analysis

December 7, 2024

Author: Jase Banta
Date: 2024-12-06
Description: The following python codes are written for an application to University of Chicago M.S. Applied Data Science program.
1. I shall write a function to randomly select 20% from `bank-additional-full.csv` to form `.csv`. 2. I want to I want to perform some data transformation and standarization. - Add a new column for age range - remove hidden characters from `job` column - Create dict in `Education` - Treat `unknown` value as None type for columns `Default`, `Housing`, `Loan`. - Transform columns to binary `Default`, `Housing`, `Loan`, `Y` - Calculate basic stats: - overall martial status, success rate, job categories, education level, house - Age distribution (visualization) colored by marital status - Job and housing

Moro, S., P. Rita, and P. Cortez. 2014. Bank Marketing. UCI Machine Learning Repository. https://doi.org/10.24432/C5K306.

[1]:

```python
import csv, random
import pandas as pd
import numpy as np

class Utility:
    def __init__(self):
        pass

    def random_select(self, source: str, target: str, size: float) -> str:
        with open(source, 'r') as file:
            reader = csv.reader(file, delimiter= ';')
            header = next(reader)
            data = list(reader)
            size = round(size * len(data))
            sample_rows = random.sample(data, min(size, len(data)))

        with open(target, 'w', newline= '') as outfile:
            writer = csv.writer(outfile, delimiter= ',')
            writer.writerow(header)
            writer.writerows(sample_rows)

        return target
    def formatEducation(self, value):
        pass

    def transformBinary(self, df, columns: list):
        mapping = {
            'yes': True,
            'no': False,
            'unknown': np.nan
        }
        for column in columns:
            df[column] = df[column].replace(mapping)

        return df

    def AgeMasking(self, age: int):
        age_gorup = np.nan
        if age < 18:
            age_gorup = '< 18'
        elif 18 <= age < 37:
            age_gorup = '18-36'
        elif 37 <= age < 55:
            age_gorup = '37-54'
        elif 54 <= age < 73:
            age_gorup = '54-72'
        elif 73 <= age < 91:
            age_gorup = '73-90'
        elif age >= 90:
            age_gorup = '90+'
        return age_gorup
```

[2]:

```python
source='bank-additional-full.csv'
target='bank-sample.csv'
size = 0.1
with open(source, 'r') as file:
    reader = csv.reader(file, delimiter= ';')
    header = next(reader)
    data = list(reader)
    size = round(size * len(data))
    sample_rows = random.sample(data, min(size, len(data)))

with open(target, 'w', newline= '') as outfile:
    writer = csv.writer(outfile, delimiter= ',')
    writer.writerow(header)
    writer.writerows(sample_rows)
```

[3]:

```python
import pandas as pd
```

```
util = Utility()
sample = util.random_select(source='bank-additional-full.csv',␣
 ↪target='bank-sample.csv', size=0.15)
data = pd.read_csv(sample)
data.head()
```

```
[3]:    age           job   marital           education default housing loan  \
     0   38    technician   married  professional.course      no     yes   no
     1   36        admin.  divorced    university.degree      no     yes   no
     2   37      services  divorced             basic.9y      no      no   no
     3   44 self-employed   married             basic.9y      no     yes   no
     4   45   blue-collar  divorced             basic.9y      no      no   no

        contact month day_of_week  …  campaign  pdays  previous     poutcome  \
     0 cellular   mar         fri  …         5    999         0  nonexistent
     1 cellular   aug         mon  …         2    999         2      failure
     2 cellular   jul         wed  …         1    999         0  nonexistent
     3 cellular   nov         thu  …         4    999         1      failure
     4 cellular   apr         fri  …         3    999         0  nonexistent

       emp.var.rate  cons.price.idx  cons.conf.idx  euribor3m  nr.employed    y
     0         -1.8          92.843          -50.0      1.726       5099.1   no
     1         -2.9          92.201          -31.4      0.884       5076.2  yes
     2          1.4          93.918          -42.7      4.957       5228.1   no
     3         -0.1          93.200          -42.0      4.076       5195.8   no
     4         -1.8          93.075          -47.1      1.405       5099.1   no

     [5 rows x 21 columns]
```

```
[4]: data['job'].apply(lambda val: val.rstrip(".,!?"))
```

```
[4]: 0          technician
     1               admin
     2            services
     3       self-employed
     4         blue-collar
                 …
     6173            admin
     6174      blue-collar
     6175       technician
     6176            admin
     6177      blue-collar
     Name: job, Length: 6178, dtype: object
```

```
[5]: data["education"].apply(lambda val: val.replace('.', '-'))
```

[5]:
```
0       professional-course
1         university-degree
2                  basic-9y
3                  basic-9y
4                  basic-9y
               …
6173            high-school
6174               basic-4y
6175    professional-course
6176            high-school
6177               basic-9y
Name: education, Length: 6178, dtype: object
```

[7]:
```
new_data = util.transformBinary(df = data, columns = ['default',␣
 ↪'housing','loan','y'])
new_data
```

[7]:
```
      age            job   marital            education default housing  \
0      38     technician   married  professional.course   False    True
1      36         admin.  divorced    university.degree   False    True
2      37       services  divorced             basic.9y   False   False
3      44  self-employed   married             basic.9y   False    True
4      45    blue-collar  divorced             basic.9y   False   False
…   …            …       …                  …     …     …
6173   58         admin.  divorced          high.school   False   False
6174   34    blue-collar   married             basic.4y   False    True
6175   34     technician   married  professional.course   False     NaN
6176   37         admin.   married          high.school   False   False
6177   32    blue-collar   married             basic.9y   False    True

       loan    contact month day_of_week  …  campaign  pdays  previous  \
0     False   cellular   mar         fri  …         5    999         0
1     False   cellular   aug         mon  …         2    999         2
2     False   cellular   jul         wed  …         1    999         0
3     False   cellular   nov         thu  …         4    999         1
4     False   cellular   apr         fri  …         3    999         0
…     …        …   …         …  …       …    …       …
6173  False  telephone   oct         thu  …         1    999         0
6174  False  telephone   may         tue  …         2    999         0
6175    NaN   cellular   may         thu  …         3    999         0
6176   True   cellular   nov         mon  …         3    999         0
6177  False  telephone   may         fri  …         1    999         0

         poutcome emp.var.rate  cons.price.idx  cons.conf.idx  euribor3m  \
0     nonexistent         -1.8          92.843          -50.0      1.726
1         failure         -2.9          92.201          -31.4      0.884
2     nonexistent          1.4          93.918          -42.7      4.957
```

```
3         failure         -0.1         93.200          -42.0      4.076
4     nonexistent         -1.8         93.075          -47.1      1.405
…           …          …             …            …        …
6173  nonexistent         -0.1         93.798          -40.4      4.794
6174  nonexistent          1.1         93.994          -36.4      4.857
6175  nonexistent         -1.8         92.893          -46.2      1.327
6176  nonexistent         -3.4         92.649          -30.1      0.722
6177  nonexistent          1.1         93.994          -36.4      4.855

      nr.employed      y
0          5099.1  False
1          5076.2   True
2          5228.1  False
3          5195.8  False
4          5099.1  False
…           …    …
6173       5195.8  False
6174       5191.0  False
6175       5099.1  False
6176       5017.5  False
6177       5191.0  False

[6178 rows x 21 columns]
```

[8]:
```
new_data['age_group']=new_data['age'].apply(lambda val: util.AgeMasking(val))
```

[9]:
```
new_data['age'].describe()
```

[9]:
```
count    6178.000000
mean       39.825834
std        10.461371
min        18.000000
25%        32.000000
50%        38.000000
75%        47.000000
max        98.000000
Name: age, dtype: float64
```

Calculate stats

[11]:
```
new_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6178 entries, 0 to 6177
Data columns (total 22 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             6178 non-null   int64
 1   job             6178 non-null   object
 2   marital         6178 non-null   object
 3   education       6178 non-null   object
 4   default         4890 non-null   object
 5   housing         6026 non-null   object
 6   loan            6026 non-null   object
 7   contact         6178 non-null   object
 8   month           6178 non-null   object
 9   day_of_week     6178 non-null   object
 10  duration        6178 non-null   int64
 11  campaign        6178 non-null   int64
 12  pdays           6178 non-null   int64
 13  previous        6178 non-null   int64
 14  poutcome        6178 non-null   object
 15  emp.var.rate    6178 non-null   float64
 16  cons.price.idx  6178 non-null   float64
 17  cons.conf.idx   6178 non-null   float64
 18  euribor3m       6178 non-null   float64
 19  nr.employed     6178 non-null   float64
 20  y               6178 non-null   bool
 21  age_group       6178 non-null   object
dtypes: bool(1), float64(5), int64(5), object(11)
memory usage: 1019.7+ KB
```

[12]:

```
new_data.pivot_table(
    values='duration',
    index=['education', 'housing'],
    columns=['marital'],
    aggfunc='mean',
    fill_value="Null"
    )
```

[12]:

```
marital                         divorced     married      single unknown
education           housing
basic.4y            False     173.827586  248.829787       287.0    Null
                    True      317.864865  276.590717       180.0    Null
basic.6y            False     427.285714  260.942029  349.166667    Null
                    True      214.777778  264.066667  323.318182    Null
basic.9y            False     264.333333   252.69863  249.063158    Null
                    True          241.38   253.84984  260.612613   139.0
high.school         False     203.255556   283.79697  257.248908    95.0
                    True          251.25  245.051345  264.927419    Null
illiterate          False          146.0        Null        Null    Null
                    True            Null        51.0       259.0    Null
professional.course False     217.057143  240.297674  298.695238    Null
                    True      316.189655  259.964567  186.789855    Null
university.degree   False     249.340909  242.041284  243.841085   152.5
```

```
                    True     304.139535  225.542533       247.6    251.5
unknown             False    143.222222  227.236842    293.9375    170.0
                    True         202.75  289.457831  331.181818     49.0
```

[72]:

```python
new_data['age_group']=new_data['age'].apply(lambda val: util.AgeMasking(val))
```

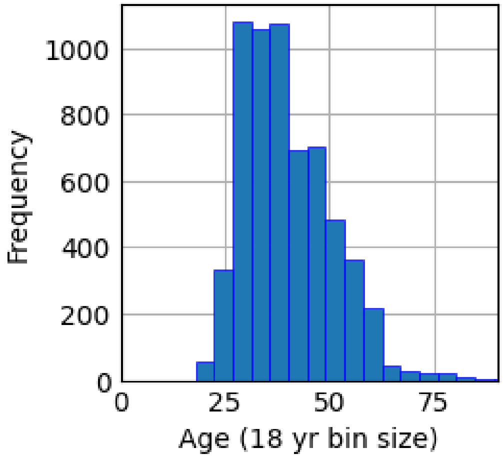
[25]:

```python
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

ax.hist(new_data['age'], bins = 18, linewidth = 0.5, edgecolor = 'blue')
ax.set(xlim=(0, 90), xlabel= "Age (18 yr bin size)", ylabel= "Frequency")

plt.show()
```



[ ]:

```python
plt.style.use('_mpl-gallery')
bar_fig, bar_ax = plt.subplot()
bar_ax.bar(new_data['job'], new_data, edgecolor = 'white', linewidth = 0.5)
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[29], line 2
      1 plt.style.use('_mpl-gallery')
----> 2 bar_fig, bar_ax = plt.subplot()
      3 bar_ax.bar(new_data['job'].to_list(), new_data, edgecolor = 'white',␣
  ↪linewidth = 0.5)
      4 plt.show()
```

```
TypeError: cannot unpack non-iterable Axes object
```