COGNIZANCE TASKS
TASK 6 [PYTHON - MEDIOCRE LVL]

Question-1

Write a python program that reads the contents from the given file ‘onelinefile.txt’. The file contains a single line which is of the format (int)(string)(float)(string) repeatedly. For e.g.

Your main task is to split the contents of the given file based on their format and write it into a .csv file say ‘Filename2.csv’. For e.g. the above txt file should be converted into a csv file such that the contents look like this:

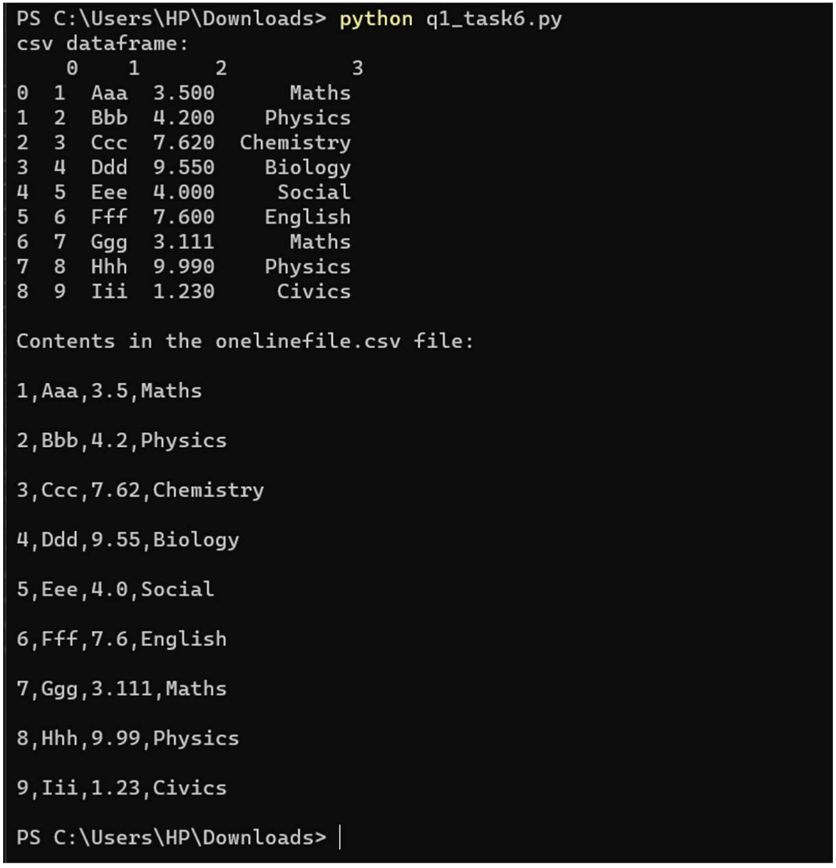
1,Aaa,3.5,Maths
2,Bbb,4.2,Physics
3,Ccc,7.62,Chemistry

Contents of ‘onelinefile.txt’

1Aaa3.5Maths2Bbb4.2Physics3Ccc7.62Chemistry4Ddd9.55Biology5Eee4.0Social6Fff7.6English7Ggg3.111Maths8Hhh9.99Physics9Iii1.23Civics

OUTPUT:

Q1_task6.py



```
PS C:\Users\HP\Downloads> python q1_task6.py
csv dataframe:
   0    1      2          3
0  1  Aaa  3.500      Maths
1  2  Bbb  4.200    Physics
2  3  Ccc  7.620  Chemistry
3  4  Ddd  9.550    Biology
4  5  Eee  4.000     Social
5  6  Fff  7.600    English
6  7  Ggg  3.111      Maths
7  8  Hhh  9.990    Physics
8  9  Iii  1.230     Civics

Contents in the onelinefile.csv file:

1,Aaa,3.5,Maths

2,Bbb,4.2,Physics

3,Ccc,7.62,Chemistry

4,Ddd,9.55,Biology

5,Eee,4.0,Social

6,Fff,7.6,English

7,Ggg,3.111,Maths

8,Hhh,9.99,Physics

9,Iii,1.23,Civics

PS C:\Users\HP\Downloads> 
```

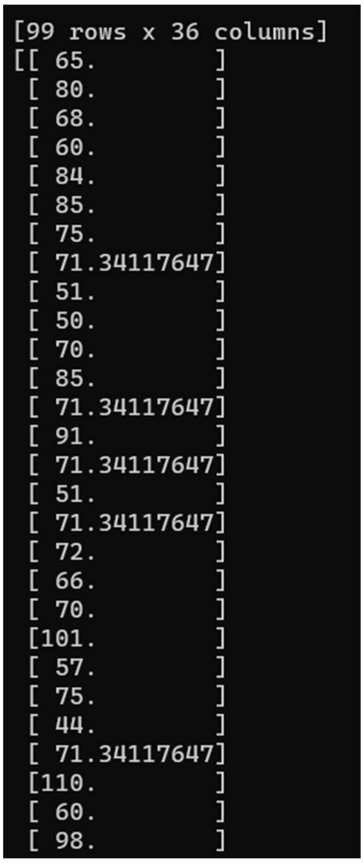
Question-2

Data formatting

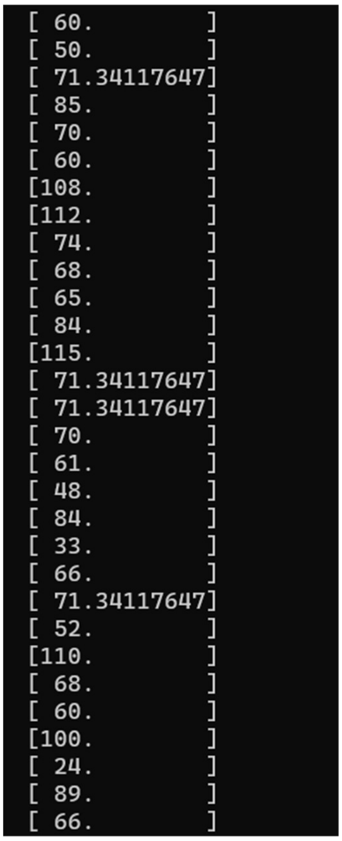
Python libraries represent missing numbers as nan which is short for “not a number”. Most libraries (including scikit-learn) will give you an error if you try to build a model using data with missing values. One of the common solution to get around this issue is to impute or fill in the missing value with a number or value of same format. From the given dataset, find the missing values(Nan/NA/-/Nil) and change those values into an appropriate number.Dataset Link

OUTPUT: q2_task6.py

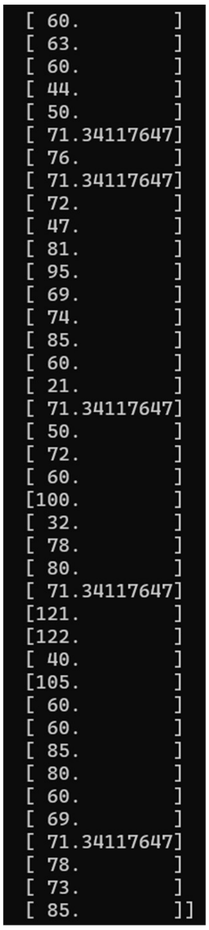
Column 3: LotFrontage



```
[99 rows x 36 columns]
[[ 65.        ]
 [ 80.        ]
 [ 68.        ]
 [ 60.        ]
 [ 84.        ]
 [ 85.        ]
 [ 75.        ]
 [ 71.34117647]
 [ 51.        ]
 [ 50.        ]
 [ 70.        ]
 [ 85.        ]
 [ 71.34117647]
 [ 91.        ]
 [ 71.34117647]
 [ 51.        ]
 [ 71.34117647]
 [ 72.        ]
 [ 66.        ]
 [ 70.        ]
 [101.        ]
 [ 57.        ]
 [ 75.        ]
 [ 44.        ]
 [ 71.34117647]
 [110.        ]
 [ 60.        ]
 [ 98.        ]
```

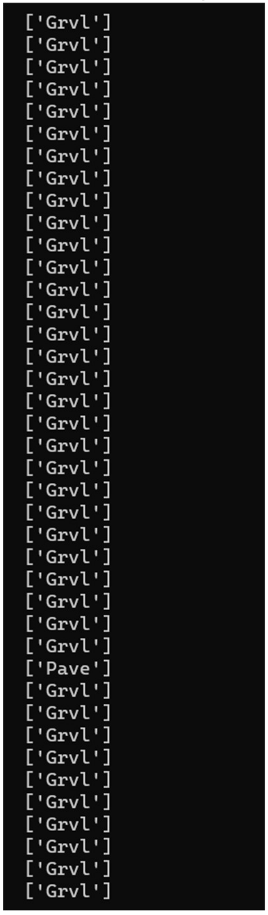


```
 [ 60.        ]
 [ 50.        ]
 [ 71.34117647]
 [ 85.        ]
 [ 70.        ]
 [ 60.        ]
 [108.        ]
 [112.        ]
 [ 74.        ]
 [ 68.        ]
 [ 65.        ]
 [ 84.        ]
 [115.        ]
 [ 71.34117647]
 [ 71.34117647]
 [ 70.        ]
 [ 61.        ]
 [ 48.        ]
 [ 84.        ]
 [ 33.        ]
 [ 66.        ]
 [ 71.34117647]
 [ 52.        ]
 [110.        ]
 [ 68.        ]
 [ 60.        ]
 [100.        ]
 [ 24.        ]
 [ 89.        ]
 [ 66.        ]
```

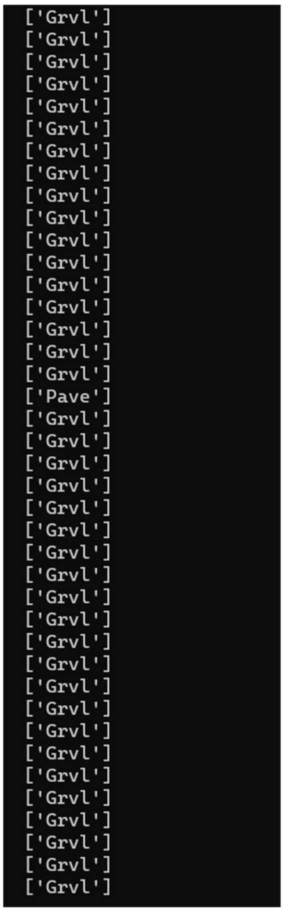


```
 [ 60.        ]
 [ 63.        ]
 [ 60.        ]
 [ 44.        ]
 [ 50.        ]
 [ 71.34117647]
 [ 76.        ]
 [ 71.34117647]
 [ 72.        ]
 [ 47.        ]
 [ 81.        ]
 [ 95.        ]
 [ 69.        ]
 [ 74.        ]
 [ 85.        ]
 [ 60.        ]
 [ 21.        ]
 [ 71.34117647]
 [ 50.        ]
 [ 72.        ]
 [ 60.        ]
 [100.        ]
 [ 32.        ]
 [ 78.        ]
 [ 80.        ]
 [ 71.34117647]
 [121.        ]
 [122.        ]
 [ 40.        ]
 [105.        ]
 [ 60.        ]
 [ 60.        ]
 [ 85.        ]
 [ 80.        ]
 [ 60.        ]
 [ 69.        ]
 [ 71.34117647]
 [ 78.        ]
 [ 73.        ]
 [ 85.        ]]
```

Column 6: Alley

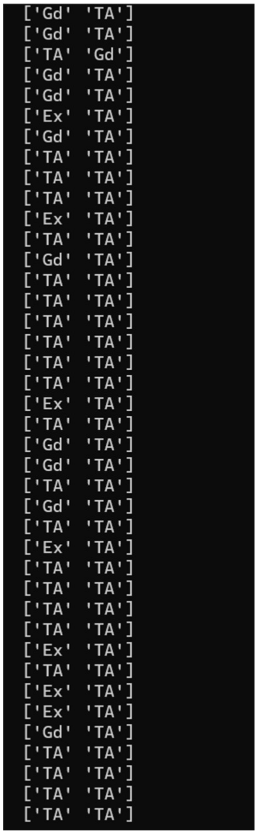


```
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Pave']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
```

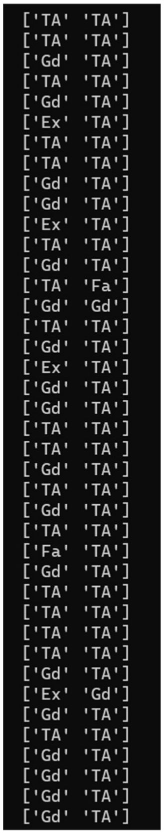


```
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Pave']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
 ['Grvl']
```

Columns: 30, 31: Bsmt Qual, Bsmt Cond

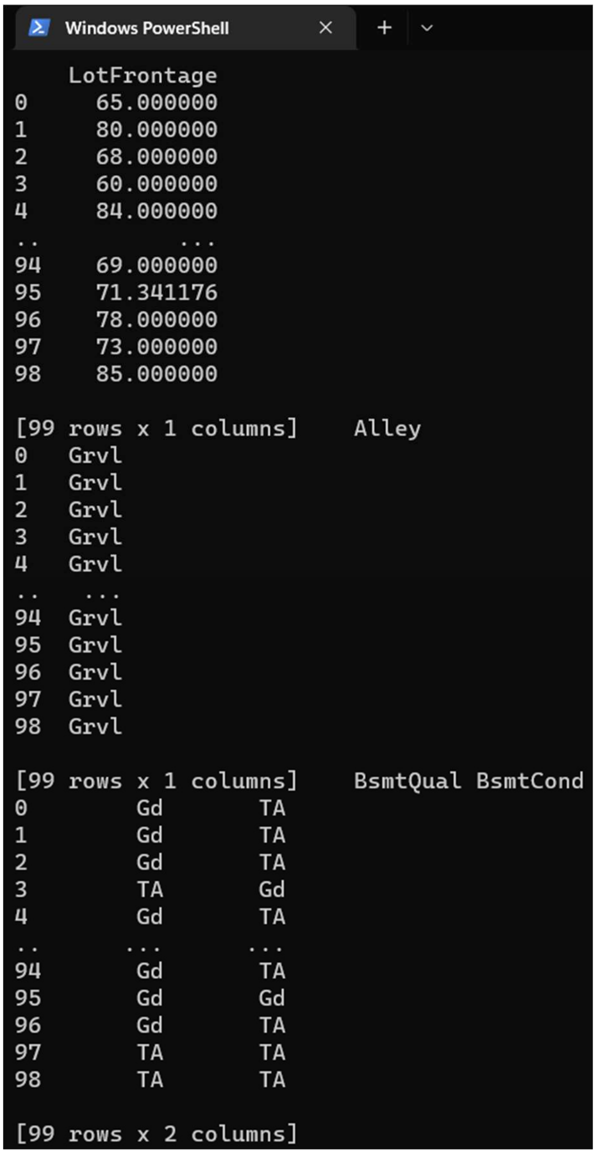


```
 ['Gd' 'TA']
 ['Gd' 'TA']
 ['TA' 'Gd']
 ['Gd' 'TA']
 ['Gd' 'TA']
 ['Ex' 'TA']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['Ex' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['Ex' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['Ex' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['Ex' 'TA']
 ['TA' 'TA']
 ['Ex' 'TA']
 ['Ex' 'TA']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
```



```
 ['TA' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['Ex' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['Gd' 'TA']
 ['Ex' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['TA' 'Fa']
 ['Gd' 'Gd']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['Ex' 'TA']
 ['Gd' 'TA']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['Fa' 'TA']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['Ex' 'Gd']
 ['Gd' 'TA']
 ['TA' 'TA']
 ['Gd' 'TA']
 ['Gd' 'TA']
 ['Gd' 'TA']
 ['Gd' 'TA']
```

Columns which had null values imputed using SimpleImputer after transformation,



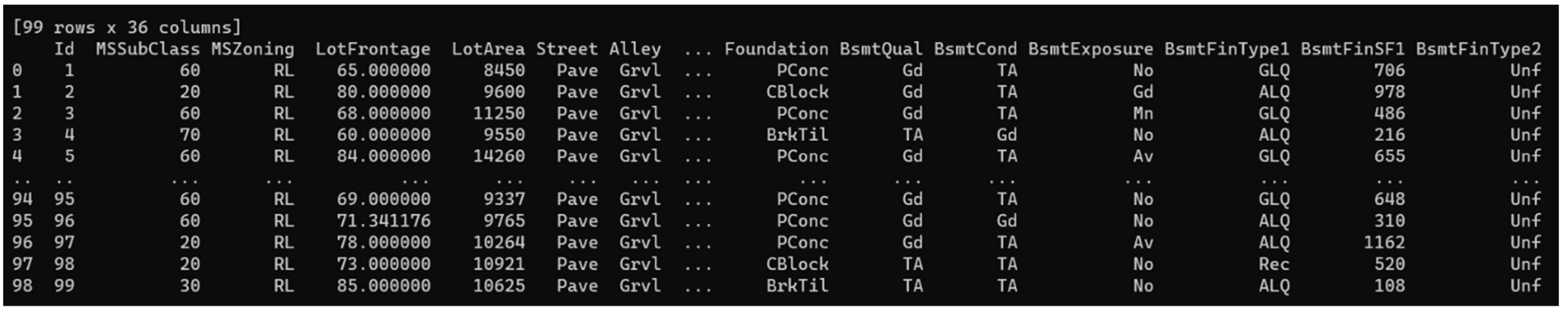
```
    LotFrontage
0     65.000000
1     80.000000
2     68.000000
3     60.000000
4     84.000000
..          ...
94    69.000000
95    71.341176
96    78.000000
97    73.000000
98    85.000000

[99 rows x 1 columns]    Alley
0   Grvl
1   Grvl
2   Grvl
3   Grvl
4   Grvl
..   ...
94  Grvl
95  Grvl
96  Grvl
97  Grvl
98  Grvl

[99 rows x 1 columns]    BsmtQual BsmtCond
0        Gd       TA
1        Gd       TA
2        Gd       TA
3        TA       Gd
4        Gd       TA
..      ...      ...
94       Gd       TA
95       Gd       Gd
96       Gd       TA
97       TA       TA
98       TA       TA

[99 rows x 2 columns]
```

Dataframe after data formatting:



```
[99 rows x 36 columns]
    Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley  ... Foundation BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2
0    1          60       RL    65.000000     8450   Pave  Grvl  ...      PConc       Gd       TA           No          GLQ        706          Unf
1    2          20       RL    80.000000     9600   Pave  Grvl  ...     CBlock       Gd       TA           Gd          ALQ        978          Unf
2    3          60       RL    68.000000    11250   Pave  Grvl  ...      PConc       Gd       TA           Mn          GLQ        486          Unf
3    4          70       RL    60.000000     9550   Pave  Grvl  ...     BrkTil       TA       Gd           No          ALQ        216          Unf
4    5          60       RL    84.000000    14260   Pave  Grvl  ...      PConc       Gd       TA           Av          GLQ        655          Unf
..  ..         ...      ...          ...      ...    ...   ...  ...        ...      ...      ...          ...          ...        ...          ...
94  95          60       RL    69.000000     9337   Pave  Grvl  ...      PConc       Gd       TA           No          GLQ        648          Unf
95  96          60       RL    71.341176     9765   Pave  Grvl  ...      PConc       Gd       Gd           No          ALQ        310          Unf
96  97          20       RL    78.000000    10264   Pave  Grvl  ...      PConc       Gd       TA           Av          ALQ       1162          Unf
97  98          20       RL    73.000000    10921   Pave  Grvl  ...     CBlock       TA       TA           No          Rec        520          Unf
98  99          30       RL    85.000000    10625   Pave  Grvl  ...     BrkTil       TA       TA           No          ALQ        108          Unf
```

Question-3

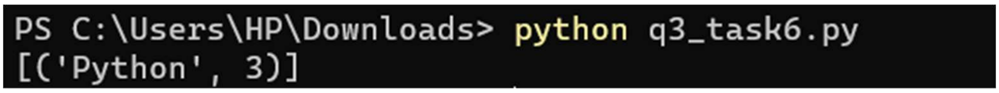
Read the file ‘about.txt’ and find the words with atleast 6 letters and the most frequently used word.

Contents of the file ‘about.txt’:

Python has tools for almost every aspect of scientific computing. The Bank of America uses Python to crunch its financial data and Facebook looks upon the Python library Pandas for its data analysis. While there are many libraries available to perform data analysis in Python, here are a few: NumPy, SciPy, Pandas and Matplotlib.

OUTPUT:

Q3_task.py



```
PS C:\Users\HP\Downloads> python q3_task6.py
[('Python', 3)]
```