Project: Other Analitycs

Course: Massive Data Analytics

Proffesor: Dr. Iman Gholampoor

Mohammad Mehdi Zare

- **Introduction**:
  In this report, I aim to address four questions utilizing the available data, including Twitter data and Persian news data, with the help of algorithms covered in the course. In the provided notebook, various libraries have been used, but the algorithms have also been implemented manually. Overall, the PySpark library and RDD API have been employed for data processing.
- **The questions that I will be answering next:**
  1. If Twitter decides to verify some users for free, which users are candidates for verifying, and which users have lost their verified label?
  2. How many distinct words are present in news data? (batch processing and stream processing approach)
  3. Each user, based on their tweets and behavior on Twitter, is interested in a specific set of categories. News also belongs to various categories. If Twitter intends to suggest daily news to its users, which news does it recommend to each user?
  4. Some news agencies copy news from others, which is a form of Plagiarism. I want to identify these instances in the news data.
- **Sections:**

  1. **Dataset preparation and preprocessing:**
     a) At first, I clean news data. I remove punctuation and stopwords. Then, remove unnecessary columns from the data and keep columns: uid, body, source, date_published, key_words, and categories. The result dataset is:

```
[8]: [('68feae4bbbedc2d54adbb2369',
  'سرمربی ملوان تصاویر وایرالی دیدار نساجی پست جالب منتشر گزارش ورزش مهدی تارتار خوشحالی عجیب غریب دقیقه چهره جالب هفته لیگ برتر اختصاص موردتوجه هواداران فوتبال قرار تارتار انتشار پست اینستاگرامی
تصاویر جشن خوشحالی نوشته جادوی طرفداران انزلی سرمربی ملوان وعده تیمش ادامه فصل توان خواهد جنگید رضایت هواداران جلب نوشته ادامه میخوانید عکسها نگاه میکنم متحیر میشوم یادم نمیآید هرگز اینطور شوری باشم
جادوی شماست سکو آنچنان حس انرژی میدهید لحظه پیروزی برایمان وصفناشدنی سکوها لب خط پای تلویزیونها مستطیل سبز همدل یکصدا میرویم باهم میجنگیم باهم میخندیم هرروز تلاش میکنیم خنده صورت نگه پیروزی شماست
مردان نیک چمن مردمان نیک سکو \U0001fa75♡',
  'ورزش سه',
  'varzesh3.com',
  '10/10/2023, 07:16:46',
  ['سرمربی ملوان', 'لیگ برتر', 'مستطیل سبز', 'انزلی'],
  ['sports']),
 ('434e9837dd761812773fa4627',
  'سخنگوی کمیسیون تلفیق برنامه هفتم توسعه موافقت ضمنی دولت اجرای همسانسازی برنامه خبر گزارش ایلنا محسن زنگنه سخنگوی کمیسیون تلفیق برنامه هفتم توسعه حاشیه علنی امروز سهشنبه مهر ماه مجلس خبرنگاران اشا
ره وضعیت همسانسازی حقوق بازنشستگان شب گذشته جلس ساعته رئیس مجلس شورای اسلامی رئیس کمیسیون تلفیق برنامه هفتم توسعه رئیس سازمان برنامه بودجه برگزار نت جمعبندی اولیه برنامه هفتم توسعه بتوانیم بحث متناس
ب حقوق بازنشستگان سرانجام برسانیم هدف پایان برنامه هفتم شاهد تحقق موضوع باشیم سخنگوی کمیسیون تلفیق برنامه هفتم توسعه عنوان موضوع اصلاحات ساختاری صندوق بازنشستگی انجام مشکل ورشکستگی صندوق بازنشستگ
ی مرتفع',
  'خبرگزاری ایلنا',
  'ilna.ir',
  '10/10/2023, 07:17:06'
```

     b) Second, I clean the tweet dataset and remove incomplete rows in the data. I remove 214 rows:

```
# In data, some tweets have incomplete data
incomp=tweets_json_rdd.filter(lambda x: x['tweet_type']=="quoted" and x.get('quoted_status').get('id',0)==0).collect()
print("sample of incomplete data: ")
print(incomp[1])
print("number of incomplete data : "+str(len(incomp)))
# filter them
tweets_json_rdd_cleaned=tweets_json_rdd.filter(lambda x:  not (x['tweet_type']=="quoted" and x.get('quoted_status').get('user',0)==0))
tweets_json_rdd_cleaned=tweets_json_rdd_cleaned.filter(lambda x : not x.get('nlp',0)== 0)
```

```
sample of incomplete data:
{'publish_source': None, 'in_reply_to_status_id_str': None, 'in_reply_to_user_id_str': None, 'in_reply_to_status_id': None, 'in_reply_to_user_id': None,
'in_reply_to_screen_name': None, 'truncated': False, 'is_quote_status': True, 'retweet_count': 1, 'reply_count': 0, 'quote_count': 1, 'favorite_count':
3, 'favorited': False, 'retweeted': False, 'possibly_sensitive': None, 'lang': 'fa', 'geo': None, 'view_count': 235, 'tweet_type': 'quoted', 'emojis': No
ne, 'text': 'یا ازدواج نکنید یا از مهر بالا فرار کنید\nاونی که به مهر بالا اصرار داره دیگه به چه زبونی بهتون بگه میخواد بعدا بدبختتون کنه؟\n!ننویسید اقا اون مهریه کوفتی رو تعداد بالا ننویسید', 'user': {'i
d': '1363594775539159045', 'id_str': '1363594775539159045', 'created_at': 1613941316, 'name': 'وکیل الرعایا', 'screen_name': 'md_s_law', 'location': 'tehra
n', 'description': 'حسبنا الله و نعم الوکیل', 'verified': False, 'followers_count': 1124, 'friends_count': 1431, 'listed_count': 1, 'favourites_count': 3955, 'st
atuses_count': 1334, 'profile_image_url': 'https://pbs.twimg.com/profile_images/1729131836213800960/VbZFyJs-_normal.jpg', 'geo_enabled': True, 'profile_i
mage_url_https': 'https://pbs.twimg.com/profile_images/1729131836213800960/VbZFyJs-_normal.jpg', 'profile_banner_url': 'https://pbs.twimg.com/profile_ban
ners/1363594775539159045/1647984275', 'default_profile': True, 'default_profile_image': False}, 'entities': {'urls': [], 'hashtags': [], 'symbols': [],
'user_mentions': []}, 'quoted_status': {}, 'engagement': 4, 'impression': 235, 'reach': 0.2091, 'influence_v2': 0.0036, 'impression_rate_v2': 0.1762, 'en
gagement_rate_v2': 0.003, 'twitter_engagement_rate_v2': 1.7021, 'timestamp': 1702393654, 'index_version': 1, 'created_at': 1698831343, 'id': '17196494109
87028916', 'tweet_source_type': 'tweet_timeline_v2', 'text_lang': 'fa', 'nlp': {'offensive': 'non_offensive', 'sentiment': 'Negative', 'emotion': ['antic
ipation'], 'keywords': [{'keyword': 'مهریه'}], 'question': 'question', 'hashtag_polarity': [], 'event': 'NotEvent', 'classification': ['social', 'persona
l'], 'ner_polarity': []}}
number of incomplete data : 214
```

2. **Exploration**

Let’s go to answer the above questions:

a) **If Twitter decides to verify some users for free, which users are candidates for verifying, and which users have lost their verified label?**

To answer this question, I use PageRank and TrustRank algorithms. users are linked to each other by their interaction with tweets. So with the PageRank algorithm, we can find a weight for the user. Also, verified users can be used as trust sets and then use the TrustRank algorithm. I use Networkx library to create a directed graph.

Steps:

- Find verified users as trust set. In data, some users have verified labels, so I filter them. You can see the number of users and verified users in the data, below picture:

```
[11]: # find trust set:
      verified_users=tweets_json_rdd_cleaned.filter(lambda x : x['user']['verified']==True).map(lambda x: x['user']['id']).distinct()
      verified_users.take(10)
[11]: ['3345517306',
       '286844404',
       '1518921260360630273',
       '1008316359405637633',
       '770390482929250304',
       '1246111801462496321',
       '947107471985926145',
       '1454596183406510081',
       '1023664436173856770',
       '1392116892274462720']
[12]: verified_users.count()
[12]: 1720
      number of users in data
[13]: tweets_json_rdd_cleaned.map(lambda x: x['user']['id']).distinct().count()
[13]: 45873
```

- Make a directed graph. Users interact with each other by replying, retweeting, and quoting. Therefore, I make edges from user interaction in data. And make a directed graph. In the below pictures you can see some edges and a part of the graph:

- Run page Rank algorithm. The result is:

```
[19]: [('38154763', 0.019069219532820137),
       ('2682820352', 0.007503240498135082),
       ('810859229267378176', 0.005947767146534361 5),
       ('4075582205', 0.004502849214073731),
       ('1440806264255045638', 0.003923641660580251),
       ('867126568451280896', 0.003609715762625 5503),
       ('945886639603310592', 0.002927343671582436),
       ('1612794343911297025', 0.002476016227295738),
       ('2739506010', 0.002474665074495339),
       ('2331584635', 0.002310059354181786),
       ('1408312021', 0.002264017531270494 7),
       ('1290359738585288704', 0.002215514043808075 7),
       ('1227976819011399681', 0.0019409359004997234),
       ('49127938', 0.0019044675409917117),
       ('46482865', 0.0018732180562105003),
       ('722757627877335040', 0.001818383265052587),
       ('1222773302441148416', 0.001779711449750 11),
```

- Run trust Rank algorithm. The result is:

```
[22]: [('38154763', 0.06050678180503591),
       ('49127938', 0.010724595312254231),
       ('2682820352', 0.01015260715102248),
       ('3414505683', 0.009560659429508 09),
       ('909512912', 0.008712373671555865),
       ('1227976819011399681', 0.007860097748656215),
       ('1440806264255045638', 0.007367714883802768),
       ('2739506010', 0.006894261694057001),
       ('742451244028039168', 0.006762648959106495),
       ('160940788', 0.006732432143866629),
       ('1458337727536144386', 0.006006501964188 4956),
       ('4075582205', 0.005764689497048296),
       ('1408312021', 0.005681094902041521),
       ('1575805035501735936', 0.005270689332861157),
       ('945886639603310592', 0.005179399830478 27),
       ('722960833853173760', 0.005040165916354118 5),
       ('867126568451280896', 0.005038288959481802),
```

- Compare results to find users that candidates to verify and lost their verify labels:

  Verified users whose rank difference in the two algorithms is greater than .03 are fake users. And users that rank 1 to 100 are new verified users.

  ### compare results to find fake and good users

```
[23]: dict_G1=dict(pagerank_sorted)
      dict_G2=dict(pagerank_sorted1)
      result_dict = {key: dict_G1[key] - dict_G2[key] for key in dict_G1}
      alpha = .03
      fake_user = [key for key, value in result_dict.items() if abs(value) > alpha]
      fake_user

[23]: ['38154763']

[24]: # fake verified user
      set(verified_nodes).intersection(fake_user)

[24]: {'38154763'}

[25]: # user that can verify
      first_100_user=list(zip(*pagerank_sorted1))[0][1:100]
      new_verified_user=set(first_100_user).difference(verified_nodes)
      new_verified_user

[25]: {'1065658304884547585',
       '1222773302441148416',
       '1515956552',
       '1588740471546847238',
       '1612794343911297025',
       '2509360928',
       '3189754680',
       '46482865',
       '722757627877335040',
       '802011472964546560',
       '940962097130889216'}
```

b) **How many distinct words are present in news data? (batch processing and stream processing approach)**

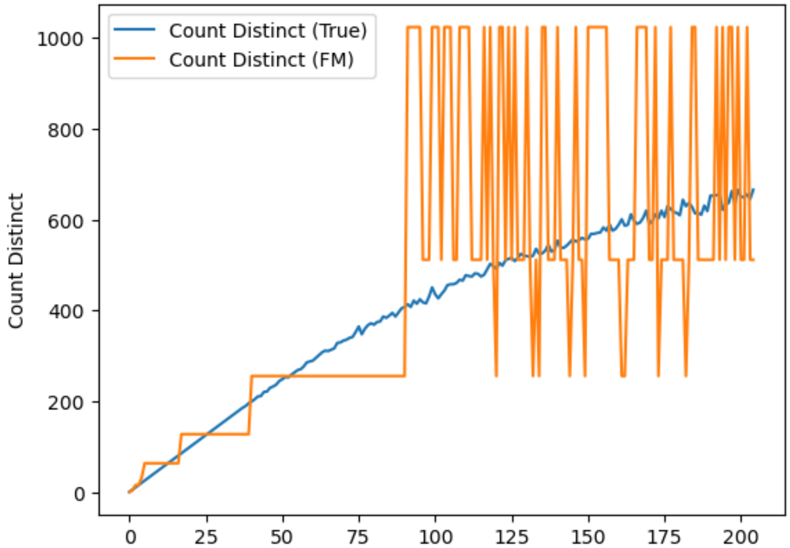
To answer this question, I implement the Flajolet Martin algorithm. Steps:

- create a tokens list using rdd (all words in all news)

```
[52]: print(tokens[1:100])

['ملوان', 'تصاویر', 'وایرالی', 'دیدار', 'نساجی', 'پست', 'جالب', 'منتشر', 'گزارش', 'ورزش', 'مهدی', 'تارتار', 'خوشحالی', 'عجیب', 'غریب', 'دقیقه', 'چهره', 'جالب', 'هفته', 'لیگ', 'ب',
'رتر', 'اختصاص', 'موردتوجه', 'هواداران', 'فوتبال', 'قرار', 'تارتار', 'انتشار', 'پست', 'اینستاگرامی', 'تصاویر', 'جشن', 'خوشحالی', 'نوشته', 'جادوی', 'طرفداران', 'انزلی', 'سرمربی', 'ملوان',
'وعده', 'تیمش', 'ادامه', 'فصل', 'توان', 'خواهد', 'جنگید', 'رضایت', 'هواداران', 'جلب', 'نوشته', 'ادامه', 'میخوانید', 'عکسها', 'نگاه', 'میکنم', 'متحیر', 'میشوم', 'یادم', 'نمیآید', 'هرگز',
'اینطور', 'شوری', 'باشم', 'جادوی', 'شماست', 'سکو', 'آنچنان', 'حس', 'انرژی', 'میدهید', 'لحظه', 'پیروزی', 'برایمان', 'وصفناشدنی', 'سکوها', 'لب', 'خط', 'پای', 'تلویزیونها', 'مستطیل',
'سبز', 'همدل', 'یکصدا', 'میرویم', 'باهم', 'میجنگیم', 'باهم', 'میخندیم', 'هرروز', 'تلاش', 'میکنیم', 'خنده', 'صورت', 'نگه', 'پیروزی', 'شماست', 'مردان', 'نیک', 'چمن']
```

- partition tokens into 3 parts.
- make 7 hash functions
- run the algorithm to estimate and compare that with the true answer.
  In each partition, I start with defining a closed hash range, big enough to hold the maximum number of unique values possible (2 ^ 64). Every word of the tokens is passed through a hash function that permutes the elements in a uniform distribution. For this hash value, I find the position of the rightmost 1s bit and mark the corresponding position in the bit vector as 1. Once all the words of the partition are processed, the bit vector will have 1s at all the positions corresponding to the position of every rightmost 1s bit for all words in the partition. Now I find the position, b, of the leftmost 1 in this bit vector. This position b corresponds to the max length of unset bits that I have seen while processing the words. Then find the median of the estimated value for each hash function. Finally, estimate distinct words by averaging values of partitions.
  In the below picture, you can see the fluctuation of estimated values for generated data.



After running the algorithm in news data, the result is:

```
tokens=news_cleaned1.flatMap(lambda x :x[1].split()).collect()
part_tokens=np.array_split(tokens, 3)
res=[]
for i in part_tokens:
    tok=list(i)
    tokens_hashed = [int(hashlib.md5(token.encode()).hexdigest(), 16) % (2 ** 64) for token in tok]
    res.append(count_distinct_fm(tokens_hashed,params))

print(len(tokens), len(set(tokens)))
print("estimated words: ")
print(statistics.mean(res))
```

```
1017082 47456
estimated words:
49152
```

True distinct words

We can see that the estimated count-distinct using the Flajolet-Martin algorithm is very close to the actual deterministic words.

c) **Each user, based on their tweets and behavior on Twitter, is interested in a specific set of categories. News also belongs to various categories. If Twitter intends to suggest daily news to its users, which news does it recommend to each user?**

In this part, I use stream processing. users like some topics. news has topics too. so we can use the bloom filter to recommend news to users each day. Steps:

- Make data as a stream data.
- Implement bloom filter
- Make categories in news data and tweet data the same. Twitter and Wikipedia use different words for categories. For example, economy and economical. You can see the categories in tweets and news:

```
# list of categories in new and tweets
categories=news_cleaned1.flatMap(lambda x :x[6]).distinct().collect()
classifications=tweets_rdd_ok.flatMap(lambda x : x['nlp'].get('classification')).distinct().collect()
```

```
print(categories)
print(classifications)
```

```
['politics', 'social', 'economy', 'security', 'culture', 'sports', 'science_and_technology', 'religious', 'health', 'military']
['entertainment', 'security', 'military', 'health', 'offensive', 'culture', 'religious', 'others', 'politics', 'science_and_technology', 'sports', 'econo
my', 'personal', 'advertising', 'social']
```

- I use categories that exist in the intersection set.

```
print(set(categories).intersection(classifications))
print(set(categories).difference(classifications))
```

```
{'politics', 'security', 'social', 'health', 'economy', 'sports', 'culture', 'military', 'religious', 'science_and_technology'}
set()
```

- For each user, we can run a query. In a query, for each day make a bloom filter for users according to their liked categories. In the stream data of news, filter news and recommend one news.
  As we know, The bloom filter has false positive. I calculate the number of hash functions and the size of the bit array. According to the number of categories and False positive Probability.
  The result of the sample query is:

```
[40]: recom_news2user('1004052664882118657',list_news_day,tweets_rdd_ok)

Size of bit array:19
False positive Probability:0.0001
Number of hash functions:13
category for day 10/09/2023 : ['politics']
in day :10/09/2023
user 1004052664882118657  maybe like news :
('0184a2f6b9df84f2203d79f19', 'آنجایی واشنگتن تلاش میکند مانع افزایش قیمت نفت جهان پالایشگاه چین مطمئن صادرات نفت ایران ادامه خواهد گزارش گروه اقتصاد بینالملل فارس نقل بلومبرگ آنجاییکه واشنگتن ن
تلاش میکند مانع افزایش قیمت نفت جهان پالایشگاه چین مطمئن صادرات نفت ایران ادامه خواهد پالایشگاه مستقل چین موسوم تیپاتها مشتاق خرید نفت ایران بلومبرگ مدعی ارتباط ایران حماس ارتباط ادعایی حمله غافلگیرانه میتواند
تجارت نفت کشور موضوع سختتر آمریکا قبال تحریم ایران معرض تهدید قرار حاضر چندین پالایشگاه تاجر چینی نفت خام ایران میخرند گفتند هیچگونه اختلال سختگیری پیشبینی نمیکنند نخواستند نامشان فاش گفتند فشاری واشنگتن
دولت بایدن فشار تورمی آستانه انتخابات ریاستجمهوری سال آینده بررسی خواهد گذشته تحریمها چندان اشتیاق چین خرید نفت ایران تاثیری نداشته چراکه نفت خام مبادی مالزی تامین آمار شرکت رهیابی محموله نفتی نشان خرید نف
ت ایران چین ماه آگوست روزانه میلیون بشکه روز رسیده بیشترین رقم گذشته میرود بنا ادعای شبکه خبری تهدیدات درحالیکه رسانه آمریکایی ادعا میکنند ایران حملات مقاومت نقش آنتونی بلینکن وزیر خارجه آمریکا شبکه خبر
ی انبیسی حاضر شواهدی مبنی ایران مستقیم دخالت پایان پیام', 'farsnews.ir', '10/09/2023', ['انتخابات ریاست جمه', 'آنتونی بلینکن', 'وزیر خارجه آمریکا', 'گروه اقتصاد الملل فارس', 'خبرگزاری فارس
وری'], ['politics'])
category for day 10/10/2023 : ['security']
in day :10/10/2023
user 1004052664882118657  maybe like news :
('e7e44d4fb2f82802cebed90bf', 'گزارش ایلنا امیر سرتیپ علیرضا الهامی جانشین فرمانده نیروی پدافند هوایی ارتش دومین کنگره هزار شهید استان همدان اظهارداشت خانواده شهدا تقدیم فرزندان کشور مردم امنیت آ
رامش حیثیت عزت تبیین اهداف انگیزه حرکت شهدا نسل امروز یادوارهها کنگرهها جزو اهداف اصلی قرار فرموده رهبر معظم انقلاب مدظلهالعالی اجر ثواب اقدامات اجر شهدا کمتر اشاره گفتمان دفاع مقدس ثابت اصل صیانت ک
```

d) **Some news agencies copy news from others, which is a form of Plagiarism. I want to identify these instances in the news data.**
Some Twitter users do plagiarism, meaning they send out tweets that are essentially identical to another user's tweet, with slight modifications in sentence structure. Due to constraints in processing tweet text, I implement the algorithm on news data, leveraging the results of Exercise 1 in this section. Additionally, the first news agency to publish the text is considered trustworthy. note: I implemented tf-idf in hw1. To save time I used the Mllib library in the project.

The similarity between the two news can be obtained by computing the cosine similarity of their TF-IDF vectors. We know that the cosine similarity is a number between 0 and 1, due to normalization. If the cosine similarity is 1, the new are identical; if the similarity is 0, the new have nothing in common.

I chose news from hw2 results that I have found similar news for that. After calculating tf-idf for all news and candidate news, calculate the cosine similarity between the candidate and other news and find the top five news that has high cosine similarity. According to their published date, we can say which ones are plagiarism.

Cosine similarity of top five news:

```
Code                                         JupyterLab    Python 3 (ipykernel)

for t in topFive:
    print("doc '%s' has score %.4f" % (t[0], t[1]))

doc 'a58188b693d8167ecf144685a' has score 1.0000
doc '6789ed5448fe3bcb8e9bf4bcb' has score 0.9947
doc '49a301cb98246ca6d89e33755' has score 0.9947
doc 'c672d416dfda66a9b979e1202' has score 0.9913
doc '35a62bf8c7becf6a2cfca6357' has score 0.9909
```

News are:

```
[47]: x,y=zip(*topFive)
      y=list(x)
      top_news=news_cleaned1.filter(lambda x: x[0] in y)
      top_news.take(10)
```

```
[47]: [('a58188b693d8167ecf144685a',
  'مراسم افتتاح ایستگاه مترو خطوط آغاز عملیات حفاری مکانیزه تونل خط حضور رئیسجمهور انجام گزارشحوزه دولت خبرگزاری تسنیم سید ابراهیم رئیسی رئیسجمهور صبح امروز سهشنبه حضور ایستگاه شهید آرمان علیوردی منطقه کوهسار افتتاح رسمی ایستگاه خط مترو تهران قطار ایستگاه شهران ایستگاه شکل رسمی افتتاح مراسم افتتاح ایستگاه شهید علیوردی حضور خانواده برگزار رئیس جمهور پیشنهاد تابلویی زندگینامه شهید والامقام ورودی ایستگاه نصب همزمان مراسم ایستگاه شهر زیبا خط میدان کتاب خط متروی تهران بهرهبرداری رسیدند مراسم امروز عملیات حفاری مکانیزه تونل خط متروی تهران کارگاه دریاچه شهدای خلیج فارس آغاز خط شرق پایتخت ادامه مییابد خط کیلومتر ایستگاه جابجایی مسافران مناطق پوشش خواهد',
  'خبرگزاری تسنیم',
  'www.tasnimnews.com',
  '10/10/2023, 07:33:10',
  ['کارگاه دریاچه شهدای خلیج فارس',
   'ایستگاه شهید آرمان علی وردی',
   'ایستگاه شهید علی وردی',
   'ایستگاه شهران'],
  ['politics']),
 ('35a62bf8c7becf6a2cfca6357',
  'گزارش خبرگزاری مهر حجت الاسلام سید ابراهیم رئیسی رئیسجمهور صبح امروز سهشنبه مهر حضور ایستگاه شهید آرمان علیوردی منطقه کوهسار افتتاح رسمی ایستگاه خط مترو تهران قطار ایستگاه شهران ایستگاه شکل رسمی افتتاح اساس گزارش مراسم افتتاح ایستگاه شهید علیوردی حضور خانواده برگزار رئیس جمهور پیشنهاد تابلویی زندگینامه شهید والامقام ورودی ایستگاه نصب همزمان مراسم ایستگاه شهر زیبا خط میدان کتاب خط متروی تهران بهرهبرداری رسیدند مراسم امروز عملیات حفاری مکانیزه تونل خط متروی تهران کارگاه دریاچه شهدای خلیج فارس آغاز خط شرق پایتخت ادامه مییابد گفتنی خط کیلومتر ایستگاه جابجایی مسافران مناطق پوشش خواهد',
  'خبرگزاری مهر | اخبار ایران و جهان | Mehr News Agency',
  'mehrnews.com',
  '10/10/2023, 07:46:15',
  ['کارگاه دریاچه شهدای خلیج فارس',
   'ایستگاه شهید آرمان علی وردی',
   'ایستگاه شهید علی وردی',
   'مهر'],
  ['social']),
 ('6789ed5448fe3bcb8e9bf4bcb',
  'مراسم افتتاح ایستگاه خطوط مترو حفاری مکانیزه تونل خط متروی تهران حضور رئیس جمهور آغاز گزارش همشهری آنلاین حضور سید ابراهیم رئیسی ایستگاه شهید آرمان علیوردی ایستگاه کوهسار مراسم افتتاح ایستگاه مترو رسما مراسم خانواده شهید علی وردی حضور رئیس جمهور پشنهاد تابلویی زندگینامه شهید ورودی ایستگاه نصب رئیس جمهور قطار مترو ایستگاه ایستگاه شهران همزمان مراسم ایستگاه شهر زیبا خط میدان کتاب خط متروی تهران بهرهبرداری رسیدند امروز همزمان مراسم افتتاح ایستگاه مترو عملیات حفاری مکانیزه تونل خط متروی تهران کارگاه دریاچه شهدای خلیج فارس آغاز خط شرق پایتخت ادامه مییابد خط کیلومتر ایستگاه جابجایی مسافران مناطق پوشش خواهد',
  'همشهری آنلاین',
  'hamshahrionline.ir',
  '10/10/2023, 07:48:02',
  ['کارگاه دریاچه شهدای خلیج فارس',
   'آرمان علی وردی',
   'ابراهیم رئیسی',
   'ایستگاه شهران'],
  ['social']),
```

you can see that all the top five news are the same and about specific topics and published on specific dates.

```
top_news.map(lambda x : (x[0],x[4])).take(10)
```

```
[('a58188b693d8167ecf144685a', '10/10/2023, 07:33:10'),
 ('35a62bf8c7becf6a2cfca6357', '10/10/2023, 07:46:15'),
 ('6789ed5448fe3bcb8e9bf4bcb', '10/10/2023, 07:48:02'),
 ('49a301cb98246ca6d89e33755', '10/10/2023, 08:21:00'),
 ('c672d416dfda66a9b979e1202', '10/10/2023, 09:06:22')]
```

Therefore, news with Id a58188b693d8167ecf144685a is a source, and others may copy from that.

The news agency of this news is :

```
59]: top_news.filter(lambda x: x[0]=='a58188b693d8167ecf144685a').map(lambda x:(x[2],x[3])).collect()
```

```
59]: [('خبرگزاری تسنیم', 'www.tasnimnews.com')]
```

Conclusion:

In the surrounding datasets, there is a wealth of information that can be uncovered through various data analysis methods. We discussed and explored different approaches for processing stream and persistent data, each having its own advantages. In the project, we demonstrated that stream processing methods may have errors, but they still provide good estimates of information, enabling us to make strong assumptions backed by the science of statistics and probability regarding data behavior.

We also emphasized the importance of data and how it can enhance the user experience in applications. It can assist us in identifying fraudulent activities and plagiarism by individuals from other sources. Moreover, based on users' behavioral history, we can even suggest news or tweets that align with their preferences and interests.

Thanks - winter 2024