COGNIZANCE CLUB

PRELIMINARY

Name: Dharan Kumar

Roll No: CH.EN.U4CSE20019

Topic: Task-6(Python-Medicore L3V3L)

Task-6(Python Programming)

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. 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'.

Code:

```
f=open("onelinefile.txt","r")
string=str(f.read())
n=len(string)
temp=""
c=0
for i in range(n):
    if string[i].isalpha()==True:
        temp+=string[i]
        if i+1<n:
            if string[i+1].isalpha()!=True:
                if c!=3:
                    temp+=","
                c+=1
    elif string[i].isdigit()==True:
        temp+=string[i]
        if string[i+1]==".":
            temp+="."
        elif string[i+1].isdigit()!=True and
string[i+1]!=".":
```

```
            if c!=3:
                temp+=","
            c+=1
    if c%4==0:
        temp+="\n"
        c=0
f.close()
f1=open("Filename2.csv","w")
f1.writelines(temp)
f1.close()
print("The lines written in the file are\n"+temp)
```

Output:

```
========== RESTART: D:\DHARAN\Task6\Q1.py ==========
The lines written in the file are
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
```

Filename2.csv

	A	B	C	D	E	F	G	H
1	1	Aaa	3.5	Maths				
2	2	Bbb	4.2	Physics				
3	3	Ccc	7.62	Chemistry				
4	4	Ddd	9.55	Biology				
5	5	Eee	4	Social				
6	6	Fff	7.6	English				
7	7	Ggg	3.111	Maths				
8	8	Hhh	9.99	Physics				
9	9	Iii	1.23	Civics				
10								
11								
12								
13								

Result:

Hence the program to write the text values in the .csv file is working successfully.

2) 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.

Code:

```
import pandas as pd


dk = pd.read_csv("dataset.csv")

print("\nMissing values in the given csv file are: ")

print(dk.isnull().sum())


print("\nMissing values in LotFrontage: \n")

print(dk['LotFrontage'].isnull())

print("\nupdated LotFrontage values(changed '-99'
values for the LotFrontage instead of NA): \n")

dk['LotFrontage'].fillna('-99',inplace = True)

print(dk['LotFrontage'])


print("\nMissing values in Alley: \n")

print(dk['Alley'].isnull())

print("\nupdated Alley values(changed 'empty' values
for the Alley instead of NA): \n")

dk['Alley'].fillna('empty',inplace = True)

print(dk['Alley'])
```

```
print("\n",dk[dk['BsmtQual'].isnull()])


print("\nupdated BsmtQual values(changed 'empty1'
values for the BsmtQual instead of NA): \n")

dk['BsmtQual'].fillna('empty1',inplace = True)

print(dk[dk['BsmtQual'].isnull()])


print("\n",dk[dk['BsmtQual'].isnull()])


print("\nupdated BsmtCond values(changed 'empty2'
values for the BsmtCond instead of NA): \n")

dk['BsmtCond'].fillna('empty2',inplace = True)

print(dk[dk['BsmtCond'].isnull()])


print("\n",dk[dk['BsmtExposure'].isnull()])


print("\nupdated BsmtExposure values(changed 'empty3'
values for the BsmtExposure instead of NA): \n")

dk['BsmtExposure'].fillna('empty3',inplace = True)

print(dk[dk['BsmtExposure'].isnull()])


print("\n",dk[dk['BsmtFinType1'].isnull()])


print("\nupdated BsmtFinType1 values(changed 'empty4'
values for the BsmtFinType1 instead of NA): \n")

dk['BsmtFinType1'].fillna('empty4',inplace = True)
```

```
print(dk[dk['BsmtFinType1'].isnull()])


print("\n",dk[dk['BsmtFinType2'].isnull()])


print("\nupdated BsmtFinType2 values(changed 'empty5' values for the BsmtFinType2 instead of NA): \n")
dk['BsmtFinType2'].fillna('empty5',inplace = True)
print(dk[dk['BsmtFinType2'].isnull()])


print("\n\nUpdated final csv file: \n")
print(dk.isnull().sum())
```

Output:

```
Missing values in the given csv file are:
Id               0
MSSubClass       0
MSZoning         0
LotFrontage     14
LotArea          0
Street           0
Alley           93
LotShape         0
LandContour      0
Utilities        0
LotConfig        0
LandSlope        0
Neighborhood     0
Condition1       0
Condition2       0
BldgType         0
HouseStyle       0
OverallQual      0
OverallCond      0
YearBuilt        0
YearRemodAdd     0
RoofStyle        0
RoofMatl         0
Exterior1st      0
Exterior2nd      0
MasVnrType       0
MasVnrArea       0
ExterQual        0
ExterCond        0
Foundation       0
BsmtQual         3
BsmtCond         3
BsmtExposure     3
BsmtFinType1     3
BsmtFinSF1       0
BsmtFinType2     3
dtype: int64
```

```
Missing values in LotFrontage:

0     False
1     False
2     False
3     False
4     False
      ...
94    False
95     True
96    False
97    False
98    False
Name: LotFrontage, Length: 99, dtype: bool

updated LotFrontage values(changed '-99' values for the LotFrontage instead of NA):

0     65.0
1     80.0
2     68.0
3     60.0
4     84.0
      ...
94    69.0
95     -99
96    78.0
97    73.0
98    85.0
Name: LotFrontage, Length: 99, dtype: object


Missing values in Alley:

0     True
1     True
2     True
3     True
4     True
      ...
94    True
95    True
96    True
97    True
98    True
Name: Alley, Length: 99, dtype: bool

updated Alley values(changed 'empty' values for the Alley instead of NA):

0     empty
1     empty
2     empty
3     empty
4     empty
      ...
94    empty
95    empty
96    empty
97    empty
98    empty
Name: Alley, Length: 99, dtype: object
```

```
     Id  MSSubClass MSZoning  ... BsmtFinType1  BsmtFinSF1 BsmtFinType2
17  18          90       RL  ...          NaN           0          NaN
39  40          90       RL  ...          NaN           0          NaN
90  91          20       RL  ...          NaN           0          NaN

[3 rows x 36 columns]

updated BsmtQual values(changed 'empty1' values for the BsmtQual instead of NA):

Empty DataFrame
Columns: [Id, MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Conditio
n2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMat1, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, Exte
rCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2]
Index: []
```

```
updated BsmtCond values(changed 'empty2' values for the BsmtCond instead of NA):

Empty DataFrame
Columns: [Id, MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Conditio
n2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMat1, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, Exte
rCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2]
Index: []

     Id  MSSubClass MSZoning  ... BsmtFinType1  BsmtFinSF1 BsmtFinType2
17  18          90       RL  ...          NaN           0          NaN
39  40          90       RL  ...          NaN           0          NaN
90  91          20       RL  ...          NaN           0          NaN

[3 rows x 36 columns]
```

```
updated BsmtExposure values(changed 'empty3' values for the BsmtExposure instead of NA):

Empty DataFrame
Columns: [Id, MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Conditio
n2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMat1, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, Exte
rCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2]
Index: []

     Id  MSSubClass MSZoning  ... BsmtFinType1  BsmtFinSF1 BsmtFinType2
17  18          90       RL  ...          NaN           0          NaN
39  40          90       RL  ...          NaN           0          NaN
90  91          20       RL  ...          NaN           0          NaN

[3 rows x 36 columns]
```

```
updated BsmtFinType1 values(changed 'empty4' values for the BsmtFinType1 instead of NA):

Empty DataFrame
Columns: [Id, MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Conditio
n2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMat1, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, Exte
rCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2]
Index: []

     Id  MSSubClass MSZoning  ... BsmtFinType1  BsmtFinSF1 BsmtFinType2
17  18          90       RL  ...       empty4           0          NaN
39  40          90       RL  ...       empty4           0          NaN
90  91          20       RL  ...       empty4           0          NaN

[3 rows x 36 columns]
```

```
Empty DataFrame
Columns: [Id, MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Conditio
n2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMat1, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, Exte
rCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2]
Index: []
```

Updated list

```
Updated final csv file:

Id              0
MSSubClass      0
MSZoning        0
LotFrontage     0
LotArea         0
Street          0
Alley           0
LotShape        0
LandContour     0
Utilities       0
LotConfig       0
LandSlope       0
Neighborhood    0
Condition1      0
Condition2      0
BldgType        0
HouseStyle      0
OverallQual     0
OverallCond     0
YearBuilt       0
YearRemodAdd    0
RoofStyle       0
RoofMatl        0
Exterior1st     0
Exterior2nd     0
MasVnrType      0
MasVnrArea      0
ExterQual       0
ExterCond       0
Foundation      0
BsmtQual        0
BsmtCond        0
BsmtExposure    0
BsmtFinType1    0
BsmtFinSF1      0
BsmtFinType2    0
dtype: int64
```

Result:

Hence the program find the missing value (Nan/NA/-/Nil) to the approximate values is working successfully.

3) Read the file 'about.txt' and find the words with atleast 6 letters and the most frequently used word.

Code:

```
import re


print("\nThe atleast 6 character words are");
file = open("data.txt", "r")
text=file.read()
lis=re.findall(r"\b\w{6,}\b", text)
print(*lis,sep='\n')
file.close()


count = 0
word = ""
maxCount = 0
words = []
file = open("data.txt", "r")
for line in file:
    string =
line.lower().replace(',','').replace('.','').split("
");
    for s in string:
        words.append(s);
for i in range(0, len(words)):
    count = 1;
    for j in range(i+1, len(words)):
```

```
        if(words[i] == words[j]):
            count = count + 1;
    if(count > maxCount):
        maxCount = count;
        word = words[i];
print("Most repeated word: " + word);
file.close();
```

Output:

```
The atleast 6 character words are
Python
almost
aspect
scientific
computing
America
Python
crunch
financial
Facebook
Python
library
Pandas
analysis
libraries
available
perform
analysis
Python
Pandas
Matplotlib
Most repeated word: python
```

Result:

Hence the program finds all the words with at least 6 characters and the most frequently used word are working successfully.