Assignment 2

Name : Sanket Rajaram Mote

PRN : 2019BTECS00113

Batch : T5

Course : Software Engineering Tools Lab

Anaconda

1. Original author

Ans. Peter Wang and Travis Oliphant

2. Developers

Ans. Anaconda, Inc. (previously Continuum Analytics)

3. Initial release

Ans. 0.8.0/17 July 2012

4. Stable release

Ans. 2021.11 / 17 November 2021

5. Repository (with cloud support)

Ans. Anaconda Repository is an enterprise server on your network or your private cloud where open source and proprietary packages may be stored, retrieved, and shared.

6. Written in (Languages)

Ans. Python

7. Operating System support

Ans. Windows, Linux, Mac OS

8. Platform, portability

Ans. Platform: Anaconda Navigator

9. Available in (Total languages)

Ans. Python & R

10.List of languages supported

Ans. Python & R

11.Type (Programming tool, integrated development environment etc.)

Ans. Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, etc.), that aims to simplify package management and deployment.

12.Website

Ans. https://www.anaconda.com/

13.Features

Ans.

- It has more than 1500 Python/R data science packages.
- Anaconda simplifies package management and deployment.
- It has tools to easily collect data from sources using machine learning and AI.
- It is free and open-source.

14.Size (in MB, GB etc.)

Ans. The current download size sits at around 600MB and will take up over 2GB of disk space once installed

15.Type of software (Open source/License):

Ans. Free and Open-Software Software.

16.If License- Provide details.

Ans. Anaconda Individual Edition 2021.11.

17.Latest version

Ans.

18.Cloud support (Yes/No)

Ans. Yes

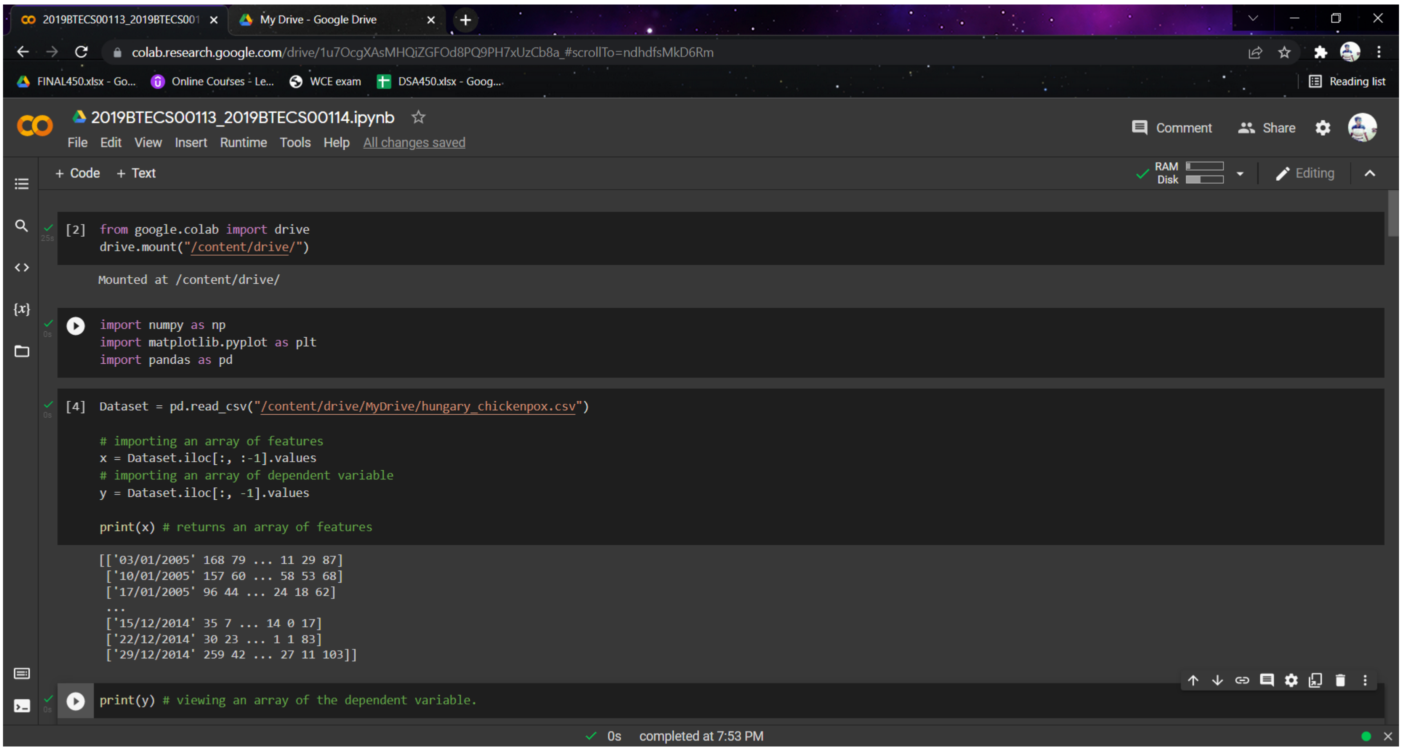
I. Implement linear regression problem using Google colab (Perform pre-processing, training, and testing)

Dataset 6- https://archive.ics.uci.edu/ml/datasets/Hungarian+Chickenpox+Cases

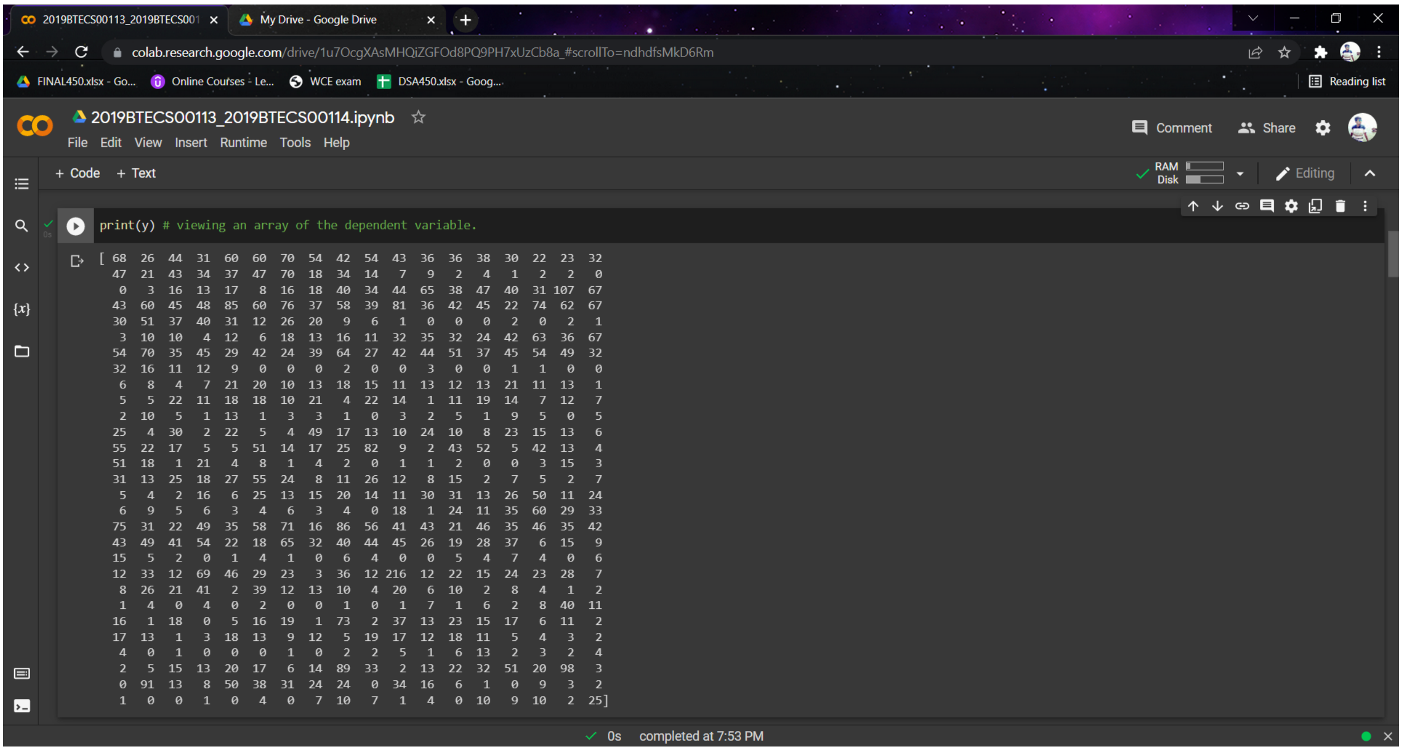
Implementation:

1.Pre-processing:

a. Getting the dataset, reading it, and importing required libraries.


2019BTECS00113_2019BTECS001
My Drive - Google Drive
colab.research.google.com/drive/1u7OcgXAsMHQiZGFOd8PQ9PH7xUzCb8a_#scrollTo=ndhdfsMkD6Rm
FINAL450.xlsx - Go...
Online Courses - Le...
WCE exam
DSA450.xlsx - Goog...
Reading list
2019BTECS00113_2019BTECS00114.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing

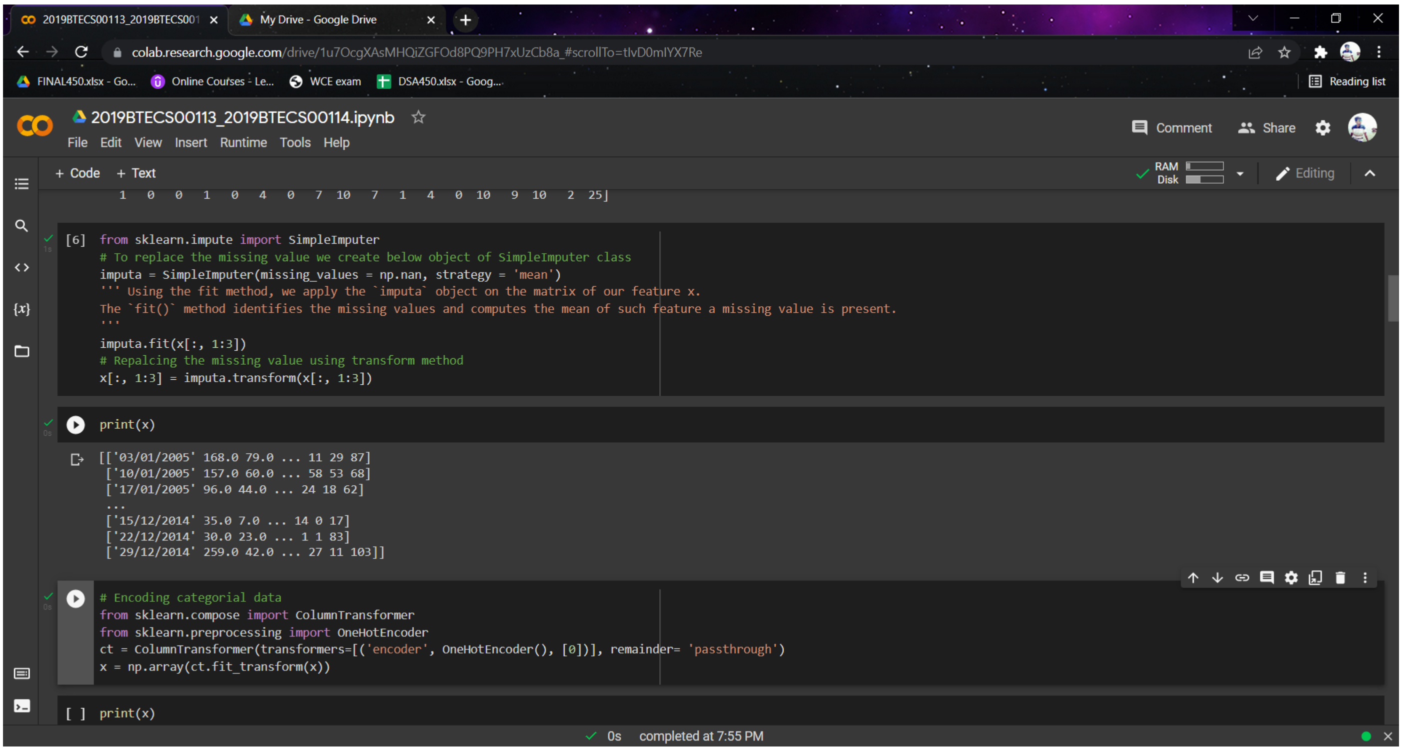
```
[2] from google.colab import drive
    drive.mount("/content/drive/")

    Mounted at /content/drive/

    import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd

[4] Dataset = pd.read_csv("/content/drive/MyDrive/hungary_chickenpox.csv")

    # importing an array of features
    x = Dataset.iloc[:, :-1].values
    # importing an array of dependent variable
    y = Dataset.iloc[:, -1].values

    print(x) # returns an array of features

    [['03/01/2005' 168 79 ... 11 29 87]
     ['10/01/2005' 157 60 ... 58 53 68]
     ['17/01/2005' 96 44 ... 24 18 62]
     ...
     ['15/12/2014' 35 7 ... 14 0 17]
     ['22/12/2014' 30 23 ... 1 1 83]
     ['29/12/2014' 259 42 ... 27 11 103]]

    print(y) # viewing an array of the dependent variable.
```



```
print(y) # viewing an array of the dependent variable.
```

```
[ 68  26  44  31  60  60  70  54  42  54  43  36  36  38  30  22  23  32
  47  21  43  34  37  47  70  18  34  14   7   9   2   4   1   2   2   0
   0   3  16  13  17   8  16  18  40  34  44  65  38  47  40  31 107  67
  43  60  45  48  85  60  76  37  58  39  81  36  42  45  22  74  62  67
  30  51  37  40  31  12  26  20   9   6   1   0   0   0   2   0   2   1
   3  10  10   4  12   6  18  13  16  11  32  35  32  24  42  63  36  67
  54  70  35  45  29  42  24  39  64  27  42  44  51  37  45  54  49  32
  32  16  11  12   9   0   0   0   2   0   0   3   0   0   1   1   0   0
   6   8   4   7  21  20  10  13  18  15  11  13  12  13  21  11  13   1
   5   5  22  11  18  18  10  21   4  22  14   1  11  19  14   7  12   7
   2  10   5   1  13   1   3   3   1   0   3   2   5   1   9   5   0   5
  25   4  30   2  22   5   4  49  17  13  10  24  10   8  23  15  13   6
  55  22  17   5   5  51  14  17  25  82   9   2  43  52   5  42  13   4
  51  18   1  21   4   8   1   4   2   0   1   1   2   0   0   3  15   3
  31  13  25  18  27  55  24   8  11  26  12   8  15   2   7   5   2   7
   5   4   2  16   6  25  13  15  20  14  11  30  31  13  26  50  11  24
   6   9   5   6   3   4   6   3   4   0  18   1  24  11  35  60  29  33
  75  31  22  49  35  58  71  16  86  56  41  43  21  46  35  46  35  42
  43  49  41  54  22  18  65  32  40  44  45  26  19  28  37   6  15   9
  15   5   2   0   1   4   1   0   6   4   0   0   5   4   7   4   0   6
  12  33  12  69  46  29  23   3  36  12 216  12  22  15  24  23  28   7
   8  26  21  41   2  39  12  13  10   4  20   6  10   2   8   4   1   2
   1   4   0   4   0   2   0   0   1   0   1   7   1   6   2   8  40  11
  16   1  18   0   5  16  19   1  73   2  37  13  23  15  17   6  11   2
  17  13   1   3  18  13   9  12   5  19  17  12  18  11   5   4   3   2
   4   0   1   0   0   0   1   0   2   2   5   1   6  13   2   3   2   4
   2   5  15  13  20  17   6  14  89  33   2  13  22  32  51  20  98   3
   0  91  13   8  50  38  31  24  24   0  34  16   6   1   0   9   3   2
   1   0   0   1   0   4   0   7  10   7   1   4   0  10   9  10   2  25]
```

b. Handling missing data.



```
   1   0   0   1   0   4   0   7  10   7   1   4   0  10   9  10   2  25]
```

```
[6] from sklearn.impute import SimpleImputer
    # To replace the missing value we create below object of SimpleImputer class
    imputa = SimpleImputer(missing_values = np.nan, strategy = 'mean')
    ''' Using the fit method, we apply the `imputa` object on the matrix of our feature x.
    The `fit()` method identifies the missing values and computes the mean of such feature a missing value is present.
    '''
    imputa.fit(x[:, 1:3])
    # Repalcing the missing value using transform method
    x[:, 1:3] = imputa.transform(x[:, 1:3])
```

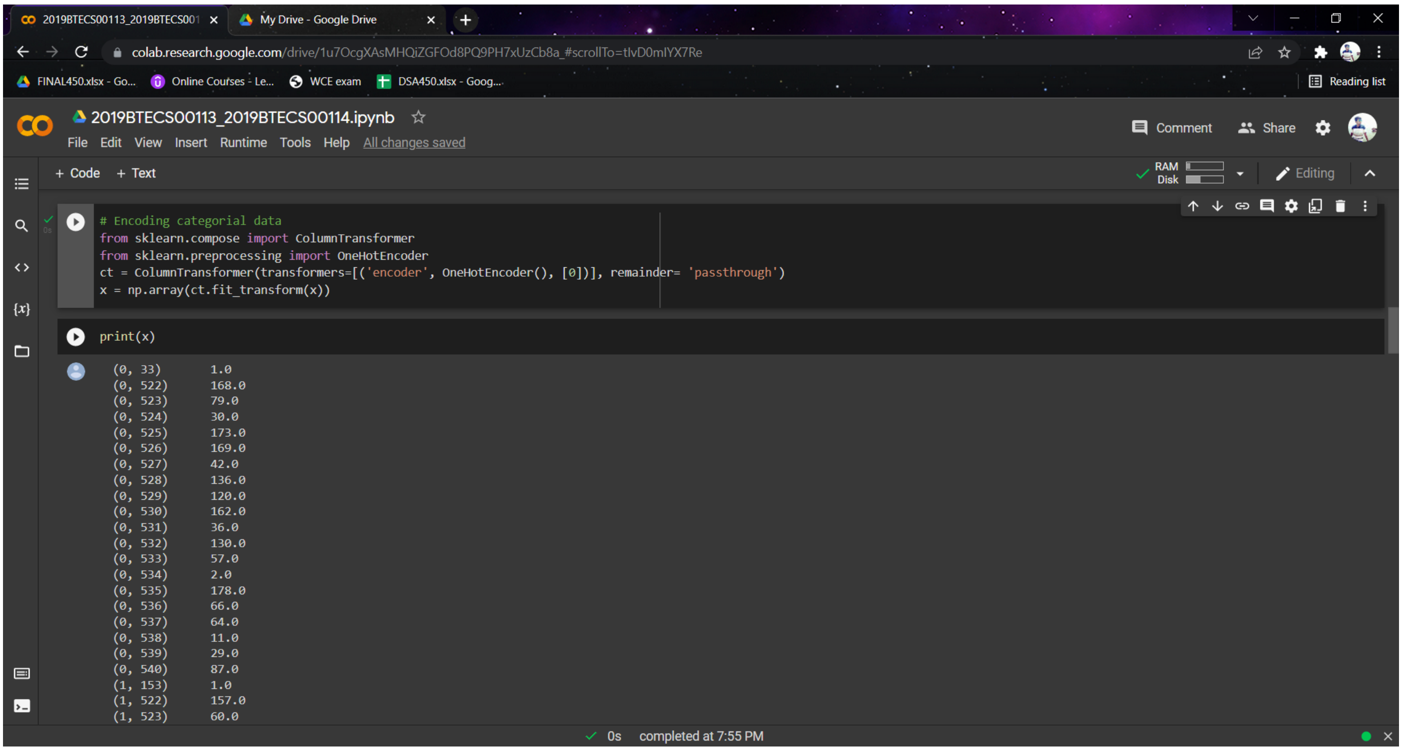
```
print(x)
```

```
[['03/01/2005' 168.0 79.0 ... 11 29 87]
 ['10/01/2005' 157.0 60.0 ... 58 53 68]
 ['17/01/2005' 96.0 44.0 ... 24 18 62]
 ...
 ['15/12/2014' 35.0 7.0 ... 14 0 17]
 ['22/12/2014' 30.0 23.0 ... 1 1 83]
 ['29/12/2014' 259.0 42.0 ... 27 11 103]]
```

```
# Encoding categorial data
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder= 'passthrough')
x = np.array(ct.fit_transform(x))
```

```
[ ] print(x)
```

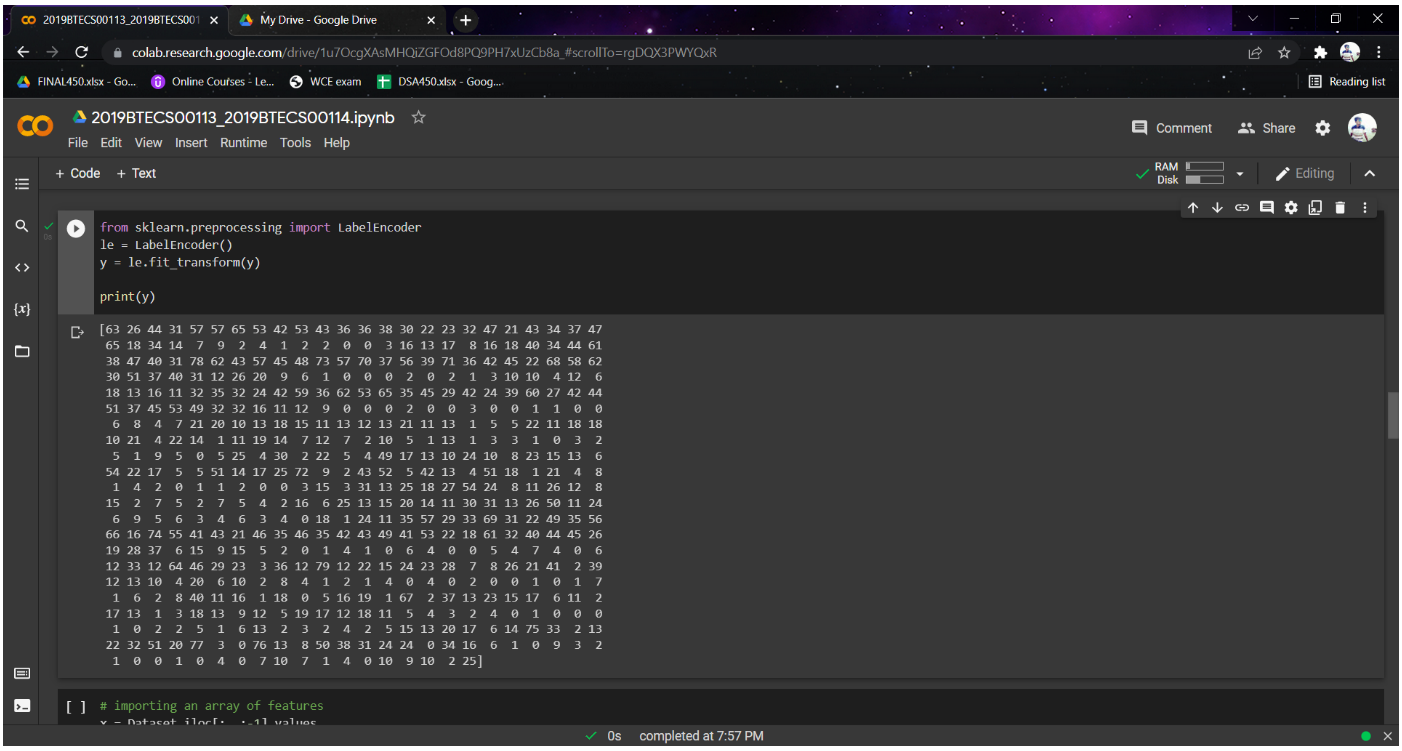
c. Encoding Categorial Data


2019BTECS00113_2019BTECS00114.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
RAM Disk
Editing
0s completed at 7:55 PM

```
# Encoding categorial data
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder= 'passthrough')
x = np.array(ct.fit_transform(x))
```

```
print(x)
```

```
  (0, 33)	1.0
  (0, 522)	168.0
  (0, 523)	79.0
  (0, 524)	30.0
  (0, 525)	173.0
  (0, 526)	169.0
  (0, 527)	42.0
  (0, 528)	136.0
  (0, 529)	120.0
  (0, 530)	162.0
  (0, 531)	36.0
  (0, 532)	130.0
  (0, 533)	57.0
  (0, 534)	2.0
  (0, 535)	178.0
  (0, 536)	66.0
  (0, 537)	64.0
  (0, 538)	11.0
  (0, 539)	29.0
  (0, 540)	87.0
  (1, 153)	1.0
  (1, 522)	157.0
  (1, 523)	60.0
```

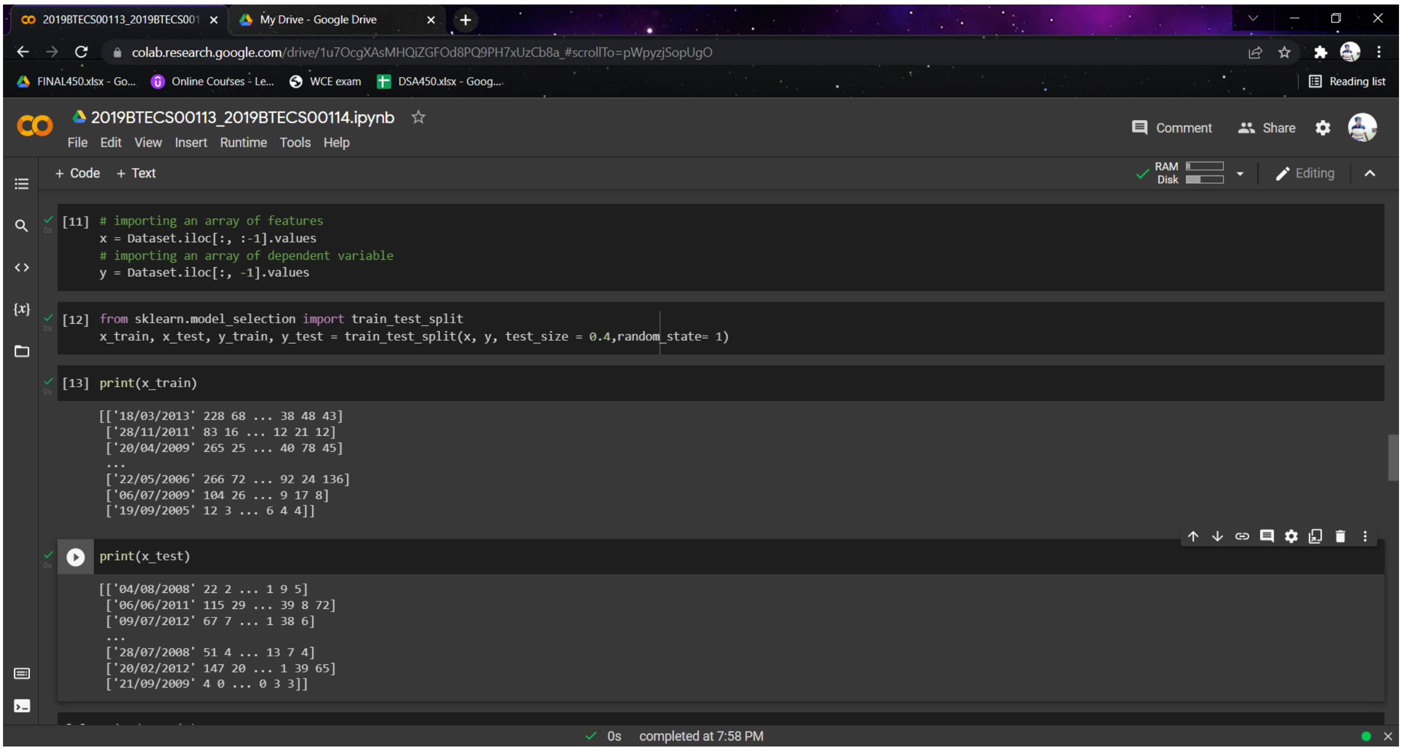

2019BTECS00113_2019BTECS00114.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Comment Share
RAM Disk
Editing
0s completed at 7:57 PM

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)

print(y)
```

```
[63 26 44 31 57 57 65 53 42 53 43 36 36 38 30 22 23 32 47 21 43 34 37 47
 65 18 34 14  7  9  2  4  1  2  2  0  0  3 16 13 17  8 16 18 40 34 44 61
 38 47 40 31 78 62 43 57 45 48 73 57 70 37 56 39 71 36 42 45 22 68 58 62
 30 51 37 40 31 12 26 20  9  6  1  0  0  0  2  0  2  1  3 10 10  4 12  6
 18 13 16 11 32 35 32 24 42 59 36 62 53 65 35 45 29 42 24 39 60 27 42 44
 51 37 45 53 49 32 32 16 11 12  9  0  0  0  2  0  0  3  0  0  1  1  0  0
  6  8  4  7 21 20 10 13 18 15 11 13 12 13 21 11 13  1  5  5 22 11 18 18
 10 21  4 22 14  1 11 19 14  7 12  7  2 10  5  1 13  1  3  3  1  0  3  2
  5  1  9  5  0  5 25  4 30  2 22  5  4 49 17 13 10 24 10  8 23 15 13  6
 54 22 17  5  5 51 14 17 25 72  9  2 43 52  5 42 13  4 51 18  1 21  4  8
  1  4  2  0  1  1  2  0  0  3 15  3 31 13 25 18 27 54 24  8 11 26 12  8
 15  2  7  5  2  7  5  4  2 16  6 25 13 15 20 14 11 30 31 13 26 50 11 24
  6  9  5  6  3  4  6  3  4  0 18  1 24 11 35 57 29 33 69 31 22 49 35 56
 66 16 74 55 41 43 21 46 35 46 35 42 43 49 41 53 22 18 61 32 40 44 45 26
 19 28 37  6 15  9 15  5  2  0  1  4  1  0  6  4  0  0  5  4  7  4  0  6
 12 33 12 64 46 29 23  3 36 12 79 12 22 15 24 23 28  7  8 26 21 41  2 39
 12 13 10  4 20  6 10  2  8  4  1  2  1  4  0  4  0  2  0  0  1  0  1  7
  1  6  2  8 40 11 16  1 18  0  5 16 19  1 67  2 37 13 23 15 17  6 11  2
 17 13  1  3 18 13  9 12  5 19 17 12 18 11  5  4  3  2  4  0  1  0  0  0
  1  0  2  2  5  1  6 13  2  3  2  4  2  5 15 13 20 17  6 14 75 33  2 13
 22 32 51 20 77  3  0 76 13  8 50 38 31 24 24  0 34 16  6  1  0  9  3  2
  1  0  0  1  0  4  0  7 10  7  1  4  0 10  9 10  2 25]
```

```
# importing an array of features
x = Dataset.iloc[:, :-1].values
```

2. Training and Testing



```
[11] # importing an array of features
     x = Dataset.iloc[:, :-1].values
     # importing an array of dependent variable
     y = Dataset.iloc[:, -1].values
```

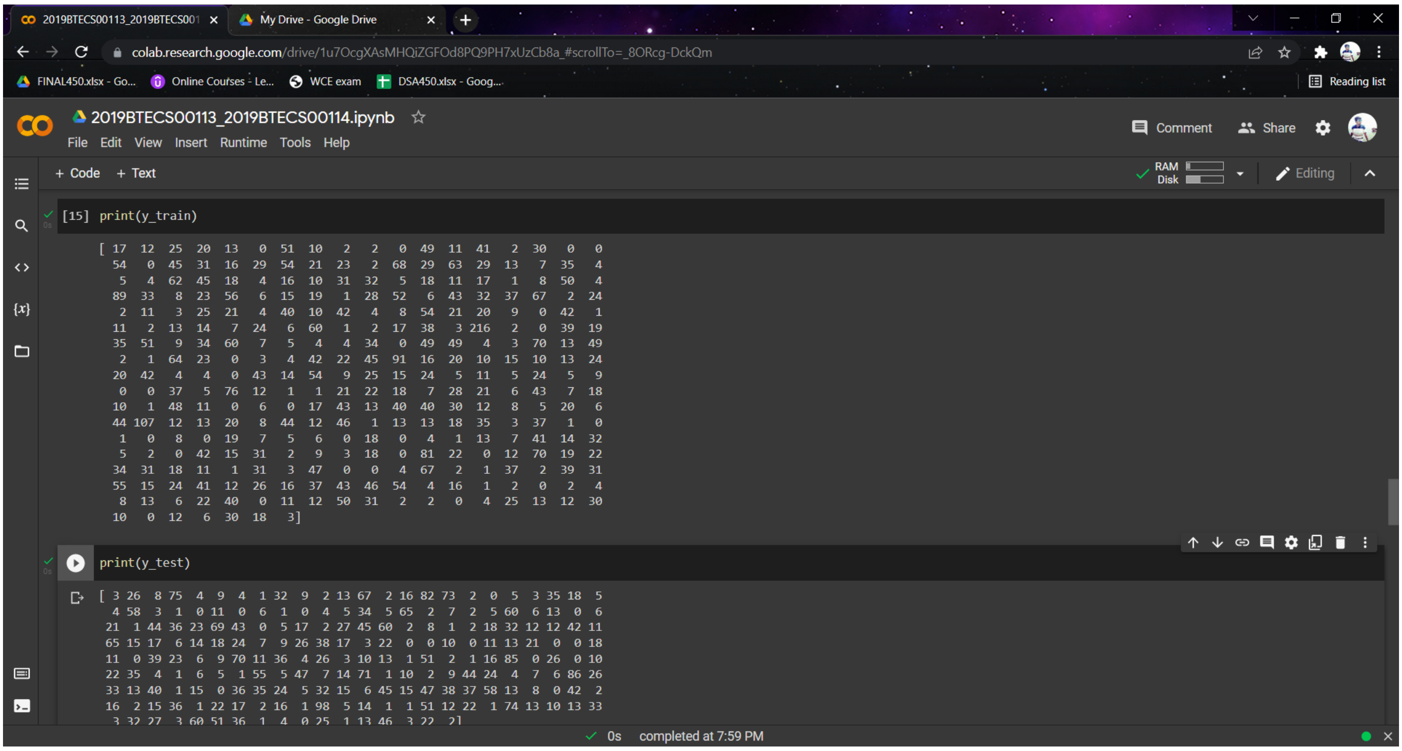
```
[12] from sklearn.model_selection import train_test_split
     x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.4,random_state= 1)
```

```
[13] print(x_train)
```

```
[['18/03/2013' 228 68 ... 38 48 43]
 ['28/11/2011' 83 16 ... 12 21 12]
 ['20/04/2009' 265 25 ... 40 78 45]
 ...
 ['22/05/2006' 266 72 ... 92 24 136]
 ['06/07/2009' 104 26 ... 9 17 8]
 ['19/09/2005' 12 3 ... 6 4 4]]
```

```
print(x_test)
```

```
[['04/08/2008' 22 2 ... 1 9 5]
 ['06/06/2011' 115 29 ... 39 8 72]
 ['09/07/2012' 67 7 ... 1 38 6]
 ...
 ['28/07/2008' 51 4 ... 13 7 4]
 ['20/02/2012' 147 20 ... 1 39 65]
 ['21/09/2009' 4 0 ... 0 3 3]]
```



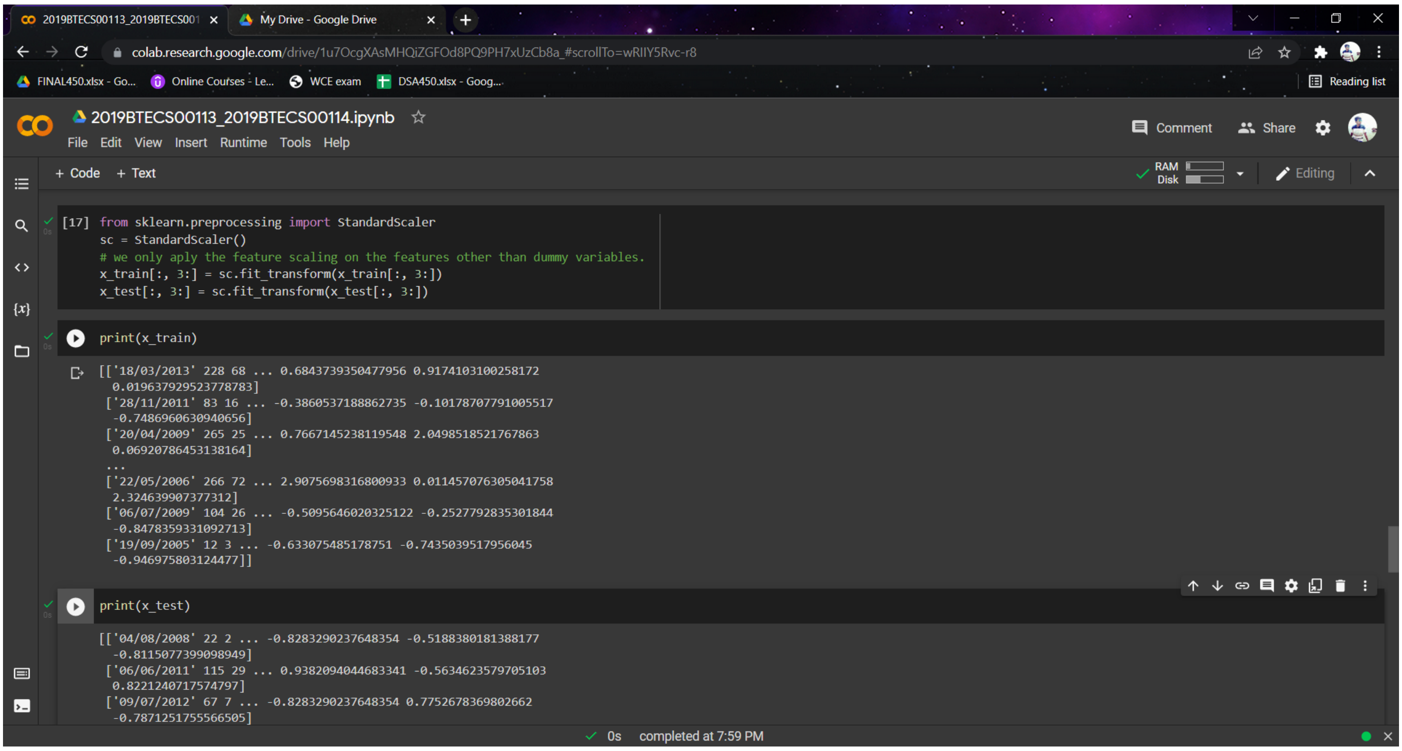
```
[15] print(y_train)
```

```
[ 17  12  25  20  13   0  51  10   2   2   0  49  11  41   2  30   0   0
  54   0  45  31  16  29  54  21  23   2  68  29  63  29  13   7  35   4
   5   4  62  45  18   4  16  10  31  32   5  18  11  17   1   8  50   4
  89  33   8  23  56   6  15  19   1  28  52   6  43  32  37  67   2  24
   2  11   3  25  21   4  40  10  42   4   8  54  21  20   9   0  42   1
  11   2  13  14   7  24   6  60   1   2  17  38   3 216   2   0  39  19
  35  51   9  34  60   7   5   4   4  34   0  49  49   4   3  70  13  49
   2   1  64  23   0   3   4  42  22  45  91  16  20  10  15  10  13  24
  20  42   4   4   0  43  14  54   9  25  15  24   5  11   5  24   5   9
   0   0  37   5  76  12   1   1  21  22  18   7  28  21   6  43   7  18
  10   1  48  11   0   6   0  17  43  13  40  40  30  12   8   5  20   6
  44 107  12  13  20   8  44  12  46   1  13  13  18  35   3  37   1   0
   1   0   8   0  19   7   5   6   0  18   0   4   1  13   7  41  14  32
   5   2   0  42  15  31   2   9   3  18   0  81  22   0  12  70  19  22
  34  31  18  11   1  31   3  47   0   0   4  67   2   1  37   2  39  31
  55  15  24  41  12  26  16  37  43  46  54   4  16   1   2   0   2   4
   8  13   6  22  40   0  11  12  50  31   2   2   0   4  25  13  12  30
  10   0  12   6  30  18   3]
```

```
print(y_test)
```

```
[ 3 26  8 75  4  9  4  1 32  9  2 13 67  2 16 82 73  2  0  5  3 35 18  5
  4 58  3  1  0 11  0  6  1  0  4  5 34  5 65  2  7  2  5 60  6 13  0  6
 21  1 44 36 23 69 43  0  5 17  2 27 45 60  2  8  1  2 18 32 12 12 42 11
 65 15 17  6 14 18 24  7  9 26 38 17  3 22  0  0 10  0 11 13 21  0  0 18
 11  0 39 23  6  9 70 11 36  4 26  3 10 13  1 51  2  1 16 85  0 26  0 10
 22 35  4  1  6  5  1 55  5 47  7 14 71  1 10  2  9 44 24  4  7  6 86 26
 33 13 40  1 15  0 36 35 24  5 32 15  6 45 15 47 38 37 58 13  8  0 42  2
 16  2 15 36  1 22 17  2 16  1 98  5 14  1  1 51 12 22  1 74 13 10 13 33
  3 32 27  3 60 51 36  1  4  0 25  1 13 46  3 22 21]
```

3. Feature Scaling



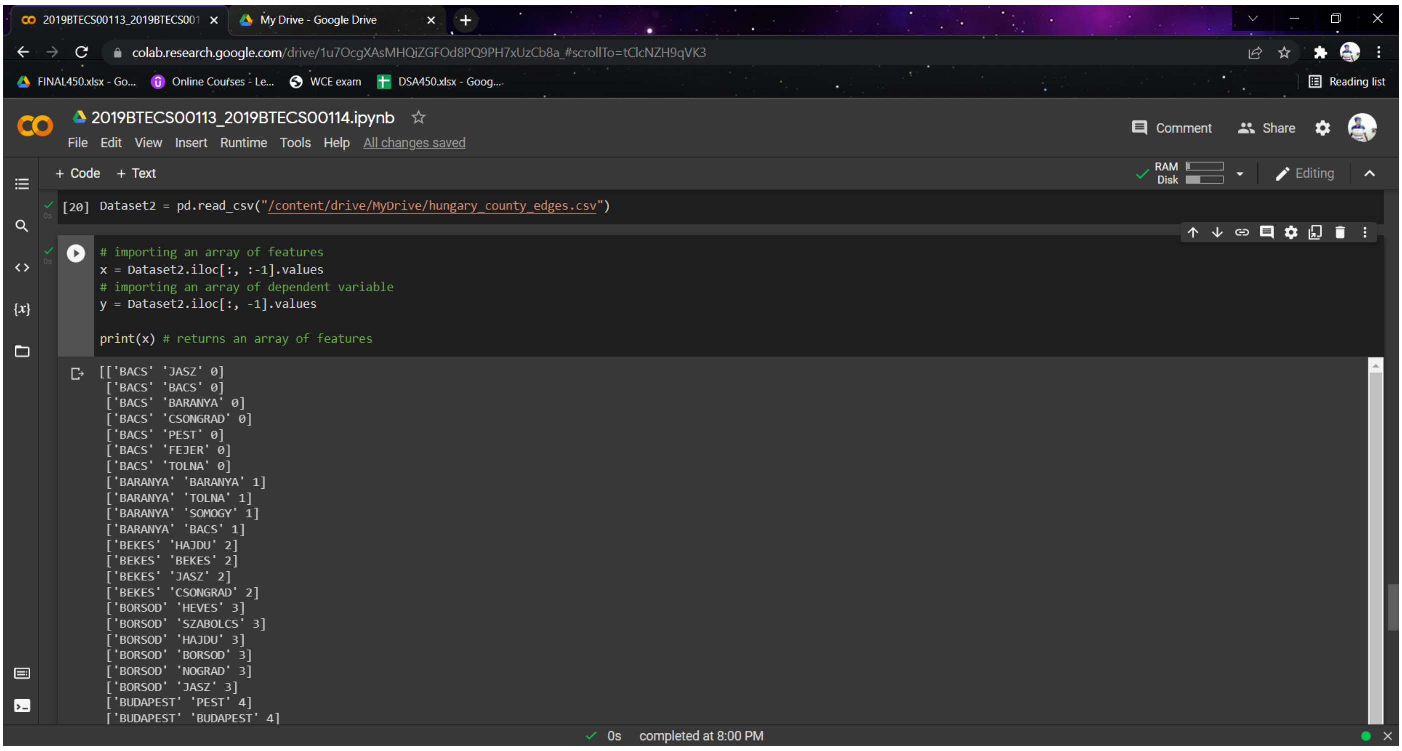
```
[17] from sklearn.preprocessing import StandardScaler
     sc = StandardScaler()
     # we only aply the feature scaling on the features other than dummy variables.
     x_train[:, 3:] = sc.fit_transform(x_train[:, 3:])
     x_test[:, 3:] = sc.fit_transform(x_test[:, 3:])
```

```
print(x_train)
```

```
[['18/03/2013' 228 68 ... 0.6843739350477956 0.9174103100258172
  0.019637929523778783]
 ['28/11/2011' 83 16 ... -0.3860537188862735 -0.10178707791005517
  -0.7486960630940656]
 ['20/04/2009' 265 25 ... 0.7667145238119548 2.0498518521767863
  0.06920786453138164]
 ...
 ['22/05/2006' 266 72 ... 2.9075698316800933 0.011457076305041758
  2.324639907377312]
 ['06/07/2009' 104 26 ... -0.5095646020325122 -0.2527792835301844
  -0.8478359331092713]
 ['19/09/2005' 12 3 ... -0.633075485178751 -0.7435039517956045
  -0.946975803124477]]
```

```
print(x_test)
```

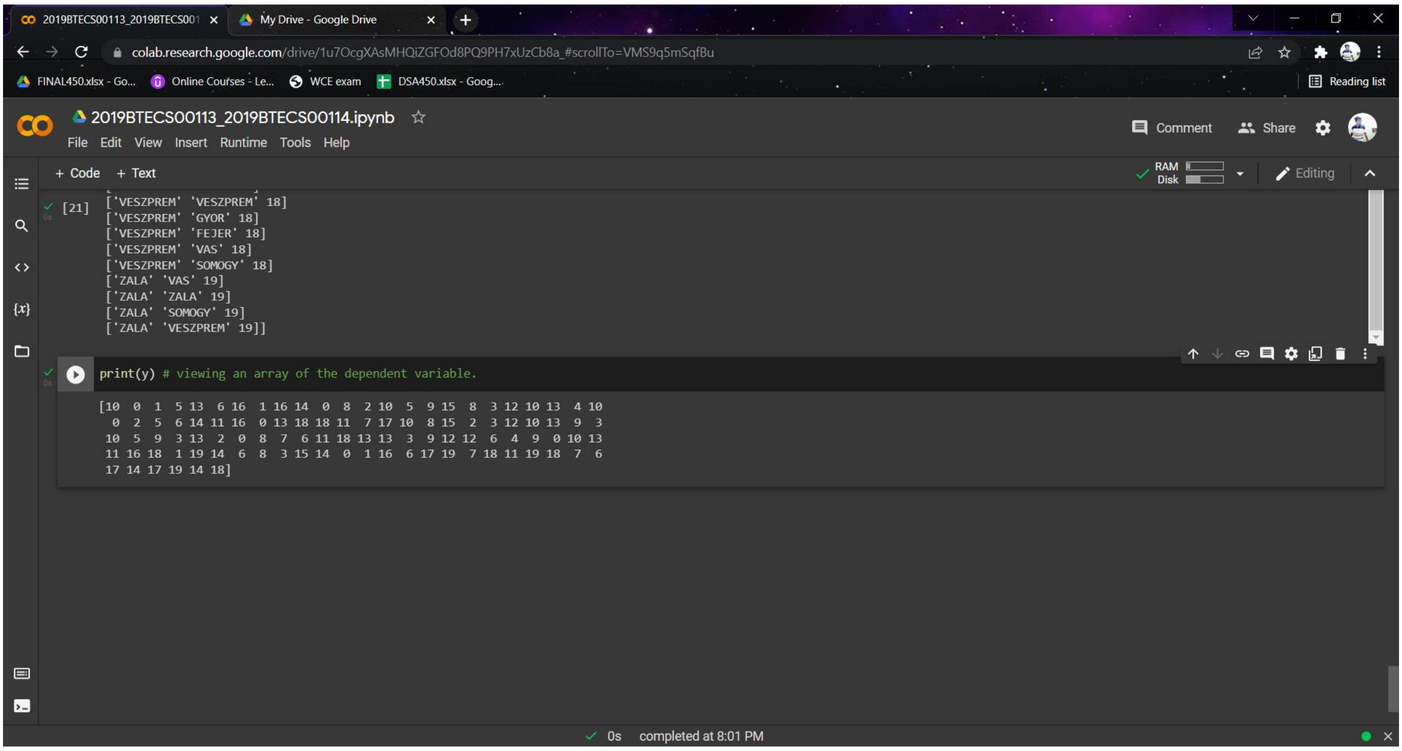
```
[['04/08/2008' 22 2 ... -0.8283290237648354 -0.5188380181388177
  -0.8115077399098949]
 ['06/06/2011' 115 29 ... 0.9382094044683341 -0.5634623579705103
  0.8221240717574797]
 ['09/07/2012' 67 7 ... -0.8283290237648354 0.7752678369802662
  -0.7871251755566505]
```



```
[20] Dataset2 = pd.read_csv("/content/drive/MyDrive/hungary_county_edges.csv")
```

```
# importing an array of features
x = Dataset2.iloc[:, :-1].values
# importing an array of dependent variable
y = Dataset2.iloc[:, -1].values

print(x) # returns an array of features
```

```
[['BACS' 'JASZ' 0]
 ['BACS' 'BACS' 0]
 ['BACS' 'BARANYA' 0]
 ['BACS' 'CSONGRAD' 0]
 ['BACS' 'PEST' 0]
 ['BACS' 'FEJER' 0]
 ['BACS' 'TOLNA' 0]
 ['BARANYA' 'BARANYA' 1]
 ['BARANYA' 'TOLNA' 1]
 ['BARANYA' 'SOMOGY' 1]
 ['BARANYA' 'BACS' 1]
 ['BEKES' 'HAJDU' 2]
 ['BEKES' 'BEKES' 2]
 ['BEKES' 'JASZ' 2]
 ['BEKES' 'CSONGRAD' 2]
 ['BORSOD' 'HEVES' 3]
 ['BORSOD' 'SZABOLCS' 3]
 ['BORSOD' 'HAJDU' 3]
 ['BORSOD' 'BORSOD' 3]
 ['BORSOD' 'NOGRAD' 3]
 ['BORSOD' 'JASZ' 3]
 ['BUDAPEST' 'PEST' 4]
 ['BUDAPEST' 'BUDAPEST' 4]
```


2019BTECS00113_2019BTECS001
My Drive - Google Drive
colab.research.google.com/drive/1u7OcgXAsMHQiZGFOd8PQ9PH7xUzCb8a_#scrollTo=VMS9q5mSqfBu
FINAL450.xlsx - Go...
Online Courses - Le...
WCE exam
DSA450.xlsx - Goog...
Reading list
2019BTECS00113_2019BTECS00114.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
[21]
0s completed at 8:01 PM

```
 ['VESZPREM' 'VESZPREM' 18]
 ['VESZPREM' 'GYOR' 18]
 ['VESZPREM' 'FEJER' 18]
 ['VESZPREM' 'VAS' 18]
 ['VESZPREM' 'SOMOGY' 18]
 ['ZALA' 'VAS' 19]
 ['ZALA' 'ZALA' 19]
 ['ZALA' 'SOMOGY' 19]
 ['ZALA' 'VESZPREM' 19]]
```

```
print(y) # viewing an array of the dependent variable.
```

```
[10  0  1  5 13  6 16  1 16 14  0  8  2 10  5  9 15  8  3 12 10 13  4 10
  0  2  5  6 14 11 16  0 13 18 18 11  7 17 10  8 15  2  3 12 10 13  9  3
 10  5  9  3 13  2  0  8  7  6 11 18 13 13  3  9 12 12  6  4  9  0 10 13
 11 16 18  1 19 14  6  8  3 15 14  0  1 16  6 17 19  7 18 11 19 18  7  6
 17 14 17 19 14 18]
```

*****