statistical_tests_rescaling_feaure_selection_Eva

May 15, 2024

0.0.1 Statistical tests exercise

1) A
2) C
3) B
4) B
5) A
6) B
7) A
8) D
9) C
10) B

7/10

0.0.2 Rescaling data exercise

1) B
2) C
3) C
4) C
5) D
6) D
7) D
8) D
9) C
10) B

10/10

0.0.3 Feature selection of student alcohol consumption dataset from:

https://www.kaggle.com/datasets/uciml/student-alcohol-consumption

[ ]:

```python
#needed packages:
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
from sklearn.feature_selection import f_regression
from sklearn.feature_selection import chi2
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

```
[ ]: #let's import both datasets:
     portug_df = pd.read_csv("../datasets/student-por.csv")
     maths_df = pd.read_csv("../datasets/student-mat.csv")
```

```
[ ]: portug_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 649 entries, 0 to 648
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      649 non-null    object
 1   sex         649 non-null    object
 2   age         649 non-null    int64
 3   address     649 non-null    object
 4   famsize     649 non-null    object
 5   Pstatus     649 non-null    object
 6   Medu        649 non-null    int64
 7   Fedu        649 non-null    int64
 8   Mjob        649 non-null    object
 9   Fjob        649 non-null    object
 10  reason      649 non-null    object
 11  guardian    649 non-null    object
 12  traveltime  649 non-null    int64
 13  studytime   649 non-null    int64
 14  failures    649 non-null    int64
 15  schoolsup   649 non-null    object
 16  famsup      649 non-null    object
 17  paid        649 non-null    object
 18  activities  649 non-null    object
 19  nursery     649 non-null    object
 20  higher      649 non-null    object
 21  internet    649 non-null    object
 22  romantic    649 non-null    object
 23  famrel      649 non-null    int64
 24  freetime    649 non-null    int64
 25  goout       649 non-null    int64
 26  Dalc        649 non-null    int64
 27  Walc        649 non-null    int64
 28  health      649 non-null    int64
 29  absences    649 non-null    int64
 30  G1          649 non-null    int64
 31  G2          649 non-null    int64
 32  G3          649 non-null    int64
dtypes: int64(16), object(17)
```

```
memory usage: 167.4+ KB
```

[ ]:
```
portug_df.shape
```

[ ]:
```
(649, 33)
```

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

[ ]:
```
  school sex  age address famsize Pstatus  Medu  Fedu     Mjob      Fjob  …  \
0     GP   F   18       U     GT3       A     4     4  at_home   teacher  …
1     GP   F   17       U     GT3       T     1     1  at_home     other  …
2     GP   F   15       U     LE3       T     1     1  at_home     other  …
3     GP   F   15       U     GT3       T     4     2   health  services  …
4     GP   F   16       U     GT3       T     3     3    other     other  …

  famrel freetime  goout  Dalc  Walc health absences  G1  G2  G3
0      4        3      4     1     1      3        4   0  11  11
1      5        3      3     1     1      3        2   9  11  11
2      4        3      2     2     3      3        6  12  13  12
3      3        2      2     1     1      5        0  14  14  14
4      4        3      2     1     2      5        0  11  13  13

[5 rows x 33 columns]
```

[ ]:
```
maths_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      395 non-null    object
 1   sex         395 non-null    object
 2   age         395 non-null    int64
 3   address     395 non-null    object
 4   famsize     395 non-null    object
 5   Pstatus     395 non-null    object
 6   Medu        395 non-null    int64
 7   Fedu        395 non-null    int64
 8   Mjob        395 non-null    object
 9   Fjob        395 non-null    object
 10  reason      395 non-null    object
 11  guardian    395 non-null    object
 12  traveltime  395 non-null    int64
 13  studytime   395 non-null    int64
 14  failures    395 non-null    int64
 15  schoolsup   395 non-null    object
 16  famsup      395 non-null    object
 17  paid        395 non-null    object
 18  activities  395 non-null    object
 19  nursery     395 non-null    object
 20  higher      395 non-null    object
 21  internet    395 non-null    object
 22  romantic    395 non-null    object
 23  famrel      395 non-null    int64
 24  freetime    395 non-null    int64
 25  goout       395 non-null    int64
 26  Dalc        395 non-null    int64
 27  Walc        395 non-null    int64
 28  health      395 non-null    int64
 29  absences    395 non-null    int64
 30  G1          395 non-null    int64
 31  G2          395 non-null    int64
 32  G3          395 non-null    int64
dtypes: int64(16), object(17)
memory usage: 102.0+ KB
```

[ ]:

```
maths_df.shape
```

[ ]:

```
(395, 33)
```

[ ]:

```
#check for missing values:
portug_df.isna().sum()
```

[ ]:

```
school        0
sex           0
age           0
address       0
famsize       0
Pstatus       0
Medu          0
Fedu          0
Mjob          0
Fjob          0
reason        0
guardian      0
traveltime    0
studytime     0
failures      0
schoolsup     0
famsup        0
paid          0
activities    0
nursery       0
higher        0
```

```
internet      0
romantic      0
famrel        0
freetime      0
goout         0
Dalc          0
Walc          0
health        0
absences      0
G1            0
G2            0
G3            0
dtype: int64
```

[ ]:

```
#check for missing values:
maths_df.isna().sum()
```

[ ]:

```
school        0
sex           0
age           0
address       0
famsize       0
Pstatus       0
Medu          0
Fedu          0
Mjob          0
Fjob          0
reason        0
guardian      0
traveltime    0
studytime     0
failures      0
schoolsup     0
famsup        0
paid          0
activities    0
nursery       0
higher        0
internet      0
romantic      0
famrel        0
freetime      0
goout         0
Dalc          0
Walc          0
health        0
absences      0
```

```
G1            0
G2            0
G3            0
dtype: int64
```

Close inspection of each variable's description suggests that there only five numerical features namely: age, absences, G1, G2 and G3. Let's first select "goout" as our categorical target.

Need to convert the object type features to factor type

```
[ ]: #how many categorical columns need to be converted to a numeric representation:␣
      ↪not all categorical columns need this treatment

     port_cat = portug_df.
      ↪drop(['age','absences','G1','G2','G3','Medu','Fedu','traveltime','studytime','failures','fa
                                'Dalc','Walc','health','absences'], axis=1)
     port_cat.columns
```

```
[ ]: Index(['school', 'sex', 'address', 'famsize', 'Pstatus', 'Mjob', 'Fjob',
            'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities',
            'nursery', 'higher', 'internet', 'romantic'],
           dtype='object')
```

```
[ ]: math_cat = maths_df.
      ↪drop(['age','absences','G1','G2','G3','Medu','Fedu','traveltime','studytime','failures','fa
                                'Dalc','Walc','health','absences'], axis=1)
     math_cat.columns
```

```
[ ]: Index(['school', 'sex', 'address', 'famsize', 'Pstatus', 'Mjob', 'Fjob',
            'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities',
            'nursery', 'higher', 'internet', 'romantic'],
           dtype='object')
```

```
[ ]: #create copies to factorise the object type columns in port_df and mat_df:
     portug_df_fact = portug_df.copy()
     maths_df_fact = maths_df.copy()
```

```
[ ]: #factorise:
     portug_df_fact[['school', 'sex', 'address', 'famsize', 'Pstatus', 'Mjob',␣
      ↪'Fjob', 'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities',
             'nursery', 'higher', 'internet', 'romantic']] =␣
      ↪portug_df_fact[['school', 'sex', 'address', 'famsize', 'Pstatus', 'Mjob',␣
      ↪'Fjob', 'reason',
                                                                                ␣
      ↪'guardian', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher',
                                                                                ␣
      ↪'internet', 'romantic']].apply(lambda x: pd.factorize(x)[0])
```

```
[ ]: #factorise:
     maths_df_fact[['school', 'sex', 'address', 'famsize', 'Pstatus', 'Mjob',␣
      ↪'Fjob', 'reason', 'guardian', 'schoolsup', 'famsup', 'paid', 'activities',
            'nursery', 'higher', 'internet', 'romantic']] = maths_df_fact[['school',␣
      ↪'sex', 'address', 'famsize', 'Pstatus', 'Mjob', 'Fjob', 'reason',
                                                                             ␣
      ↪'guardian', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher',
                                                                             ␣
      ↪'internet', 'romantic']].apply(lambda x: pd.factorize(x)[0])
```

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

```
[ ]:    school  sex  age  address  famsize  Pstatus  Medu  Fedu  Mjob  Fjob  …  \
     0       0    0   18        0        0        0     4     4     0     0  …
     1       0    0   17        0        0        1     1     1     0     1  …
     2       0    0   15        0        1        1     1     1     0     1  …
     3       0    0   15        0        0        1     4     2     1     2  …
     4       0    0   16        0        0        1     3     3     2     1  …

        famrel  freetime  goout  Dalc  Walc  health  absences  G1  G2  G3
     0       4         3      4     1     1       3         6   5   6   6
     1       5         3      3     1     1       3         4   5   5   6
     2       4         3      2     2     3       3        10   7   8  10
     3       3         2      2     1     1       5         2  15  14  15
     4       4         3      2     1     2       5         4   6  10  10

     [5 rows x 33 columns]
```

```
[ ]: maths_df_fact.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      395 non-null    int64
 1   sex         395 non-null    int64
 2   age         395 non-null    int64
 3   address     395 non-null    int64
 4   famsize     395 non-null    int64
 5   Pstatus     395 non-null    int64
 6   Medu        395 non-null    int64
 7   Fedu        395 non-null    int64
 8   Mjob        395 non-null    int64
 9   Fjob        395 non-null    int64
 10  reason      395 non-null    int64
 11  guardian    395 non-null    int64
 12  traveltime  395 non-null    int64
 13  studytime   395 non-null    int64
 14  failures    395 non-null    int64
 15  schoolsup   395 non-null    int64
 16  famsup      395 non-null    int64
 17  paid        395 non-null    int64
 18  activities  395 non-null    int64
 19  nursery     395 non-null    int64
 20  higher      395 non-null    int64
 21  internet    395 non-null    int64
 22  romantic    395 non-null    int64
 23  famrel      395 non-null    int64
 24  freetime    395 non-null    int64
 25  goout       395 non-null    int64
 26  Dalc        395 non-null    int64
 27  Walc        395 non-null    int64
 28  health      395 non-null    int64
 29  absences    395 non-null    int64
 30  G1          395 non-null    int64
 31  G2          395 non-null    int64
 32  G3          395 non-null    int64
dtypes: int64(33)
memory usage: 102.0 KB
```

Feature selection with categorical X and categorical Y: chi2

- From the quick inspection above, it seems there are no missing values in both datasets, so proceed with feature selection.
- For categorical X and categorical Y, we can use chi2. For this, lets use “goout” as the target variable of the datasets

[ ]:

```
#selecting the features and target variable for both datasets:
X_port = portug_df_fact.drop('goout', axis=1)
y_port = portug_df_fact['goout']

print(X_port)
print(y_port)
```

```
     school  sex  age  address  famsize  Pstatus  Medu  Fedu  Mjob  Fjob  …  \
0         0    0   18        0        0        0     4     4     0     0  …
1         0    0   17        0        0        1     1     1     0     1  …
2         0    0   15        0        1        1     1     1     0     1  …
3         0    0   15        0        0        1     4     2     1     2  …
4         0    0   16        0        0        1     3     3     2     1  …
..      …    …   …       …        …        …     …     …     …     …  …
644       1    0   19        1        0        1     2     3     3     1  …
645       1    0   18        0        1        1     3     1     4     2  …
646       1    0   18        0        0        1     1     1     2     1  …
```

```
647       1    1   17        0        1        1     3     1     3     2  ...
648       1    1   18        1        1        1     3     2     3     1  ...

     romantic  famrel  freetime  Dalc  Walc  health  absences  G1  G2  G3
0           0       4         3     1     1       3         4   0  11  11
1           0       5         3     1     1       3         2   9  11  11
2           0       4         3     2     3       3         6  12  13  12
3           1       3         2     1     1       5         0  14  14  14
4           0       4         3     1     2       5         0  11  13  13
..        ...     ...       ...   ...   ...     ...       ...  ..  ..  ..
644         0       5         4     1     2       5         4  10  11  10
645         0       4         3     1     1       1         4  15  15  16
646         0       1         1     1     1       5         6  11  12   9
647         0       2         4     3     4       2         6  10  10  10
648         0       4         4     3     4       5         4  10  11  11

[649 rows x 32 columns]
0      4
1      3
2      2
3      2
4      2
      ..
644    2
645    4
646    1
647    5
648    1
Name: goout, Length: 649, dtype: int64
```

[ ]:

```python
X_math = maths_df_fact.drop('goout', axis=1)
y_math = maths_df_fact['goout']

print(X_math)
print(y_math)
```

```
     school  sex  age  address  famsize  Pstatus  Medu  Fedu  Mjob  Fjob  ...  \
0         0    0   18        0        0        0     4     4     0     0  ...
1         0    0   17        0        0        1     1     1     0     1  ...
2         0    0   15        0        1        1     1     1     0     1  ...
3         0    0   15        0        0        1     4     2     1     2  ...
4         0    0   16        0        0        1     3     3     2     1  ...
..      ...  ...  ...      ...      ...      ...   ...   ...   ...   ...  ...
390       1    1   20        0        1        0     2     2     3     2  ...
391       1    1   17        0        1        1     3     1     3     2  ...
392       1    1   21        1        0        1     1     1     2     1  ...
393       1    1   18        1        1        1     3     2     3     1  ...
394       1    1   19        0        1        1     1     1     2     4  ...
```

```
     romantic  famrel  freetime  Dalc  Walc  health  absences  G1  G2  G3
0           0       4         3     1     1       3         6   5   6   6
1           0       5         3     1     1       3         4   5   5   6
2           0       4         3     2     3       3        10   7   8  10
3           1       3         2     1     1       5         2  15  14  15
4           0       4         3     1     2       5         4   6  10  10
..         …       …         …   …   …       …         …  ..  ..  ..
390         0       5         5     4     5       4        11   9   9   9
391         0       2         4     3     4       2         3  14  16  16
392         0       5         5     3     3       3         3  10   8   7
393         0       4         4     3     4       5         0  11  12  10
394         0       3         2     3     3       5         5   8   9   9

[395 rows x 32 columns]
0      4
1      3
2      2
3      2
4      2
      ..
390    4
391    5
392    3
393    1
394    3
Name: goout, Length: 395, dtype: int64
```

[ ]:

```
#splitting into test and train sets for evaluation purposes:
np.random.seed(0)
X_port_train, X_port_test, y_port_train, y_port_test = train_test_split(X_port,␣
 ↪y_port, test_size=0.25)
X_math_train, X_math_test, y_math_train, y_math_test = train_test_split(X_math,␣
 ↪y_math, test_size=0.25)
```

[ ]:

```
#see dimenisons of X_train:
print(X_port_train.shape)
print(X_math_train.shape)
```

```
(486, 32)
(296, 32)
```

[ ]:

```
#selecting only the categorical features for the chi test:
X_port_train_categorical = X_port_train.drop(['age','absences','G1','G2','G3'],␣
 ↪axis=1)
X_math_train_categorical = X_math_train.drop(['age','absences','G1','G2','G3'],␣
 ↪axis=1)
```

```
X_port_test_categorical = X_port_test.drop(['age','absences','G1','G2','G3'],␣
 ↪axis=1)
X_math_test_categorical = X_math_test.drop(['age','absences','G1','G2','G3'],␣
 ↪axis=1)
```

[ ]:

```
chi_test = SelectKBest(score_func=chi2, k=8)
port_fit = chi_test.fit(X_port_train_categorical, y_port_train)
port_scores = port_fit.scores_
port_features = port_fit.transform(X_port_train_categorical)
port_selected_indices = port_fit.get_support(indices=True)

print('Portuguese Feature Scores: ', port_scores)
print('Portuguese Selected Features Indices: ', port_selected_indices)
```

```
Portuguese Feature Scores:  [ 6.62475875  1.51390969  1.88626634  0.5988218
0.60149951  0.20948724
  0.92726784  2.08309025  1.01950487  8.03165596  6.80950657  4.63283814
  4.81015861  4.84235238  0.18840611  2.51010519  5.57271722  2.84680069
  3.93329055 15.78951902  2.15528678  2.25034615  0.54160453 23.28574422
 19.1971468  55.54826915  1.45367199]
Portuguese Selected Features Indices:  [ 0  9 10 16 19 23 24 25]
```

[ ]:

```
#see which columns:
port_selected = X_port_train_categorical.iloc[:, port_selected_indices]
port_selected.columns
```

[ ]:

```
Index(['school', 'reason', 'guardian', 'paid', 'higher', 'freetime', 'Dalc',
       'Walc'],
      dtype='object')
```

[ ]:

```
chi_test = SelectKBest(score_func=chi2, k=8)
math_fit = chi_test.fit(X_math_train_categorical, y_math_train)
math_scores = math_fit.scores_
math_features = math_fit.transform(X_math_train_categorical)
math_selected_indices = math_fit.get_support(indices=True)

print('Maths Feature Scores: ', math_scores)
print('Maths Selected Features Indices: ', math_selected_indices)
```

```
Maths Feature Scores:  [ 1.05566612  1.60691947  3.13388931  2.7412583
0.87881599  1.8681377
  0.26617947  0.76152336  0.78763151  5.16870844  3.20792325  3.23090354
  4.80444833 18.60902278  0.6297252   0.66106833  1.55274792  1.07109118
  7.5897972  11.93130224  0.4469659   0.50713607  0.82403374  7.67675278
 16.86285677 43.05935793  2.33268942]
Maths Selected Features Indices:  [ 9 12 13 18 19 23 24 25]
```

[ ]:

```
#see which columns:
maths_selected = X_math_train_categorical.iloc[:, math_selected_indices]
maths_selected.columns
```

[ ]:

```
Index(['reason', 'studytime', 'failures', 'nursery', 'higher', 'freetime',
       'Dalc', 'Walc'],
      dtype='object')
```

0.0.4 Building decision tree models to see the effect of the above feature selection

After selection

[ ]:

```
#portuguese columns:
X_port_train_selected = X_port_train[['age','absences','G1','G2','G3','school',␣
 ↪'reason', 'guardian', 'paid', 'higher', 'freetime', 'Dalc', 'Walc']]
print(X_port_train_selected.columns)
print(X_port_train_selected.shape)
X_port_test_selected = X_port_test[['age','absences','G1','G2','G3','school',␣
 ↪'reason', 'guardian', 'paid', 'higher', 'freetime', 'Dalc', 'Walc']]
```

```
Index(['age', 'absences', 'G1', 'G2', 'G3', 'school', 'reason', 'guardian',
       'paid', 'higher', 'freetime', 'Dalc', 'Walc'],
      dtype='object')
(486, 13)
```

[ ]:

```
#maths columns:
X_math_train_selected = X_math_train[['age','absences','G1','G2','G3','reason',␣
 ↪'studytime', 'failures', 'nursery', 'higher', 'freetime','Dalc', 'Walc']]
print(X_math_train_selected.columns)
print(X_math_train_selected.shape)
X_math_test_selected = X_math_test[['age','absences','G1','G2','G3','reason',␣
 ↪'studytime', 'failures', 'nursery', 'higher', 'freetime','Dalc', 'Walc']]
```

```
Index(['age', 'absences', 'G1', 'G2', 'G3', 'reason', 'studytime', 'failures',
       'nursery', 'higher', 'freetime', 'Dalc', 'Walc'],
      dtype='object')
(296, 13)
```

[ ]:

```
#portuguese dataset model:
port_decision_tree = DecisionTreeClassifier()
port_decision_tree.fit(X_port_train_selected, y_port_train)
port_decision_tree.score(X_port_test_selected, y_port_test)*100
```

[ ]:

```
28.834355828220858
```

[ ]:

```
#maths dataset model:
math_decision_tree = DecisionTreeClassifier()
math_decision_tree.fit(X_math_train_selected, y_math_train)
math_decision_tree.score(X_math_test_selected, y_math_test)*100
```

```
[ ]: 25.252525252525253
```

Before selection

```
[ ]: #portuguese dataset model:
     port_decision_tree_2 = DecisionTreeClassifier()
     port_decision_tree_2.fit(X_port_train, y_port_train)
     port_decision_tree_2.score(X_port_test, y_port_test)*100
```

```
[ ]: 34.96932515337423
```

```
[ ]: #maths dataset model:
     math_decision_tree_2 = DecisionTreeClassifier()
     math_decision_tree_2.fit(X_math_train, y_math_train)
     math_decision_tree_2.score(X_math_test, y_math_test)*100
```

```
[ ]: 26.262626262626267
```