# question-3

November 15, 2023

## 1 QUESTION 3

**NAME: RISHAV KUMAR**

**ROLL NO. 2301560042** MY GITHUB LINK: click me

---

### 1.1 Data Validation with Voluptuous (Schema Definitions)

[ ]:
```
import logging
import pandas as pd
from datetime import datetime
from voluptuous import Schema, Required, Range, All, ALLOW_EXTRA
from voluptuous.error import MultipleInvalid, Invalid
```

[ ]:
```
logger = logging.getLogger(0)
logger.setLevel(logging.WARNING)
```

[ ]:
```
path =r'C:\Users\risha\Documents\KRMU\AIML_assigment\datasets\sales_data.csv'
sales = pd.read_csv(path)
sales.head()
```

```
   Unnamed: 0            timestamp          city  store_id  sale_number  \
0           0  2018-09-10 05:00:45  Williamburgh         6         1530
1           1  2018-09-12 10:01:27    Ibarraberg         1         2744
2           2  2018-09-13 12:01:48   Sarachester         2         1908
3           3  2018-09-14 20:02:19  Caldwellbury        14          771
4           4  2018-09-16 01:03:21     Erikaland        11         1571

   sale_amount         associate
0       1167.0          Gary Lee
1        258.0      Daniel Davis
2        266.0      Michael Roth
3       -108.0  Michaela Stewart
4       -372.0       Mark Taylor
```

[ ]:
```
sales=sales.drop(['Unnamed: 0'], axis=1)
```

[ ]:
```
sales.dtypes
```

```
timestamp       object
city            object
store_id         int64
sale_number      int64
sale_amount    float64
associate       object
dtype: object
```

[ ]:
```
sales['timestamp'].map(lambda x: datetime.strptime(x,'%Y-%m-%d %H:%M:%S'))
```

```
0     2018-09-10 05:00:45
1     2018-09-12 10:01:27
2     2018-09-13 12:01:48
3     2018-09-14 20:02:19
4     2018-09-16 01:03:21
              …
208   2019-09-01 06:46:44
209   2019-09-03 12:47:26
210   2019-09-05 18:47:30
211   2019-09-07 23:48:08
212   2018-09-09 04:48:48
Name: timestamp, Length: 213, dtype: datetime64[ns]
```

### 1.1.1 Data Quality Check

[ ]:
```
sales.head()
```

```
             timestamp          city  store_id  sale_number  sale_amount  \
0  2018-09-10 05:00:45  Williamburgh         6         1530       1167.0
1  2018-09-12 10:01:27    Ibarraberg         1         2744        258.0
2  2018-09-13 12:01:48   Sarachester         2         1908        266.0
3  2018-09-14 20:02:19  Caldwellbury        14          771       -108.0
4  2018-09-16 01:03:21     Erikaland        11         1571       -372.0

          associate
0         Gary Lee
1     Daniel Davis
2     Michael Roth
3  Michaela Stewart
4      Mark Taylor
```

[ ]:
```
sales.dtypes
```

```
timestamp       object
city            object
store_id         int64
```

```
sale_number      int64
sale_amount    float64
associate       object
dtype: object
```

### 1.2 Defining our first schema

[ ]:
```
schema = Schema({ Required('sale_amount'): All(float, Range(min=2.50, max=1450.
↪99)),}, extra=ALLOW_EXTRA)
```

[ ]:
```
error_count = 0
for s_id, sale in sales.T.to_dict().items():
    try:
        schema(sale)
    except MultipleInvalid as e:
        logging.warning('issue with sale: %s (%s) - %s', s_id,␣
↪sale['sale_amount'], e)
        error_count += 1
```

```
WARNING:root:issue with sale: 3 (-108.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 4 (-372.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 5 (-399.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 6 (-304.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 7 (-295.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 10 (-89.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 13 (-303.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 15 (-432.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 19 (-177.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 20 (-154.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 22 (-130.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 23 (1487.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 25 (-145.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 28 (1471.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 31 (-259.0) - value must be at least 2.5 for
```

```
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 38 (-241.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 40 (-4.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 41 (1581.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 45 (1529.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 46 (-238.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 48 (-284.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 51 (-164.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 55 (-184.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 56 (-304.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 59 (1579.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 60 (-455.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 63 (1551.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 65 (-397.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 69 (-400.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 70 (1482.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 71 (-321.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 74 (-47.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 76 (-68.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 86 (1454.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 101 (-213.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 103 (-144.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 104 (-265.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 107 (-349.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 111 (-78.0) - value must be at least 2.5 for
```

```
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 112 (-310.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 116 (1570.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 120 (1490.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 123 (-179.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 124 (-391.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 129 (1504.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 130 (-91.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 132 (-372.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 141 (1512.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 142 (-449.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 149 (1494.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 152 (-405.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 155 (1599.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 156 (1527.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 157 (-462.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 162 (-358.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 164 (-78.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 167 (-358.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 171 (-391.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 178 (-304.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 180 (-9.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 187 (1475.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 194 (-433.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 195 (-329.0) - value must be at least 2.5 for
```

```
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 196 (-147.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 203 (-319.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 206 (-132.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 207 (-20.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 209 (1539.0) - value must be at most 1450.99 for
dictionary value @ data['sale_amount']
WARNING:root:issue with sale: 211 (-167.0) - value must be at least 2.5 for
dictionary value @ data['sale_amount']
```

[ ]:
```
error_count
```

```
69
```

[ ]:
```
sales.shape
```

```
(213, 6)
```

### 1.2.1 Questions we might want to answer:

- Do we have an improperly defined schema?
- Are negative values possibly returns or falsely marked? (data entry proceedures)
- Are higher values combined purchases or special sales? (or potentially fraud?)
- What should we do with our schema and our failing data points?

### 1.2.2 Adding a custom Validation Case

[ ]:
```
def ValidDate(fmt='%Y-%m-%d %H:%M:%S'):
    return lambda v: datetime.strptime(v, fmt)
```

[ ]:
```
schema = Schema({
    Required('timestamp'): All(ValidDate()),}, extra=ALLOW_EXTRA)
```

[ ]:
```
error_count = 0
for s_id, sale in sales.T.to_dict().items():
    try:
        schema(sale)
    except MultipleInvalid as e:
        logging.warning('issue with sale: %s (%s) - %s', s_id,␣
 ↪sale['timestamp'], e)
        error_count += 1
```

[ ]:
```
error_count
```

0

### 1.3 So we have valid date structures, what about actual valid dates?

[ ]:
```
def ValidDate(fmt='%Y-%m-%d %H:%M:%S'):
    def validation_func(v):
        try:
            assert datetime.strptime(v, fmt) <= datetime.now()
        except AssertionError:
            raise Invalid('date is in the future! %s' % v)
    return validation_func
```

[ ]:
```
schema = Schema({
    Required('timestamp'): All(ValidDate()),}, extra=ALLOW_EXTRA)
```

[ ]:
```
error_count = 0
for s_id, sale in sales.T.to_dict().items():
    try:
        schema(sale)
    except MultipleInvalid as e:
        logging.warning('issue with sale: %s (%s) - %s',
                        s_id, sale['timestamp'], e)
        error_count += 1
```

[ ]:
```
error_count
```

0

[ ]:

[ ]:
```
import pandas as pd
import numpy as np
```

[ ]:
```
df = pd.read_csv(r'C:
 ↪\Users\risha\Documents\KRMU\AIML_assigment\datasets\HVAC_with_nulls.csv',␣
 ↪encoding='utf-8')
```

[ ]:
```
df.head()
```

| | Date | Time | TargetTemp | ActualTemp | System | SystemAge | BuildingID | 10 |
|---|---|---|---|---|---|---|---|---|
| 0 | 6/1/13 | 0:00:01 | 66.0 | 58 | 13 | 20.0 | 4 | NaN |
| 1 | 6/2/13 | 1:00:01 | NaN | 68 | 3 | 20.0 | 17 | NaN |
| 2 | 6/3/13 | 2:00:01 | 70.0 | 73 | 17 | 20.0 | 18 | NaN |
| 3 | 6/4/13 | 3:00:01 | 67.0 | 63 | 2 | NaN | 15 | NaN |
| 4 | 6/5/13 | 4:00:01 | 68.0 | 74 | 16 | 9.0 | 3 | NaN |

[ ]:
```
df.isnull().sum()
```

```
Date             0
Time             0
TargetTemp     760
ActualTemp       0
System           0
SystemAge      753
BuildingID       0
10            8000
dtype: int64
```

[ ]:
```
df.isna()
```

```
       Date   Time  TargetTemp  ActualTemp  System  SystemAge  BuildingID  \
0     False  False       False       False   False      False       False
1     False  False        True       False   False      False       False
2     False  False       False       False   False      False       False
3     False  False       False       False   False       True       False
4     False  False       False       False   False      False       False
…     …    …          …          …      …         …          …
7995  False  False       False       False   False      False       False
7996  False  False       False       False   False      False       False
7997  False  False        True       False   False      False       False
7998  False  False       False       False   False      False       False
7999  False  False       False       False   False      False       False

        10
0     True
1     True
2     True
3     True
4     True
…    …
7995  True
7996  True
7997  True
7998  True
7999  True

[8000 rows x 8 columns]
```

[ ]:
```
df.drop_duplicates()
```

```
         Date     Time  TargetTemp  ActualTemp  System  SystemAge  BuildingID  \
0      6/1/13  0:00:01        66.0          58      13       20.0           4
1      6/2/13  1:00:01         NaN          68       3       20.0          17
2      6/3/13  2:00:01        70.0          73      17       20.0          18
3      6/4/13  3:00:01        67.0          63       2        NaN          15
4      6/5/13  4:00:01        68.0          74      16        9.0           3
```

```
…       …       …           …          …      …         …          …
7995  6/16/13  1:33:07        66.0          58      17       18.0          20
7996  6/17/13  2:33:07        68.0          72      17       27.0          12
7997  6/18/13  3:33:07         NaN          69      10        4.0           3
7998  6/19/13  4:33:07        65.0          63       7       23.0          20
7999  6/20/13  5:33:07        66.0          66       9       21.0           3

      10
0     NaN
1     NaN
2     NaN
3     NaN
4     NaN
…    ..
7995 NaN
7996 NaN
7997 NaN
7998 NaN
7999 NaN

[8000 rows x 8 columns]
```

[ ]:
```
df=df.drop(['10'], axis=1)
```

[ ]:
```
df['TargetTemp'] = df['TargetTemp'].fillna(df['TargetTemp'].mean())
```

[ ]:
```
df.TargetTemp.isna().sum()
```

```
0
```

[ ]:
```
df.ActualTemp.isna().sum()
```

```
0
```

[ ]:
```
df.SystemAge.isna().sum()
```

```
753
```

[ ]:
```
df.SystemAge= df.SystemAge.fillna(df.SystemAge.mean())
```

[ ]:
```
df.SystemAge.isna().sum()
```

```
0
```

[ ]:
```

```

### 1.3.1 Managing Nulls

[ ]:
```
import pandas as pd
from numpy import random
```

[ ]:
```
df = pd.read_csv(r'C:
↪\Users\risha\Documents\KRMU\AIML_assigment\datasets\iot_example_with_nulls.
↪csv')
```

### 1.3.2 Data Quality Check

[ ]:
```
df.head()
```

```
             timestamp        username  temperature  heartrate  \
0  2017-01-01T12:00:23    michaelsmith         12.0         67
1  2017-01-01T12:01:09       kharrison          6.0         78
2  2017-01-01T12:01:34       smithadam          5.0         89
3  2017-01-01T12:02:09  eddierodriguez         28.0         76
4  2017-01-01T12:02:36       kenneth94         29.0         62

                                  build  latest      note
0  4e6a7805-8faa-2768-6ef6-eb3198b483ac     0.0  interval
1  7256b7b0-e502-f576-62ec-ed73533c9c84     0.0      wake
2  9226c94b-bb4b-a6c8-8e02-cb42b53e9c90     0.0       NaN
3                                   NaN     0.0    update
4  122f1c6a-403c-2221-6ed1-b5caa08f11e0     NaN       NaN
```

[ ]:
```
df.dtypes
```

```
timestamp       object
username        object
temperature    float64
heartrate        int64
build           object
latest         float64
note            object
dtype: object
```

[ ]:
```
df.note.value_counts()
```

```
note
wake        16496
user        16416
interval    16274
sleep       16226
update      16213
test        16068
Name: count, dtype: int64
```

### 1.3.3 Let's remove all null values (including the note: n/a)

[ ]:
```
df = pd.read_csv(r'C:
↪\Users\risha\Documents\KRMU\AIML_assigment\datasets\iot_example_with_nulls.
↪csv', na_values=['n/a'])
```

### 1.3.4 Test to see if we can use dropna

[ ]:
```
df.shape
```

```
(146397, 7)
```

[ ]:
```
df.dropna().shape
```

```
(46116, 7)
```

[ ]:
```
df.dropna(how='all', axis=1).shape
```

```
(146397, 7)
```

### 1.3.5 Test to see if we can drop columns

[ ]:
```
my_columns = list(df.columns)
```

[ ]:
```
my_columns
```

```
['timestamp',
 'username',
 'temperature',
 'heartrate',
 'build',
 'latest',
 'note']
```

[ ]:
```
list(df.dropna(thresh=int(df.shape[0] * .9), axis=1).columns)
```

```
['timestamp', 'username', 'heartrate']
```

### 1.3.6 I want to find all columns that have missing data

[ ]:
```
missing_info = list(df.columns[df.isnull().any()])
```

[ ]:
```
missing_info
```

```
['temperature', 'build', 'latest', 'note']
```

[ ]:
```
for col in missing_info:
    num_missing = df[df[col].isnull() == True].shape[0]
```

```
    print('number missing for column {}: {}'.format(col,
                                                    num_missing))
```

```
number missing for column temperature: 32357
number missing for column build: 32350
number missing for column latest: 32298
number missing for column note: 48704
```

[ ]:
```
for col in missing_info:
    percent_missing = df[df[col].isnull() == True].shape[0] / df.shape[0]
    print('percent missing for column {}: {}'.format(
        col, percent_missing))
```

```
percent missing for column temperature: 0.22102228870810195
percent missing for column build: 0.22097447352063226
percent missing for column latest: 0.22061927498514314
percent missing for column note: 0.332684412931959
```

### 1.3.7 Can I easily substitute majority values in for missing data?

[ ]:
```
df.note.value_counts()
```

```
note
wake        16496
user        16416
interval    16274
sleep       16226
update      16213
test        16068
Name: count, dtype: int64
```

[ ]:
```
df.build.value_counts().head()
```

```
build
4e6a7805-8faa-2768-6ef6-eb3198b483ac    1
12aefc6b-272c-751e-6117-134ee73e2649    1
fd4049c3-2297-14ac-a27e-6da57129dd10    1
0bcfab8f-bc25-3f8f-8585-0614e1555fd1    1
b0de05dd-2860-abbb-8be6-f5c0e30ca063    1
Name: count, dtype: int64
```

[ ]:
```
df.latest.value_counts()
```

```
latest
0.0    75735
1.0    38364
Name: count, dtype: int64
```

[ ]:
```
df.latest = df.latest.fillna(0)
```

### 1.3.8 Have not yet addressed temperature missing values... Let's find a way to fill

[ ]:
```
df.username.value_counts().head()
```

```
username
esmith    45
zsmith    43
vsmith    41
ysmith    40
jsmith    37
Name: count, dtype: int64
```

[ ]:
```
df = df.set_index('timestamp')
```

[ ]:
```
df.head()
```

| timestamp | username | temperature | heartrate | \ |
|---|---|---|---|---|
| 2017-01-01T12:00:23 | michaelsmith | 12.0 | 67 | |
| 2017-01-01T12:01:09 | kharrison | 6.0 | 78 | |
| 2017-01-01T12:01:34 | smithadam | 5.0 | 89 | |
| 2017-01-01T12:02:09 | eddierodriguez | 28.0 | 76 | |
| 2017-01-01T12:02:36 | kenneth94 | 29.0 | 62 | |

| timestamp | build | latest | note |
|---|---|---|---|
| 2017-01-01T12:00:23 | 4e6a7805-8faa-2768-6ef6-eb3198b483ac | 0.0 | interval |
| 2017-01-01T12:01:09 | 7256b7b0-e502-f576-62ec-ed73533c9c84 | 0.0 | wake |
| 2017-01-01T12:01:34 | 9226c94b-bb4b-a6c8-8e02-cb42b53e9c90 | 0.0 | NaN |
| 2017-01-01T12:02:09 | NaN | 0.0 | update |
| 2017-01-01T12:02:36 | 122f1c6a-403c-2221-6ed1-b5caa08f11e0 | 0.0 | NaN |

[ ]:
```
df.temperature = df.groupby(df['username']).temperature.fillna(df.temperature.
↪mean())
```

### 1.3.9 Exercise: How many temperature values did I fill? What percentage of values are still missing (for temperature)?

[ ]:
```
rows_filled = 32357 - df[df.temperature.isnull() == True].shape[0]
still_missing = df[df.temperature.isnull() == True].shape[0] / df.shape[0]
```

[ ]:
```
rows_filled
```

```
32357
```

[ ]:
```
still_missing
```

0.0

[ ]:
```

```

[ ]:
```
from fuzzywuzzy import fuzz, process
```

[ ]:
```
berlin = ['Berlin, Germany',
          'Berlin, Deutschland',
          'Berlin',
          'Berlin, DE']
```

[ ]:
```
fuzz.partial_ratio(berlin[0], berlin[1])
```

60

[ ]:
```
fuzz.ratio?
```

```
Signature:
fuzz.ratio(s1,
s2)
Docstring: <no docstring>
File:
c:\users\risha\appdata\local\programs\python\python312\lib\site-
packages\fuzzywuzzy\fuzz.py
Type:      function
```

[ ]:
```
fuzz.ratio(berlin[0], berlin[1])
```

65

[ ]:
```
fuzz.token_set_ratio(berlin[0], berlin[1])
```

62

[ ]:
```
fuzz.token_sort_ratio(berlin[0], berlin[1])
```

62

[ ]:
```
fuzz.partial_ratio(berlin[1], berlin[2])
```

100

[ ]:
```
fuzz.ratio(berlin[1], berlin[2])
```

48

[ ]:
```
fuzz.token_sort_ratio(berlin[1], berlin[2])
```

```
50
```

[ ]:
```
fuzz.token_set_ratio(berlin[2], berlin[3])
```

```
100
```

[ ]:
```
choices = ['Germany', 'Deutschland', 'France',
           'United Kingdom', 'Great Britain',
           'United States']
```

[ ]:
```
process.extract('DE', choices, limit=2)
```

```
[('Deutschland', 90), ('United States', 57)]
```

[ ]:
```
process.extract('UK', choices)
```

```
[('Deutschland', 45),
 ('United Kingdom', 45),
 ('United States', 45),
 ('Germany', 0),
 ('France', 0)]
```

[ ]:
```
process.extract('frankreich', choices)
```

```
[('France', 62),
 ('Great Britain', 41),
 ('Germany', 35),
 ('United Kingdom', 25),
 ('United States', 25)]
```

[ ]:
```
process.extract('U.S.', choices)
```

```
[('United States', 86),
 ('Deutschland', 60),
 ('United Kingdom', 57),
 ('Great Britain', 30),
 ('Germany', 0)]
```

[ ]:
```
```

[ ]:
```
from sklearn import preprocessing
import pandas as pd
from datetime import datetime
from sklearn.impute import SimpleImputer
```

[ ]:
```
hvac = pd.read_csv(r"C:
↪\Users\risha\Documents\KRMU\AIML_assigment\datasets\HVAC_with_nulls.csv")
```

### 1.3.10 Checking Data Quality

[ ]:
```
hvac.dtypes
```

```
Date           object
Time           object
TargetTemp    float64
ActualTemp      int64
System          int64
SystemAge     float64
BuildingID      int64
10            float64
dtype: object
```

[ ]:
```
hvac.shape
```

```
(8000, 8)
```

[ ]:
```
hvac= hvac.drop(['10'], axis=1)
```

[ ]:
```
hvac.head()
```

| | Date | Time | TargetTemp | ActualTemp | System | SystemAge | BuildingID |
|---|---|---|---|---|---|---|---|
| 0 | 6/1/13 | 0:00:01 | 66.0 | 58 | 13 | 20.0 | 4 |
| 1 | 6/2/13 | 1:00:01 | NaN | 68 | 3 | 20.0 | 17 |
| 2 | 6/3/13 | 2:00:01 | 70.0 | 73 | 17 | 20.0 | 18 |
| 3 | 6/4/13 | 3:00:01 | 67.0 | 63 | 2 | NaN | 15 |
| 4 | 6/5/13 | 4:00:01 | 68.0 | 74 | 16 | 9.0 | 3 |

### 1.3.11 Impute missing values with mean

[ ]:
```
# imp = SimpleImputer(missing_values='NaN', strategy='mean')
hvac.TargetTemp= hvac.TargetTemp.fillna(hvac.TargetTemp.mean())
```

[ ]:
```
hvac_numeric = hvac[['TargetTemp', 'SystemAge']]
```

[ ]:
```
hvac.head()
```

| | Date | Time | TargetTemp | ActualTemp | System | SystemAge | BuildingID |
|---|---|---|---|---|---|---|---|
| 0 | 6/1/13 | 0:00:01 | 66.000000 | 58 | 13 | 20.0 | 4 |
| 1 | 6/2/13 | 1:00:01 | 67.507735 | 68 | 3 | 20.0 | 17 |
| 2 | 6/3/13 | 2:00:01 | 70.000000 | 73 | 17 | 20.0 | 18 |
| 3 | 6/4/13 | 3:00:01 | 67.000000 | 63 | 2 | NaN | 15 |
| 4 | 6/5/13 | 4:00:01 | 68.000000 | 74 | 16 | 9.0 | 3 |

### 1.3.12 Scale temperature values

[ ]:
```
hvac['ScaledTemp'] = preprocessing.scale(hvac['ActualTemp'])
```

[ ]:
```
hvac['ScaledTemp'].head()
```

```
0   -1.293272
1    0.048732
2    0.719733
3   -0.622270
4    0.853934
Name: ScaledTemp, dtype: float64
```

### 1.3.13 Scale using a min and max scaler

[ ]:
```
min_max_scaler = preprocessing.MinMaxScaler()
```

[ ]:
```
temp_minmax = min_max_scaler.fit_transform(hvac[['ActualTemp']])
```

[ ]:
```
temp_minmax
```

```
array([[0.12],
       [0.52],
       [0.72],
       …,
       [0.56],
       [0.32],
       [0.44]])
```

### 1.3.14 Exercise: add the temp_minmax back to the dataframe as a new column

[ ]:
```
hvac['MinMaxScaledTemp'] = temp_minmax[:,0]
hvac['MinMaxScaledTemp'].head()
```

```
0    0.12
1    0.52
2    0.72
3    0.32
4    0.76
Name: MinMaxScaledTemp, dtype: float64
```

[ ]:
```

```

## 1.4 Case Study: Preparing Lobste.rs

[ ]:
```
import pandas as pd
import requests
from fuzzywuzzy import fuzz
from collections import Counter
from sklearn import preprocessing
```

### 1.4.1 If you'd rather read from the API to get the latest, uncomment the details (and add comment to the final line)

[ ]:
```
stories = pd.read_json(r'C:
↪\Users\risha\Documents\KRMU\AIML_assigment\datasets\all_lobsters.json')
```

[ ]:
```
stories.head()
```

```
        comment_count                                        comments_url  \
09zw7r              0           https://lobste.rs/s/09zw7r/edited_truth
0bdne7             17  https://lobste.rs/s/0bdne7/rise_social_media_v…
1bhbod             11  https://lobste.rs/s/1bhbod/tcl_misunderstood_a…
1xkje1              0  https://lobste.rs/s/1xkje1/interview_4_jonatha…
2dasvh             19     https://lobste.rs/s/2dasvh/return_hipster_pda

                created_at                                        description  \
09zw7r 2017-08-08 20:11:09
0bdne7 2017-08-08 21:12:38
1bhbod 2017-04-30 20:28:52  <p>Did any language end up taking that "highly…
1xkje1 2017-05-01 02:31:35  <p>Rust's own Jonathan Turner on his backgroun…
2dasvh 2017-08-08 14:25:29

        downvotes               last_updated  score  \
09zw7r          0  2017-08-09T11:03:57.014269      3
0bdne7          9  2017-08-09T11:03:57.014269     -1
1bhbod          0  2017-05-01T06:29:11.725518     17
1xkje1          0  2017-05-01T06:29:11.725518      1
2dasvh          0  2017-08-09T11:03:56.287654     20

                    short_id_url  \
09zw7r  https://lobste.rs/s/09zw7r
0bdne7  https://lobste.rs/s/0bdne7
1bhbod  https://lobste.rs/s/1bhbod
1xkje1  https://lobste.rs/s/1xkje1
2dasvh  https://lobste.rs/s/2dasvh

                                          submitter_user  \
09zw7r  {'avatar_url': 'https://lobste.rs/avatars/trn-…
0bdne7  {'avatar_url': 'https://lobste.rs/avatars/nkhu…
```

```
1bhbod  {'is_moderator': False, 'is_admin': False, 'us…
1xkje1  {'is_moderator': False, 'is_admin': False, 'us…
2dasvh  {'created_at': '2017-01-19T14:56:50.000-06:00'…

                            tags                                      title  \
09zw7r              [crypto, pdf]                          The Edited Truth
0bdne7            [law, privacy]    The Rise of The Social Media Vigilante
1bhbod             [programming]           Tcl the misunderstood - antirez
1xkje1  [audio, javascript, rust]   Interview 4 - Jonathan Turner: Part 1/3
2dasvh               [practices]             The Return of the Hipster PDA

        upvotes                                              url
09zw7r        3              https://eprint.iacr.org/2017/714.pdf
0bdne7        8  https://medium.com/@nkhumphreys_89452/the-rise…
1bhbod       17  http://antirez.com/articoli/tclmisunderstood.html
1xkje1        1  http://www.newrustacean.com/show_notes/intervi…
2dasvh       20  http://www.agilesysadmin.net/return-of-the-hip…
```

[ ]:
```
stories.dtypes
```

```
comment_count              int64
comments_url              object
created_at        datetime64[ns]
description               object
downvotes                  int64
last_updated              object
score                      int64
short_id_url              object
submitter_user            object
tags                      object
title                     object
upvotes                    int64
url                       object
dtype: object
```

### 1.4.2 Let's take a look at the submitter_user field, as it appears like a dict

[ ]:
```
stories.submitter_user.iloc[3]
```

```
{'is_moderator': False,
 'is_admin': False,
 'username': 'chriskrycho',
 'karma': 27,
 'avatar_url': 'https://secure.gravatar.com/avatar/
 ↪c096ed07142659408dc6651f8320acd3?r=pg&d=identicon&s=100',
 'created_at': '2016-08-15T09:33:28.000-05:00',
```

```
 'about': "I'm a husband and father; a theologian, composer, poet, and essayist;␣
 ↪a front end developer at [Olo](http://www.olo.com); a [Rust](https://www.
 ↪rust-lang.org/en-US/) enthusiast host; and the host of the [Winning␣
 ↪Slowly](http://www.winningslowly.org), [New Rustacean](http://www.newrustacean.
 ↪com/), [Sap.py](http://www.sap-py.com), and [Run With Me](http://runwith.
 ↪chriskrycho.com/) podcasts."}
```

[ ]:
```
user_df = stories['submitter_user'].apply(pd.Series)
```

[ ]:
```
user_df.head()
```

```
                                                avatar_url  \
09zw7r              https://lobste.rs/avatars/trn-100.png
0bdne7      https://lobste.rs/avatars/nkhumphreys-100.png
1bhbod  https://secure.gravatar.com/avatar/85002353297…
1xkje1  https://secure.gravatar.com/avatar/c096ed07142…
2dasvh              https://lobste.rs/avatars/trn-100.png

                         created_at  is_admin     username  karma  \
09zw7r  2017-01-19T14:56:50.000-06:00     False          trn    429
0bdne7  2014-07-02T06:36:39.000-05:00     False  nkhumphreys     -1
1bhbod  2016-11-30T10:14:24.000-06:00     False     yumaikas    578
1xkje1  2016-08-15T09:33:28.000-05:00     False  chriskrycho     27
2dasvh  2017-01-19T14:56:50.000-06:00     False          trn    429

        is_moderator                                              about  \
09zw7r         False
0bdne7         False  Web developer and previously embedded C developer
1bhbod         False  I blog infrequently at https://junglecoder.com…
1xkje1         False  I'm a husband and father; a theologian, compos…
2dasvh         False

       github_username
09zw7r             NaN
0bdne7             NaN
1bhbod             NaN
1xkje1             NaN
2dasvh             NaN
```

### 1.4.3 Can we combine the user data without potential column overlap?

[ ]:
```
set(user_df.columns).intersection(stories.columns)
```

```
{'created_at'}
```

[ ]:
```
user_df = user_df.rename(columns={'created_at':
                                  'user_created_at'})
```

[ ]:
```
stories = pd.concat([stories.drop(['submitter_user'], axis=1),
                     user_df], axis=1)
```

[ ]:
```
stories.head()
```

```
        comment_count                                      comments_url  \
09zw7r              0           https://lobste.rs/s/09zw7r/edited_truth
0bdne7             17  https://lobste.rs/s/0bdne7/rise_social_media_v…
1bhbod             11  https://lobste.rs/s/1bhbod/tcl_misunderstood_a…
1xkje1              0  https://lobste.rs/s/1xkje1/interview_4_jonatha…
2dasvh             19     https://lobste.rs/s/2dasvh/return_hipster_pda

                created_at                                        description  \
09zw7r 2017-08-08 20:11:09
0bdne7 2017-08-08 21:12:38
1bhbod 2017-04-30 20:28:52  <p>Did any language end up taking that "highly…
1xkje1 2017-05-01 02:31:35  <p>Rust's own Jonathan Turner on his backgroun…
2dasvh 2017-08-08 14:25:29

        downvotes               last_updated  score  \
09zw7r          0  2017-08-09T11:03:57.014269      3
0bdne7          9  2017-08-09T11:03:57.014269     -1
1bhbod          0  2017-05-01T06:29:11.725518     17
1xkje1          0  2017-05-01T06:29:11.725518      1
2dasvh          0  2017-08-09T11:03:56.287654     20

                     short_id_url                       tags  \
09zw7r  https://lobste.rs/s/09zw7r              [crypto, pdf]
0bdne7  https://lobste.rs/s/0bdne7             [law, privacy]
1bhbod  https://lobste.rs/s/1bhbod              [programming]
1xkje1  https://lobste.rs/s/1xkje1  [audio, javascript, rust]
2dasvh  https://lobste.rs/s/2dasvh                [practices]

                                        title  upvotes  \
09zw7r                          The Edited Truth        3
0bdne7      The Rise of The Social Media Vigilante        8
1bhbod             Tcl the misunderstood - antirez       17
1xkje1    Interview 4 - Jonathan Turner: Part 1/3        1
2dasvh               The Return of the Hipster PDA       20

                                                  url  \
09zw7r               https://eprint.iacr.org/2017/714.pdf
0bdne7  https://medium.com/@nkhumphreys_89452/the-rise…
1bhbod  http://antirez.com/articoli/tclmisunderstood.html
1xkje1  http://www.newrustacean.com/show_notes/intervi…
2dasvh  http://www.agilesysadmin.net/return-of-the-hip…
```

```
                                                avatar_url  \
09zw7r              https://lobste.rs/avatars/trn-100.png
0bdne7      https://lobste.rs/avatars/nkhumphreys-100.png
1bhbod  https://secure.gravatar.com/avatar/85002353297…
1xkje1  https://secure.gravatar.com/avatar/c096ed07142…
2dasvh              https://lobste.rs/avatars/trn-100.png

                      user_created_at  is_admin     username  karma  \
09zw7r  2017-01-19T14:56:50.000-06:00     False          trn    429
0bdne7  2014-07-02T06:36:39.000-05:00     False  nkhumphreys     -1
1bhbod  2016-11-30T10:14:24.000-06:00     False     yumaikas    578
1xkje1  2016-08-15T09:33:28.000-05:00     False  chriskrycho     27
2dasvh  2017-01-19T14:56:50.000-06:00     False          trn    429

        is_moderator                                              about  \
09zw7r         False
0bdne7         False  Web developer and previously embedded C developer
1bhbod         False  I blog infrequently at https://junglecoder.com…
1xkje1         False  I'm a husband and father; a theologian, compos…
2dasvh         False

       github_username
09zw7r             NaN
0bdne7             NaN
1bhbod             NaN
1xkje1             NaN
2dasvh             NaN
```

#### 1.4.4 Let's check for nulls

[ ]:
```
stories.shape
```

```
(74, 20)
```

[ ]:
```
stories.dropna().shape
```

```
(8, 20)
```

[ ]:
```
stories.dropna(thresh=10, axis=1).shape
```

```
(74, 19)
```

#### 1.4.5 Exercise: which columns would be dropped?

[ ]:
```
set(stories.columns) - set(stories.dropna(thresh=10, axis=1).columns)
```

```
{'github_username'}
```

### 1.5 Let's make the tags easier to use by having them as features in the columns.

[ ]:
```
tag_df = stories.tags.apply(pd.Series)
```

[ ]:
```
tag_df.head()
```

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 09zw7r | crypto | pdf | NaN | NaN | NaN |
| 0bdne7 | law | privacy | NaN | NaN | NaN |
| 1bhbod | programming | NaN | NaN | NaN | NaN |
| 1xkje1 | audio | javascript | rust | NaN | NaN |
| 2dasvh | practices | NaN | NaN | NaN | NaN |

[ ]:
```
pd.unique(tag_df.values.ravel())
```

```
array(['crypto', 'pdf', nan, 'law', 'privacy', 'programming', 'audio',
       'javascript', 'rust', 'practices', 'ruby', 'devops', 'web',
       'hardware', 'science', 'reversing', 'security', 'openbsd',
       'windows', 'design', 'compilers', 'haskell', 'c++', 'assembly',
       'games', 'math', 'release', 'event', 'netbsd', 'unix', 'c',
       'linux', 'testing', 'lua', 'job', 'video', 'philosophy', 'android',
       'networking', 'erlang', 'emacs', 'historical', 'browsers',
       'person', 'culture', 'java', 'go', 'book', 'css', 'debugging',
       'education', 'art', 'compsci', 'databases'], dtype=object)
```

[ ]:
```
set(tag_df.values.ravel())
```

```
{'android',
 'art',
 'assembly',
 'audio',
 'book',
 'browsers',
 'c',
 'c++',
 'compilers',
 'compsci',
 'crypto',
 'css',
 'culture',
 'databases',
 'debugging',
 'design',
 'devops',
 'education',
 'emacs',
 'erlang',
 'event',
```

```
 'games',
 'go',
 'hardware',
 'haskell',
 'historical',
 'java',
 'javascript',
 'job',
 'law',
 'linux',
 'lua',
 'math',
 nan,
 'netbsd',
 'networking',
 'openbsd',
 'pdf',
 'person',
 'philosophy',
 'practices',
 'privacy',
 'programming',
 'release',
 'reversing',
 'ruby',
 'rust',
 'science',
 'security',
 'testing',
 'unix',
 'video',
 'web',
 'windows'}
```

[ ]:
```
len(pd.unique(tag_df.values.ravel()))
```

```
54
```

[ ]:
```
# most common tags

Counter(tag_df.values.ravel()).most_common(5)
```

```
[(nan, 231),
 ('programming', 13),
 ('hardware', 10),
 ('security', 10),
 ('practices', 8)]
```

### 1.5.1 Let's create a dummy df with our tags

[ ]:
```
tag_df = pd.get_dummies(tag_df.apply(pd.Series).stack()).sum()
```

[ ]:
```
tag_df.head()
```

```
android     1
art         1
assembly    3
audio       1
book        2
dtype: int64
```

### 1.5.2 Now we can add it back to our stories DataFrame

[ ]:
```
stories = pd.concat([stories.drop('tags', axis=1),
                     tag_df], axis=1)
```

[ ]:
```
stories.head()
```

```
        comment_count                                    comments_url  \
09zw7r            0.0           https://lobste.rs/s/09zw7r/edited_truth
0bdne7           17.0  https://lobste.rs/s/0bdne7/rise_social_media_v…
1bhbod           11.0  https://lobste.rs/s/1bhbod/tcl_misunderstood_a…
1xkje1            0.0  https://lobste.rs/s/1xkje1/interview_4_jonatha…
2dasvh           19.0      https://lobste.rs/s/2dasvh/return_hipster_pda

                created_at                                        description  \
09zw7r 2017-08-08 20:11:09
0bdne7 2017-08-08 21:12:38
1bhbod 2017-04-30 20:28:52  <p>Did any language end up taking that "highly…
1xkje1 2017-05-01 02:31:35  <p>Rust's own Jonathan Turner on his backgroun…
2dasvh 2017-08-08 14:25:29

        downvotes               last_updated  score  \
09zw7r        0.0  2017-08-09T11:03:57.014269    3.0
0bdne7        9.0  2017-08-09T11:03:57.014269   -1.0
1bhbod        0.0  2017-05-01T06:29:11.725518   17.0
1xkje1        0.0  2017-05-01T06:29:11.725518    1.0
2dasvh        0.0  2017-08-09T11:03:56.287654   20.0

                     short_id_url  \
09zw7r  https://lobste.rs/s/09zw7r
0bdne7  https://lobste.rs/s/0bdne7
1bhbod  https://lobste.rs/s/1bhbod
1xkje1  https://lobste.rs/s/1xkje1
2dasvh  https://lobste.rs/s/2dasvh
```

```
                                          title  upvotes  \
09zw7r                          The Edited Truth      3.0
0bdne7      The Rise of The Social Media Vigilante      8.0
1bhbod             Tcl the misunderstood - antirez     17.0
1xkje1    Interview 4 - Jonathan Turner: Part 1/3      1.0
2dasvh               The Return of the Hipster PDA     20.0

                                                  url  \
09zw7r               https://eprint.iacr.org/2017/714.pdf
0bdne7  https://medium.com/@nkhumphreys_89452/the-rise…
1bhbod  http://antirez.com/articoli/tclmisunderstood.html
1xkje1  http://www.newrustacean.com/show_notes/intervi…
2dasvh  http://www.agilesysadmin.net/return-of-the-hip…

                                           avatar_url  \
09zw7r              https://lobste.rs/avatars/trn-100.png
0bdne7      https://lobste.rs/avatars/nkhumphreys-100.png
1bhbod  https://secure.gravatar.com/avatar/85002353297…
1xkje1  https://secure.gravatar.com/avatar/c096ed07142…
2dasvh              https://lobste.rs/avatars/trn-100.png

                     user_created_at is_admin     username  karma  \
09zw7r  2017-01-19T14:56:50.000-06:00    False          trn  429.0
0bdne7  2014-07-02T06:36:39.000-05:00    False  nkhumphreys   -1.0
1bhbod  2016-11-30T10:14:24.000-06:00    False     yumaikas  578.0
1xkje1  2016-08-15T09:33:28.000-05:00    False  chriskrycho   27.0
2dasvh  2017-01-19T14:56:50.000-06:00    False          trn  429.0

       is_moderator                                              about  \
09zw7r        False
0bdne7        False  Web developer and previously embedded C developer
1bhbod        False  I blog infrequently at https://junglecoder.com…
1xkje1        False  I'm a husband and father; a theologian, compos…
2dasvh        False

       github_username   0
09zw7r             NaN NaN
0bdne7             NaN NaN
1bhbod             NaN NaN
1xkje1             NaN NaN
2dasvh             NaN NaN
```

### 1.5.3 Another potentially useful feature is the post times...

[ ]:
```
stories['created_hour'] = stories.created_at.map(
    lambda x: x.hour)
```

[ ]:
```
stories['created_dow'] = stories.created_at.map(
    lambda x: x.weekday())
```

### 1.5.4 Let's analyze some of the correlations in our features so far...

[ ]:
```
stories[['created_hour', 'score']].corr()
```

```
              created_hour     score
created_hour      1.000000  0.253917
score             0.253917  1.000000
```

[ ]:
```
stories[['created_dow', 'score']].corr()
```

```
             created_dow     score
created_dow     1.000000 -0.113918
score          -0.113918  1.000000
```

[ ]:
```
stories[['karma', 'score']].corr()
```

```
          karma     score
karma  1.000000 -0.061921
score -0.061921  1.000000
```

[ ]:
```
stories[['comment_count', 'score']].corr()
```

```
               comment_count     score
comment_count       1.000000  0.637632
score               0.637632  1.000000
```

[ ]:
```
stories[[ 'score']].corr()
```

```
       score
score    1.0
```

### 1.5.5 We might also want/need to normalize scores. We can use a Scaler / MinMaxScaler or Normalizer

[ ]:
```
stories['score']=stories['score'].fillna(stories.score.mean())
```

[ ]:
```
normed_score = preprocessing.normalize(stories[['score']])
```

[ ]:
```
normed_score[:5]
```

```
array([[ 1.],
       [-1.],
       [ 1.],
       [ 1.],
       [ 1.]])
```

**hmm... maybe a min-max scaler works better for our needs!**

[ ]:
```
scaler = preprocessing.MinMaxScaler()
```

[ ]:
```
scaled_score = scaler.fit_transform(stories[['score']])
```

[ ]:
```
scaled_score[:5]
```

```
array([[0.07272727],
       [0.        ],
       [0.32727273],
       [0.03636364],
       [0.38181818]])
```

[ ]:
```
stories['scaled_score'] = scaled_score[:,0]
stories['scaled_score']
```

```
09zw7r     0.072727
0bdne7     0.000000
1bhbod     0.327273
1xkje1     0.036364
2dasvh     0.381818
             …
testing    0.155037
unix       0.155037
video      0.155037
web        0.155037
windows    0.155037
Name: scaled_score, Length: 127, dtype: float64
```

[ ]:
```

```