

```
In [348]: import pandas as pd
import numpy as np
import operator
def dict_sort_des(diction):
    sorted_d = sorted(diction.items(), key=operator.itemgetter(1), reverse = True)
    return sorted_d

In [349]: df = pd.read_csv('BM.csv')
df.rename(columns = {'V1': 'Age','V2': 'Job','V3': 'Marital','V4': 'Education','V5': 'c_default','V6': 'Balance','V7': 'Housi
ng','V8': 'Loan','V9': 'contact','V10': 'pay','V11': 'Month','V12': 'Duration','V13': 'Total times called', 'V14': 'pdays','V
15': 'previous','V16': 'poutcome','class': 'subscribed'},inplace = True)

In [350]: len(df)
Out[350]: 45211

In [351]: df = df[df.Job != 'unknown']
df = df[df.Education != 'unknown']
df = df[df.poutcome != 'unknown']
df = df[df.poutcome != 'other']

In [352]: len(df)
Out[352]: 6133

In [353]: temp = df[['loan', 'housing', 'c_default', 'Balance']]
loans = []
for index, row in temp.iterrows():
    if row['loan'] == 'yes' or row['housing'] == 'yes' or row['c_default'] == 'yes' or row['Balance'] < 0:
        loans.append('Negative')
    else:
        loans.append('Positive')
df.insert(6, 'Balance Status', loans, True)

In [354]: df = df.drop(['Loan', 'housing', 'Day', 'pdays', 'previous', 'contact', 'c_default', 'Balance'],axis=1)

In [355]: df.head()
Out[355]:
```

	Age	Job	Marital	Education	Balance Status	Month	Duration	Total times called	poutcome	subscribed
24060	33	admin.	married	tertiary	Positive	oct	39	1	failure	1
24064	33	services	married	secondary	Negative	oct	144	1	failure	2
24077	36	management	married	tertiary	Negative	oct	140	1	failure	2
24080	56	technician	married	secondary	Negative	oct	518	1	success	2
24127	51	admin.	single	secondary	Positive	nov	449	1	failure	1

```
In [356]: xt = dict.fromkeys(['retired', 'unemployed', 'student'], 'no-pay')
df['Job'] = df['Job'].replace(xt)
xt = dict.fromkeys(['blue-collar', 'technician', 'services', 'housemaid'], 'low-pay')
df['Job'] = df['Job'].replace(xt)
xt = dict.fromkeys(['self-employed', 'entrepreneur'], 'varincome')
df['Job'] = df['Job'].replace(xt)
xt = dict.fromkeys(['admin.', 'management'], 'high-pay')
df['Job'] = df['Job'].replace(xt)

In [357]: df['Age'] = np.where(df['Age'].between(18, 30), 1, df['Age'])
df['Age'] = np.where(df['Age'].between(31, 60), 2, df['Age'])
df['Age'] = np.where(df['Age'].between(61, 95), 3, df['Age'])
xt = dict.fromkeys([1], 'Young')
df['Age'] = df['Age'].replace(xt)
xt = dict.fromkeys([2], 'Mature')
df['Age'] = df['Age'].replace(xt)
xt = dict.fromkeys([3], 'old')
df['Age'] = df['Age'].replace(xt)

In [358]: xt = dict.fromkeys(['secondary', 'primary'], 'Basic')
df['Education'] = df['Education'].replace(xt)
xt = dict.fromkeys(['tertiary'], 'Advanced')
df['Education'] = df['Education'].replace(xt)

In [359]: xt = dict.fromkeys(['jan', 'feb', 'mar', '1'])
df['Month'] = df['Month'].replace(xt)
xt = dict.fromkeys(['apr', 'may', 'jun', '2'])
df['Month'] = df['Month'].replace(xt)
xt = dict.fromkeys(['jul', 'aug', 'sep', '3'])
df['Month'] = df['Month'].replace(xt)
xt = dict.fromkeys(['oct', 'nov', 'dec', '4'])
df['Month'] = df['Month'].replace(xt)

In [360]: df['Duration'] = np.where(df['Duration'].between(0, 180), 0, df['Duration'])
df['Duration'] = np.where(df['Duration'].between(181, 300), 1, df['Duration'])
df['Duration'] = np.where(df['Duration'].between(301, 5000), 2, df['Duration'])
xt = dict.fromkeys([0], 'L')
df['Duration'] = df['Duration'].replace(xt)
xt = dict.fromkeys([1], 'M')
df['Duration'] = df['Duration'].replace(xt)
xt = dict.fromkeys([2], 'H')
df['Duration'] = df['Duration'].replace(xt)

In [362]: xt = dict.fromkeys([1, 2, 3, 4, 5], 'Low')
df['Total times called'] = df['Total times called'].replace(xt)
xt = dict.fromkeys([6, 7, 8, 9, 10, 11, 12, 14, 15], 'High')
df['Total times called'] = df['Total times called'].replace(xt)

In [363]: xt = dict.fromkeys([1], 'No')
df['subscribed'] = df['subscribed'].replace(xt)
xt = dict.fromkeys([2], 'Yes')
df['subscribed'] = df['subscribed'].replace(xt)

In [364]: df
Out[364]:
```

	Age	Job	Marital	Education	Balance Status	Month	Duration	Total times called	poutcome	subscribed
24060	Mature	high-pay	married	Advanced	Positive	4	L	Low	failure	No
24064	Mature	low-pay	married	Basic	Negative	4	L	Low	failure	Yes
24077	Mature	high-pay	married	Advanced	Negative	4	L	Low	failure	Yes
24080	Mature	low-pay	married	Basic	Negative	4	M	Low	success	Yes
24127	Mature	high-pay	single	Basic	Positive	4	M	Low	failure	No
24151	Mature	no-pay	divorced	Basic	Negative	4	L	Low	failure	No
24160	Mature	high-pay	married	Advanced	Negative	4	L	Low	failure	Yes
24165	Young	high-pay	married	Basic	Negative	4	L	Low	success	No
24170	Young	high-pay	single	Advanced	Negative	4	L	Low	failure	No
24184	Mature	varincome	married	Advanced	Negative	4	L	Low	failure	No
24186	Mature	high-pay	divorced	Advanced	Positive	4	M	Low	failure	No
24187	Mature	low-pay	married	Basic	Negative	4	L	Low	failure	No
24189	Mature	high-pay	married	Advanced	Negative	4	L	Low	failure	No
24198	Mature	low-pay	single	Basic	Negative	4	M	Low	failure	No
24203	Mature	high-pay	married	Advanced	Positive	4	M	Low	failure	No
24207	Mature	low-pay	married	Advanced	Negative	4	L	Low	failure	No
24208	Mature	high-pay	single	Basic	Negative	4	L	Low	failure	No
24230	Mature	low-pay	married	Basic	Positive	4	M	Low	failure	No
24236	Mature	low-pay	single	Basic	Negative	4	L	Low	failure	No
24239	Mature	high-pay	divorced	Basic	Negative	4	L	Low	success	No
24250	Mature	low-pay	married	Basic	Negative	4	L	Low	failure	No
24253	Mature	high-pay	married	Basic	Negative	4	M	Low	failure	No
24254	Mature	low-pay	married	Basic	Positive	4	L	Low	failure	No
24257	Mature	high-pay	married	Advanced	Negative	4	L	Low	failure	No
24264	Mature	high-pay	married	Advanced	Positive	4	M	Low	success	No
24266	Mature	low-pay	married	Basic	Negative	4	L	Low	failure	No
24299	Mature	high-pay	married	Basic	Negative	4	M	Low	failure	No
24311	Mature	no-pay	married	Basic	Negative	4	L	Low	failure	No
24312	Mature	varincome	married	Advanced	Positive	4	L	Low	failure	No
24315	Mature	low-pay	single	Basic	Negative	4	M	Low	failure	No
...	...	...	...	...	...	...	...	...	...	...
45160	Mature	low-pay	married	Basic	Negative	4	M	Low	success	Yes
45162	Young	high-pay	single	Basic	Positive	4	M	Low	success	Yes
45163	old	no-pay	married	Basic	Positive	4	M	Low	failure	No
45166	Mature	no-pay	single	Advanced	Positive	4	M	Low	success	Yes
45168	old	no-pay	married	Advanced	Negative	4	M	Low	success	Yes
45169	Mature	high-pay	married	Advanced	Negative	4	M	Low	failure	No
45171	Mature	high-pay	single	Basic	Positive	4	M	Low	failure	No
45172	Young	low-pay	single	Advanced	Positive	4	L	Low	success	No
45173	Mature	high-pay	single	Advanced	Positive	4	L	Low	failure	No
45174	old	low-pay	married	Basic	Positive	4	M	Low	success	Yes
45175	Mature	varincome	single	Basic	Positive	4	M	Low	success	Yes
45177	Mature	high-pay	divorced	Basic	Negative	4	M	Low	success	Yes
45178	Mature	low-pay	married	Basic	Negative	4	M	Low	success	Yes
45179	Mature	low-pay	married	Basic	Positive	4	M	Low	failure	No
45180	old	no-pay	married	Basic	Positive	4	M	Low	failure	No
45181	Mature	low-pay	married	Basic	Positive	4	L	Low	failure	No
45182	Mature	low-pay	married	Basic	Positive	4	M	Low	success	Yes
45183	old	no-pay	married	Basic	Positive	4	L	Low	success	No
45184	old	no-pay	married	Basic	Positive	4	L	Low	success	No
45185	Mature	low-pay	married	Advanced	Negative	4	M	Low	success	Yes
45188	Young	high-pay	single	Basic	Negative	4	M	Low	success	Yes
45189	Young	low-pay	single	Basic	Positive	4	L	Low	failure	No
45190	Mature	low-pay	married	Basic	Positive	4	M	Low	success	Yes
45191	old	no-pay	divorced	Advanced	Negative	4	M	Low	failure	Yes
45193	Young	varincome	single	Advanced	Positive	4	M	Low	success	Yes
45194	Mature	high-pay	married	Advanced	Negative	4	L	Low	failure	No
45195	old	no-pay	married	Basic	Positive	4	M	Low	success	Yes
45201	Mature	high-pay	married	Advanced	Positive	4	M	Low	success	Yes
45204	old	no-pay	married	Basic	Positive	4	M	Low	failure	Yes
45208	old	no-pay	married	Basic	Positive	4	H	Low	success	Yes

6133 rows × 10 columns

```
In [365]: len(df)
Out[365]: 6133

In [366]: temp = df[['Education', 'subscribed']]
temp.groupby(temp.columns.tolist(),as_index=False).size()

Out[366]: Education subscribed
Advanced No 1450
         Yes 658
         No 3179
         Yes 846
dtype: int64

In [367]: df = df.drop(['Education'],axis=1)

In [368]: temp = df[['Duration', 'subscribed']]
temp.groupby(temp.columns.tolist(),as_index=False).size()

Out[368]: Duration subscribed
H No 57
  Yes 96
L No 2582
  Yes 248
M No 1900
  Yes 1160
dtype: int64

In [369]: temp = df[['Balance Status', 'subscribed']]
temp.groupby(temp.columns.tolist(),as_index=False).size()

Out[369]: Balance Status subscribed
Negative No 3489
         Yes 560
         No 1140
         Yes 944
dtype: int64

In [370]: temp = df[['poutcome', 'subscribed']]
temp.groupby(temp.columns.tolist(),as_index=False).size()

Out[370]: poutcome subscribed
failure No 4122
        Yes 587
success No 507
        Yes 917
dtype: int64

In [371]: temp = df[['Job', 'subscribed']]
temp.groupby(temp.columns.tolist(),as_index=False).size()

Out[371]: Job subscribed
high-pay No 1600
         Yes 610
low-pay No 2317
        Yes 498
no-pay No 400
       Yes 323
varincome No 312
          Yes 73
dtype: int64

In [372]: temp = df[['Marital', 'subscribed']]
temp.groupby(temp.columns.tolist(),as_index=False).size()

Out[372]: Marital subscribed
divorced No 541
         Yes 139
married No 2753
        Yes 838
single No 1335
        Yes 527
dtype: int64

In [373]: temp = df[['Age', 'subscribed']]
temp.groupby(temp.columns.tolist(),as_index=False).size()

Out[373]: Age subscribed
Mature No 3870
        Yes 1046
Young No 603
       Yes 288
old No 156
   Yes 170
dtype: int64

In [375]: temp = df[['Month', 'poutcome', 'Total times called', 'subscribed']]
temp.groupby(temp.columns.tolist(),as_index=False).size()

Out[375]: Month poutcome Total times called subscribed
1 failure High No 31
   Low No 725
   Yes 94
   success High No 4
   Low Yes 2
   Low No 74
   Low Yes 168
2 failure High No 105
   Low Yes 4
   Low No 2129
   success High Yes 235
   Low No 11
   Low Yes 8
   Low No 186
3 failure High Yes 260
   Low No 9
   Low Yes 1
   Low No 287
   success High Yes 137
   Low No 4
   Low Yes 7
   Low No 122
4 failure High Yes 278
   Low No 27
   success High Yes 116
   Low No 4
   Low Yes 1
   Low No 102
   Low Yes 193
dtype: int64

In [376]: temp = df[['Age', 'Job', 'Balance Status', 'subscribed']]
temp.groupby(temp.columns.tolist(),as_index=False).size()

Out[376]: Age Job Balance Status subscribed
Mature high-pay Negative 1022
         high-pay Positive 206
         low-pay Negative 363
         low-pay Positive 291
         no-pay Negative 1708
         no-pay Positive 214
         no-pay Negative 307
         no-pay Positive 181
         no-pay Negative 116
         no-pay Positive 25
         varincome Negative 86
         varincome Positive 216
         varincome Negative 21
         varincome Positive 64
Young high-pay Negative 22
         high-pay Positive 133
         low-pay Negative 31
         low-pay Positive 68
         low-pay Negative 69
         low-pay Positive 238
         no-pay Negative 38
         no-pay Positive 48
         no-pay Negative 50
         no-pay Positive 27
         varincome Negative 7
         varincome Positive 58
         varincome Negative 75
         varincome Positive 5
         old high-pay Negative 10
         old high-pay Positive 13
         low-pay Negative 2
         low-pay Positive 12
         no-pay Negative 13
         no-pay Positive 1
         no-pay Negative 15
         no-pay Positive 15
         no-pay Negative 5
         no-pay Positive 10
         varincome Negative 120
         varincome Positive 120
         varincome Negative 3
         varincome Positive 9
dtype: int64

In [377]: df = df.drop(['Marital', 'Duration', 'poutcome', 'Month', 'Age'],axis=1)

In [380]: temp = df.groupby(df.columns.tolist(),as_index=False).size()

In [379]: df.drop_duplicates()

Out[379]:
```

	Job	Balance Status	Total times called	subscribed
24060	high-pay	Positive	Low	No
24064	low-pay	Negative	Low	Yes
24077	high-pay	Negative	Low	Yes
24151	no-pay	Negative	Low	No
24165	high-pay	Negative	Low	No
24184	varincome	Negative	Low	No
24187	low-pay	Negative	Low	No
24230	low-pay	Positive	Low	No
24312	varincome	Positive	Low	No
24364	no-pay	Positive	Low	No
24622	no-pay	Positive	Low	Yes
25627	varincome	Positive	Low	Yes
25752	low-pay	Negative	High	No
25845	no-pay	Negative	Low	Yes
26400	varincome	Negative	High	No
26434	low-pay	Positive	Low	Yes
26485	high-pay	Positive	High	No
26618	high-pay	Negative	High	No
26997	high-pay	Positive	Low	Yes
27129	low-pay	Positive	High	No
27399	varincome	Negative	High	No
28845	no-pay	Positive	High	No
30599	no-pay	Positive	High	No
34317	high-pay	Negative	High	Yes
34391	no-pay	Positive	High	Yes
38725	low-pay	Negative	High	Yes
40233	high-pay	Positive	High	Yes
43526	low-pay	Positive	High	Yes

```
In [388]: temp.to_csv('Final Data.csv', sep=',')

/home/gunno/local/lib/python3.6/site-packages/ipykernel_launcher.py:1: FutureWarning: The signature of 'Series.to_csv' w
as aligned to that of 'DataFrame.to_csv', and argument 'header' will change its default value from False to True: please
pass an explicit value to suppress this warning.
***Entry point for launching an Iython kernel.
```

```
In [ ]:
```