ADVANCED DATA MINING

Assignment Report

SUMANTH POBALA
UMID 71652304
DATA SCIENCE SEMESTER 1

DATA CLEANING 01 EXERCISE:

1. Normalizing Publisher and Title

1.1. For cleaning the title, I used a function with clean_title and by using the 'find' inbuilt function we will search for the symbols and keywords that we need to recognize and remove. After finding them I with help of split and join prebuilt functions we have cleaned the title.

```
    title = list(map(str.capitalize, title.split()))
    return ' '.join(title)

df['Title'] = df['Title'].apply(clean_title)
df.head()
```

Out[246]:

Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL	Cost
206	London	1879.0	S. Tinsley & Co.	Walter Forbes	A. A.	http://www.flickr.com/photos/britishlibrary/ta...	94.805614
216	London	1868.0	Virtue & Co.	All For Greed	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...	77.514704
218	London	1869.0	Bradbury, Evans & Co.	Love The Avenger	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta...	94.761914
472	London	1851.0	James Darling	Welsh Sketches, Chiefly Ecclesiastical, To The...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...	43.181231
480	London	1857.0	Wertheim & Macintosh	The World In Which I Live, And My Place In It	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta...	9.379887

Fig 1: Clean Publisher

1.1.1. For Cleaning the Publisher, Regex is being used to remove the symbols where we used the string replace function along with regex i.e., **str.replace('[^A-Za-z0-9]+', ' ')**, this regex cleans the publisher and returns the alphanumeric part of Publisher.

1.2. Now replacing the missing values (NaN) is done which can be completed with same replace function where we need to use the numpy and select the missing values with np.nan and replace with word 'UNK'. Code is **"replace (np.nan, 'UNK', regex=True)"**

```
df.Publisher = df.Publisher.str.replace('[^A-Za-z0-9]+', ' ')
import numpy as npy
df.Publisher = df.Publisher.replace(np.nan, 'UNK', regex=True)
df.Publisher[:10]
```

Out[276]:
```
0         S Tinsley Co
1            Virtue Co
2    Bradbury Evans Co
3         James Darling
4    Wertheim Macintosh
5     William Macintosh
6            The Author
7                   UNK
8                   UNK
9                   UNK
Name: Publisher, dtype: object
```

Fig 1.1. Cleaning Publisher and Replacing missing values

2. Clean Date of Publication and categorize as “Old” or “Modern”

2.1. Our first task in this is cleaning the Date of Publication, we can achieve this by using function and we will declare symbols in a new variable called ‘Unwanted Characters’, we use a for loop where we go character wise in the Date of Publication by taking the reference of Unwanted Characters. Then we apply the function with **apply(clean_dates),** where clean_dates is defined function.

```
df.head(4)
#df['Date of Publication'].head(10)
```

Out[279]:

	Identifier	Edition Statement	Place of Publication	Date of Publication	Publisher	Title	Author	Contributors	Corporate Author	Corporate Contributors	Former owner	Engraver	Issuance type
0	206	NaN	London	1879	S Tinsley Co	Walter Forbes. [A novel.] By A. A	A. A.	FORBES, Walter.	NaN	NaN	NaN	NaN	monographic
1	216	NaN	London; Virtue & Yorston	1868	Virtue Co	All for Greed. [A novel. The dedication signed...	A., A. A.	BLAZE DE BURY, Marie Pauline Rose - Baroness	NaN	NaN	NaN	NaN	monographic
2	218	NaN	London	1869	Bradbury Evans Co	Love the Avenger. By the author of “All for Gr...	A., A. A.	BLAZE DE BURY, Marie Pauline Rose - Baroness	NaN	NaN	NaN	NaN	monographic
3	472	NaN	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	Appleyard, Ernest Silvanus.	NaN	NaN	NaN	NaN	monographic

Fig 2.1. Cleaning Date of Publication

2.2. By using numpy we are allocating the dates to an array called doparray, by using the if loop we compare the date with 1750, where below 1750 the category is OLD and above 1750 the category is MODERN.

2.3.I used the threshold as year 1750 and compared the values with 1750 and allocated all the values to Publication Category.

Out[252]: 8287

In [280]:
```
df['Publication Catogery'] = cat
df['Publication Catogery'].head(10)
df.head(10)
```

Out[280]:

	Title	Author	Contributors	Corporate Author	Corporate Contributors	Former owner	Engraver	Issuance type	Flickr URL	Shelfmarks	Cost	Publication Catogery
	Walter Forbes. [A novel.] By A. A	A. A.	FORBES, Walter.	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com/photos/britishlibrary/ta...	British Library HMNTS 12641.b.30.	94.805614	Modern
	All for Greed. [A novel. The dedication signed...	A., A. A.	BLAZE DE BURY, Marie Pauline Rose - Baroness	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com/photos/britishlibrary/ta...	British Library HMNTS 12626.cc.2.	77.514704	Modern
	Love the Avenger. By the author of “All for Gr...	A., A. A.	BLAZE DE BURY, Marie Pauline Rose - Baroness	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com/photos/britishlibrary/ta...	British Library HMNTS 12625.dd.1.	94.761914	Modern
	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	Appleyard, Ernest Silvanus.	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com/photos/britishlibrary/ta...	British Library HMNTS 10369.bbb.15.	43.181231	Modern
	[The World in which I live, and my place in it...	A., E. S.	BROOME, John Henry.	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com/photos/britishlibrary/ta...	British Library HMNTS 9007.d.28.	9.379887	Modern
	[The World in which I live, and my place in it...	A., E. S.	BROOME, John Henry.	NaN	NaN	NaN	NaN	monographic	http://www.flickr.com/photos/britishlibrary/ta...	British Library HMNTS 9006.ee.10.	33.303515	Modern

Fig 2.2. Publication Category as Old and Modern

3. Cost value Normalization and Distribution

3.1. We import the package "***from*** *sklearn* ***import*** *preprocessing",* and numpy, by using the numpy we will store the values of cost in an array called ***x_array.***

3.1.1. We will round the cost values by using the round () function. And reshape the array as its not an 2D array.

3.2. Now we need to rescale the weight and categorize the cost values according to rescaled weights which can be obtained by using the MinMaxScaler() function which is being imported from ***sklearn.preprocessing,*** we use the created array and store it into an variable called weights and declare the scaler as MinMaxScaler() and by using scaler.fit_transform(weights) we will get the rescaled weights.

```
from sklearn.preprocessing import MinMaxScaler
import numpy as np
np.set_printoptions(threshold=np.nan)
weights = x_array
scaler = MinMaxScaler()
rescaled_weight = scaler.fit_transform(test2d)
rescaled_weight
```

```
[0.24],
[0.03],
[0.09],
[0.93],
[0.61],
[0.78],
[0.47],
[0.82],
[0.73],
[0.86],
[0.26],
[0.15],
[0.04],
[0.84],
[0.61],
[0.06],
[0.08],
[0.46],
[0.2 ],
```

Fig 3.1. Rescaled weights

3.3. Final step is creating the cost category based on the threshold of rescaled weights. I created an empty list costcategory and using the if, elif loops and for loops catogerized them into:

- Cheap (rescaledweight < 0.25)
- Average (rescaledweight >0.25 and rescaledweight < 0.75)
- Expensive (rescaledweight > 0.75)

```
In [54]: costcatogery
```

```
Out[54]: ['expensive',
          'expensive',
          'expensive',
          'average',
          'Cheap',
          'average',
          'average',
          'average',
          'expensive',
          'expensive',
          'average',
          'average',
          'average',
          'Cheap',
          'Cheap',
          'average',
          'average',
          'expensive',
          'average',
```

Fig 3.2 Cost Category

3.4.COMPLETE OUTPUT

In [292]:
```
e of Publication','Date of Publication','Publisher','Title','Author','Cost','Publication Catogery','CostCatogery'
```

Out[292]:

	Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Cost	Publication Catogery	CostCatogery
0	206	London	1879	S Tinsley Co	Walter Forbes	A. A.	94.805614	Modern	expensive
1	216	London; Virtue & Yorston	1868	Virtue Co	All For Greed	A., A. A.	77.514704	Modern	expensive
2	218	London	1869	Bradbury Evans Co	Love The Avenger	A., A. A.	94.761914	Modern	expensive
3	472	London	1851	James Darling	Welsh Sketches, Chiefly Ecclesiastical, To The...	A., E. S.	43.181231	Modern	average
4	480	London	1857	Wertheim Macintosh	The World In Which I Live, And My Place In It	A., E. S.	9.379887	Modern	Cheap
5	481	London	1875	William Macintosh	The World In Which I Live, And My Place In It	A., E. S.	33.303515	Modern	average
6	519	London	1872	The Author	Lagonells	A., F. E.	44.524752	Modern	average

In [ ]:

Fig 3.3. Complete output of exercise

DATA CLEANING EXERCISE 2

4. Creating Region Position extracted from RegionName attribute

4.1. We define a function to extract the position from region name which is situated in between square brackets [ ]. After finding them we create an array using numpy and append the values to it.

	State	RegionName	RegionPositon
0	Alabama[edit]\n	Auburn (Auburn University)[1]\n	1
1	Alabama[edit]\n	Florence (University of North Alabama)\n	NA
2	Alabama[edit]\n	Jacksonville (Jacksonville State University)[2]\n	2
3	Alabama[edit]\n	Livingston (University of West Alabama)[2]\n	2
4	Alabama[edit]\n	Montevallo (University of Montevallo)[2]\n	2

Fig 4.1 Region Position Extraction

4.2. Its same as the first one but university name is situated in between parentheses and we write a new function find the strings and later store into new variable called colg[].

	State	RegionName	RegionPositon	University Name
0	Alabama	Auburn	1	Auburn University
1	Alabama	Florence	NA	University of North Alabama
2	Alabama	Jacksonville	2	Jacksonville State University
3	Alabama	Livingston	2	University of West Alabama
4	Alabama	Montevallo	2	University of Montevallo

Fig 4.2. University Name Extraction

4.3. Complete Output

This is the complete output of exercise 2

	State	RegionName	RegionPositon	University Name
0	Alabama	Auburn	1	Auburn University
1	Alabama	Florence	NA	University of North Alabama
2	Alabama	Jacksonville	2	Jacksonville State University
3	Alabama	Livingston	2	University of West Alabama
4	Alabama	Montevallo	2	University of Montevallo
5	Alabama	Troy	2	Troy University
6	Alabama	Tuscaloosa	3	University of Alabama, Stillman College, Shelt...
7	Alabama	Tuskegee	5	Tuskegee University
8	Alaska	Fairbanks	2	University of Alaska Fairbanks
9	Arizona	Flagstaff	6	Northern Arizona University

Fig 4.3. Complete Output

DATA CLEANING EXERCISE 3

5. **Working on Olympics data**

5.1. Firstly we need to extract the tag from the country in given csv file so to achieve that we will slice the first three letters of the country.

```
['AFG',
 'ALG',
 'ARG',
 'ARM',
 'AUS',
 'AUS',
 'AUS',
 'AZE',
 'BAH',
 'BAH',
 'BAR',
 'BEL',
 'BEL',
 'BER',
 'BOH',
 'BOT',
 'BRA',
 'BRI',
 'BUL',
```

Fig 5.1 Extracting tag names from country

5.2.Next we need to get the country name and remove the tag from it to obtain cleaned data and we will use the split function and append the country names to a separate list and we will replace the country list in dataframe with newly created list so that tags wont appear in the country name.

```
['Afghanistan',
 'Algeria',
 'Argentina',
 'Armenia',
 'Australasia',
 'Australia',
 'Austria',
 'Azerbaijan',
 'Bahamas',
 'Bahrain',
 'Barbados',
 'Belarus',
 'Belgium',
 'Bermuda',
 'Bohemia',
 'Botswana',
 'Brazil',
 'British',
 'Bulgaria',
```

Fig 5.2. Extracting Country name without tags

5.3. Finally we are creating a new dataframe ***"extracted_df"*** which contain Country Tags ,Summer Olympics, Winter Olympics and Combined Total (Which we need to calculate).

	Tag	Summer Olympics	Winter Olympics
0	AFG	13	0
1	ALG	12	3
2	ARG	23	18
3	ARM	5	6
4	AUS	2	0

Fig 5.3 New Dataframe

5.4.Now we need to calculate the Combined total of medals which is an addition of Summer Olympics and Winter Olympics. We create a new attribute 'Combined Total" and add the medals of summer and winter Olympics and store it into newly created dataframe.

	Tag	Summer Olympics	Winter Olympics	Combined_Total
0	AFG	13	0	13
1	ALG	12	3	15
2	ARG	23	18	41
3	ARM	5	6	11
4	AUS	2	0	2

Fig 5.4. Data Frame with required details