنیم.
3. تعداد ابعاد مسئله را با استفاده از روش های مختلف کاهش دهیم (pca اینا) و یک قائده و قانون روی این معیار تعریف کنیم.
4. از یک مدل اماری تک متغییره استفاده کنیم به نحوی که هر دسته از رویداد را به یک توزیع اماری نسبت دهد.
5. تحلیل مسئله به عنوان یک anomaly detection

روش های متفاوتی برای تشخیص دیتا های پرت یا outlier وجود دارد یکی از ساده ترین روش ها LOF است که خروجی آن به صورت 1و -1 می باشد. میتوان از تکنیک های متفاوت تری مانند  [Histogram Based Outlier Score](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.401.5686) (HBOS) که خروجی آن ها یک امتیاز است نیز استفاده نمود برای مثال و سادگی در صورتی که روی دیتای نهایی lof را که خروجی باینری دارد استفاده کنیم به خروجی زیر میرسیم ، نقاط قرمز رانندگان خطرناک هستند.



همانطور که از چارت بالا قابل مشاهده است با این روش Lof رانندگان خیلی خطرناک و خیلی ایمن را میتوان مشخص نمود. اطلاعات خیلی معنا داری نمیدهد. {بررسی شود روش HBOS ایا همین مشکل را دارد یا خیر که مفهوما داره ولی چیز جالبی بنظر میاد به عنوان ابزار}

1. راهکار دوم تخمین احتمال امنیت رانندگی راننده است، به نحوی که یک توزیع چند متغیره بدست بیاریم ، جوری که با میانگین امنیت رانندگی رانندگان محاسبه شده باشد .

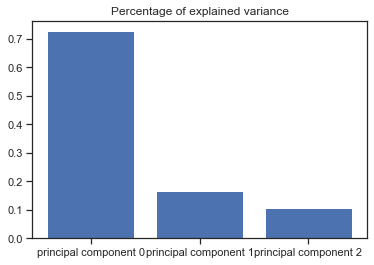
<https://en.wikipedia.org/wiki/Multivariate_kernel_density_estimation>

که همون شکل پیچ پیچی است که تو شکل ها تو مثلث پایین قرار میگرفت. دلیل استفاده نکردن از این روش ، اینه که توزیعی که داخلش استفاده میکنن نرماله ولی اطلاعاتی که داریم نرمال نیست.

1. کاهش ابعاد

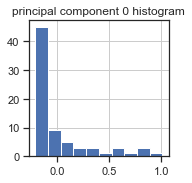
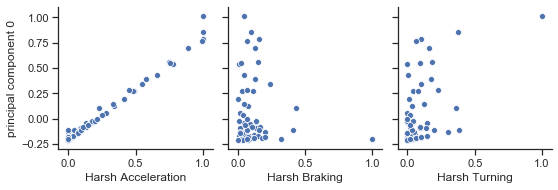
راه کار سوم ، کاهش ابعاد مسئله به یک بعد است ، برای مثال از 3 بعد به یک بعد تبدیل میکنیم تا رفتار راننده را به صورت خلاصه با یک عدد بررسی کنیم.برای این کار میتوان از ابزار های متفاوتی استفاده نمود ولی ساده ترین آنها PCA است که به سادگی میتوان ابعاد مسئله را کاهش داد.

میاد ابعاد مسئله که 3 تاس (سرعت گیری هارش ، دور زدن هارش و ترمز هارش ) رو به 3 تای دیگه مپ میکنه ، بعد ، واریانس هر کدوم از المان های خروجی رو حساب میکنه:



المان صفر رو به عنوان شاخص ارزیابی انتخاب میکنه ، توزیعش رو بررسی میکنه ، البته میگه دلیل انتخابش اینه هم هست که با همه ویژگی ها کلوریشن مثبت داره

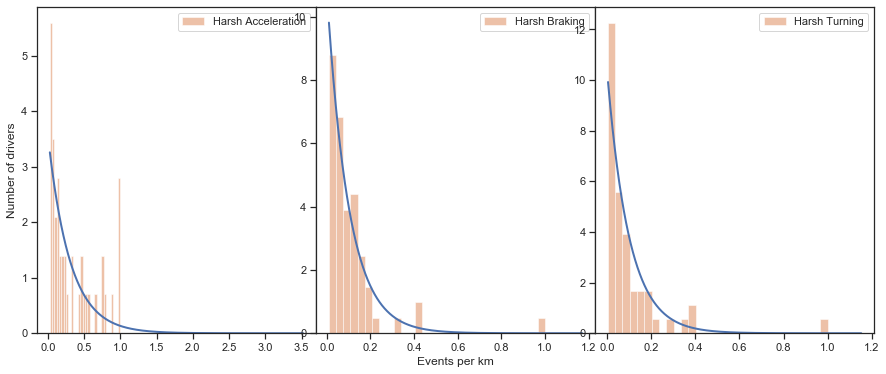
ولی چون واریانس زیادی داره (70 درصد) یه شاخص مناسب برای ارزیابی نیست.کولوریشن مثبت منظور اینه که این ایونتا که زیاد میشن ، این یارو هم زیاد میشه.



1. مدل انتخابی ، مقایسه هر راننده با جمعیت همگن

در پایان ، روشی که انتخاب شد به این نحو است که توزیعی را به هر ویژگی تخصیص داده و سپس هر راننده را با جمعیت همگن میانگین مقایسه میکنیم.

چون اکثر اتفاقات توزیع skewed دارند. ما برای تخصیص یک توزیع از ، توزیع نمایی یا توزیع گاما استفاده کردیم.

توزیع نمایی : 

Harsh Acceleration

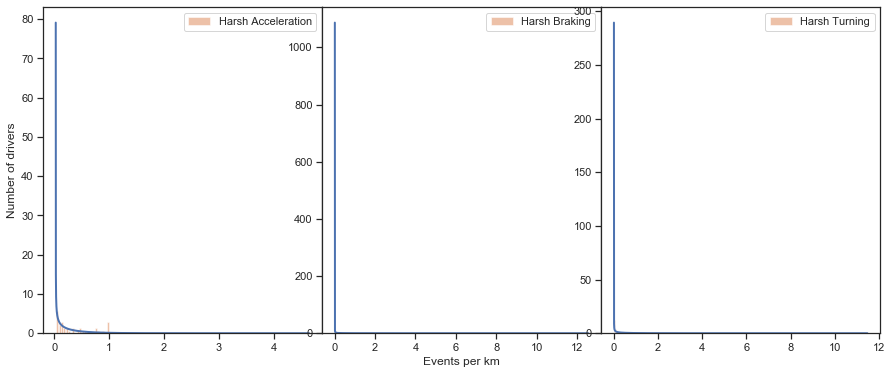
sse: 88.82176421245042

Harsh Braking

sse: 24.71635029453169

Harsh Turning

sse: 98.48049904324613

توزیع گاما : 

Harsh Acceleration

sse: 60.86883139447292

Harsh Braking

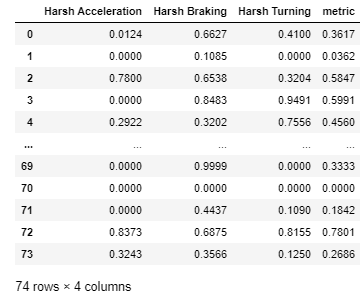
sse: 103.96724777511837

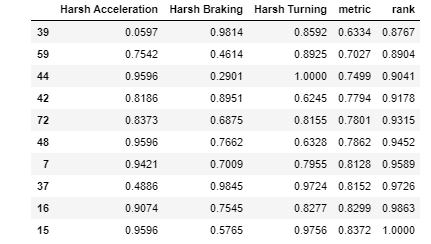
Harsh Turning

sse: 176.62138755403464

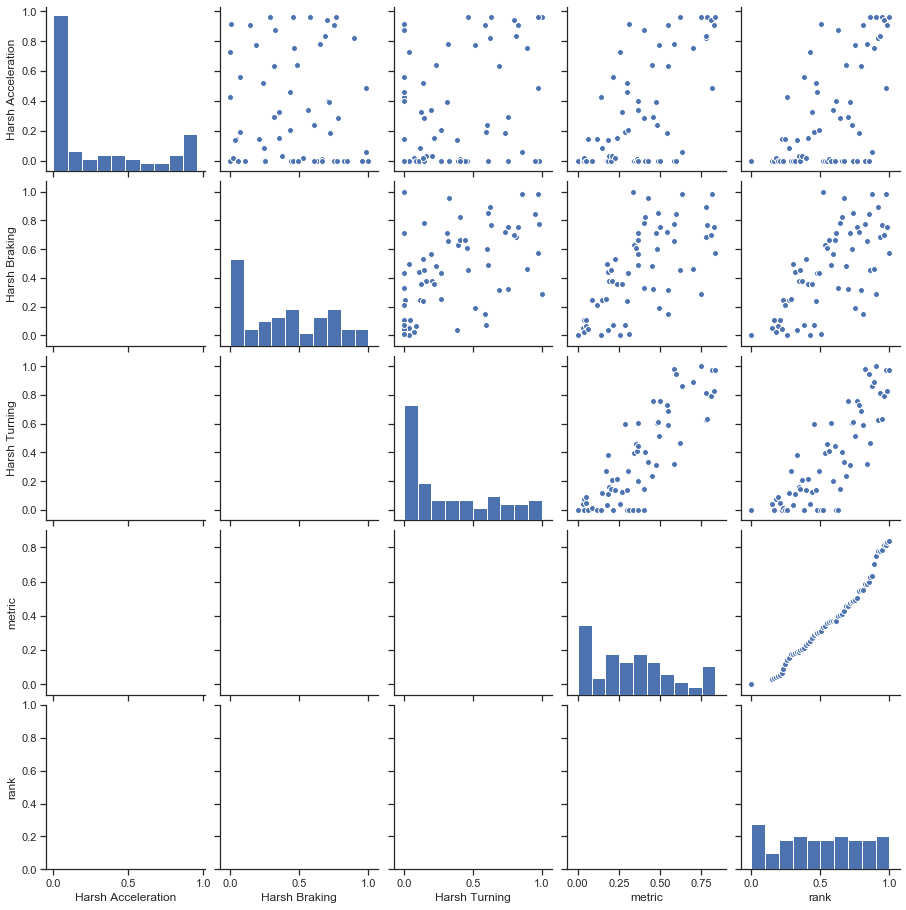
همانطور که مشاهده میشود توزیع نمایی توزیع بهتری است چرا که میزان خطای sse کمتری دارد.

{اینجا اگر بنا به توزیع فیت کردن باشه میشه یه توزیع خوب فیت کرد که هم جنرالیزیشن حفظ شه هم میزان sse رو مینیمم کرد}



حال باید یه متریک یا شاخص به عنوان رانندگی هر فرد در نظر بگیریم ، به عنوان ساده ترین روش ، میانگین احتمال رخ داد در هر کلیومتر را ، با یکدیگر میانگین میگیریم و به عنوان شاخص کل رانندگی هر شخص معرفی میکنیم و رانندگان را رتبه بندی میکنیم و بر تعداد کل تقسیم میکنیم و به عنوان خروجی در نظر میگیریم ، حالا با توجه به اطلاعات موجود یک رگرسیون خطی در نظر میگیریم که به ازای اطلاعات ورودی جدید رتبه ی وی را مشخص کند. {به قولی یک مپینگ بین صفر یک ، به صورت رباست تهیه میکنیم اینجا باید تست و ترین رو جدا کرد و بررسی کرد ولی در مقاله نکرده فک کنم چون دیتا کم هست}خروجی: 

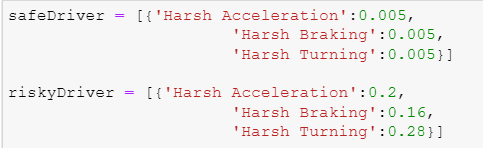
به صورت کلی نمودار توزیع ها را میتوانیم مشاهده کنیم :



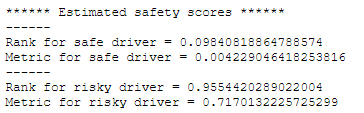
به طور کلی میتوان مشاهده نمود که با افزایش میزان ، احتمال رخ یک اتفاق هاش ، متریک و رنکینکگ بالا میرود.

مثال :

دو راننده داریم که نرخ ایجاد رویداد برا کیلومترشون به شکل زیر است(یعنی بعد از تبدیل boxcox و تقسیم به مسافت طی شده) :



برای این کار ابتدا اطلاعات موجود را با توجه به مقادیر الگوریتم یادگیری ، نرمالیزه میکنیم ، سپس بر اساس توزیع های از پیش محاسبه شده متریک یا شاخص را محاسبه و سپس رتبه یا رنک آن را بدست می آوریم ، خروجی به شکل زیر حاصل شد:



همانطور که مشاهده میشود رنک راننده خطرناک بالاست ، و رنک راننده راننده ایمن ، پایین است.

انتخاب پارامتر های مدل :

در مدل آخر برای ترکیب احتمال میانگین رخ داد ها ، به صورت ساده میانگین گرفتیم ، ولی در عمل برای تشخیص انتخاب پارامتر مناسب ، به یک تحقیق که بر روی میانگین نرخ تصادفات و دلایل آن ها ، منتشر شده است ، رجوع میکنیم. به طور نمونه  [National Motor Vehicle Crash Causation Survey](https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/811059)

این مقاله دلایل تصادف را مشخص و دسته بندی کرده برای نمونه ؛ سرعت گیری شدید ، سرعت غیر مجاز و ...

فرضیات مدل :

فرض اصلی این رویکرد این است که جامعه یک دست و هموژن ، همگن است و پایه استدلال است. به این معنی که هیچ اتفاق غیر عادی باعث تمایز رانندگان نمیشود ، همچنین این رویکرد برای محیط هایی که تعداد رفتار هارش و رانندگان خطرناک آن زیاد است مناسب نیست ، برای همین بجای استفاده از توزیع های از پیش تعریف شده ، بهتر است از توزیع های [goodness of fit statistical tests](https://www.itl.nist.gov/div898/handbook/eda/section3/eda35f.htm), such as the [Anderson-Darling](https://www.itl.nist.gov/div898/handbook/eda/section3/eda35e.htm)test or the [Kolmogorov-Smirnov](https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm) test.

استفاده کنیم ، همچنین تعداد رانندگان در این تحلیل بسیار مهم است.

برای رفع مشکل فرض اول ، می بایست ابتدا رانندگان را یک تقسیم بندی کنیم. و آن ها را در دسته بندی های هموژن و همگن مقایسه کنیم. برای همین ، این شرکت پوینتر که دستگاه کالکتور اطلاعات رو آماده کرده ، یه سری متادیتا برای ما گزاشته مانند نوع ماشین ، کشور ، نوع دستگاه و سنور و... که با کمک این ها سعی میکنیم مقایسه ای منصفانه انجام دهیم.

عملیاتی شدن نتایج : چگونه مسئله را به صورت یک مسئله دادگان انبوه بررسی کنیم:

این شرکت پوینتر به کمک ماکروسافت ، تونسته به عدد 70 ملیون رویداد در روز بررسه ، برای آنالیز رفتار راننده و سایر تخمین ها ، یک پایپ لاین دیتا طراحی کردن که از بخش های مختلفی تشکیل شده :

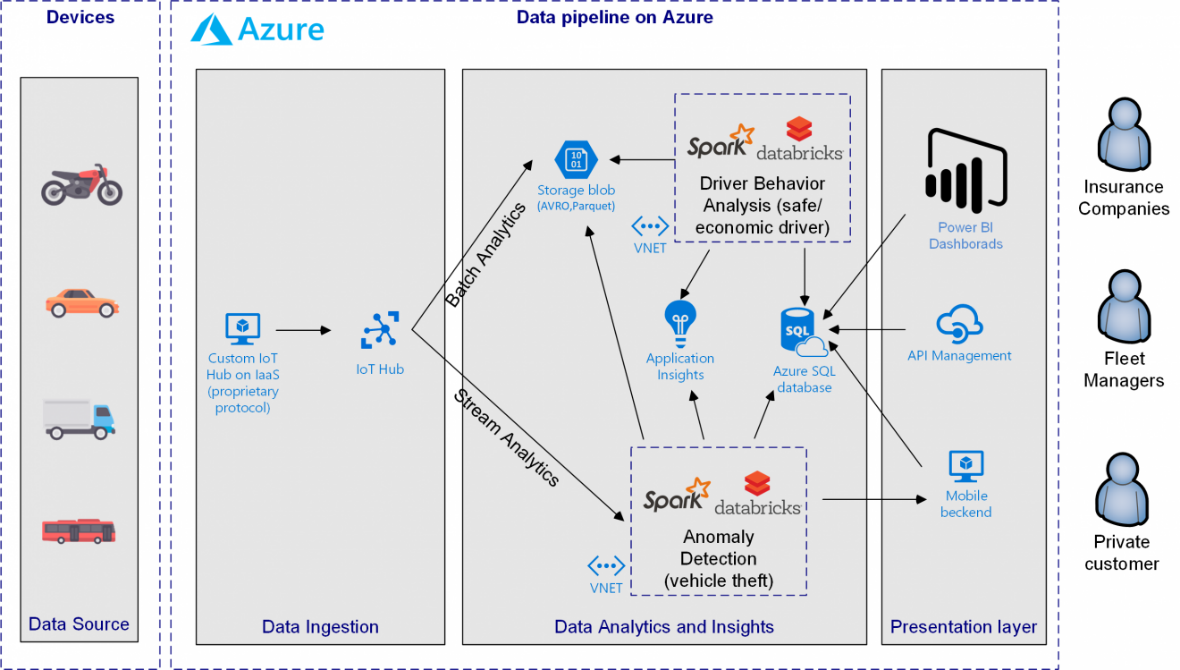
که توضیح میدم {سعی بر اینه اینا رو به صورت لوکال خودم رو رزیری پای پیاده کنم}

[Event Hubs](https://azure.microsoft.com/en-us/services/event-hubs/): A data ingestion tool {مال من میشه Kafka cluster}

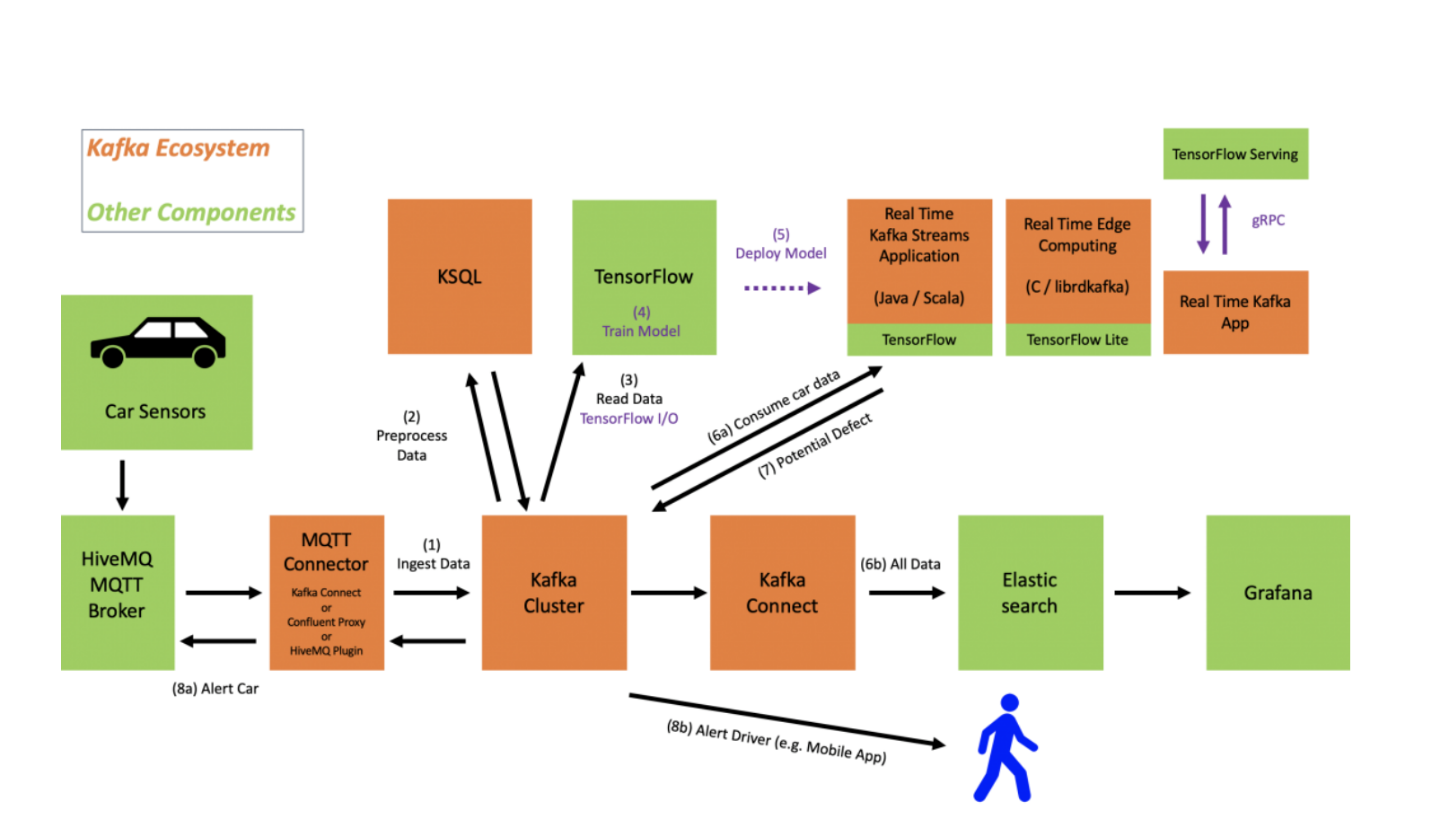
[zure DataBricks](https://azure.microsoft.com/en-us/services/event-hubs/): A managed spark platform which facilitates the creation, management, monitoring and deployment of Spark jobs {مال من میشه مجموعه رزبری هام که با اسپارک به هم وصل شدن}

[Azure Blob Storage](https://azure.microsoft.com/en-us/services/storage/blobs/): For storing raw events before analysis {مال من میشه دیتا بیسی که استفاده میکنم احتمالا KSQL }

[Application Insights](https://azure.microsoft.com/en-us/services/application-insights/): A tool for monitoring jobs in production{برا این یه وب سرور باید بیاد بالا که هندل شه عجله ای نیست }



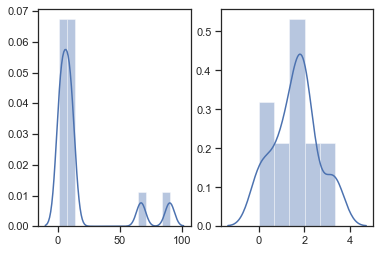
چیزی که این بر بستر سرور های آژور میگه

چیزی که من توذهنم هست

<https://www.youtube.com/watch?time_continue=2&v=7oVSLt0AZ3M&feature=emb_logo>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
حالا بریم رو مقاله هه که داشتیم میگفتیم ، بعد تو یه ریپازیتوری دیگه ، میریم سراغ ،کلودینگ و رزبری­پای

تبدیل box-cox transform

یک تبدیل آماری است که در 1964 توسط دونفر George Box and Sir David Roxbee Cox طراحی شده ، سعی میکند که توزیع داده شده را به توزیع نرمال نزدیک کند به عنوان مثال

توزیع سمت چپ را به توزیع سمت راست تبدیل میکند. روابط ریاضیش هم به شکل زیر است که از طریق پارامتر لاندا این تبدیل قابل بازگشت هست برای بدست آورن لاندای مناسب هم باید یک تابع خطا نوشته و آون رو اپتیمم رو بدست آورد که راه حل بسته دارد و در کتابخانه­­ی scipy.stats.boxcox پیاده سازی شده همچنین تابع معکوس آن نیز در scipy.special.inv\_boxcox قرار دارد.

