cse572-homework1-key

November 4, 2022

1 CSE 572: Homework 1

This notebook provides a template and starting code to implement Part 2 of the Homework 1 assignment.

To execute and make changes to this notebook, click File > Save a copy to save your own version in your Google Drive or Github. Read the step-by-step instructions below carefully. To execute the code, click on each cell below and press the SHIFT-ENTER keys simultaneously or by clicking the Play button.

When you finish executing all code/exercises, save your notebook then download a copy (.ipynb file). Submit the following **three** things: 1. a link to your Colab notebook, 2. the .ipynb file, and 3. a pdf of the executed notebook on Canvas.

To generate a pdf of the notebook, click File > Print > Save as PDF.

1.1 Prepare the dataset

You will use the Census Income dataset for Part 2 of the Homework 1 assignment. The task for this dataset is to predict whether an individual's income exceeds $50,000 per year or not based on a set of attributes/features (including age, occupation, education, and other factors). The dataset was constructed from the 1994 US Census database.

[1]:
```
import pandas as pd
import numpy as np

data = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/
 ↪adult/adult.data', header=None)
data.columns = ['age', 'workclass', 'fnlwgt', 'education', 'education-num',
                'marital-status', 'occupation', 'relationship', 'race', 'sex',
                'capital-gain', 'capital-loss', 'hours-per-week',␣
 ↪'native-country',
                'class']

data = data.drop(['fnlwgt', 'education-num'], axis=1)

data = data.replace(' ?', np.nan)

data.sample(10)
```

[1]:
```
       age          workclass      education       marital-status  \
28304   49            Private        HS-grad   Married-civ-spouse
6507    51        Federal-gov     Assoc-acdm   Married-civ-spouse
4993    23            Private   Some-college        Never-married
134     46            Private     Assoc-acdm             Divorced
16903   40        Federal-gov      Bachelors   Married-civ-spouse
23654   34       Self-emp-inc      Bachelors             Divorced
31881   52        Federal-gov   Some-college   Married-civ-spouse
32234   41            Private        HS-grad   Married-civ-spouse
1314    32            Private   Some-college   Married-civ-spouse
22682   43   Self-emp-not-inc        Masters             Divorced

                occupation    relationship                 race      sex  \
28304   Machine-op-inspct         Husband                White     Male
6507         Tech-support         Husband                Black     Male
4993       Exec-managerial   Not-in-family                White     Male
134           Tech-support   Not-in-family                Black   Female
16903         Adm-clerical         Husband                White     Male
23654      Farming-fishing   Not-in-family                White     Male
31881      Exec-managerial            Wife   Asian-Pac-Islander   Female
32234                Sales         Husband                White     Male
1314    Handlers-cleaners         Husband                White     Male
22682       Prof-specialty   Not-in-family                White   Female

       capital-gain  capital-loss  hours-per-week  native-country   class
28304             0             0              40   United-States   <=50K
6507              0             0              40   United-States   <=50K
4993              0             0              40   United-States   <=50K
134               0             0              36   United-States   <=50K
16903          7298             0              48   United-States    >50K
23654             0             0              60   United-States   <=50K
31881             0             0              40   United-States    >50K
32234             0             0              40   United-States    >50K
1314              0             0              50          Canada    >50K
22682             0             0              20   United-States   <=50K
```

[2]:
```
data.shape
```

[2]:
```
(32561, 13)
```

[3]:
```
SEED = 0
```

[4]:
```
continuous_feaures = ['age', 'capital-gain', 'capital-loss', 'hours-per-week']
categorical_features = ['workclass', 'education', 'marital-status',␣
 ↪'occupation', 'relationship', 'race', 'sex', 'native-country']
```

2 Data Preparation

[5]:
```
data.columns = data.columns.str.strip()
```

[6]:
```
def inds_nans(df):
  inds = df.isna().any(axis=1)
  # print('Found {} rows that had NaN values.'.format(inds.sum()))
  return inds

def inds_dups(df):
    inds = df.duplicated()
    # print('Found {} rows that were duplicates.'.format(inds.sum()))
    return inds
```

[7]:
```
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler

def normalize(train, test):
  columns = train.columns
  normalizer = preprocessing.Normalizer()
  train[continuous_feaures] = normalizer.
 ↪fit_transform(train[continuous_feaures])
  test[continuous_feaures] = normalizer.transform(test[continuous_feaures])
  # df[continuous_feaures] = preprocessing.normalize(df[continuous_feaures],␣
 ↪norm='l2')
  return pd.DataFrame(train, columns=columns), pd.DataFrame(test,␣
 ↪columns=columns)

def featurewise_standardize(train, test):
    columns = train.columns
    scaler = StandardScaler()
    train[continuous_feaures] = scaler.fit_transform(train[continuous_feaures])
    test[continuous_feaures] = scaler.transform(test[continuous_feaures])
    return pd.DataFrame(train, columns=columns), pd.DataFrame(test,␣
 ↪columns=columns)
```

[8]:
```
from tensorflow import keras
from sklearn.preprocessing import OneHotEncoder

enc=OneHotEncoder()

def one_hot_encode(df, features):
  dummies = pd.get_dummies(df[features].astype(str))
  df_new = pd.concat([df[continuous_feaures],dummies],axis=1)
  print(df_new)
  return df_new
```

[9]:
```
data = data.loc[~(inds_nans(data) | inds_dups(data)), :]

target = data['class'].astype('category').cat.codes
data = data.drop('class', axis=1)
```

[10]:
```
for col in categorical_features:
  data[col] = data[col].astype('category').cat.codes

data
```

[10]:

	age	workclass	education	marital-status	occupation	relationship	\
0	39	5	9	4	0	1	
1	50	4	9	2	3	0	
2	38	2	11	0	5	1	
3	53	2	1	2	5	0	
4	28	2	9	2	9	5	
…	…	…	…	…	…	…	
32554	53	2	12	2	3	0	
32555	22	2	15	4	10	1	
32556	27	2	7	2	12	5	
32558	58	2	11	6	0	4	
32560	52	3	11	2	3	5	

	race	sex	capital-gain	capital-loss	hours-per-week	native-country
0	4	1	2174	0	40	38
1	4	1	0	0	13	38
2	4	1	0	0	40	38
3	2	1	0	0	40	38
4	2	0	0	0	40	4
…	…	…	…	…	…	…
32554	4	1	0	0	40	38
32555	4	1	0	0	40	38
32556	4	0	0	0	38	38
32558	4	0	0	0	40	38
32560	4	0	15024	0	40	38

[26904 rows x 12 columns]

[11]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(data, target, test_size = 0.
  ↪4, random_state = SEED)
```

[12]:
```
X_train_new = one_hot_encode(X_train, categorical_features)
X_test_new = one_hot_encode(X_test, categorical_features)
```

```
       age  capital-gain  capital-loss  hours-per-week  workclass_0  \
2401    20             0             0              40            0
2984    28             0             0              40            0
11443   32             0             0              40            0
2       38             0             0              40            0
26422   54             0             0              40            0
…    …             …             …             …           …
15117   49         14344             0              45            0
23216   61             0             0              40            0
11218   29             0             0              40            0
12350   32             0             0              35            0
3011    37             0             0              45            0

       workclass_1  workclass_2  workclass_3  workclass_4  workclass_5  …  \
2401             0            1            0            0            0  …
2984             0            1            0            0            0  …
11443            0            1            0            0            0  …
2                0            1            0            0            0  …
26422            0            1            0            0            0  …
…             …           …          …           …          …  …
15117            0            1            0            0            0  …
23216            0            1            0            0            0  …
11218            0            0            0            0            1  …
12350            0            1            0            0            0  …
3011             0            1            0            0            0  …

       native-country_37  native-country_38  native-country_39  \
2401                   0                  1                  0
2984                   0                  1                  0
11443                  0                  1                  0
2                      0                  1                  0
26422                  0                  1                  0
…                   …                  …                  …
15117                  0                  1                  0
23216                  0                  0                  0
11218                  0                  1                  0
12350                  0                  1                  0
3011                   0                  1                  0

       native-country_4  native-country_40  native-country_5  \
2401                  0                  0                 0
2984                  0                  0                 0
11443                 0                  0                 0
2                     0                  0                 0
26422                 0                  0                 0
…                  …                  …                 …
15117                 0                  0                 0
23216                 0                  0                 0
```

```
11218                0                 0                 0
12350                0                 0                 0
3011                 0                 0                 0

       native-country_6  native-country_7  native-country_8  native-country_9
2401                  0                 0                 0                 0
2984                  0                 0                 0                 0
11443                 0                 0                 0                 0
2                     0                 0                 0                 0
26422                 0                 0                 0                 0
…                   …                 …                 …                 …
15117                 0                 0                 0                 0
23216                 0                 0                 0                 0
11218                 0                 0                 0                 0
12350                 0                 0                 0                 0
3011                  0                 0                 0                 0

[16142 rows x 101 columns]
       age  capital-gain  capital-loss  hours-per-week  workclass_0  \
15777   48          7688             0              70            0
19658   24             0             0              65            0
12605   56             0             0              40            0
7944    54             0             0              45            0
21845   34             0          1573              40            0
…     …              …             …               …            …
5742    42             0             0              40            0
1273    25             0             0              40            0
30289   24             0             0              40            0
20548   23             0             0              25            1
25701   31             0             0              40            0

       workclass_1  workclass_2  workclass_3  workclass_4  workclass_5  …  \
15777            0            1            0            0            0  …
19658            0            1            0            0            0  …
12605            0            1            0            0            0  …
7944             0            1            0            0            0  …
21845            0            1            0            0            0  …
…              …            …            …            …            …  …
5742             0            1            0            0            0  …
1273             0            1            0            0            0  …
30289            0            1            0            0            0  …
20548            0            0            0            0            0  …
25701            0            1            0            0            0  …

       native-country_37  native-country_38  native-country_39  \
15777                  0                  1                  0
19658                  0                  1                  0
12605                  0                  1                  0
```

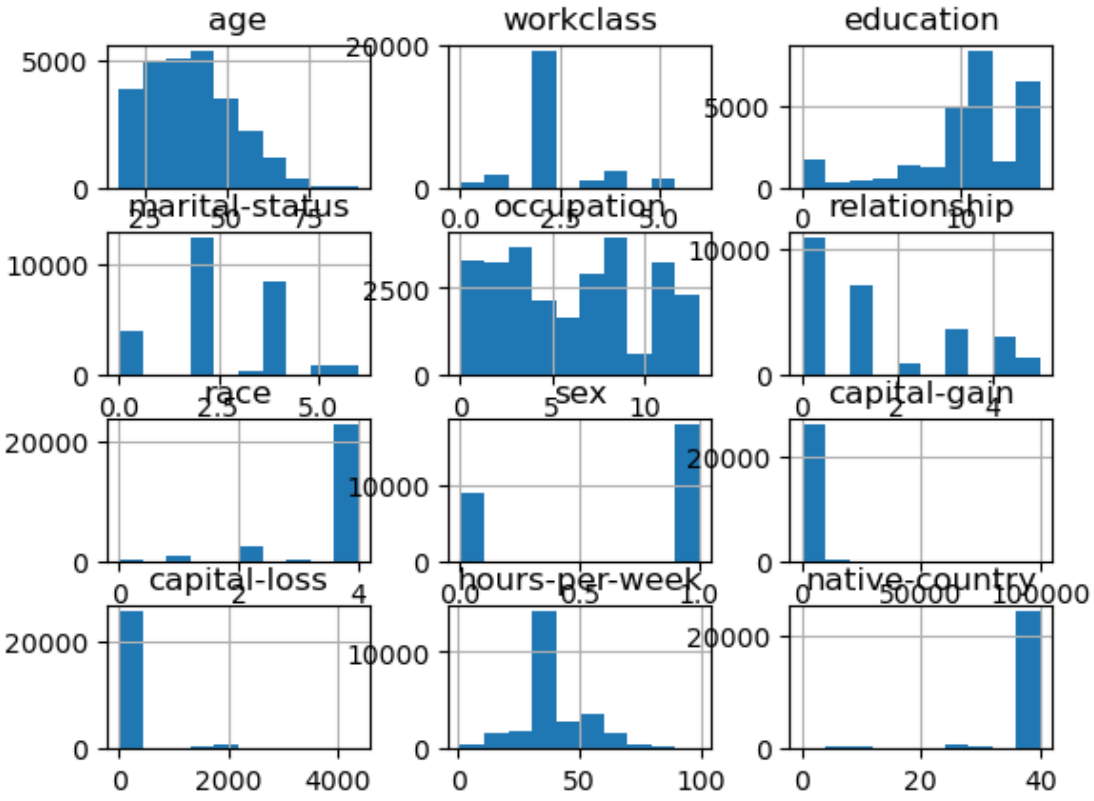
```
7944                    0                    1                    0
21845                   0                    1                    0
…                     …                    …                    …
5742                    0                    1                    0
1273                    0                    1                    0
30289                   0                    0                    0
20548                   0                    1                    0
25701                   0                    1                    0

       native-country_4  native-country_40  native-country_5  \
15777                 0                  0                 0
19658                 0                  0                 0
12605                 0                  0                 0
7944                  0                  0                 0
21845                 0                  0                 0
…                   …                  …                 …
5742                  0                  0                 0
1273                  0                  0                 0
30289                 0                  0                 0
20548                 0                  0                 0
25701                 0                  0                 0

       native-country_6  native-country_7  native-country_8  native-country_9
15777                 0                 0                 0                 0
19658                 0                 0                 0                 0
12605                 0                 0                 0                 0
7944                  0                 0                 0                 0
21845                 0                 0                 0                 0
…                   …                 …                 …                 …
5742                  0                 0                 0                 0
1273                  0                 0                 0                 0
30289                 0                 0                 0                 0
20548                 0                 0                 0                 0
25701                 0                 0                 0                 0

[10762 rows x 102 columns]
```

3 Data Visualization

[13]:
```
data.hist()
```

[13]:
```
array([[<AxesSubplot: title={'center': 'age'}>,
        <AxesSubplot: title={'center': 'workclass'}>,
        <AxesSubplot: title={'center': 'education'}>],
       [<AxesSubplot: title={'center': 'marital-status'}>,
        <AxesSubplot: title={'center': 'occupation'}>,
        <AxesSubplot: title={'center': 'relationship'}>],
```

```
[<AxesSubplot: title={'center': 'race'}>,
 <AxesSubplot: title={'center': 'sex'}>,
 <AxesSubplot: title={'center': 'capital-gain'}>],
[<AxesSubplot: title={'center': 'capital-loss'}>,
 <AxesSubplot: title={'center': 'hours-per-week'}>,
 <AxesSubplot: title={'center': 'native-country'}>]], dtype=object)
```



[15]:
```
data.corr()
```

[15]:

	age	workclass	education	marital-status	occupation	\
age	1.000000	0.075188	0.002693	-0.245075	-0.004569	
workclass	0.075188	1.000000	0.021279	-0.031968	0.014047	
education	0.002693	0.021279	1.000000	-0.040886	-0.025890	
marital-status	-0.245075	-0.031968	-0.040886	1.000000	0.015093	
occupation	-0.004569	0.014047	-0.025890	0.015093	1.000000	
relationship	-0.231116	-0.070998	-0.010785	0.155562	-0.063950	
race	0.040449	0.051476	0.005590	-0.073612	0.008183	
sex	0.070460	0.079388	-0.029904	-0.108972	0.066917	
capital-gain	0.077238	0.033218	0.034542	-0.042092	0.019724	
capital-loss	0.052562	0.002698	0.020685	-0.033702	0.010865	
hours-per-week	0.072530	0.047169	0.058703	-0.175692	0.023073	

```
native-country  0.010184   0.012728   0.075671       -0.029410    0.002798

                relationship      race       sex  capital-gain  capital-loss  \
age                -0.231116  0.040449  0.070460      0.077238      0.052562
workclass          -0.070998  0.051476  0.079388      0.033218      0.002698
education          -0.010785  0.005590 -0.029904      0.034542      0.020685
marital-status      0.155562 -0.073612 -0.108972     -0.042092     -0.033702
occupation         -0.063950  0.008183  0.066917      0.019724      0.010865
relationship        1.000000 -0.113702 -0.586799     -0.063324     -0.070257
race               -0.113702  1.000000  0.085761      0.020046      0.032171
sex                -0.586799  0.085761  1.000000      0.053496      0.052797
capital-gain       -0.063324  0.020046  0.053496      1.000000     -0.036035
capital-loss       -0.070257  0.032171  0.052797     -0.036035      1.000000
hours-per-week     -0.248745  0.056459  0.228764      0.079403      0.049053
native-country     -0.006305  0.116916 -0.004330      0.012896      0.015937

                hours-per-week  native-country
age                   0.072530        0.010184
workclass             0.047169        0.012728
education             0.058703        0.075671
marital-status       -0.175692       -0.029410
occupation            0.023073        0.002798
relationship         -0.248745       -0.006305
race                  0.056459        0.116916
sex                   0.228764       -0.004330
capital-gain          0.079403        0.012896
capital-loss          0.049053        0.015937
hours-per-week        1.000000        0.014046
native-country        0.014046        1.000000
```

4 K Nearest Neighbours

4.1 Data Preparation

[16]:
```
X_train_knn, X_test_knn = featurewise_standardize(X_train_new, X_test_new)
```

4.2 Model training

The details of training your model will vary depending on which model you choose to implement. For all models, you will find the optimal hyperparameters of your model using 5-fold cross validation and Grid Search.

[17]:
```
# K-fold cross validation and Grid Search

from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

```python
knn = KNeighborsClassifier()

k_range = list(range(1, 31))
param_grid = {'n_neighbors': list(range(8, 10)),
              'weights': ['uniform', 'distance'],
              'metric': ['l1', 'l2']
             }

# define parameter range
grid = GridSearchCV(knn, param_grid=param_grid, cv=5, scoring='accuracy',␣
 ↪return_train_score=False,verbose=1)

# Fit the model for grid search
grid_search_knn=grid.fit(X_train_knn, Y_train)
```

```
Fitting 5 folds for each of 8 candidates, totalling 40 fits
```

```
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
```

```
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
```

```
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
```

```
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
/Users/hkerner/anaconda3/envs/cse572/lib/python3.9/site-
packages/sklearn/neighbors/_classification.py:228: FutureWarning: Unlike other
reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`
typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will
change: the default value of `keepdims` will become False, the `axis` over which
the statistic is taken will be eliminated, and the value None will no longer be
accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

[18]:
```
grid_search_knn.best_params_
```

[18]:
```
{'metric': 'l2', 'n_neighbors': 9, 'weights': 'uniform'}
```

4.3 Evaluation

Your final model evaluation should be performed on the test set. Report the following metrics: - Overall accuracy - Precision - Recall - F1 score

[19]:
```
knn_best = grid_search_knn.best_estimator_

Y_predTest=knn_best.predict(X_test_knn)
```

```
test_accuracy=accuracy_score(Y_test,Y_predTest)*100
```

[20]:
```
# YOUR CODE HERE
print("Overall Accuracy : {:.2f}%".format(test_accuracy) )

from sklearn.metrics import precision_recall_fscore_support as score

precision, recall, fscore, support = score(Y_test, Y_predTest, average='micro')
print('precision: {:.2f}'.format(precision))
print('recall: {:.2f}'.format(recall))
print('fscore: {:.2f}'.format(fscore))
```

```
Overall Accuracy : 82.60%
precision: 0.83
recall: 0.83
fscore: 0.83
```

5 Naive Bayes Classifier

5.1 Data Preparation

[21]:
```
X_train_nb, X_test_nb = featurewise_standardize(X_train_new, X_test_new)
```

5.2 Model training

The details of training your model will vary depending on which model you choose to implement. For all models, you will find the optimal hyperparameters of your model using 5-fold cross validation and Grid Search.

[22]:
```
# K-fold cross validation and Grid Search

from sklearn.model_selection import GridSearchCV
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

gnb = GaussianNB()

param_grid = {
            'var_smoothing': [1e-9, 1e-5, 1e-2, 0.5, 0.7, 1, 2],
        }


# defining parameter range
gaussian_nb_grid = GridSearchCV(gnb, param_grid=param_grid, scoring='accuracy',␣
 ↪cv=5, return_train_score=False, verbose=1)

# fitting the model for grid search
```

```
grid_search_gnb=gaussian_nb_grid.fit(X_train_nb, Y_train)
```

```
Fitting 5 folds for each of 7 candidates, totalling 35 fits
```

[23]:
```
print(grid_search_gnb.best_params_)
```

```
{'var_smoothing': 0.5}
```

5.3 Evaluation

Your final model evaluation should be performed on the test set. Report the following metrics: - Overall accuracy - Precision - Recall - F1 score

[24]:
```
gnb_best = grid_search_gnb.best_estimator_

Y_predTest=gnb_best.predict(X_test_nb)

test_accuracy=accuracy_score(Y_test,Y_predTest)*100
```

[25]:
```
# YOUR CODE HERE
print("Overall Accuracy : {:.2f}%".format(test_accuracy) )

from sklearn.metrics import precision_recall_fscore_support as score

precision, recall, fscore, support = score(Y_test, Y_predTest, average='micro')
print('precision: {:.2f}'.format(precision))
print('recall: {:.2f}'.format(recall))
print('fscore: {:.2f}'.format(fscore))
```

```
Overall Accuracy : 80.64%
precision: 0.81
recall: 0.81
fscore: 0.81
```

6 Support Vector Machines

6.1 Data Preparation

[26]:
```
X_train_nb, X_test_nb = normalize(X_train_new, X_test_new)
```

6.2 Model training

The details of training your model will vary depending on which model you choose to implement. For all models, you will find the optimal hyperparameters of your model using 5-fold cross validation and Grid Search.

[27]:
```
# K-fold cross validation and Grid Search
```

```
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

svm  = SVC()
param_grid = {'C': [1, 10, 100],
              # 'gamma': [1, 0.1, 0.01],
              'kernel': ['rbf', 'poly']}

# grid = GridSearchCV(svm, param_grid, refit = True, verbose = 3)


# defining parameter range
svm_grid = GridSearchCV(svm, param_grid=param_grid, scoring='accuracy', cv=5,␣
 ↪return_train_score=False, verbose=1)

# fitting the model for grid search
grid_search_svm=svm_grid.fit(X_train_nb, Y_train)
```

```
Fitting 5 folds for each of 6 candidates, totalling 30 fits
```

[28]:
```
print(grid_search_svm.best_params_)
```

```
{'C': 1, 'kernel': 'poly'}
```

6.3 Evaluation

Your final model evaluation should be performed on the test set. Report the following metrics: - Overall accuracy - Precision - Recall - F1 score

[29]:
```
svm_best = grid_search_svm.best_estimator_

Y_predTest=svm_best.predict(X_test_nb)

test_accuracy=accuracy_score(Y_test,Y_predTest)*100
```

[30]:
```
# YOUR CODE HERE
print("Overall Accuracy : {:.2f}%".format(test_accuracy) )

from sklearn.metrics import precision_recall_fscore_support as score

precision, recall, fscore, support = score(Y_test, Y_predTest, average='micro')
print('precision: {:.2f}'.format(precision))
print('recall: {:.2f}'.format(recall))
print('fscore: {:.2f}'.format(fscore))
```

```
Overall Accuracy : 84.16%
precision: 0.84
```

```
recall: 0.84
fscore: 0.84
```

7 Decision Tree

7.1 Model Training

[45]:
```
# K-fold cross validation and Grid Search

from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

dtc = DecisionTreeClassifier()

param_grid = {
            "criterion": ["gini", "entropy"],
            "max_depth" : range(4,10)
        }


# defining parameter range
dtc_grid = GridSearchCV(dtc, param_grid=param_grid, scoring='accuracy', cv=5,␣
 ↪return_train_score=False, verbose=1)

# fitting the model for grid search
grid_search_dtc=dtc_grid.fit(X_train, Y_train)
```

```
Fitting 5 folds for each of 12 candidates, totalling 60 fits
```

[46]:
```
print(grid_search_dtc.best_params_)
```

```
{'criterion': 'gini', 'max_depth': 9}
```

7.2 Evaluation

Your final model evaluation should be performed on the test set. Report the following metrics: - Overall accuracy - Precision - Recall - F1 score

[47]:
```
dtc_best = grid_search_dtc.best_estimator_

Y_predTest=dtc_best.predict(X_test)

test_accuracy=accuracy_score(Y_test,Y_predTest)*100
```

[48]:
```
# YOUR CODE HERE
print("Overall Accuracy : {:.2f}%".format(test_accuracy) )
```

```
from sklearn.metrics import precision_recall_fscore_support as score

precision, recall, fscore, support = score(Y_test, Y_predTest, average='micro')
print('precision: {:.2f}'.format(precision))
print('recall: {:.2f}'.format(recall))
print('fscore: {:.2f}'.format(fscore))
```

```
Overall Accuracy : 84.49%
precision: 0.84
recall: 0.84
fscore: 0.84
```