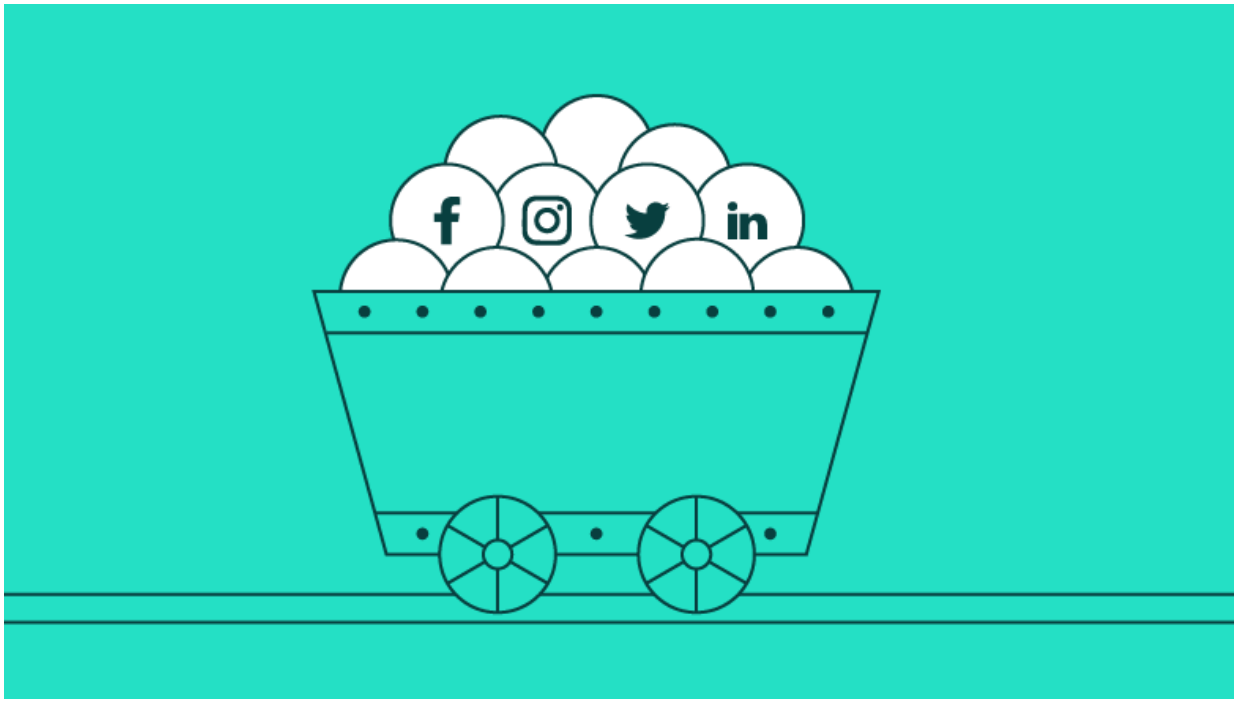
بسم الله الرحمن الرحیم

آنالیز شبکه های اجتماعی با استفاده از روش های داده کاوی

Social Networks Data Mining Techniques & Method



مقدمه

امروزه گسترش شبکه های اجتماعی به حدی شده است که تقریبا در همه امور روزمره خود با آن سروکار داریم.اهمیت شبکه های اجتماعی از آن جهت بالاست که این شبکه ها نمودی از رفتار،علایق ،کنش ها و سلایق جوامع امروزی را مشخص می کنند.از این روست که توجه به مقوله شبکه های اجتماعی و بررسی هرچه بیشتر آن می تواند به درک کلی از سمت و سوی علایق و رفتار یک جامعه و قومیت را پیش روی جامعه شناسان، تحلیلگران و دانشمندان علوم اجتماعی روشن سازد.

شرکت های بزرگ ، دولت ها ، دانشگاهها همه و همه با استفاده از روشهای نوین درصدد شناخت هرچه بیشتر جوامع امروزی هستند.در همین راستا اهمیت شبکه های اجتماعی که محلی برای اجتماع افکار و عقاید مختلف می باشد مورد توجه آنان واقع شده است.شرکتهای بزرگ در راستای اهداف تجاری خود به بررسی دقیق شبکه های اجتماعی می پردازند تا بازخورد ها ، میزان محبوبیت و رویکرد جوامع نسبت به محصولات و سرویس های خود را مورد واکاوی قراردهند.دولت ها نیز با واکاوی مطالب در این شبکه ها به دنبال شناخت بیشتر از جامعه خود و بهبود عملکرد هستند و این به خودی خود اهمیت بالای شبکه های اجتماعی را مشخص می کند.

در این مقاله سعی شده به روش های نوین در جمع آوری و داده کاوی در شبکه های اجتماعی مدرن پرداخته شود.جمع آوری اطلاعات از شبکه های اجتماعی ، دسته بندی ، ترتیب دهی و حرص داده ها ، پردازش و داده نمایی از جمله روش هایی است که داده های نامرتبط در فضای شبکه های اجتماعی را به معیارهایی مهم برای تصمیم گیری و مشاهده مبدل می کند.

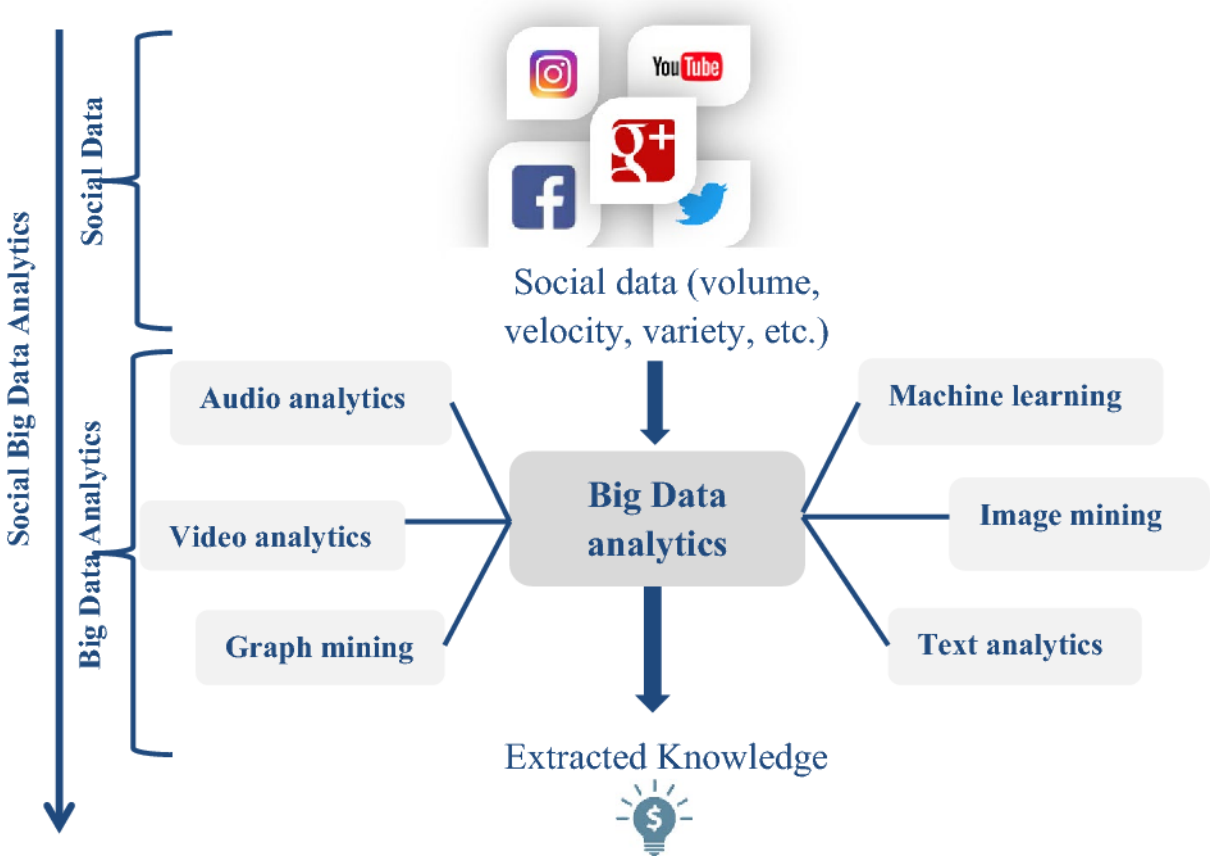
فرهاد کریمی

داده کاوی شبکه های اجتماعی چیست و چه کسانی از آن بهره می برند؟

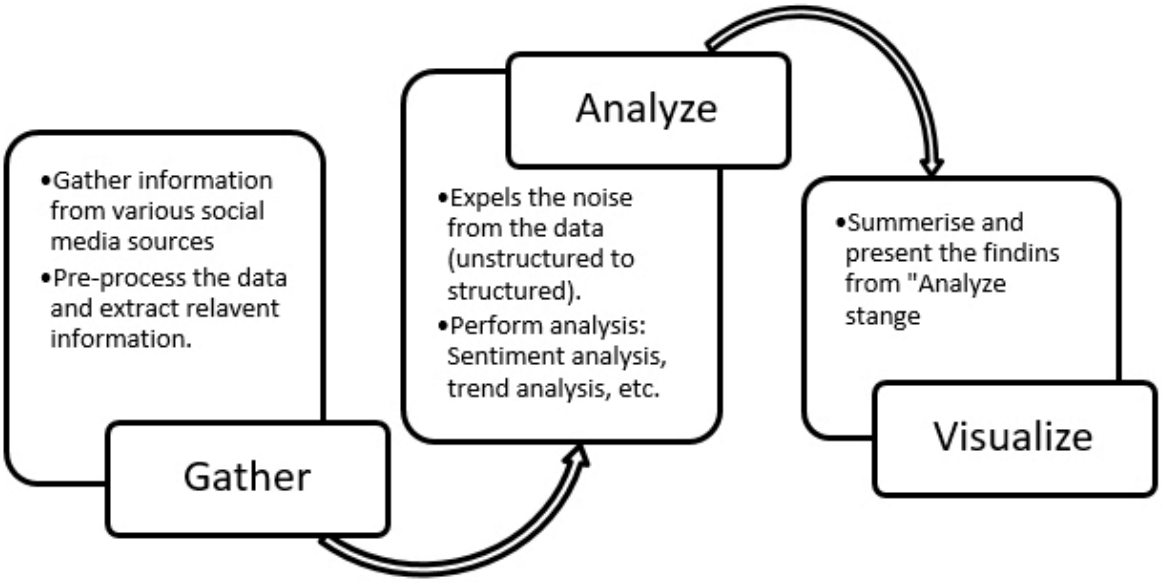
شبکه های اجتماعی امروزه فعالتر از سابق شده اند.بنا بر آخرین محاسبات صورت گرفته در سال 2019 ، جمعیت حاضر در شبکه های اجتماعی مختلف بالغ بر 3 میلیارد نفر هستند.آمارها نشان می دهد در هر ثانیه 11 اکانت جدید ایجاد می شود و این یعنی در زمانی که شما این متن را مطالعه می کنید شاید صدها هزار توییت، ویدئو و محتوا ارسال شده باشد.

این اطلاعات در نهایت یک **بیگ دیتا** را تشکیل می دهند.بیگ دیتا مفهومی است که به تجمع داده های ساختار یافته و غیر ساختار یافته در سطح اینترنت گفته می شود.در اینجا منظور ما از بیگ دیتا ، اطلاعات عظیم قابل جمع آوری از شبکه های اجتماعی است.

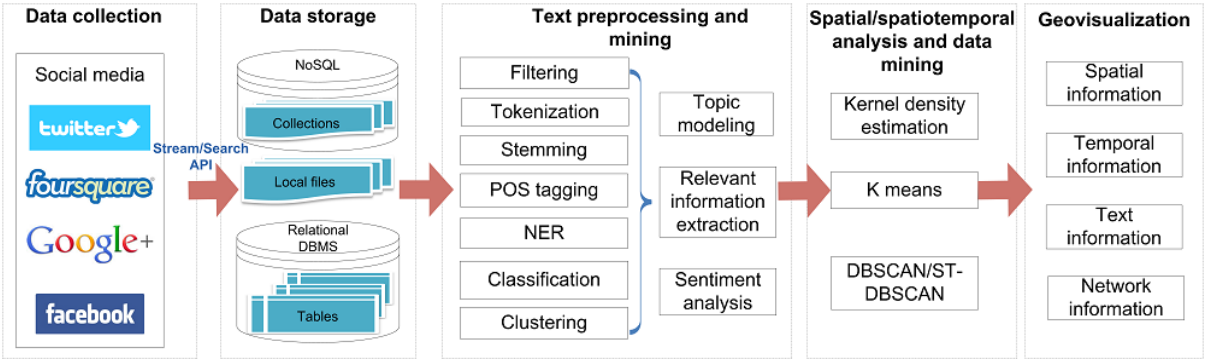
داده کاوی به خودی خود به معنی استخراج اطلاعات از درون یک دیتابیس می باشد.اما در مورد داده کاوی شبکه های اجتماعی بحث کمی متفاوت است.داده کاوی در شبکه های اجتماعی به تکنیک ها و روشهای یافتن الگوهایی خاص و پنهان از میان داده های پنهان در محتوای شبکه های اجتماعی اشاره دارد که لازمه آن به کارگیری هوش مصنوعی ، تکنیک های یادگیری ماشین، ریاضیات و آمار می باشد.



داده های جمع آوری شده از شبکه های اجتماعی طی مراحل مختلف به داده هایی قابل فهم و مفید تبدیل می شوند مثلا در شبکه اجتماعی توییتر داده های جمع آوری شده می تواند مارا از برتری موضوعات روز Trand باخبر کند یا به ما بگوید که کاربر روی کدام مطالب کلیک بیشتری کرده و یا مدلهایی به مراتب پیچیده تر را برای ما معین نماید.در تصویر زیر فرآیند داده کاوی به شکل مختصر توضیح داده شده.



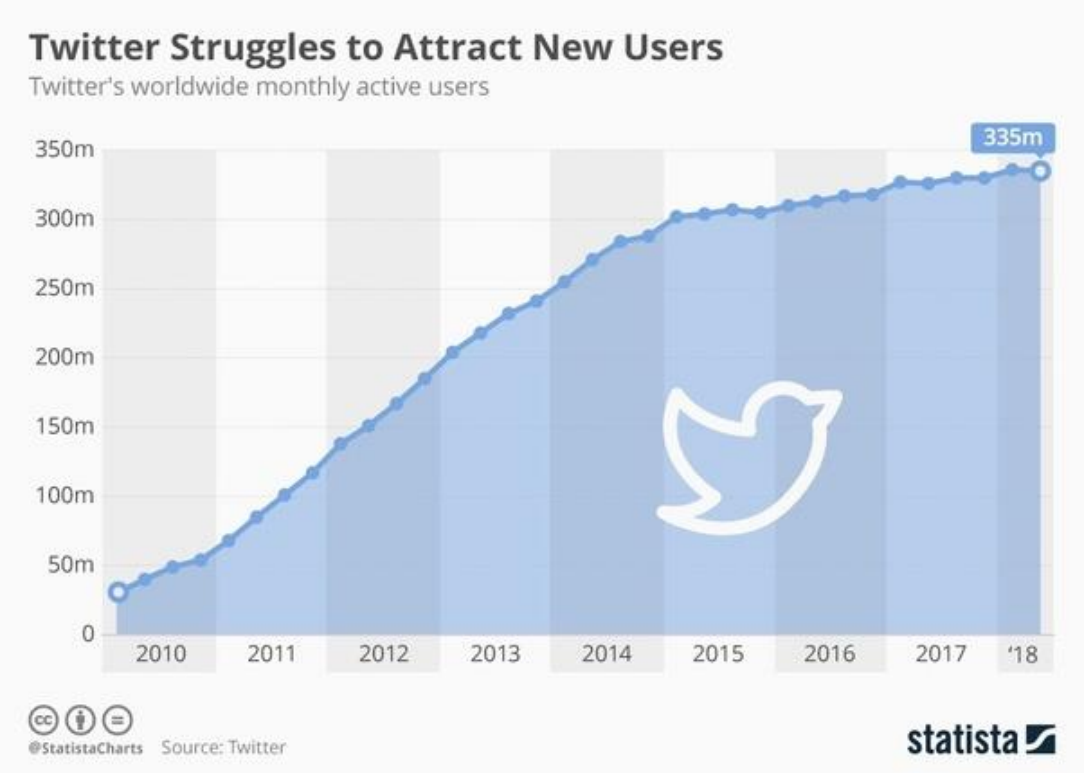
1. مرحله نخست : دریافت داده ها از شبکه های اجتماعی و پیش پردازش، در این مرحله API های ارائه شده از سوی توسعه دهندگان این شبکه ها به کمک ما خواهد آمد.
2. مرحله دوم : حذف و ویرایش ساختاری + آنالیز متون و احساسات در متون ، بررسی Trand های برتر و موضوعات با اقبال عمومی و به عبارتی موضوعات داغ شبکه های اجتماعی
3. مرحله سوم : داده نمایی به واسطه ابزار مختلف جهت ایجاد درک عمیق از داده های دریافتی برای مخاطبین



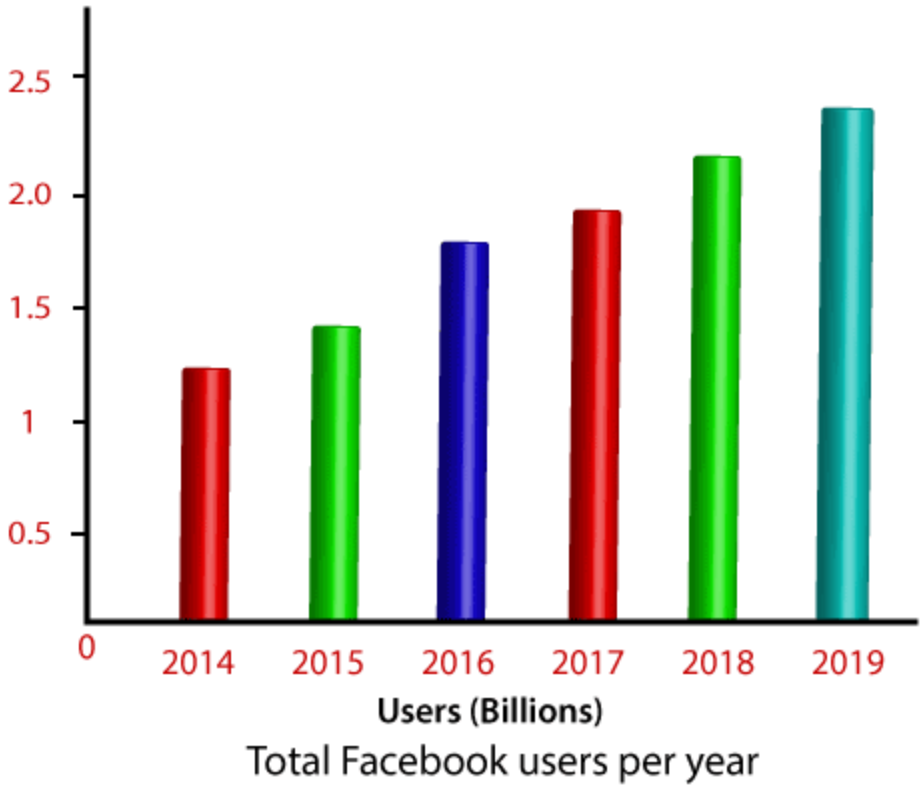
در عکس فوق جزئیات بیشتر و دقیقتر، پیرامون داده کاوی در شبکه های اجتماعی را مشاهده می نمایید که در ادامه به تفصیل به آن می پردازیم.

ضریب نفوذ شبکه های اجتماعی

شبکه های اجتماعی اعم از پیامرسانها و شبکه های اشتراک محتوا و عکس همانند توییتر و اینستاگرام دارای ضریب نفوذ متغییر در کشورهای مختلف دنیا می باشند.و اقبال عمومی نسبت به هر یک از آنها در کشورهای مختلف تقاوت دارد به عنوان مثال پیامرسان اصلی در آمریکا واتس اپ و در ایران تلگرام است.طی سالهای گذشته و با گسترده تر شدن استفاده از تلفن های همراه هوشمند کاربران این شبکه ها به شکل سرسام آوری افزایش یافته است.به عنوان نمونه کاربران توییتر در سال 2010 کمتر از پنجاه میلیون نفر بود که این میزان در سال 2018 به بیش از 335 میلیون نفر افزایش یافته است.



ضریب نفوذ جهانی توییتر



تعداد کاربران فیس بوک از سال 2014 تا 2019

در ادامه قصد داریم روش داده کاوی ، متن کاوی و داده نمایی را با مثالهایی از این شبکه اجتماعی و با استفاده از API آن به شکل قابل درکتری بر روی یک دیتاست توضیح دهیم.

مراحل داده کاوی

مرحله نخست : Data Collecting (جمع آوری اطلاعات)

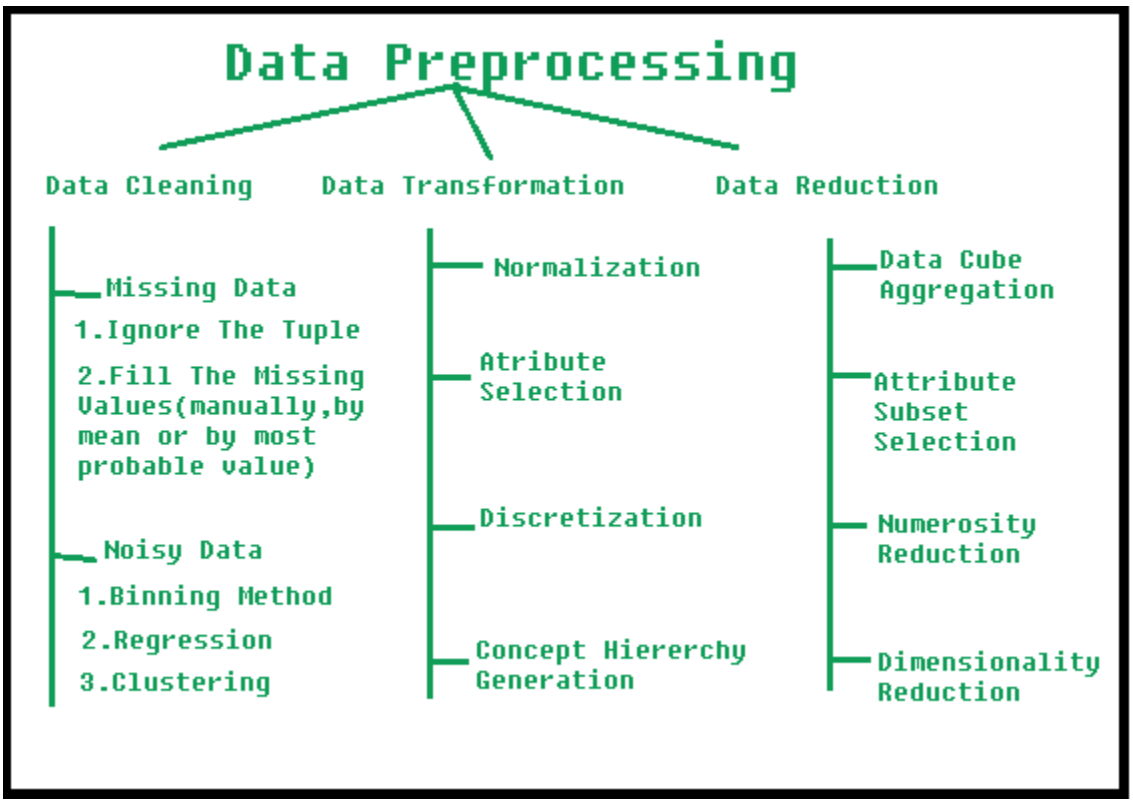
اصلی ترین بخش جمع آوری اطلاعات مورد نظر از مسیر شبکه های اجتماعی هدف است.این بخش به ما کمک می کند تا DataSet خود را شکل دهیم و به واسطه آن نخستین قدم را برداریم.اگر داده کاوری را با استفاد از متد KDD یا Knowledge Discovery in Databases بررسی نماییم.

پس نیاز است که:

- در مرحله نخست اهداف خود را به شکل دقیق معین نماییم: باید مشخص کنیم که نمای کلی کار چیست و چه هدف کلی را دنبال می کنیم و در همین راستا اطلاعات مورد نیاز را استخراج نماییم.
- امکان استفاده از اطلاعات ترکیبی :پس از مشخص شدن هدف کلی در جمع آوری داده ها امکان دارد نیاز به استخراج داده از منابع مختلف باشیم که در نهایت می تواند به شکل ترکیبی دیتاست مورد نظر مارا شکل دهد.

Data Cleaning تمیزسازی دادها(تکنیک های پیش پردازش)

این مرحله پس از جمع آوری اطلاعات Data Gathering انجام میگیرد.در این فرآیند داده های تکراری ، نال، ناسازگار، ناقص و در نهایت نویز ها حذف شده و داده ها در قالبی قابل درک برای موتور آنالیزور داده در خواهند آمد.



برخی از متدهای متداول برای تمیزسازی داده ها:

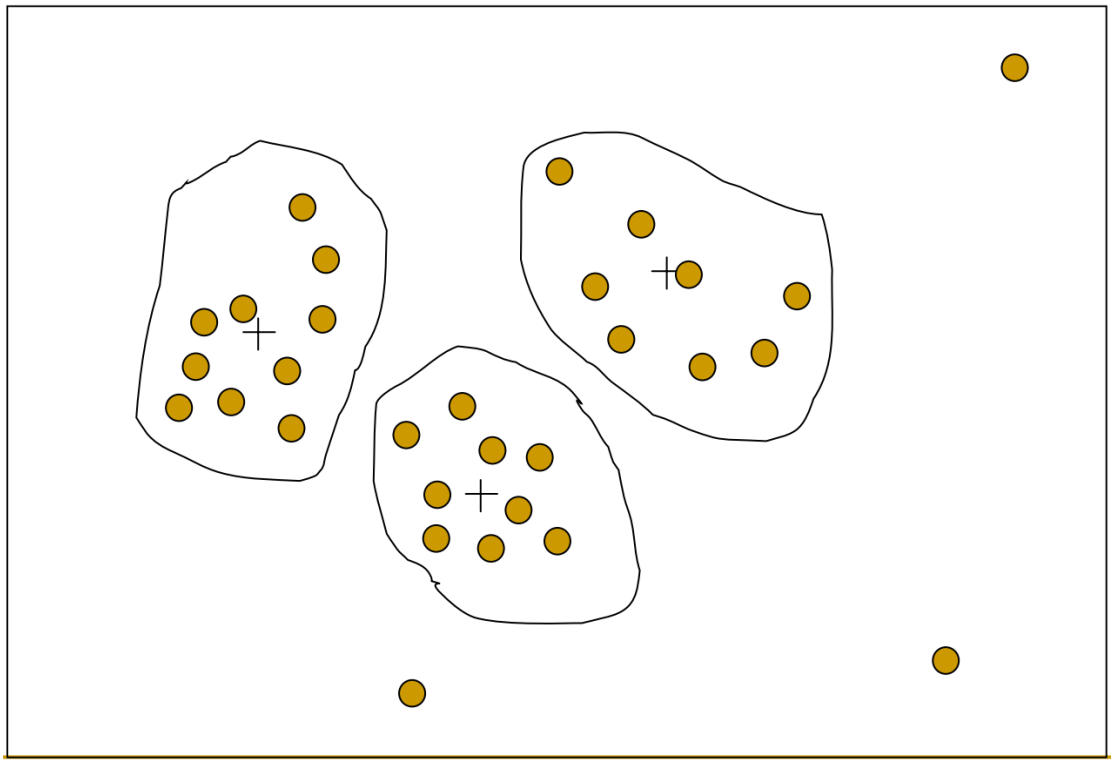
حذف داده های ناقص

1. امکان تمیزسازی دستی داده های ناقص در دیتاست هایی با ابعاد کوچک
2. عدم دریافت داده های Tuple : به دلیل اینکه اینگونه داده ها خود حاوی مجموعه ای از داده های دیگر می باشد.میتوان برای تمیزتر ماندن داده ها از دریافت تاپل ها خودداری نمود.(البته این راهکار متدوال نیست و توصیه نمی شود.)
3. جایگذینی داده های ناقص با مقادیر CONSTANT نظیر Unknown
4. استفاده از موارد "احتمالی" به جای داده های ناقص(در مواردی با استفاده از هوش مصنوعی و یادگیری ماشین می توان نزدیکترین جایگذین احتمالی را قرار داد تا موجب تمیزسازی بیشتر داده ها گردد.)

حذف داده های نویزی Noisy Data

این گروه داده های دریافتی شامل اطلاعاتی می شود که به واسطه مشکلات در دریافت در مرحله Data Collection موجب خطا شده اند.

1. استفاده از روش Bining برای دسته بندی داده ها و استفاده از الگوریتم جایگذینی داده های خالی و یا دریافت نشده با نزدیکترین داده دریافت شده.
2. استفاده از روش های Regression
3. استفاده از خوشه بندی Data Clustering : در این روش خوشه های نزدیک و همگروه جدا شده و داده های خارج از خوشه بندی پرت می شوند.



خوشه بندی داده ها روشی برای نویزگیری از اطلاعات

Data Transformationانتقال داده ها

در این مرحله داده ها با تکنیک های مختلف به فرم مورد نظر برای پردازش آسانتر درخواهند آمد.عمده تکنیک های مرسوم در انتقال داده ها عبارتند از:

1. نرمالسازی : به عنوان مثال برای پردازش توییت های دریافتی برای پردازش مشخص خواهیم کرد که چه بازه ای از توییت ها دریافت شود و در نهایت با حذف شاخص های نامتناسب با پترن های خود اقدام به نرمالسازی در قالب مدل قابل اطمینان خواهیم نمود.
2. تعمیم خصوصیات : در این سطح می توان با جمع برخی خصوصیات و با استفاده از راهکارهای یادگیری ماشین به خصوصیات جدیدی رسید.
3. مفاهیم سلسله مراتبی نسلی : این مفهموم مسئله گسترش یک مفهوم با استفاده از AI برای رسیدن به مفاهیم بالاتر را فرض می داند.به عنوان مثال می توان از بررسی شهرها به کشور رسید.

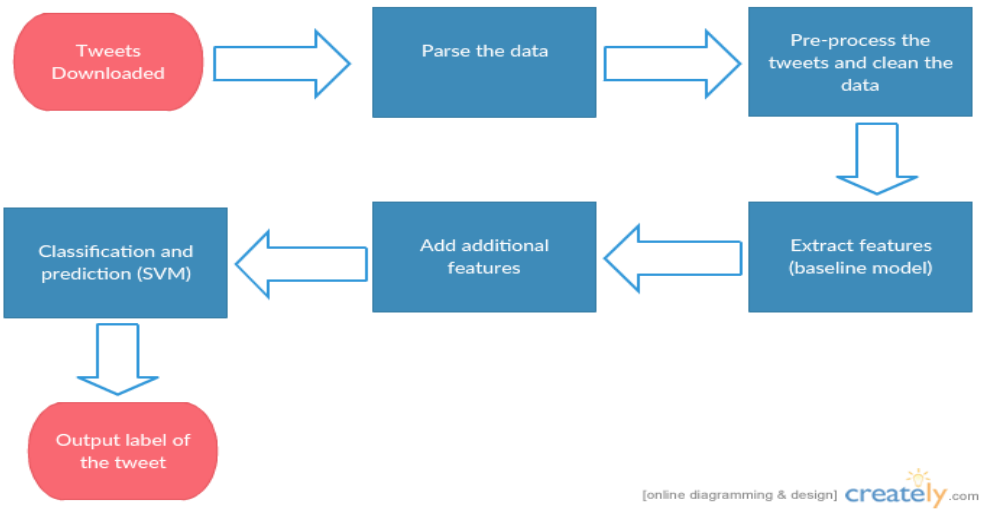
داده کاهی Data Reduction

به دلیل اینکه داده کاوی تکنیکی برای پردازش حجم عظیمی از داده هاست قطعا پردازش این حجم داده در ابعاد بزرگ موجب کاهش کارایی خواهش شد و قطعا منجر به سخت شدن پردازش می شود.تکنیک های داده کاهی برای کاستن از این سختی ها مورد استفاده قرار می گیرند.که شامل تکنیک های زیر می باشد که هر یک به فراخور مورد استفاده قرار می گیرند:

1. تجمیع داده های مکعبی D.C.A
2. بررسی خصوصیات زیر مجموعه ای یا A.S.S
3. کاهش اعداد
4. کاهش ابعاد

پردازش داده ها یا Data Processing

در اصل داده کاوی هنر یافتن الگوهای مخفی از دل داده های حجیم است.این هنر با کمک گرفتن از الگوریتم های AI و تحلیل های آماری حاصل می شود.مرحله پردازش داده در اصل شامل طراحی یک موتور برای کاوش بر روی داده ها و یافتن این الگوها است.به عنوان مثال توییت های استخراج شده بنا بر موضوعی خاص دسته بندی می شوند و سپس برای اعمال برخی ملاخضات باید مورد آنالیز قرار بگیرند.



آنالیز داده ها و داده نمایی Data Analysis and Data Visualization

در فاز پایانی بار استفاده از الگوریتم های آنالیز و با استفاده از ابزای نظیر R و پایتون می توان مکانیزمی را جهت آنالیز داده ها در نظر گرفت.الگوریتم ها بنا بر ماهیت عملکردی و هدف ما می توانند متغییر باشند.

الگوریتم‌های گوناگونی برای تحلیل داده Data Analysis موجود هستند و لذا انتخاب الگوریتم داده‌کاوی Data Mining مناسب یک مساله، برای پژوهشگران و تحلیلگران کاری دشوار است. برخی از سازمان‌ها به دلیل دشواری انتخاب الگوریتم داده کاوی مناسب، به طور مکرر از برخی الگوریتم‌های داده‌کاوی استفاده می‌کنند. هنگامی‌که یک الگوریتم نامناسب پیاده‌سازی می‌شود، دانش کشف شده اغلب برای سازمان بدون معنا است زیرا از اطلاعات صحیحی پرده‌برداری نکرده و این امر می‌تواند منجر به تصمیم‌گیری‌های غلط در کسب‌وکار شود.

انتخاب الگوریتم صحیح درآنالیز بسیار حیاتی است!

البته باید دانست که راهنمای مشخصی برای آنکه پژوهشگران یا تحلیلگران چگونه الگوریتم انتخاب کنند وجود ندارد. انتخاب الگوریتم مناسب را یکی از چالش‌های موجود برای اغلب پژوهشگران حوزه داده‌کاوی بیان می‌کنند. انتخاب یک الگوریتم مشخص امری بسیار پیچیده است، لذا برخی از پژوهشگران برای ارتقای نتایج داده‌کاوی از چندین الگوریتم استفاده کرده و پردازش‌ها را با الگوریتم‌های مختلف تکرار می‌کنند.

داده‌کاوان ممکن است از الگوریتمی که نتایج صحیح را تولید می‌کند، سریع است و کد آن به خوبی مستند و شفاف‌سازی شده استفاده کنند.در صورت امکان بهتر است پیش از انجام پردازش روی مجموعه داده‌های حقیقی، الگوریتم را روی یک مجموعه داده ورودی به‌صورت آزمایشی پیاده کنند تا عملکرد الگوریتم برای حل یک نوع مساله خاص سنجیده شود.گاه نیاز به استفاده از چندین الگوریتم برای حل یک مساله واحد جهت حل فازهای مختلف مساله است.

لازم به ذکر است برای انتخاب یک الگوریتم داده‌کاوی مناسب، ممکن است نیاز باشد پژهشگران بیش از یک مولفه را در نظر بگیرند. انتخاب الگوریتم با توجه به تنها یکی از مولفه‌ها، می‌تواند امکان بازدهی نامطلوب نتایج را افزایش داده و حتی خروجی فرآیند داده‌کاوی را بی‌معنی کند.

ساختار داده: ساختار مجموعه داده مولفه دیگری است که در تعیین الگوریتم مورد استفاده نقش مهمی دارد. ارتباط بین اشیا/داده‌ها، ارتباط بین متغیرها و روشی که داده‌ها بر اساس آن ذخیره شده‌اند، انتخاب الگوریتم مناسب برای داده‌کاوی را تحت تاثیر قرار می‌دهد. برخی الگوریتم‌ها ویژه مجموعه داده‌هایی با انواع مشخص هستند و لذا سازمان‌ها را مجبور به داشتن ابزارهای مختلف برای اهداف گوناگون می‌کنند.

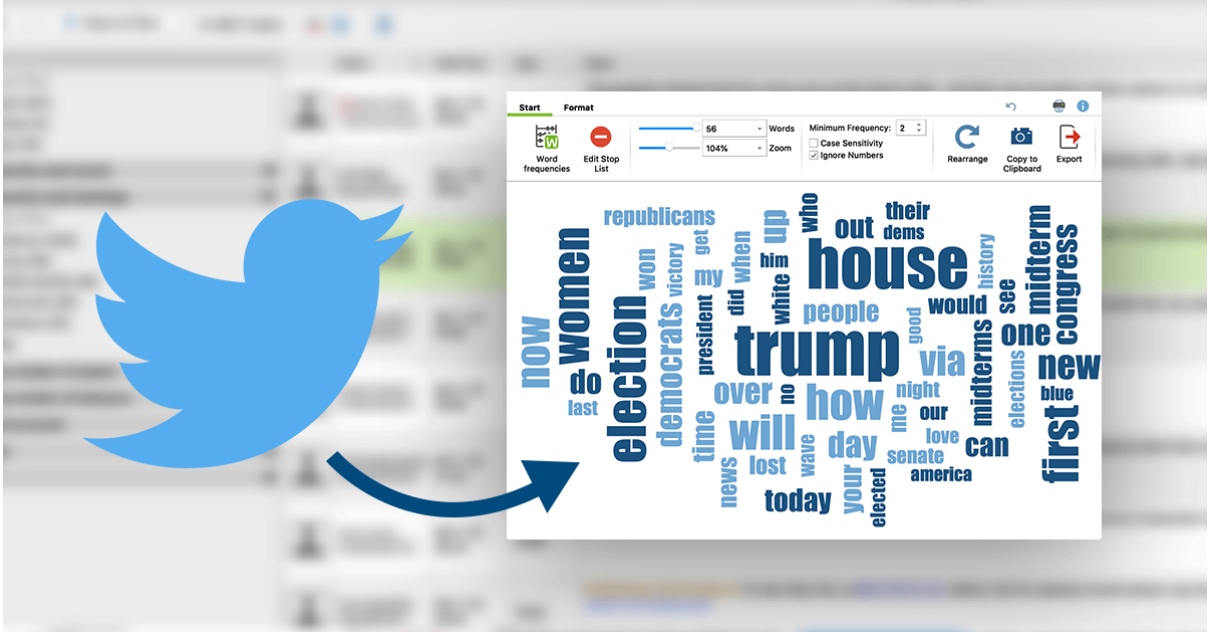
نتایج مورد انتظار: هر فرآیند داده‌کاوی باید یک خروجی به عنوان راه حل مساله داشته باشد که در واقع نتیجه مورد انتظار مساله است. هدف اصلی داده‌کاوی شناسایی الگوها و گرایش‌ها در داده‌ها است تا از این دانش در تصمیم‌گیری‌ها استفاده شود. بسته به‌نوع نتایج مورد انتظار، داده‌کاوان الگوریتمی را انتخاب می‌کنند که قادر به تولید آن نتایج است.

وظایف قابل انجام: وظایفی که ضمن فرآیند داده‌کاوی انجام می‌شوند انتخاب الگوریتم را تحت تاثیر قرار می‌دهند. برای مثال اگر وظیفه قابل انجام، دسته‌بندی باشد، الگوریتم‌ها احتمالا از روش‌های دسته‌بندی (Classification)یا رگرسیون (Regression) خواهند بود. هنگامی که وظیفه استخراج قواعد وابستگی (Association Rules)است، امکان دارد الگوریتم آپریوری (Apriori) مورد استفاده قرار بگیرد.در پروژه داده کاوی توییتر از الگوریتم های رگرسیون و خوشه بندی یا دسته بندی استفاده می شود.

نتایج داده‌کاوی تصمیم‌گیری‌های کسب‌وکار را آگاهانه می‌سازد، بنابراین اگر اطلاعاتی که ضمن فرآیند داده‌کاوی کشف شده اشتباه باشند ممکن است موجب اتخاذ تصمیمات اشتباه توسط تصمیم‌سازان شود. اطلاعات برای آنکه مفید باشند باید با ساختار درستی ارائه شوند.

پیاده سازی یک سیستم داده کاوی شبکه های اجتماعی به شکل عملی

در ادامه به روش پیاده سازی عملی یک الگو برای داده کاوی در شبکه اجتماعی توییتر و نحوه آنالیز و داده نمایی آن اشاره خواهد شد.سعی ما این است تا با ارائه راهکاری عملی به بررسی هرچه بیشتر ابعاد و شیوه کار با API توییتر برای داده کاوی و آنالیز این شبکه اجتماعی بپردازیم.



بخش اول:جمع آوری اطلاعات و ساخت Dataset

در این قسمت به روشهای جمع آوری اطلاعات و ساخت Collection و Dataset مورد نظر با استفاده از کتابخانه پایتونی Tweepy می پردازیم.

در قدم اول برای دسترسی به API توییتر باید یک اپ برو روی سایت توییتر بسازیم.جهت ساخت اپ به آدرس : https://apps.tweeter.com مراجعه میکنیم. توجه داشته باشید که پیش از ساخت اپ باید در توییتر عضو شده و لاگین کرده باشید.

سپس توضیحاتی درباره اپ خود را نوشته و آن را نامگذاری میکنید.مثلا myminingapp.پس از آن دو توکن رشته ای تحت نامهای consumer key و consumer secret دریافت خواهید کرد و همینطور ACCESS TOKEN و ACCESS TOKEN SECRET که از قسمت Configuration اپ شما قابل دسترسی خواهند بود.این 4 رشته مجوز استفاده شما از سرویس توییتر خواهد بود.و باید به صورت امن

توسط شما نگهداری شود.در ابتدا دسترسی شما به توییتر با سطح دسترسی read only می باشد و در صورتی که بخواهید توانایی نوشتن در توییتر را نیز بیابید نیاز به ایجاد دسترسی سطح مورد نظر در سایت توییتر خواهید بود.

توییتر محدودیت هایی را هم اعمال کرده که از آنجمله می توان به محدودیت در دانلود Dataset اشاره کرد.در لینکهای زیر شرح کاملی از این محدودیت ها آورده شده:

https://dev.twitter.com/overview/terms/agreement-and-policy

https://dev.twitter.com/rest/public/rate-limiting

توییتر REST API مناسبی فراهم کرده تا به راحتی بتوان از سرویس های آن بهره مند شد.همینطور در سمت کلاینت هم کتابخانه های پایتونی فراوانی ایجاد شده که به راحتی و بدون اینکه نیاز باشد دست به کد شوید و از نو بنویسید؛ می توانید از آنها استفاده کنید.از جمله این کتابخانه ها و شاید یکی از بهترین کتابخانه ها برای این منظور کتابخانه Tweepy می باشد که ما از آن استفاده خواهیم کرد.

pip install tweepy==3.3.0 برای نصب Tweepy از طریق ترمینال یا cmd دستورزیر را وارد کنید تا با استفاده از pip کتابخانه Tweepy نصب 3.3.0 به ورژن مورد نظر این مقاله اشاره دارد.درصورتی که ورژن درج نشود.آخرین ورژن نصب خواهد شد.لازم به ذکر است که Tweepy برای هر دو ورژن 2 و3 پایتون قابل دسترس می باشد.با توجه به اینکه نسخه 3 پایتون دارای ویژگی های جدیدی است و میتوان گفت تغییرات فراوانی نسبت به نسخه 2 داشته، توسعه دهندگان این کتابخانه آن را برای این ورژن نیز بازنویسی کرده اند.

با استفاده از کد پایتون زیر و 4 رشته ای که دریافت کردیم به REST Service توییتر متصل می شویم.

```python
import tweepy
from tweepy import OAuthHandler
 consumer_key = 'YOUR-CONSUMER-KEY'
consumer_secret = 'YOUR-CONSUMER-SECRET'
access_token = 'YOUR-ACCESS-TOKEN'
access_secret = 'YOUR-ACCESS-SECRET'
 auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(محل وارد کردن اکسس توکن دریافتی از توییتر)
api = tweepy.API(auth)

```

این اتصال از طریق پرتکل OAuth صورت میگیرد. OAuth درواقع پرتکلی برای برقرای ارتباط با استفاده از احراز هویت بر روی بستر وب ؛ موبایل و... می باشد.این پرتکل مجوز(Authorization) را مورد بررسی قرار نمیدهد و تنها در سطح احراز هویت کاربردیست.

متغییر api در تکه کد فوق.بخش اصلی ارتباطی ما با توییترهست.مثلا تصور کنید که می خواهیم 10 تا توییت های تایملاین خود را بیرون بکشیم.کافیست اینطور عمل کنیم:

```
for status in tweepy.Cursor(api.home_timeline).items(10):
    # Process a single status
    print(status.text)
```

interface کتابخانه ما Cursor به صورت تکرار شونده و با محدودیت 10 تا از توییت های مارا درون status میریزد و به صورت text به خروجی میفرستد.حتی اگر بخواهیم اطلاعات ما به صورت json نمایش داده شود. کافی است از متد json_ استفاده کنیم.

```
for status in tweepy.Cursor(api.home_timeline).items(10):
    # Process a single status
    process_or_store(status._json)
```

تصور کنید بخواهیم لیست تمامی follower (دنبال کننده ها) را بدست بیاوریم.کافیست به این شکل عمل کنیم:

```
for friend in tweepy.Cursor(api.friends).items():
```

```
    process_or_store(friend._json)
```

کد بالا ؛ تمامی دنبال کنندگان اکانت مارا به شکل فرمت json در خروجی نمایش خواهد داد.

درصورتی که بخواهیم لیست تمامی توییت هایمان را نمایش دهد :

```
for tweet in tweepy.Cursor(api.user_timeline).items():
    process_or_store(tweet._json)
```

به راحتی میتوان اطلاعات را که به شـــکل json در آمده با تعامل با یک پایگاه داده NoSQL ذخیره و دیتاست خود را تامین کنیم.

توجه داشـــته باشـــید که: تابع process_or_store() تنها یک تابع برای پرینت خروجی در قالب jsonهست و کاربر می تواند بنا به نیاز طرح کار و یا نمایش داده هارا پیاده سازی کند.

ساده ترین شکل این پیاده سازی نمایش در خروجی با print() هست که مشاهده می کنید.

```
def process_or_store(tweet):
    print(json.dumps(tweet))
```

استریم داده ها Data Streaming

Streaming بیانگر این اسـت که ما می خواهیم همواره به جدیدترین توییت های ارسـال شـده توسـط کاربر به صـورت آنی دسـترسی داشـته باشـیم.پس نیاز به یک StreamListener() هسـت که مدام گوش دهد که آیا توییت جدید ارسال شده و اگر شده آن را واکشی.کرده و عملیات مورد نظر ما روی آن صورت بپذیرد.به تکیه کد زیر توجه کنید :

```
mysite > blog > views.py > ...
from tweepy import Stream
from tweepy.streaming import StreamListener

class MyListener(StreamListener):

    def on_data(self, data):
        try:
            with open('python.json', 'a') as f:
                f.write(data)
                return True
        except BaseException as e:
            print("Error on_data: %s" % str(e))
        return True

    def on_error(self, status):
        print(status)
        return True

twitter_stream = Stream(auth, MyListener())
twitter_stream.filter(track=['#python'])

```

تکه کد فوق با استفاده از streaming تمامی توییت هایی که دارای هشتگ python باشند را به صورت بدون وقفه و realtime دریافت میکند و درون یک فایل json ذخیره می کند.در صـــورتی که این کد را اجرا کنید به مرور زمان با دیتابیس عظیمی روبرو خواهید شـــد که شـــامل توییت های فراروانی با فیلتر #python می باشد. و لحظه به لحظه در حال بزرگتر شدن هست.

در صورتی که علاقه داشته باشید میتوانید با مثال های بیشتری از streaming در اکانت gist زیر آشنا شوید:

https://gist.github.com/bonzanini/af0463b927433c73784d

بخش دوم : آنالیز متون و توییت ها

در بخش قبل به اســـتخراج توییت ها پرداختیم و در این بخش به آنالیز محتوای توییت ها خواهیم پرداخت.تصـورکنید که با اسـتفاده از روشـهای ذکر شـده قبلی فایلی متشـکل از توییت ها با فرمت json ایجاد شده.حالا برای نمایش محتویات فایل به شکل زیر عمل میکنیم:

```
1  import json
2
3  with open('mytweets.json', 'r') as f:
4      line = f.readline() # read only the first tweet/line
5      tweet = json.loads(line) # load it as Python dict
6      print(json.dumps(tweet, indent=4)) # pretty-print
7
```

تکه کد فوق با اســتفاده از کتابخانه json محتوای فایل mytweets.json که شــامل مجموعه توییت های بدست آمده است را بازگشایی و نمایش می دهد.

برخی از ویژگی های کلیدی توییت که می توان از آنها استفاده کرد:

نام ویژگی	عملکرد
Text	متن توییت
Create_at	تاریخ ایجاد توییت
Favorit_Count , retweet_count	تعداد لایک و ریتوییت ها
Favorited, retweeted	مقدار Boolean که مشـــخص میکند کاربر احراز هویت شـــده این توییت را ریتوییت یا لایک کرده یا نه
Lang	نمایش زبان توییت en=English
Id	Id یکتا توییت
Place, coordinate,geo	مختصات جغرافیایی در صورت وجود
User	پروفایل کامل نویسنده توییت

Entities	لیستی از موارد شامل : هشتگ ها – url -منشن ها و...
In_reply_to_user_id	Id کاربری که توییت در پاسخ به توییت اون ارسال شده است
In_reply_to_status_id	Id توییتی کع این توییت در پاسخ به آن ارسال شده است

ویژگی های فوق دارای خروجی نوع string هم هستند برای این کار می توان از جایگذین آنها مثل *_id_str استفاده کرد که خروجی را به شکل رشته ارسال می کند.اطلاعات فوق برای آنالیز بسیار کاربردی هستند.با استفاده از این ویژگی ها میتوان به هشتگ های پرطرفدار .پر ریتوییت ترین توییت ها(توییت ها با بازنشر- بالا) و محبوبترین توییت ها پی برد.همینطور بسیاری از موارد دیگر که در آنالیز به آنها نیاز خواهیم داشت.

قدم بعدی برای شروع آنالیز برسی text که همان محتوای توییت می باشد است.برای این کار نیاز به این است که متن توییت به قسمت های کوچکتر تقسیم شود که به هر قسمت توکن گفته می شود.این عمل را tokenise کردن یا توکن گذاری می نامند.

چطور متن توییت را توکن بندی کنیم؟

برای این کار از کتابخانه NLTK استفاده می کنیم.که کتابخانه ای برای توکن بندی کردن یک رشته می باشد.

```
1  from nltk.tokenize import word_tokenize
2  tweet = 'RT @marcobonzanini: just an example! :D http://example.com #NLP'
3  print(word_tokenize(tweet))
4
```

خروجی توییت ارسالی به تابع به شکل زیر است:

```
# ['RT', '@', 'marcobonzanini', ':', 'just', 'an', 'example', '!', ':', 'D', 'http', ':',
'//example.com', '#', 'NLP']
```

یک لیست با توکن های جدا شده.

البته کتابخانه NLTK برای زبان انگلیسی تعبیه شده و برخی مشکلات را ایجاد میکند به عنوان مثال یک هشتگ را از متن هشتگ جدا میکند.همینطور منشنها را و هر یک از علائم # و @ را جداگانه به عنوان یک توکن در نظر میگیرد که اشتباه است.لذا باید تغییراتی در کد اعمال شود که مثلا #NLP یک توکن مجزا باشد.

برای همین از کد زیر استفاده میکنیم که با استفاده از عبارات با قاعده این کار را راحتتر می کند:

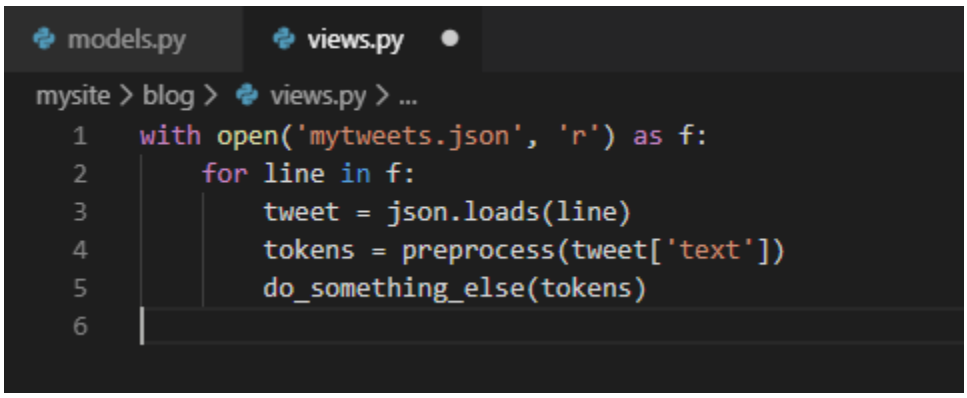
```
mysite > blog > views.py > ...
import re
emoticons_str = r"""
    (?:
        [:=;] # Eyes
        [oO\-]? # Nose (optional)
        [D\)\]\(\]/\\OpP] # Mouth
    )"""

regex_str = [
    emoticons_str,
    r'<[^>]+>', # HTML tags
    r'(?:@[\w_]+)', # @-mentions
    r"(?:\#+[\w_]+[\w\'_\-]*[\w_]+)", # hash-tags
    r'http[s]?://(?:[a-z]|[0-9]|[$-_@.&amp;+]|[!*\(\),]|(?:%[0-9a-f][0-9a-f]))+', # URLs

    r'(?:(?:\d+,?)+(?:\.?\d+)?)', # numbers
    r"(?:[a-z][a-z'\-_]+[a-z])", # words with - and '
    r'(?:[\w_]+)', # other words
    r'(?:\S)' # anything else
]

tokens_re = re.compile(r'('+'|'.join(regex_str)+')', re.VERBOSE | re.IGNORECASE)
emoticon_re = re.compile(r'^'+emoticons_str+'$', re.VERBOSE | re.IGNORECASE)

def tokenize(s):
    return tokens_re.findall(s)

def preprocess(s, lowercase=False):
    tokens = tokenize(s)
    if lowercase:
        tokens = [token if emoticon_re.search(token) else token.lower() for token in tokens]
    return tokens

⊗0 ⚠0
```

روش فوق با به کار گیری عبارات باقاعده regex توکن بندی را تمیزتر انجام خواهد داد و ما در خروجی این لیست را خواهیم داشت :

['RT', '@marcobonzanini', ':', 'just', 'an', 'example', '!', ':D', 'http://example.com', '#NLP']

همانطور که میبینید هر یک از @MARCOBONXANINI و #NLP به یک توکن سرهم تبدیل شدند.

اگر بخواهیم این پروسه روی تمامی توییت های ذخیره شده ما اعمال شود به شکل زیر عمل میکنیم:



روش استفاده از ریجکس برای توکن بندی روشی است که برخی اطلاعات اعم از شماره تلفن و.. را از قلم می اندازد برای بهبود استفاده از روش عبارات با قاعده نیاز به توسعه این روش و روی آوردن به مبحث Named Entity Recognation هست.دلیل اینکه نمیتوان به صورت قطعی اعداد و ارقامی مثل تلفن یا مبالغ را به صورت توکن در آورد ساده است.چون در کشورهای مختلف به شکل های مختلفی نگاشته می شوند.مثلا ارقام به شکل 1000 یا 1.000 و... نوشته می شوند که کار ما در تولید پترن مناسب برای اعمال روی regex سخت می شود.

آنالیز توکن ها (پرکاربردترین کلمات)

در صورتی که بخواهیم در آنالیزی سمت و سوی توییت های ذخیره شده را برسی کنیم.مثلا برسی کنیم که چه توکنهایی چندبار تکرار شدند و به این وسیله پر تکرار ترین کلمات را شناسایی کنیم.یعنی با برسی توکن های پر کاربرد به موضوع Trend شده در شبکه توییتر پی ببریم.

در کد زیر با توییت های ذخیره شده را برسی کرده و با استفاده از تابع کتابخانه collection و counter به برسی تعداد بیشترین ()most_common توکن های تکرار شده در کل توییت های ذخیره شده می پردازیم.

به تکه کد زیر توجه کنید:

خروجی این تکه کد چیزی نیست که مورد قبول باشد.چون حرف ربط را که معمولا در زبان نوشتار بیشترین تکرار را دارند کاندیدا کرده :

```
[(':', 44), ('rt', 26), ('to', 26), ('and', 25), ('on', 22)]
```

برای حل این مشکل و استثنا کردن حروف ربط باید چاره ای اندیشیده شود.

حذف حروف ربط

حروف ربط در تمام زبانها وجود دارد ولی در زبان انگلیسی پایتون و کتابخانه nltk یک لیست وجود دارد که به واسطه آن میتوان حروف ربط انگلیسی را از رشته مورد نظر حذف کرد.به تکه کد زیر توجه کنید:

models.py views.py

mysite > blog > views.py > ...

```python
import operator
import json
from collections import Counter

fname = 'mytweets.json'
with open(fname, 'r') as f:
    count_all = Counter()
    for line in f:
        tweet = json.loads(line)
        # Create a list with all the terms
        terms_all = [term for term in preprocess(tweet['text'])]
        # Update the counter
        count_all.update(terms_all)
    # Print the first 5 most frequent words
    print(count_all.most_common(5))

```

models.py views.py

mysite > blog > views.py > ...

```python
from nltk.corpus import stopwords
import string

punctuation = list(string.punctuation)
stop = stopwords.words('english') + punctuation + ['rt', 'via']

```

حالا برای عملیاتی شدن تکه کد مثال قبلی میتوانیم تکه کد زیر را جایگذین کنیم که تعداد پر کاربرد ترین کلمات واقعی را با حذف حروف ربط نمایش دهد:

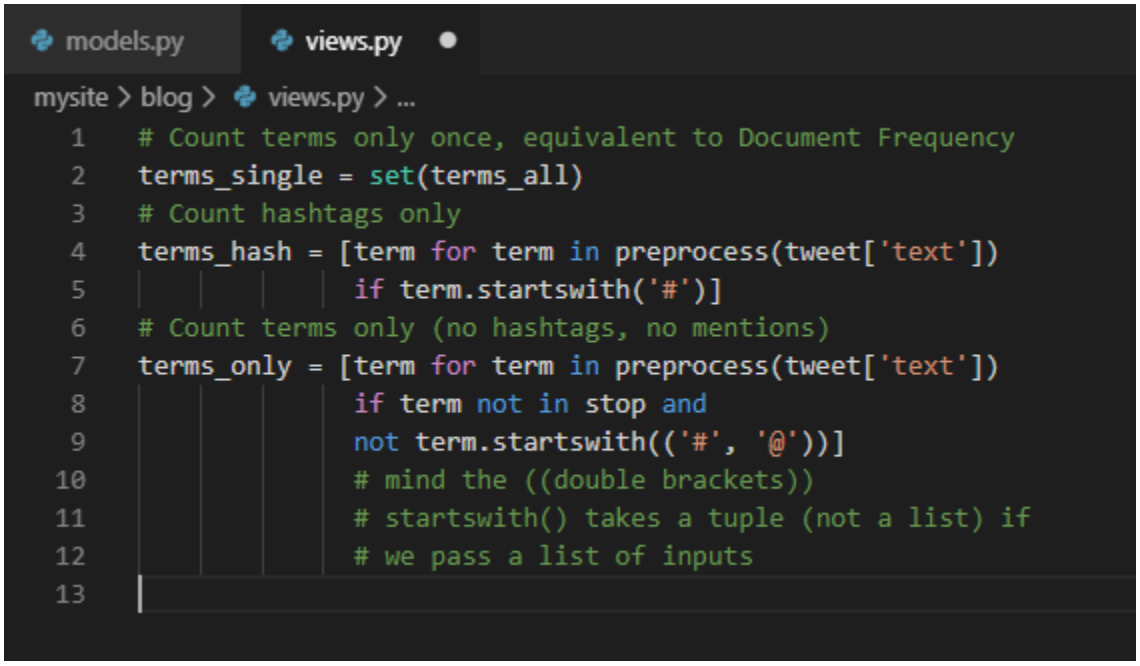
```
terms_stop = [term for term in preprocess(tweet['text']) if term not in stop]
```

خروجی به شکل زیر خواهد بود:

```
[('python', 11), ('@miguelmalvarez', 9), ('#python', 9), ('data', 8),
('@danielasfregola', 7)]
```

اعمال فیلتر دقیق تر روی داده ها

برای اعمال فیلتر دقیقتر روی داده ها و مثلا توکن بندی کردن هشتگ ها می توان تکه کد فوق را به شکل زیر تغییر داد:



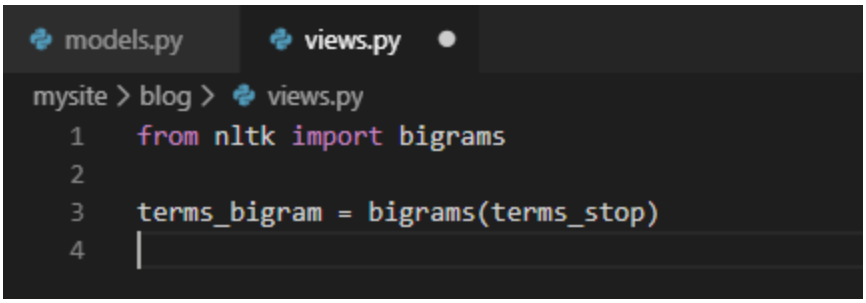
خروجی به شکل زیر تغییر خواهد کرد:

```
[('#python', 9), ('#scala', 6), ('#nosql', 4), ('#bigdata', 3), ('#nlp', 3)]
```

و بیشترین تکرار ها به این شکل خواهد بود:

```
[('python', 11), ('data', 8), ('summarisation', 6), ('twitter', 5), ('nice', 5)]
```

اینگونه برسی محتوای پر تکرار در سطح اول اطلاعات زیادی درباره موضوع مطلب توییتها به ما نمیده.برای همین در سطح دوم دوباره این پروسه تکرار می شود تا پر تکرارترین توکن دوم را هم بیابیم.این کار را با استفاده از کتابخانه nltk و تابع bigrams که به صورت تکرار شونده بزرگترین مقادیر را باز میگرداند انجام میدهیم.



با استفاده از تابع bigram کاندیدای دومین کلمه هم در توییت ها مشخص می شود و خروجی به شکل زیر درمیآید:

```
[(('nice', 'article'), 4), (('extractive', 'summarisation'), 4), (('summarisation', 'sentence'), 3), (('short', 'paper'), 3), (('paper', 'extractive'), 2)]
```

Visualization داده های بدست آمده

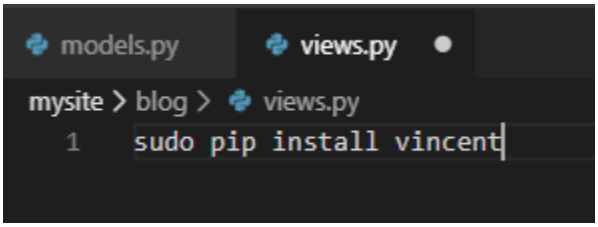
ویژوال کردن داده های بدست آمده در قالب نمودار ها می تواند در مقیاس های عظیم که برسی موردی ثمربخش نیست.مفید و حیاتی باشد.در ادامه به نحوه پیاده سازی مکانیز نمایی کردن داده های بدست آمده از توییتر اشاره خواهیم کرد.

ایجاد نمودار با استفاده از ماژول Vincent

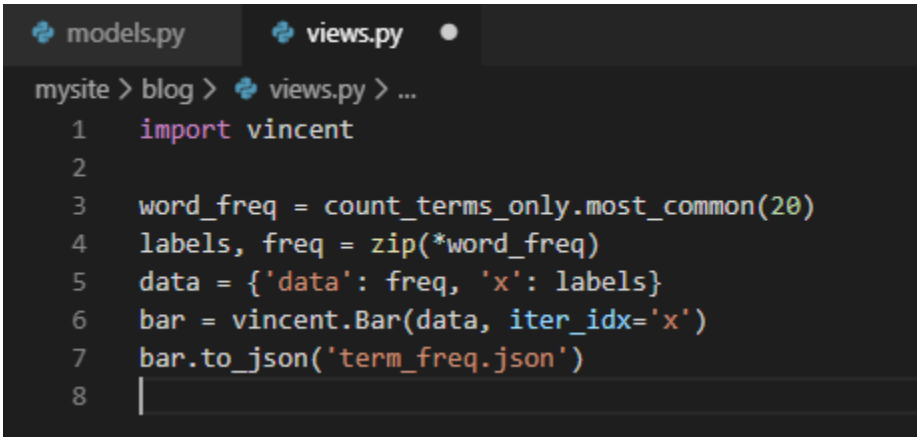
کتابخانه های فراوانی برای ایجاد طرح ویژوالیزیشن در پایتون وجود دارد مثل کتابخانه های matplotlib و ggplot ، یکی از جالبترین کتابخانه ها برای ایجاد Visualization و نمودار کتابخانه جاوا اسکریپتی D3.js هست که می توانید با استفاده از CSS و SVG نمودار های فوق العاده حرفه ای طراحی و پیاده سازی نمایید.

Vincent کتابخانه پایتونی دیگری است که ارتباط بین پایتون و D3.js فراهم کرده و کار مارا در back end و front end خیلی راحت تر کرده است.به این شکل که داده های استخراج شده و آنالیز شده را به شکل Vega (یک فرمت جی.سان قابل استفاده در D3) به این شکل ارتباط بین پایتون و D3 به راحتی برقرا شده و به راحتی میتوان از آن بهره برد.

ابتدا Vincent را نصب کنید:



سپس با استفاده از دیتاست مورد نظر خود که در اینجا term_freq.json هست را برای پیاده سازی طرح به شکل زیر برسی میکنیم:



سپس خروجی Vincent را که به صورت فرمت vega یا همان فرمت json سفارشی شده برای D3 می باشد را به شکل زیر فراخوانی میکنیم :

mysite > blog > views.py > ...

```
<html>
<head>
    <title>Vega Scaffold</title>
    <script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
    <script src="http://d3js.org/topojson.v1.min.js"></script>
    <script src="http://d3js.org/d3.geo.projection.v0.min.js" charset="utf-8"></script>
    <script src="http://trifacta.github.com/vega/vega.js"></script>
</head>
<body>
    <div id="vis"></div>
</body>
<script type="text/javascript">
// parse a spec and create a visualization view
function parse(spec) {
  vg.parse.spec(spec, function(chart) { chart({el:"#vis"}).update(); });
}
parse("term_freq.json");
</script>
</html>

```

همانطور که مشاهده میکنید دیتای تولیدی را در قابل فرمت vega به سمت کاربر ارسال می کنیم تا برای نمایش کاربر و رسم نمودار استفاده قرار بگیرد.

حالا کافیست فایل مورد نظر را با فرمت html ذخیره کنید.و با استفاده از سرور لوکال پایتون آن را اجرا نمایید.

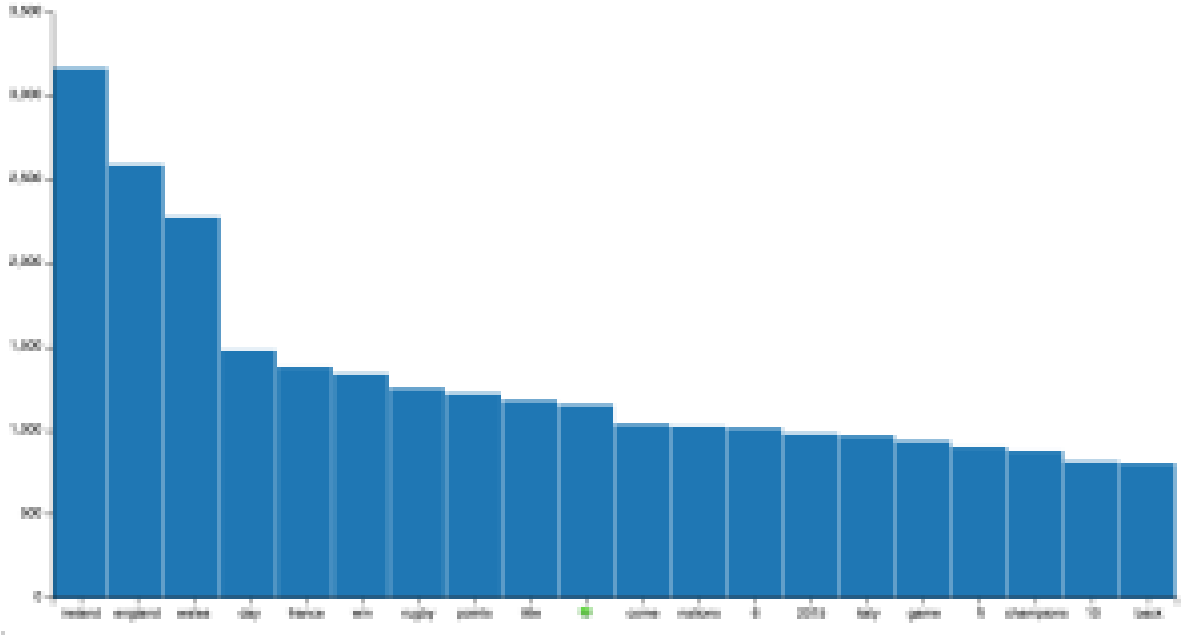
برای اینکار ابتدا وب سرور توکار پایتون را باید استارت کنیم.

```
python -m http.server 8888 # Python 3

python -m SimpleHTTPServer 8888 # Python 2
```

برای کسانی که از پایتون 3 استفاده می کنند خط اول و برای پایتون 2 از خط دوم استفاده کنید.با اجرای دستور فوق در ترمینال . بر روی لوکال هاست و پورت 8888 یک ورب سرور توکار پایتون اجرا می شود که وظیفه تفسیر فایل مورد نظر ما برای ترسیم چارت را خواهد داشت.حالا مرورگر خود را باز کنید و آدرس زیر را وارد نمایید:

http:localhost:8888/chart.html

*با فرض اینکه فایل نمودار ما با نام chart.html ذخیره شده است.



نموداری شبیه به نمودار فوق را مشاهده خواهید کرد.که صد البته با توجه به دیتاست دارای بازه متفاوتی می باشد.

ترسیم نمودار بر اساس مختصات زمانی

به عنوان مثال در نظر بگیرید بخواهیم ببینیم که واکنش فعالان توییتری نسبت به موضوع یا اتفاق جاری در کشور چگونه است.برای اینکار باید استریم real time خود را به صورت نمودار زنده دربیاوریم.در همین راستا یک کتابخانه فوق العاده دیگر به نام Panda در پایتون وجود دارد که برای آنالیز time series بسیار عالی عمل می کند.مثلا تصور کنید ما میخواهیم هشتگی با عنوان #ITAvWal را به صورت time series و real timeبرسی کنیم.

ابتدا پاندا را نصب میکنیم:

```
mysite > blog > views.py
1    Pip install panda
```

سپس هشتگ مورد نظر را با استفاده از قطعه کد زیر تحت نظر میگیریم.

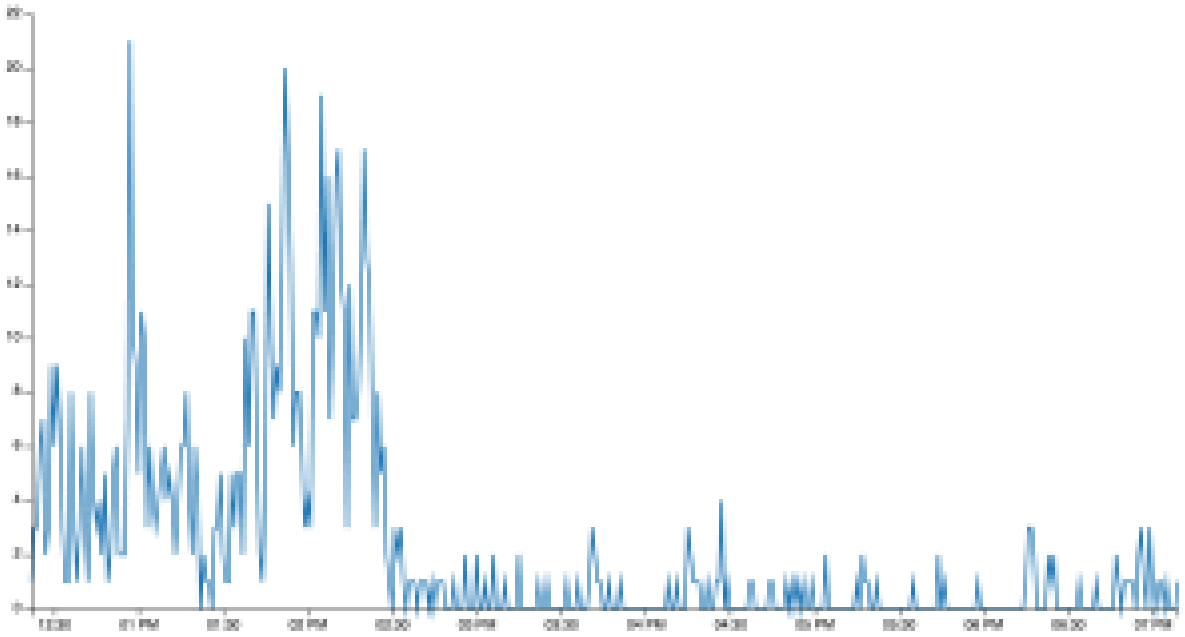
```
mysite > blog > views.py > ...
1   import pandas
2   import json
3
4   dates_ITAvWAL = []
5   # f is the file pointer to the JSON data set
6   for line in f:
7       tweet = json.loads(line)
8       # let's focus on hashtags only at the moment
9       terms_hash = [term for term in preprocess(tweet['text']) if term.startswith('#')]
10      # track when the hashtag is mentioned
11      if '#itavwal' in terms_hash:
12          dates_ITAvWAL.append(tweet['created_at'])
13
14  # a list of "1" to count the hashtags
15  ones = [1]*len(dates_ITAvWAL)
16  # the index of the series
17  idx = pandas.DatetimeIndex(dates_ITAvWAL)
18  # the actual series (at series of 1s for the moment)
19  ITAvWAL = pandas.Series(ones, index=idx)
20
21  # Resampling / bucketing
22  per_minute = ITAvWAL.resample('1Min', how='sum').fillna(0)
23
```

خط آخر به را مشـــاهده کنید .تمام توییت ها با هشـــتگ مورد نظر را به فاصـــله هر 1 دقیقه جمع آوری میکند و هر دقیقه با توییت های جدید جمع میکنند تا دیتاست ما کامل شود.درصورتی که در دقیقه مورد نظر توییتی با هشتگ مورد نظر ارسال نشده باشد مقدار 0 با استفاده از ()fillna جایگذین می شود.

حالا دیتاست مورد نظر را برای ایجاد یک طرح نمودار به Vincent میدهیم.

```
mysite > blog > views.py > ...
1   time_chart = vincent.Line(ITAvWAL)
2   time_chart.axis_titles(x='Time', y='Freq')
3   time_chart.to_json('time_chart.json')
4
```

کافیه تا فایل time_char.json را درون یک نمودار d3 فراخوانی کنید.چیزی شـــبیه به نمودار زیر را مشاهده خواهید کرد:

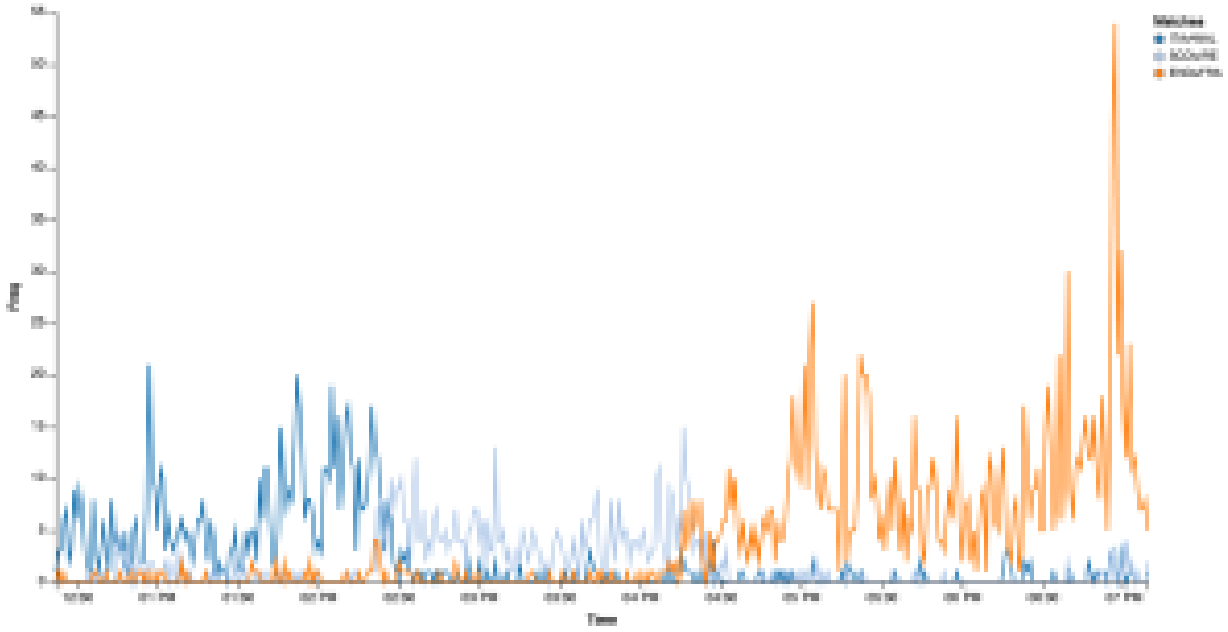


حالا تصـــور کنید که بخواهیم همزمان چند هشـــتگ دیگر را هم زیر نظر بگیریم و برای آنها نمودار رســـم کنیم.با استفاده از پاندا و کمی تغییر در کد فوق اینکار امکان پذیر خواهد بود.به این شکل :

mysite > blog > views.py > ...

```
# all the data together
match_data = dict(ITAvWAL=per_minute_i, SCOvIRE=per_minute_s, ENGvFRA=per_minute_e)
# we need a DataFrame, to accommodate multiple series
all_matches = pandas.DataFrame(data=match_data,
                               index=per_minute_i.index)
# Resampling as above
all_matches = all_matches.resample('1Min', how='sum').fillna(0)

# and now the plotting
time_chart = vincent.Line(all_matches[['ITAvWAL', 'SCOvIRE', 'ENGvFRA']])
time_chart.axis_titles(x='Time', y='Freq')
time_chart.legend(title='Matches')
time_chart.to_json('time_chart.json')
```

و خروجی چیزی شبیه به نمودار زیر خواهد شد:



همانطور که میبینید این نمودار هشـــتگهای مختلف را بنا به فعالیت های آنها و تعداد ارســـال و... مورد بررسی قرار داده و نمودار آنها را رسم کرده.