Practical 2

Aim: Data Pre-processing

Code:

```
import gzip
import csv

path = "/content/amazon_reviews_us_Gift_Card_v1_00.tsv.gz"
f = gzip.open(path, 'rt')

reader = csv.reader(f, delimiter = '\t')

header = next(reader)

dataset = []
for line in reader:
  d = dict(zip(header, line))
  for field in ['helpful_votes', 'star_rating', 'total_votes']:
      d[field] = int(d[field])
  for field in ['verified_purchase', 'vine']:
      if d[field] == 'Y':
          d[field] = True
      else:
          d[field] = False
  dataset.append(d)

print(reader)

len(dataset)

dataset.append(d)

dataset[0]

dataset = [d for d in dataset if 'review_date' in d]

len(dataset)

for d in dataset:
  d['yearInt'] = int(d['review_date'][:4])

dataset = [d for d in dataset if d['yearInt'] > 2009]
```

```
len(dataset)

dataset = [d for d in dataset if d['total_votes'] < 3 or d['helpful_vot
es']/d['total_votes'] >= 0.5]

len(dataset)

from collections import defaultdict

nReviewsPerUser = defaultdict(int)

for d in dataset:
  nReviewsPerUser[d['customer_id']] += 1

dataset = [d for d in dataset if nReviewsPerUser[d['customer_id']] >= 2
]

len(dataset)

dataset = [d for d in dataset if len(d['review_body'].split()) >= 10]

len(dataset)
```

Output:

```
<_csv.reader object at 0x7f7c0afc7200>

[ ] len(dataset)

148310

[ ]  dataset.append(d)

[ ] dataset[0]

{'marketplace': 'US',
 'customer_id': '24371595',
 'review_id': 'R27ZP1F1CD0C3Y',
 'product_id': 'B004LLIL5A',
 'product_parent': '346014806',
 'product_title': 'Amazon eGift Card - Celebrate',
 'product_category': 'Gift Card',
 'star_rating': 5,
 'helpful_votes': 0,
 'total_votes': 0,
 'vine': False,
 'verified_purchase': True,
 'review_headline': 'Five Stars',
 'review_body': 'Great birthday gift for a young adult.',
 'review_date': '2015-08-31',
 'yearInt': 2015}

[ ] dataset = [d for d in dataset if 'review_date' in d]

[ ] len(dataset)

148309
```

```
148095
```

```
[ ] dataset = [d for d in dataset if d['total_votes'] < 3 or d['helpful_votes']/d['total_votes'] >= 0.5]
```

```
[ ] len(dataset)

147801
```

```
[ ] from collections import defaultdict
```

```
[ ] nReviewsPerUser = defaultdict(int)
```

```
[ ] for d in dataset:
      nReviewsPerUser[d['customer_id']] += 1
```

```
[ ] dataset = [d for d in dataset if nReviewsPerUser[d['customer_id']] >= 2]
```

```
[ ] len(dataset)

11172
```

```
[ ] dataset = [d for d in dataset if len(d['review_body'].split()) >= 10]
```

```
len(dataset)

7033
```

Code:

```
import pandas as pd
import numpy as np
import seaborn as srs

df = pd.read_csv('train.csv')
df.info()

cols = ['Name', 'Ticket', 'Cabin']
df = df.drop(cols, axis=1)
df.info()

df1 = df.dropna()
df1.info()

dummies = []
cols = ['Pclass', 'Sex', 'Embarked']

for col in cols:
    dummies.append(pd.get_dummies(df[col]))
```

```
titanic_dummies = pd.concat(dummies, axis=1)
print(titanic_dummies)

df = pd.concat((df,titanic_dummies), axis=1)
df.shape

df = df.drop(['Pclass', 'Sex', 'Embarked'], axis=1)

df.info()

df['Age'] = df['Age'].interpolate()
df.info()

X = df.values
y = df['Survived'].values

X = np.delete(X, 1, axis=1)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3
, random_state=0)
```

Output:

```
[ ] <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 891 entries, 0 to 890
    Data columns (total 12 columns):
     #   Column       Non-Null Count  Dtype
    ---  ------       --------------  -----
     0   PassengerId  891 non-null    int64
     1   Survived     891 non-null    int64
     2   Pclass       891 non-null    int64
     3   Name         891 non-null    object
     4   Sex          891 non-null    object
     5   Age          714 non-null    float64
     6   SibSp        891 non-null    int64
     7   Parch        891 non-null    int64
     8   Ticket       891 non-null    object
     9   Fare         891 non-null    float64
     10  Cabin        204 non-null    object
     11  Embarked     889 non-null    object
    dtypes: float64(2), int64(5), object(5)
    memory usage: 83.7+ KB

[ ] cols = ['Name', 'Ticket', 'Cabin']
    df = df.drop(cols, axis=1)
    df.info()

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 891 entries, 0 to 890
    Data columns (total 9 columns):
     #   Column       Non-Null Count  Dtype
    ---  ------       --------------  -----
     0   PassengerId  891 non-null    int64
     1   Survived     891 non-null    int64
     2   Pclass       891 non-null    int64
     3   Sex          891 non-null    object
     4   Age          714 non-null    float64
     5   SibSp        891 non-null    int64
     6   Parch        891 non-null    int64
     7   Fare         891 non-null    float64
     8   Embarked     889 non-null    object
    dtypes: float64(2), int64(5), object(2)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  712 non-null    int64
 1   Survived     712 non-null    int64
 2   Pclass       712 non-null    int64
 3   Sex          712 non-null    object
 4   Age          712 non-null    float64
 5   SibSp        712 non-null    int64
 6   Parch        712 non-null    int64
 7   Fare         712 non-null    float64
 8   Embarked     712 non-null    object
dtypes: float64(2), int64(5), object(2)
memory usage: 55.6+ KB
```

```
dummies = []
cols = ['Pclass', 'Sex', 'Embarked']
```

```
for col in cols:
    dummies.append(pd.get_dummies(df[col]))
```

```
titanic_dummies = pd.concat(dummies, axis=1)
print(titanic_dummies)
```

```
     1  2  3  female  male  C  Q  S
0    0  0  1       0     1  0  0  1
1    1  0  0       1     0  1  0  0
2    0  0  1       1     0  0  0  1
3    1  0  0       1     0  0  0  1
4    0  0  1       0     1  0  0  1
..  .. .. ..     ...   ... .. .. ..
886  0  1  0       0     1  0  0  1
887  1  0  0       1     0  0  0  1
888  0  0  1       1     0  0  0  1
889  1  0  0       0     1  1  0  0
890  0  0  1       0     1  0  1  0

[891 rows x 8 columns]
```

```
[ ] df = pd.concat((df,titanic_dummies), axis=1)
    df.shape
```

```
(891, 17)
```

```
[ ] df = df.drop(['Pclass', 'Sex', 'Embarked'], axis=1)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  712 non-null    int64  
 1   Survived     712 non-null    int64  
 2   Age          712 non-null    float64
 3   SibSp        712 non-null    int64  
 4   Parch        712 non-null    int64  
 5   Fare         712 non-null    float64
 6   1            712 non-null    uint8  
 7   2            712 non-null    uint8  
 8   3            712 non-null    uint8  
 9   female       712 non-null    uint8  
 10  male         712 non-null    uint8  
 11  C            712 non-null    uint8  
 12  Q            712 non-null    uint8  
 13  S            712 non-null    uint8  
dtypes: float64(2), int64(4), uint8(8)
memory usage: 44.5 KB
```

```
] df['Age'] = df['Age'].interpolate()
   df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  712 non-null    int64
 1   Survived     712 non-null    int64
 2   Age          712 non-null    float64
 3   SibSp        712 non-null    int64
 4   Parch        712 non-null    int64
 5   Fare         712 non-null    float64
 6   1            712 non-null    uint8
 7   2            712 non-null    uint8
 8   3            712 non-null    uint8
 9   female       712 non-null    uint8
 10  male         712 non-null    uint8
 11  C            712 non-null    uint8
 12  Q            712 non-null    uint8
 13  S            712 non-null    uint8
dtypes: float64(2), int64(4), uint8(8)
memory usage: 44.5 KB
```

```
] X = df.values
   y = df['Survived'].values
```

```
] X = np.delete(X, 1, axis=1)
```

```
] from sklearn.model_selection import train_test_split
   X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```