

Advanced data mining

Assignment Report



**SUMANTH POBALA**

**UMID 71652304**

**DATA SCIENCE SEMESTER 1**

**DATA CLEANING 01 EXERCISE:**

1. **Normalizing Publisher and Title**
   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.



Fig 1: Clean Publisher

* + 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. 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)”**

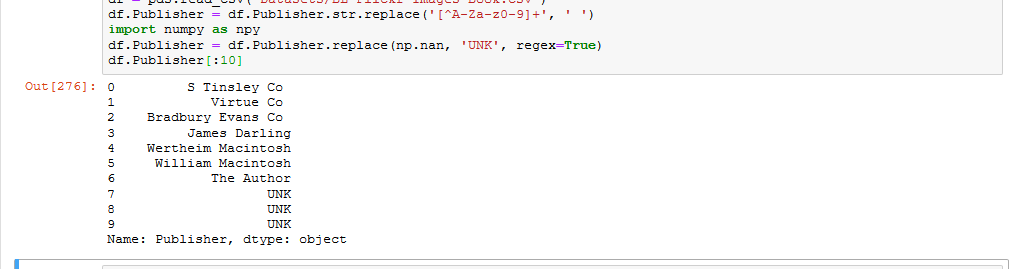


Fig 1.1. Cleaning Publisher and Replacing missing values

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

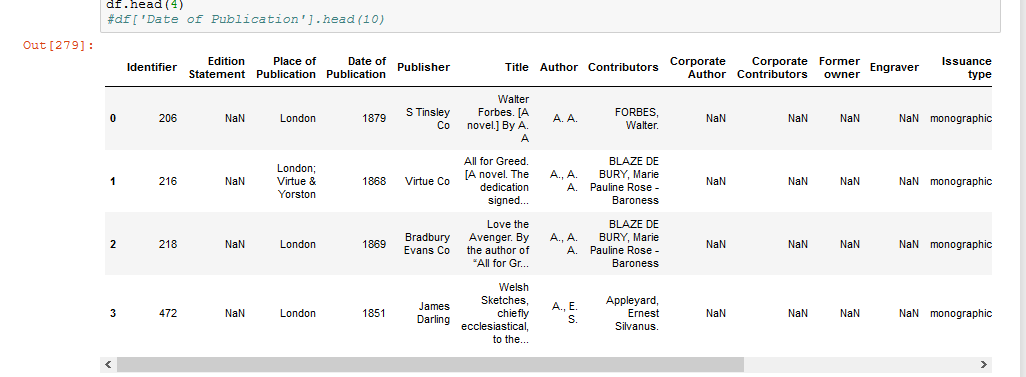
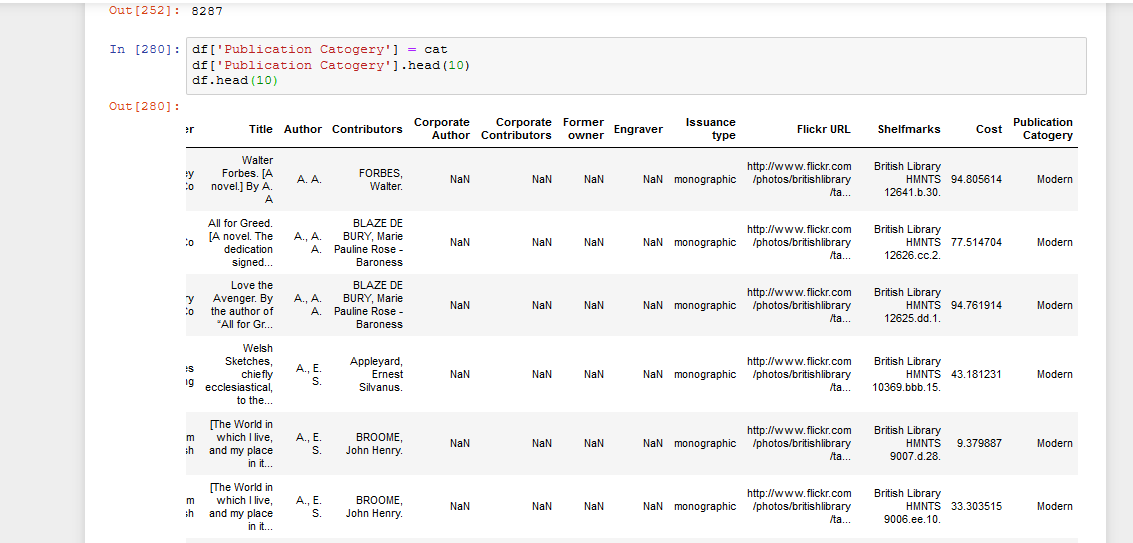


Fig 2.1. Cleaning Date of Publication

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

Fig 2.2. Publication Category as Old and Modern

1. **Cost value Normalization and Distribution**
   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.*** 
      1. We will round the cost values by using the round () function. And reshape the array as its not an 2D array.
   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.

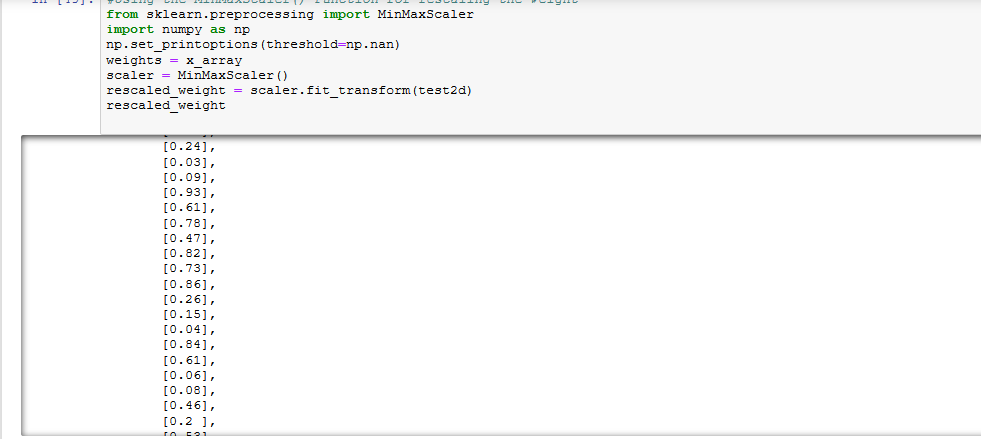
******

Fig 3.1. Rescaled weights

* 1. 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)

****

Fig 3.2 Cost Category

* 1. **COMPLETE OUTPUT**

****

Fig 3.3. Complete output of exercise

**DATA CLEANING EXERCISE 2**

1. **Creating Region Position extracted from RegionName attribute**
   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.



Fig 4.1 Region Position Extraction

* 1. 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[].

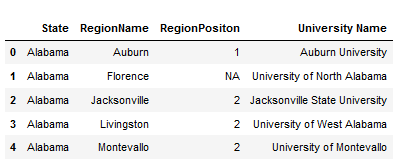


Fig 4.2. University Name Extraction

* 1. **Complete Output**

This is the complete output of exercise 2

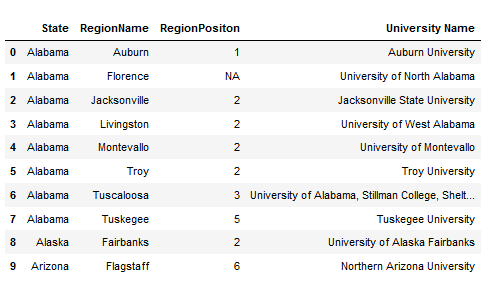
****

Fig 4.3. Complete Output

**DATA CLEANING EXERCISE 3**

1. **Working on Olympics data** 
   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.

****

Fig 5.1 Extracting tag names from country

* 1. 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.



Fig 5.2. Extracting Country name without tags

* 1. 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).

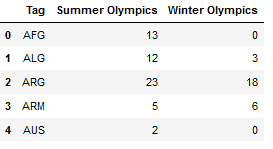


Fig 5.3 New Dataframe

* 1. 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.

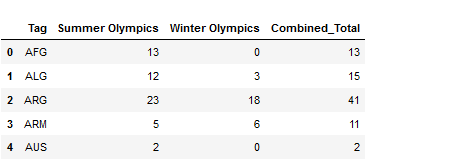


Fig 5.4. Data Frame with required details