Recommender System

May 18, 2020

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import json
from matplotlib import cm
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
```

```
/home/sanjukta/anaconda/lib/python3.7/site-
packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is
deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

[2]:
```
#read our dataset
excel_file = '/home/sanjukta/Downloads/RecommendationEngineData.xlsx'
videos_df = pd.read_excel(excel_file)
```

1 Exploratory Data Analysis

1.0.1 Data Preprocessing

[3]:
```
videos_df.head()
```

[3]:
```
   POST_ID               POST_STRING_UNIQUE_ID          CREATED_AT  \
0  5251588  ec7e9ef3246874618d617623ee07451c 2020-04-22 19:51:00
1  5539448  e38e34aa65c0c7c2ed42426fe92e6419 2020-05-10 18:00:00
2  5503440  01e4dc698aba6a4561739c58906838cc 2020-05-08 07:33:00
3  5538585  87d93e56b144f5ba7557663b2fb6218c 2020-05-10 15:18:00
4  5540220  4b20839183de924a7bc8e4bcdc9c20a2 2020-05-10 17:11:00

      Creator_Name                                           Caption  Length  \
0      Nojoto News  Know who loved your story | Tag Nojotians #Noj...      51
1      Nojoto News  Details for Day 1 (Monday) :- \nExpress Karo N...     168
2  Anand Mohan Jha  Anshh only 4 youð sorry #Nojoto #story #Poe...       0
3  Anand Mohan Jha  #krishna_flute àďőàď¿àďĄ àďďàě àďőàď¿àďĄ àďžà...       0
4    Bhawna Mishra  #SuperMom #chitthi #letter #originalmess #message     116
```

```
   Watch_Views  Total_Watch_time  Average_Watch_time  10_Sec_Watch_Time  ...  \
0        61196            732610                12.0             584192  ...
1         2751             33002                12.0              25716  ...
2         7086            126534                17.9             110171  ...
3         1119             19908                17.8              17109  ...
4         1075             15966                14.9              13091  ...

   Execution_Reach  Spammy_Views  Love  Comment  Share  Report_Abuse  \
0          1000000         28445  1720      108     35             0
1            50000          1037   130       10      4             0
2            50000          2606   337      113      9             0
3            10000           447   114       34      2             0
4            10000           376   143       49      1             0

   Repost_Count  Creation_type ContentType LANGUAGE_NAME
0            73       Uploaded       Video       English
1            17       Uploaded       Video       English
2            21       Uploaded       Video       English
3            12        Created       Video       English
4            13        Created       Video       English

[5 rows x 21 columns]
```

[4]:
```
videos_df.shape
```

[4]:
```
(1000, 21)
```

[5]:
```
videos_df.describe()
```

[5]:
```
            POST_ID       Length   Watch_Views  Total_Watch_time  \
count  1.000000e+03  1000.000000   1000.000000      1.000000e+03
mean   5.238019e+06   104.448000   2846.651000      4.301675e+04
std    2.590702e+05    87.586384   4599.174397      7.344618e+04
min    1.090611e+06     0.000000    148.000000      1.431000e+03
25%    5.125160e+06     0.000000    755.250000      9.142500e+03
50%    5.309428e+06   101.000000   1356.500000      1.985050e+04
75%    5.399322e+06   180.250000   2873.750000      4.339400e+04
max    5.540220e+06   320.000000  61196.000000      1.058837e+06

       Average_Watch_time  10_Sec_Watch_Time  10_Sec_Views  Execution_Reach  \
count         1000.000000        1000.000000    1000.00000       1000.00000
mean            14.165800       35449.142000     984.14100      11383.00000
std              3.929419       61858.176108    1667.32367      33979.30111
min              5.300000         737.000000      29.00000       1000.00000
25%             11.375000        6808.500000     214.75000       5000.00000
50%             13.600000       15964.000000     445.50000       5000.00000
75%             16.300000       35946.000000     964.00000      10000.00000
max             31.700000      929662.000000   21149.00000    1000000.00000
```

```
       Spammy_Views         Love      Comment        Share  Report_Abuse  \
count   1000.000000  1000.000000  1000.000000  1000.000000        1000.0
mean     883.922000   242.542000    39.740000     4.483000           0.0
std     1480.606192   229.016141    36.382194    10.604535           0.0
min       53.000000    37.000000     0.000000     0.000000           0.0
25%      267.750000   113.000000    14.000000     0.000000           0.0
50%      459.000000   168.000000    28.000000     1.000000           0.0
75%      868.250000   277.250000    55.000000     4.000000           0.0
max    28445.000000  2657.000000   283.000000   150.000000           0.0

       Repost_Count
count   1000.000000
mean       8.661000
std        9.156961
min        0.000000
25%        3.000000
50%        6.000000
75%       12.000000
max       91.000000
```

[6]:
```
#Statistical summary of our categorical columns
videos_df.describe(include=['O'])
```

[6]:
```
                   POST_STRING_UNIQUE_ID  Creator_Name       Caption  \
count                               1000          1000           996
unique                              1000           284           918
top     1a0e38d5e9914451afe63b4931ea6d90  Kapil Nayyar  #StoryOnline
freq                                   1            36            20

       Creation_type ContentType LANGUAGE_NAME
count           1000        1000          1000
unique             2           1             3
top          Created       Video       English
freq             676        1000           611
```

[7]:
```
# just to make sure that all Nan containing rows are deleted..
print("No of Nan values in our dataframe : ", sum(videos_df.isnull().any()))
```

```
No of Nan values in our dataframe :  1
```

[8]:
```
dup_bool = videos_df.duplicated(['POST_STRING_UNIQUE_ID','Caption'])
dups = sum(dup_bool) # by considering all columns..( including timestamp)
print("There are {} duplicate rating entries in the data..".format(dups))
```

```
There are 0 duplicate rating entries in the data..
```

[9]:
```
print("Total data ")
print("-"*50)
print("\nTotal no of videos :",videos_df.shape[0])
print("Total No of unique Users Name  :", len(np.unique(videos_df.
 ↪Creator_Name)))
print("Total No of unique Language  :", len(np.unique(videos_df.LANGUAGE_NAME)))

print("Total No of Likes  :", len(np.unique(videos_df.Love)))
print("Total No of Comment  :", len(np.unique(videos_df.Comment)))
print("Total No of Share  :", len(np.unique(videos_df.Share)))
```

```
Total data
--------------------------------------------------

Total no of videos : 1000
Total No of unique Users Name  : 284
Total No of unique Language  : 3
Total No of Likes  : 416
Total No of Comment  : 152
Total No of Share  : 48
```

1.0.2 SUMMARY

1. There is not duplicate entry
2. The videos are made in 3 different langeuage (i.e. English, Hindi & Punjabi)
3. One user (i.e. Kapil Nayyar) whose vide is shared 36 times
4. Total data set contains 1000 rows and 20 columns
5. CREATED_AT column with data type : Object
6. Missing values in the Caption column

[10]:
```
def print_top10(column_of_interest,column_stats):
    df = videos_df.groupby(column_of_interest)['POST_ID','Watch_Views',␣
 ↪'Love','Comment'].apply(lambda x: x.astype(int).sum())
    return df.sort_values(by=column_stats, ascending=False).head(10)
def visualize_top10(column_of_interest, column_stats):
    most_viewed_df = videos_df.groupby([column_of_interest])[column_stats].
 ↪sum().reset_index()
    sorted_df= most_viewed_df.sort_values(column_stats, ascending=False).iloc[:
 ↪5]
    ax = sorted_df.plot.bar(figsize = (15,15))
    # customizes the video titles, for asthetic purposes for the bar chart
    labels = []
    for item in sorted_df[column_of_interest]:
        labels.append(item[:20] + '...')
    ax.set_xticklabels(labels, rotation=45, fontsize=12)
    plt.show()
```

1.1 With respect to Watch_Views top ten content when a new user arrives

Watch_Views column defines popularity of a particular video. So when a new user comes, we recomment these top ten videos to continue with the services.

[11]:
```
print_top10('Caption','Watch_Views')
```

```
/home/sanjukta/anaconda/lib/python3.7/site-packages/ipykernel_launcher.py:2:
FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of
keys) will be deprecated, use a list instead.
```

[11]:
```
                                                     POST_ID  Watch_Views  \
Caption
Know who loved your story | Tag Nojotians #Nojo...   5251588        61196
#StoryOnline                                        106255962        59100
àďďàěàďÿàďřàď¿ àďžàěàďĄ, àďďàď¿àďšàě àďžàěà...   4633518        44679
galti #rap #nojotonews #trendingvideos #music        4649801        32335
#Rap #Song #Alag #hi #Vibe #hai \n#Nojotonews #...   4649483        32065
#RapOnline                                           9574081        30702
Aaj Hi Acha Hai | Storytelling\n\nReal happines...   4661957        27143
"yaadon ke patthar"\n\n#talk #travel #life_expe...   4654533        25869
#PoetryOnline                                       90098113        25639
#BabaYAGA #rap                                       4624971        25602

                                                    Love  Comment
Caption
Know who loved your story | Tag Nojotians #Nojo...  1720      108
#StoryOnline                                        4155      561
àďďàěàďÿàďřàď¿ àďžàěàďĄ, àďďàď¿àďšàě àďžàěà...  2657      157
galti #rap #nojotonews #trendingvideos #music       1263       75
#Rap #Song #Alag #hi #Vibe #hai \n#Nojotonews #...  1336       32
#RapOnline                                          1279      109
Aaj Hi Acha Hai | Storytelling\n\nReal happines...  1285       55
"yaadon ke patthar"\n\n#talk #travel #life_expe...  1188      147
#PoetryOnline                                       2909      476
#BabaYAGA #rap                                      1260       87
```

1.2 With respect to Love top ten content when a new user arrives

Love column defines popularity of a particular video. So when a new user comes, we recomment these top ten videos to continue with the services.

[12]:
```
print_top10('Caption','Love')
```

```
/home/sanjukta/anaconda/lib/python3.7/site-packages/ipykernel_launcher.py:2:
FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of
keys) will be deprecated, use a list instead.
```

[12]:
```
                                                    POST_ID  Watch_Views  \
Caption
#StoryOnline                                      106255962        59100
#PoetryOnline                                      90098113        25639
àďďàěàďÿàďřàď¿ àďžàěàďĄ, àďďàď¿àďšàě àďžàěà...    4633518        44679
Know who loved your story | Tag Nojotians #Nojo...    5251588        61196
#poetryonline\n\nAaj Bhi âĎðŔżâİďïÿŔ                4839623        18763
#Rap #Song #Alag #hi #Vibe #hai \n#Nojotonews #...    4649483        32065
https://youtu.be/SOzwxVsPx-8  kuch musibat hai ...    4843851        17202
Aaj Hi Acha Hai | Storytelling\n\nReal happines...    4661957        27143
#RapOnline                                          9574081        30702
galti #rap #nojotonews #trendingvideos #music         4649801        32335

                                                    Love  Comment
Caption
#StoryOnline                                        4155      561
#PoetryOnline                                       2909      476
àďďàěàďÿàďřàď¿ àďžàěàďĄ, àďďàď¿àďšàě àďžàěà...  2657      157
Know who loved your story | Tag Nojotians #Nojo...  1720      108
#poetryonline\n\nAaj Bhi âĎðŔżâİďïÿŔ              1400      132
#Rap #Song #Alag #hi #Vibe #hai \n#Nojotonews #...  1336       32
https://youtu.be/SOzwxVsPx-8  kuch musibat hai ...  1309      119
Aaj Hi Acha Hai | Storytelling\n\nReal happines...  1285       55
#RapOnline                                          1279      109
galti #rap #nojotonews #trendingvideos #music       1263       75
```

1.3 With respect to Comment top ten content when a new user arrives

Comment column defines popularity of a particular video. So when a new user comes, we recomment these top ten videos to continue with the services.

[13]:
```
print_top10('Caption','Comment')
```

```
/home/sanjukta/anaconda/lib/python3.7/site-packages/ipykernel_launcher.py:2:
FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of
keys) will be deprecated, use a list instead.
```

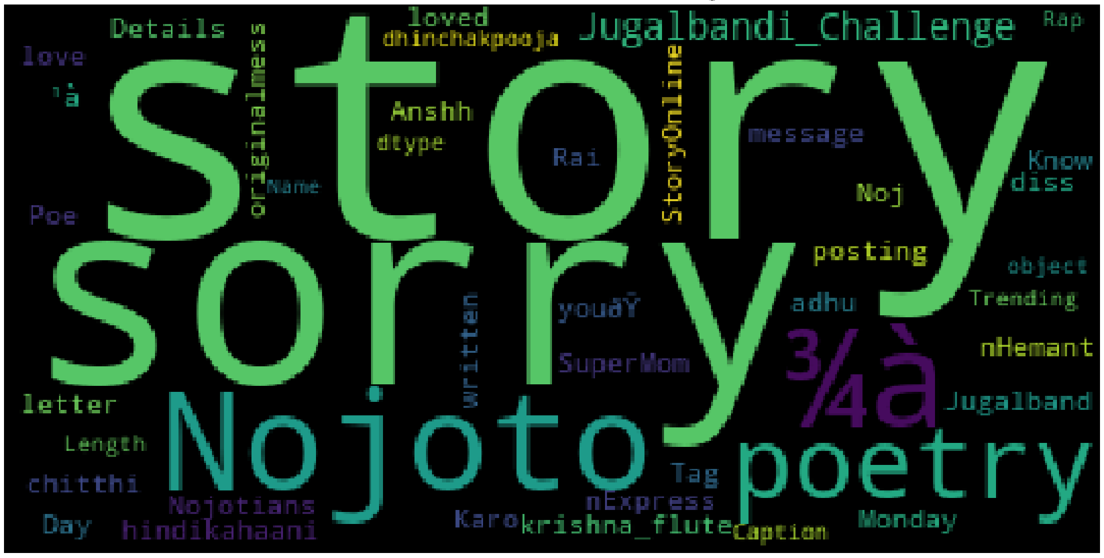
[13]:
```
                                                    POST_ID  Watch_Views  \
Caption
#StoryOnline                                      106255962        59100
#PoetryOnline                                      90098113        25639
#MessageForModi\ncomedy -only for fun \nàďÍàď£à...    5244171        19197
#PoetryOnline #SayaniChidiya #SonaUniyal #Mom &...    4962465        19695
#poetryonline                                       4957718         9787
Old Man - Lost Smile\n#StoryOnline \n#nojotofil...    5476527         1901
And I would choose you in a hundred lifetimes,i...    1090611         8081
Share Nojoto Suggestions in Captionðð\n\n...    5069530        14858
```

```
seedha bhopal se #bhopali #sourabhshresth #rapp...    3905472        15305
#Taameer #Nojoto  àďőàď¿àďĄ àďďàěàďřàě àďňàďž...    5309459        2545

                                                    Love  Comment
Caption
#StoryOnline                                        4155      561
#PoetryOnline                                       2909      476
#MessageForModi\ncomedy -only for fun \nàďĺàď£à...  1052      283
#PoetryOnline #SayaniChidiya #SonaUniyal #Mom &...  1045      240
#poetryonline                                       1057      218
Old Man - Lost Smile\n#StoryOnline \n#nojotofil...   336      209
And I would choose you in a hundred lifetimes,i...   900      195
Share Nojoto Suggestions in Captionðð\n\n...   502      189
seedha bhopal se #bhopali #sourabhshresth #rapp...  1034      187
#Taameer #Nojoto  àďőàď¿àďĄ àďďàěàďřàě àďňàďž...   342      185
```

[14]:
```python
plt.figure(figsize = (10,10))
stopwords = set(STOPWORDS)
wordcloud = WordCloud(background_color = 'black',stopwords =␣
 ↪stopwords,max_words = 1000,max_font_size = 120,random_state = 42).
 ↪generate(str(videos_df['Caption']))
plt.imshow(wordcloud)
plt.title('WORD CLOUD for Caption', fontsize = 20)
plt.axis('off')
plt.show()
```

WORD CLOUD for Caption



2 Comment Distribution

[15]:
```
from plotly.offline import init_notebook_mode, plot, iplot
import plotly.graph_objs as go
init_notebook_mode(connected=True)


init_notebook_mode(connected=True)

data = videos_df['Comment'].value_counts().sort_index(ascending=False)
trace = go.Bar(x = data.index,
               text = ['{:.1f} %'.format(val) for val in (data.values /␣
 ↪videos_df.shape[0] * 100)],
               textposition = 'auto',
               textfont = dict(color = '#000000'),
               y = data.values,
               )
# Create layout
layout = dict(title = 'Distribution Of {} Comment'.format(videos_df.shape[0]),
              xaxis = dict(title = 'Comment'),
              yaxis = dict(title = 'Count'))
# Create plot
fig = go.Figure(data=[trace], layout=layout)
iplot(fig)
```

3 Comment Distribution By Creator_Name

[16]:
```
# Number of ratings per book
data = videos_df['Creator_Name'].value_counts().sort_index(ascending=False)

# Create trace
trace = go.Histogram(x = data.values,
                     name = 'Creator_Name',
                     xbins = dict(start = 0,
                                  end = 50,
                                  size = 2))
# Create layout
layout = go.Layout(title = 'Distribution Of Number of Comments Per Creator_Name␣
 ↪(Clipped at 100)',
                   xaxis = dict(title = 'Number of Comments Per Creator_Name'),
                   yaxis = dict(title = 'Count'),
                   bargap = 0.2)

# Create plot
fig = go.Figure(data=[trace], layout=layout)
iplot(fig)
```

# Comment Distribution By Repost_Count

[17]:
```
# Number of ratings per user
data = videos_df['Repost_Count'].value_counts().sort_index(ascending=False)

# Create trace
trace = go.Histogram(x = data.values,
                     name = 'Comment',
                     xbins = dict(start = 0,
                                  end = 50,
                                  size = 2))
# Create layout
layout = go.Layout(title = 'Distribution Of Number of Comments with␣
 ↪Repost_Count',
                   xaxis = dict(title = 'Repost_Count'),
                   yaxis = dict(title = 'Count'),
                   bargap = 0.2)

# Create plot
fig = go.Figure(data=[trace], layout=layout)
iplot(fig)
```

4 Recommender System made easy with Scikit-Surprise

[77]:
```
from surprise import Reader, Dataset
reader = Reader()
data = Dataset.load_from_df(videos_df[['POST_ID', 'Caption','Comment']], reader)
```

[78]:
```
from surprise.model_selection import train_test_split

trainset, testset = train_test_split(data, test_size=0.2)
```

[79]:
```
from surprise import SVD, accuracy
algo = SVD()
algo.fit(trainset)
```

[79]:
```
<surprise.prediction_algorithms.matrix_factorization.SVD at 0x7f085a14ecc0>
```

[80]:
```
predictions = algo.test(testset)
```

5 Evaluation

Singular vector decomposition (SVD) shown here employs the use of gradient descent to minimize the squared error between predicted rating and actual rating, eventually getting the best model.

[81]:
```
from surprise import accuracy
accuracy.rmse(predictions)
```

```
RMSE: 50.3233
```

[81]:
```
50.32325506165117
```

6 Conclusion

1. With respect to Watch_Views top ten content when a new user arrives
2. With respect to Love top ten content when a new user arrives
3. With respect to Comment top ten content when a new user arrives
4. We have taken comments as a parameter to decide the more number of comments means more number of views
5. Then we used Scikitlearn-Surprise Recommender System algorithm to create a relationship between 'POST_ID', 'Caption','Comment'
6. The RMSE is 50.32% and we can further improve this while optimizing the hyperparameter tuning.

[ ]:
```

```