```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 from sklearn.model_selection import train_test_split
6
```

```
1 from sklearn.metrics import mean_squared_error, r2_score
2 #Import Data from Input
3 df= pd.read_csv(r"C:\Users\shiny\Downloads\crx_data.csv.csv")
```

```
1 df.head()
```

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
0	b	30.83	0.000	u	g	w	v	1.25	t	t	1	f	g	202	0	+
1	a	58.67	4.460	u	g	q	h	3.04	t	t	6	f	g	43	560	+
2	a	24.5	0.500	u	g	q	h	1.50	t	f	0	f	g	280	824	+
3	b	27.83	1.540	u	g	w	v	3.75	t	t	5	t	g	100	3	+
4	b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	120	0	+

df.columns

```
1 df.shape
2
```

```
(690, 16)
```

```
1 df.columns
```

```
Index(['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10', 'A11',
       'A12', 'A13', 'A14', 'A15', 'A16'],
      dtype='object')
```

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 690 entries, 0 to 689
Data columns (total 16 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   A1      690 non-null    object
 1   A2      690 non-null    object
 2   A3      690 non-null    float64
 3   A4      690 non-null    object
 4   A5      690 non-null    object
 5   A6      690 non-null    object
 6   A7      690 non-null    object
 7   A8      690 non-null    float64
 8   A9      690 non-null    object
 9   A10     690 non-null    object
 10  A11     690 non-null    int64
 11  A12     690 non-null    object
 12  A13     690 non-null    object
 13  A14     690 non-null    object
 14  A15     690 non-null    int64
 15  A16     690 non-null    object
dtypes: float64(2), int64(2), object(12)
memory usage: 86.4+ KB
```

```
1 df.describe()
2
```

```
--NORMAL--
```

	A3	A8	A11	A15
count	690.000000	690.000000	690.00000	690.000000
mean	4.758725	2.223406	2.40000	1017.385507
std	4.978163	3.346513	4.86294	5210.102598
min	0.000000	0.000000	0.00000	0.000000
25%	1.000000	0.165000	0.00000	0.000000
50%	2.750000	1.000000	0.00000	5.000000
75%	7.207500	2.625000	3.00000	395.500000

```
1 df.head()
```

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
0	b	30.83	0.000	u	g	w	v	1.25	t	t	1	f	g	202	0	+
1	a	58.67	4.460	u	g	q	h	3.04	t	t	6	f	g	43	560	+
2	a	24.5	0.500	u	g	q	h	1.50	t	f	0	f	g	280	824	+
3	b	27.83	1.540	u	g	w	v	3.75	t	t	5	t	g	100	3	+
4	b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	120	0	+

```
1 df.dtypes
2
```

```
A1      object
A2      object
A3     float64
A4      object
A5      object
A6      object
A7      object
A8     float64
A9      object
A10     object
A11      int64
A12     object
A13     object
A14     object
A15      int64
A16     object
dtype: object
```

```
1 df.A2.unique()
```

```
array(['30.83', '58.67', '24.5', '27.83', '20.17', '32.08', '33.17',
       '22.92', '54.42', '42.5', '22.08', '29.92', '38.25', '48.08',
       '45.83', '36.67', '28.25', '23.25', '21.83', '19.17', '25',
       '47.75', '27.42', '41.17', '15.83', '47', '56.58', '57.42',
       '42.08', '29.25', '42', '49.5', '36.75', '22.58', '27.25', '23',
       '27.75', '54.58', '34.17', '28.92', '29.67', '39.58', '56.42',
       '54.33', '41', '31.92', '41.5', '23.92', '25.75', '26', '37.42',
       '34.92', '34.25', '23.33', '23.17', '44.33', '35.17', '43.25',
       '56.75', '31.67', '23.42', '20.42', '26.67', '36', '25.5', '19.42',
       '32.33', '34.83', '38.58', '44.25', '44.83', '20.67', '34.08',
       '21.67', '21.5', '49.58', '27.67', '39.83', '?', '37.17', '25.67',
       '34', '49', '62.5', '31.42', '52.33', '28.75', '28.58', '22.5',
       '28.5', '37.5', '35.25', '18.67', '54.83', '40.92', '19.75',
       '29.17', '24.58', '33.75', '25.42', '37.75', '52.5', '57.83',
       '20.75', '39.92', '24.75', '44.17', '23.5', '47.67', '22.75',
       '34.42', '28.42', '67.75', '47.42', '36.25', '32.67', '48.58',
       '33.58', '18.83', '26.92', '31.25', '56.5', '43', '22.33', '32.83',
       '40.33', '30.5', '52.83', '46.67', '58.33', '37.33', '23.08',
       '32.75', '68.67', '28', '44', '25.08', '32', '60.58', '40.83',
       '19.33', '41.33', '56', '49.83', '22.67', '27', '26.08', '18.42',
       '21.25', '57.08', '22.42', '48.75', '40', '40.58', '28.67',
       '33.08', '21.33', '41.75', '34.5', '48.17', '27.58', '24.08',
       '24.83', '36.33', '35.42', '71.58', '39.5', '39.33', '24.33',
       '60.08', '55.92', '53.92', '18.92', '50.08', '65.42', '17.58',
       '18.08', '19.67', '25.17', '33.5', '58.42', '26.17', '42.83',
       '38.17', '20.5', '48.25', '28.33', '18.75', '18.5', '45', '40.25',
       '41.42', '17.83', '18.17', '20', '52.17', '50.75', '17.08',
       '18.33', '59.67', '18', '37.58', '30.67', '18.58', '16.25',
       '21.17', '17.67', '16.5', '29.5', '21.75', '18.25', '35.75',
```

```
       '16.08', '69.17', '32.92', '16.33', '22.17', '57.58', '15.92',
       '31.75', '19', '17.5', '33.67', '30.17', '33.25', '25.25', '34.75',
       '47.33', '39.08', '42.75', '38.92', '62.75', '32.25', '26.75',
       '63.33', '30.75', '16', '19.5', '32.42', '30.25', '26.83', '16.92',
       '24.42', '39.42', '23.58', '21.42', '33', '26.33', '26.25',
       '28.17', '20.83', '43.17', '56.83', '15.17', '29.83', '31',
       '51.92', '69.5', '19.58', '22.25', '38.42', '26.58', '35', '29.42',
       '49.17', '51.83', '58.58', '53.33', '27.17', '25.92', '30.58',
       '17.25', '27.33', '36.5', '29.75', '52.42', '36.17', '34.58',
       '21.92', '36.58', '31.08', '30.42', '21.08', '17.42', '39.17',
       '26.5', '17.33', '23.75', '34.67', '74.83', '45.33', '47.25',
       '24.17', '39.25', '39', '64.08', '31.33', '21', '13.75', '46',
       '20.25', '60.92', '30', '22.83', '45.17', '41.58', '55.75',
       '25.33', '31.83', '33.92', '24.92', '80.25', '30.08', '48.33',
       '76.75', '51.33', '41.92', '29.58', '32.17', '51.42', '42.17',
       '43.08', '59.5', '65.17', '20.33', '48.5', '28.08', '73.42',
       '51.58', '38.67', '46.08', '20.08', '42.25', '16.17', '47.83',
       '22', '38.33', '25.58', '21.58', '36.08', '38.75', '35.58',
       '31.58', '15.75', '17.92', '30.33', '47.17', '25.83', '50.25',
       '36.42'], dtype=object)
```

```
1 df = df[df.A2 != "?"]
2 df.A2 = df.A2.astype("float")
```

```
1 df.A14.unique()
```

```
array(['202', '43', '280', '100', '120', '360', '164', '80', '180', '52',
       '128', '260', '0', '320', '396', '96', '200', '300', '145', '500',
       '168', '434', '583', '30', '240', '70', '455', '311', '216', '491',
       '400', '239', '160', '711', '250', '520', '515', '420', '?', '980',
       '443', '140', '94', '368', '288', '188', '112', '171', '268',
       '167', '75', '152', '176', '329', '212', '410', '274', '375',
       '408', '350', '204', '40', '181', '399', '440', '93', '60', '395',
       '393', '21', '29', '102', '431', '370', '24', '20', '129', '510',
       '195', '144', '380', '49', '50', '381', '150', '117', '56', '211',
       '230', '156', '22', '228', '519', '253', '487', '220', '88', '73',
       '121', '470', '136', '132', '292', '154', '272', '340', '108',
       '720', '450', '232', '170', '1160', '460', '348', '480', '640',
       '372', '276', '221', '352', '141', '178', '600', '550', '2000',
       '225', '210', '110', '356', '45', '62', '92', '174', '17', '86',
       '454', '254', '28', '263', '333', '312', '371', '99', '252', '760',
       '560', '130', '523', '680', '163', '208', '383', '330', '422',
       '290', '840', '432', '32', '186', '303', '349', '224', '369', '76',
       '231', '309', '416', '465', '256'], dtype=object)
```

```
1 df = df[df.A14 != "?"]
2 df.A14 = df.A14.astype("float")
```

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 666 entries, 0 to 689
Data columns (total 16 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   A1      666 non-null    object 
 1   A2      666 non-null    float64
 2   A3      666 non-null    float64
 3   A4      666 non-null    object 
 4   A5      666 non-null    object 
 5   A6      666 non-null    object 
 6   A7      666 non-null    object 
 7   A8      666 non-null    float64
 8   A9      666 non-null    object 
 9   A10     666 non-null    object 
 10  A11     666 non-null    int64  
 11  A12     666 non-null    object 
 12  A13     666 non-null    object 
 13  A14     666 non-null    float64
 14  A15     666 non-null    int64  
 15  A16     666 non-null    object 
dtypes: float64(4), int64(2), object(10)
memory usage: 88.5+ KB
```

```
1 df.isnull().sum()
```

```
A1     0
A2     0
```

```
A3     0
A4     0
A5     0
A6     0
A7     0
A8     0
A9     0
A10    0
A11    0
A12    0
A13    0
A14    0
A15    0
A16    0
dtype: int64
```

```
1 X = df.iloc[:, [1,2,7,10,13,14]].values
2 Y = df.iloc[:, [15]].values
```

```
1 from sklearn.decomposition import PCA
2 pca = PCA(n_components=1)
3 pca
```

```
PCA(n_components=1)
```

```
1 principalComponents = pca.fit_transform(X)
```

```
1 principalComponents.shape
```

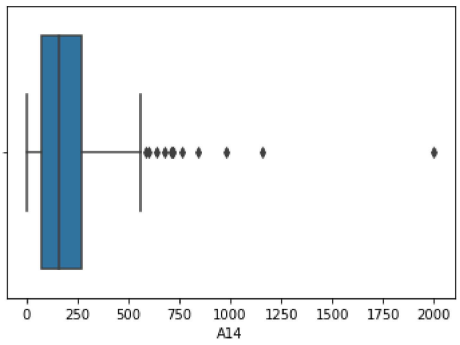
```
(666, 1)
```

```
1 pca.explained_variance_ratio_
```

```
array([0.99891275])
```

```
1 import seaborn as sns
2 sns.boxplot(df["A14"])
```

```
E:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.
  warnings.warn(
<AxesSubplot:xlabel='A14'>
```




```
1 import seaborn as sns
2 sns.heatmap(df.corr())
```

<AxesSubplot:>


```
1 df.corr()
```

	A2	A3	A8	A11	A14	A15
A2	1.000000	0.211249	0.404847	0.190790	-0.079812	0.027512
A3	0.211249	1.000000	0.303990	0.271777	-0.217364	0.120021
A8	0.404847	0.303990	1.000000	0.329274	-0.069020	0.053009
A11	0.190790	0.271777	0.329274	1.000000	-0.118024	0.059604
A14	-0.079812	-0.217364	-0.069020	-0.118024	1.000000	0.069396
A15	0.027512	0.120021	0.053009	0.059604	0.069396	1.000000

```
1 print(principalComponents[:])
```

```
[ 4.43117432e+02]
[-9.98706556e+02]
[-7.98404171e+02]
[-9.98998048e+02]
[ 1.69099546e+03]
[-9.98996519e+02]
[-9.98265120e+02]
[-9.98092421e+02]
[-7.53724262e+02]
[-9.98999758e+02]
[-9.98778764e+02]
[ 2.09457474e+02]
[-9.98312697e+02]
[ 2.61000038e+02]
[-9.87722899e+02]
[-9.98665662e+02]
[-9.98770996e+02]
[-9.98996038e+02]
[-9.98994013e+02]
[-9.98996178e+02]
[ 9.00212058e+03]
[-9.98612528e+02]
[-9.98996685e+02]
[ 4.00099076e+03]
[ 3.00099136e+03]
[-4.38999699e+02]
[-9.63004976e+02]
[-2.84665586e+02]
[-4.47312820e+02]
[-4.98405268e+02]
[-6.98927648e+02]
[-7.77997724e+02]
[ 1.28399716e+03]
[-8.98449830e+02]
[-9.98836180e+02]
[-9.83991206e+02]
[-7.14997019e+02]
[ 2.38041451e+02]
[-6.98287008e+02]
[-9.98503720e+02]
[-9.98770636e+02]
[-9.97875282e+02]
[-9.98358284e+02]
[ 4.80190198e+03]
[-7.98452009e+02]
[-9.98998428e+02]
[-6.98632799e+02]
[-9.98999722e+02]
[-9.98265492e+02]
[-9.97370349e+02]
[-9.98814120e+02]
[-9.98538007e+02]
[-2.68426235e+02]
[-5.98817339e+02]
[-9.98999692e+02]
[-9.98724060e+02]
[ 4.90020602e+04]
[-5.42999312e+02]
```

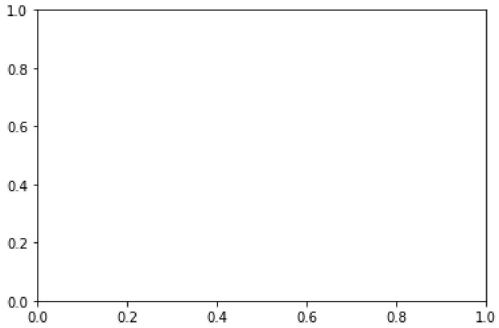
```
1 df.columns
```

```
Index(['A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8', 'A9', 'A10', 'A11',
       'A12', 'A13', 'A14', 'A15', 'A16'],
      dtype='object')
```

```
1 plt.plot(range(2), pca.explained_variance_ratio_,label="Variance-ratio")
2 plt.plot(range(2), np.cumsum(pca.explained_variance_ratio_),label="cumulative variance-ratio")
3 plt.title("Component-wise and Cumulative Explained Variance")
4 plt.xlabel("No.of components")
5 plt.ylabel("Data_explained")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [48], in <cell line: 1>()
----> 1 plt.plot(range(2), pca.explained_variance_ratio_,label="Variance-ratio")
      2 plt.plot(range(2), np.cumsum(pca.explained_variance_ratio_),label="cumulative variance-ratio")
      3 plt.title("Component-wise and Cumulative Explained Variance")

File E:\Anaconda\lib\site-packages\matplotlib\pyplot.py:2757, in plot(scalex, scaley, data, *args, **kwargs)
   2755 @_copy_docstring_and_deprecators(Axes.plot)
   2756 def plot(*args, scalex=True, scaley=True, data=None, **kwargs):
-> 2757     return gca().plot(
   2758         *args, scalex=scalex, scaley=scaley,
   2759         **({"data": data} if data is not None else {}), **kwargs)

File E:\Anaconda\lib\site-packages\matplotlib\axes\_axes.py:1632, in Axes.plot(self, scalex, scaley, data, *args, **kwargs)
   1390 """
   1391 Plot y versus x as lines and/or markers.
   1392
   (...)
   1629 (``'green'``) or hex strings (``'#008000'``).
   1630 """
   1631 kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D)
-> 1632 lines = [*self._get_lines(*args, data=data, **kwargs)]
   1633 for line in lines:
   1634     self.add_line(line)

File E:\Anaconda\lib\site-packages\matplotlib\axes\_base.py:312, in _process_plot_var_args.__call__(self, data, *args, **kwargs)
    310     this += args[0],
    311     args = args[1:]
--> 312 yield from self._plot_args(this, kwargs)

File E:\Anaconda\lib\site-packages\matplotlib\axes\_base.py:498, in _process_plot_var_args._plot_args(self, tup, kwargs, return_kwargs)
    495     self.axes.yaxis.update_units(y)
    497 if x.shape[0] != y.shape[0]:
--> 498     raise ValueError(f"x and y must have same first dimension, but "
    499                      f"have shapes {x.shape} and {y.shape}")
    500 if x.ndim > 2 or y.ndim > 2:
    501     raise ValueError(f"x and y can be no greater than 2D, but have "
    502                      f"shapes {x.shape} and {y.shape}")

ValueError: x and y must have same first dimension, but have shapes (2,) and (1,)
```

SEARCH STACK OVERFLOW




```
1 #Random Forests
2
3 from sklearn.model_selection import train_test_split
4
5 from sklearn.ensemble import RandomForestClassifier
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2,random_state=123)
df= RandomForestClassifier(max_depth=8,n_estimators=5)
df.fit(X_train,y_train)

```

```
C:\Users\shiny\AppData\Local\Temp\ipykernel_8944\1269264156.py:8: DataConversionWarning: A column-vector y was passed when a 1d array wa
  df.fit(X_train,y_train)
RandomForestClassifier(max_depth=8, n_estimators=5)
```

```
y_pred = df.predict(X_test)
```

```
df.score(X_test, Y_test)
```

```
0.7686567164179104
```

```

from sklearn.model_selection import cross_val_score

scores = cross_val_score(df, X_train, Y_train,cv=10)

scores
```

```
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
array([0.75925926, 0.66666667, 0.77358491, 0.69811321, 0.73584906,
       0.75471698, 0.77358491, 0.8490566 , 0.79245283, 0.77358491])
```

```
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), 3* scores.std() ))
```

```
Accuracy: 0.76 (+/- 0.14)
```

```
#knn classification
```

```
from sklearn.neighbors import KNeighborsClassifier
clf_entropy=KNeighborsClassifier(n_neighbors=19, p=2, weights='distance')
```

```
clf_entropy.fit(X_train,y_train)
```

```
E:\Anaconda\lib\site-packages\sklearn\neighbors\_classification.py:198: DataConversionWarning: A column-vector y was passed when a 1d ar
  return self._fit(X, y)
KNeighborsClassifier(n_neighbors=19, weights='distance')
```

```
predicted_entropy = clf_entropy.predict(X_test)
```

```
from  sklearn.metrics import confusion_matrix
cM = confusion_matrix(Y_test,predicted_entropy)
print(cM)
```

```
[[30 34]
 [ 6 64]]
```

```
tn, fn, fp, tp = cM.ravel()
print("Accuracy=",(tp+tn)/(tp+tn+fp+fn))
```

```
Accuracy= 0.7014925373134329
```

```
recall = tp/(tp+fn)
precision=tp/(tp+fp)
print("Recall = Sensitivity = ",tp/(tp+fn))
print("Specificity =", tn/(tn+fp))

print("Precision=",tp/(tp+fp))
f1score= 2 *(recall*precision)/(precision+recall)
print("f1 score=", f1score)
```

```
Recall = Sensitivity =  0.6530612244897959
Specificity = 0.8333333333333334
Precision= 0.9142857142857143
f1 score= 0.7619047619047618
```

```
from sklearn.metrics import roc_curve
from  sklearn.metrics import auc
fpr, tpr, thresholds = roc_curve(Y_test,predicted_entropy)
aucP=auc(fpr, tpr)

print('auc of ROC curve = ', aucP)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [88], in <cell line: 3>()
      1 from sklearn.metrics import roc_curve
      2 from  sklearn.metrics import auc
----> 3 fpr, tpr, thresholds = roc_curve(Y_test,predicted_entropy)
      4 aucP=auc(fpr, tpr)
      6 print('auc of ROC curve = ', aucP)

File E:\Anaconda\lib\site-packages\sklearn\metrics\_ranking.py:962, in roc_curve(y_true, y_score, pos_label, sample_weight,
drop_intermediate)
    873 def roc_curve(
    874     y_true, y_score, *, pos_label=None, sample_weight=None, drop_intermediate=True
    875 ):
```

```

from sklearn.model_selection import cross_val_score

scores = cross_val_score(df, X_train, Y_train,cv=10)

scores
```

```
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
array([0.7962963 , 0.72222222, 0.73584906, 0.67924528, 0.67924528,
       0.66037736, 0.79245283, 0.8490566 , 0.64150943, 0.81132075])
```

```
    244         f"y_true takes value in {{{classes_repr}}} and pos_label is not "
```

```
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), 3* scores.std() ))
```

```
Accuracy: 0.74 (+/- 0.20)
    249     pos_label = 1
```

```
#Logistic Regression
```

```
pos_label explicitly.
```

```
from sklearn.linear_model import LogisticRegression
clf_entropy = LogisticRegression()
```

```
clf_entropy.fit(X_train,Y_train)
```

```
E:\Anaconda\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was e
  y = column_or_1d(y, warn=True)
E:\Anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
LogisticRegression()
```

```
predicted_entropy = clf_entropy.predict(X_test)
```

```
from  sklearn.metrics import confusion_matrix
cM = confusion_matrix(Y_test,predicted_entropy)
```

```
print(cM)
```

```
[[37 27]
 [ 5 65]]
```

```
tn, fn, fp, tp = cM.ravel()
print("Accuracy=",(tp+tn)/(tp+tn+fp+fn))

```

```
Accuracy= 0.7611940298507462
```

```
recall = tp/(tp+fn)
precision=tp/(tp+fp)
print("Recall = Sensitivity = ",tp/(tp+fn))
print("Specificity =", tn/(tn+fp))

print("Precision=",tp/(tp+fp))
f1score= 2 *(recall*precision)/(precision+recall)
print("f1 score=", f1score)
```

```
Recall = Sensitivity =  0.7065217391304348
Specificity = 0.8809523809523809
Precision= 0.9285714285714286
f1 score= 0.8024691358024691
```

```
from sklearn.metrics import roc_curve
from  sklearn.metrics import auc
fpr, tpr, thresholds = roc_curve(Y_test,predicted_entropy)
aucP=auc(fpr, tpr)

print('auc of ROC curve = ', aucP)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [46], in <cell line: 3>()
      1 from sklearn.metrics import roc_curve
      2 from  sklearn.metrics import auc
----> 3 fpr, tpr, thresholds = roc_curve(Y_test,predicted_entropy)
      4 aucP=auc(fpr, tpr)
      6 print('auc of ROC curve = ', aucP)

File E:\Anaconda\lib\site-packages\sklearn\metrics\_ranking.py:962, in roc_curve(y_true, y_score, pos_label, sample_weight, drop_intermediate)
    873 def roc_curve(
    874     y_true, y_score, *, pos_label=None, sample_weight=None, drop_intermediate=True
    875 ):
    876     """Compute Receiver operating characteristic (ROC).
    877 
    878     Note: this implementation is restricted to the binary classification task.
```

```

from sklearn.model_selection import cross_val_score

scores = cross_val_score(df, X_train, Y_train,cv=10)

scores
```

```
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
array([0.75925926, 0.64814815, 0.81132075, 0.79245283, 0.75471698,
       0.75471698, 0.71698113, 0.81132075, 0.75471698, 0.77358491])
```

```
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), 3* scores.std() ))
```

```
Accuracy: 0.76 (+/- 0.14)
```

```
#decision trees
```

```
#entropy
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
clf_entropy = DecisionTreeClassifier(criterion='entropy',
                                      min_samples_leaf=6,
                                      min_weight_fraction_leaf=0.1,
                                     min_samples_split=5)
```

```
clf_entropy.fit(X_train,Y_train)
```

```
DecisionTreeClassifier(criterion='entropy', min_samples_leaf=6,
                       min_samples_split=5, min_weight_fraction_leaf=0.1)
```

```
train_entropy = clf_entropy.predict(X_train)
from sklearn.metrics import accuracy_score
```

```
acc = accuracy_score(Y_train,train_entropy)
print("Train Accuracy = " ,acc)
```

```
Train Accuracy =  0.7725563909774437
```

```
predicted_entropy = clf_entropy.predict(X_test)

```

```
from sklearn.metrics import accuracy_score
acc = accuracy_score(Y_test,predicted_entropy)
print("Accuracy = " ,acc)

```

```
Accuracy =  0.7611940298507462
```

```
from  sklearn.metrics import confusion_matrix
cM = confusion_matrix(Y_test,predicted_entropy)
print(cM)
```

```
[[43 21]
 [11 59]]
```

```
tn, fn, fp, tp = cM.ravel()
print("Accuracy=",(tp+tn)/(tp+tn+fp+fn))
```

```
Accuracy= 0.7611940298507462
```

```
recall = tp/(tp+fn)
precision=tp/(tp+fp)
print("Recall = Sensitivity = ",tp/(tp+fn))
print("Specificity =", tn/(tn+fp))

print("Precision=",tp/(tp+fp))
f1score= 2 *(recall*precision)/(precision+recall)
print("f1 score=", f1score)
```

```
Recall = Sensitivity =  0.7375
Specificity = 0.7962962962962963
Precision= 0.8428571428571429
f1 score= 0.7866666666666667
```

```
from sklearn.metrics import roc_curve
from  sklearn.metrics import auc
fpr, tpr, thresholds = roc_curve(Y_test,predicted_entropy)
aucP=auc(fpr, tpr)

print('auc of ROC curve = ', aucP)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [123], in <cell line: 3>()
      1 from sklearn.metrics import roc_curve
      2 from  sklearn.metrics import auc
----> 3 fpr, tpr, thresholds = roc_curve(Y_test,predicted_entropy)
      4 aucP=auc(fpr, tpr)
      6 print('auc of ROC curve = ', aucP)

File E:\Anaconda\lib\site-packages\sklearn\metrics\_ranking.py:962, in roc_curve(y_true, y_score, pos_label, sample_weight,
drop_intermediate)
    873 def roc_curve(
    874     y_true, y_score, *, pos_label=None, sample_weight=None, drop_intermediate=True
    875 ):
    876     """Compute Receiver operating characteristic (ROC).
    877
    878     Note: this implementation is restricted to the binary classification task.
   (...)
    960
    961     """
--> 962     fps, tps, thresholds = _binary_clf_curve(
    963         y_true, y_score, pos_label=pos_label, sample_weight=sample_weight
    964     )
    966     # Attempt to drop thresholds corresponding to points in between and
    967     # collinear with other points. These are always suboptimal and do not
    968     # appear on a plotted ROC curve (and thus do not affect the AUC).
   (...)
    973     # but does not drop more complicated cases like fps = [1, 3, 7],
    974     # tps = [1, 2, 4]; there is no harm in keeping too many thresholds.
    975     if drop_intermediate and len(fps) > 2:

File E:\Anaconda\lib\site-packages\sklearn\metrics\_ranking.py:748, in _binary_clf_curve(y_true, y_score, pos_label, sample_weight)
    745     y_score = y_score[nonzero_weight_mask]
    746     sample_weight = sample_weight[nonzero_weight_mask]
--> 748 pos_label = _check_pos_label_consistency(pos_label, y_true)
    750 # make y_true a boolean vector
    751 y_true = y_true == pos_label

File E:\Anaconda\lib\site-packages\sklearn\metrics\_base.py:243, in _check_pos_label_consistency(pos_label, y_true)
```

```python

from sklearn.model_selection import cross_val_score

scores = cross_val_score(df, X_train, Y_train,cv=10)

scores
```

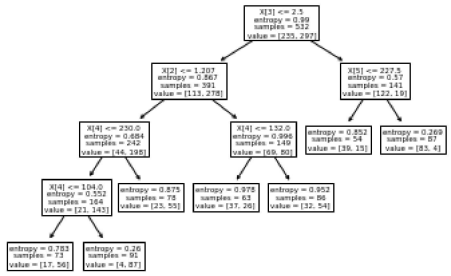
```
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
array([0.74074074, 0.74074074, 0.83018868, 0.71698113, 0.71698113,
       0.81132075, 0.73584906, 0.77358491, 0.67924528, 0.75471698])
```

```python
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), 3* scores.std() ))
```

```
Accuracy: 0.75 (+/- 0.13)
```

```python
from sklearn.tree import plot_tree
plot_tree(clf_entropy)
```

```
[Text(0.625, 0.9, 'X[3] <= 2.5\nentropy = 0.99\nsamples = 532\nvalue = [235, 297]'),
 Text(0.4166666666666667, 0.7, 'X[2] <= 1.207\nentropy = 0.867\nsamples = 391\nvalue = [113, 278]'),
 Text(0.25, 0.5, 'X[4] <= 230.0\nentropy = 0.684\nsamples = 242\nvalue = [44, 198]'),
 Text(0.16666666666666666, 0.3, 'X[4] <= 104.0\nentropy = 0.552\nsamples = 164\nvalue = [21, 143]'),
 Text(0.08333333333333333, 0.1, 'entropy = 0.783\nsamples = 73\nvalue = [17, 56]'),
 Text(0.25, 0.1, 'entropy = 0.26\nsamples = 91\nvalue = [4, 87]'),
 Text(0.3333333333333333, 0.3, 'entropy = 0.875\nsamples = 78\nvalue = [23, 55]'),
 Text(0.5833333333333334, 0.5, 'X[4] <= 132.0\nentropy = 0.996\nsamples = 149\nvalue = [69, 80]'),
 Text(0.5, 0.3, 'entropy = 0.978\nsamples = 63\nvalue = [37, 26]'),
 Text(0.6666666666666666, 0.3, 'entropy = 0.952\nsamples = 86\nvalue = [32, 54]'),
 Text(0.8333333333333334, 0.7, 'X[5] <= 227.5\nentropy = 0.57\nsamples = 141\nvalue = [122, 19]'),
 Text(0.75, 0.5, 'entropy = 0.852\nsamples = 54\nvalue = [39, 15]'),
 Text(0.9166666666666666, 0.5, 'entropy = 0.269\nsamples = 87\nvalue = [83, 4]')]
```




```
1 #gini
```

```
1 clf_entropy = DecisionTreeClassifier(
2                                       min_samples_leaf=4,
3                                      min_weight_fraction_leaf=0.1,
4                                     min_samples_split=4)
```

```
1 clf_entropy.fit(X_train,Y_train)
```

```
DecisionTreeClassifier(min_samples_leaf=4, min_samples_split=4,
                       min_weight_fraction_leaf=0.1)
```

```
1 train_entropy = clf_entropy.predict(X_train)
2 from sklearn.metrics import accuracy_score
3 acc = accuracy_score(Y_train,train_entropy)
4 print("Train Accuracy = " ,acc)
```

```
Train Accuracy =  0.7725563909774437
```

```
1 predicted_entropy = clf_entropy.predict(X_test)
```

```
1 from sklearn.metrics import accuracy_score
2 acc = accuracy_score(Y_test,predicted_entropy)
3 print("Accuracy = " ,acc)
```

```
Accuracy =  0.7611940298507462
```

```
1 from  sklearn.metrics import confusion_matrix
2 cM = confusion_matrix(Y_test,predicted_entropy)
3 print(cM)
```

```
[[43 21]
 [11 59]]
```

```
1 tn, fn, fp, tp = cM.ravel()
2 print("Accuracy=",(tp+tn)/(tp+tn+fp+fn))
```

```
Accuracy= 0.7611940298507462
```

```
1 recall = tp/(tp+fn)
2 precision=tp/(tp+fp)
3 print("Recall = Sensitivity = ",tp/(tp+fn))
4 print("Specificity =", tn/(tn+fp))
```

```
5
6 print("Precision=",tp/(tp+fp))
7 f1score= 2 *(recall*precision)/(precision+recall)
8 print("f1 score=", f1score)
```

```
Recall = Sensitivity =  0.7375
Specificity = 0.7962962962962963
Precision= 0.8428571428571429
f1 score= 0.7866666666666667
```

```
1 from sklearn.metrics import roc_curve
2 from  sklearn.metrics import auc
3 fpr, tpr, thresholds = roc_curve(Y_test,predicted_entropy)
4 aucP=auc(fpr, tpr)
5
6 print('auc of ROC curve = ', aucP)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [137], in <cell line: 3>()
      1 from sklearn.metrics import roc_curve
      2 from  sklearn.metrics import auc
----> 3 fpr, tpr, thresholds = roc_curve(Y_test,predicted_entropy)
      4 aucP=auc(fpr, tpr)
      6 print('auc of ROC curve = ', aucP)

File E:\Anaconda\lib\site-packages\sklearn\metrics\_ranking.py:962, in roc_curve(y_true, y_score, pos_label, sample_weight,
drop_intermediate)
    873 def roc_curve(
    874     y_true, y_score, *, pos_label=None, sample_weight=None, drop_intermediate=True
    875 ):
    876     """Compute Receiver operating characteristic (ROC).
    877
    878     Note: this implementation is restricted to the binary classification task.
   (...)
    960
    961     """
--> 962     fps, tps, thresholds = _binary_clf_curve(
    963         y_true, y_score, pos_label=pos_label, sample_weight=sample_weight
    964     )
    966     # Attempt to drop thresholds corresponding to points in between and
    967     # collinear with other points. These are always suboptimal and do not
    968     # appear on a plotted ROC curve (and thus do not affect the AUC).
   (...)
    973     # but does not drop more complicated cases like fps = [1, 3, 7],
    974     # tps = [1, 2, 4]; there is no harm in keeping too many thresholds.
    975     if drop_intermediate and len(fps) > 2:

File E:\Anaconda\lib\site-packages\sklearn\metrics\_ranking.py:748, in _binary_clf_curve(y_true, y_score, pos_label, sample_weight)
    745     y_score = y_score[nonzero_weight_mask]
    746     sample_weight = sample_weight[nonzero_weight_mask]
--> 748 pos_label = _check_pos_label_consistency(pos_label, y_true)
    750 # make y_true a boolean vector
    751 y_true = y_true == pos_label

File E:\Anaconda\lib\site-packages\sklearn\metrics\_base.py:243, in _check_pos_label_consistency(pos_label, y_true)
    232 if pos_label is None and (
    233     classes.dtype.kind in "OUS"
    234     or not (
   (...)
    240     )
    241 ):
    242     classes_repr = ", ".join(repr(c) for c in classes)
--> 243     raise ValueError(
    244         f"y_true takes value in {{{classes_repr}}} and pos_label is not "
    245         "specified: either make y_true take value in {0, 1} or "
    246         "{-1, 1} or pass pos_label explicitly."
    247     )
    248 elif pos_label is None:
    249     pos_label = 1

ValueError: y_true takes value in {'+', '-'} and pos_label is not specified: either make y_true take value in {0, 1} or {-1, 1} or pass
pos_label explicitly.
```

SEARCH STACK OVERFLOW

```

from sklearn.model_selection import cross_val_score

scores = cross_val_score(df, X_train, Y_train,cv=10)

scores
```

```
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
array([0.75925926, 0.74074074, 0.75471698, 0.69811321, 0.67924528,
       0.71698113, 0.73584906, 0.88679245, 0.77358491, 0.79245283])
```

```
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), 3* scores.std() ))
```

```
Accuracy: 0.75 (+/- 0.16)
```

```
#MLP Classification
```

```
from sklearn.neural_network import MLPClassifier
clf_entropy=MLPClassifier (hidden_layer_sizes=(15), activation='logistic')
```

```
clf_entropy.fit(X_train,Y_train)
```

```
E:\Anaconda\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:1109: DataConversionWarning: A column-vector y was passec
  y = column_or_1d(y, warn=True)
MLPClassifier(activation='logistic', hidden_layer_sizes=15)
```

```
train_entropy = clf_entropy.predict(X_train)
from sklearn.metrics import accuracy_score
acc = accuracy_score(Y_train,train_entropy)
print("Train Accuracy = " ,acc)

```

```
Train Accuracy =  0.7593984962406015
```

```
from  sklearn.metrics import confusion_matrix
cM = confusion_matrix(Y_test,predicted_entropy)
print(cM)
```

```
[[43 21]
 [11 59]]
```

```
tn, fn, fp, tp = cM.ravel()
print("Accuracy=",(tp+tn)/(tp+tn+fp+fn))
```

```
Accuracy= 0.7611940298507462
```

```
recall = tp/(tp+fn)
precision=tp/(tp+fp)
```

```
print("Recall = Sensitivity = ",tp/(tp+fn))
print("Specificity =", tn/(tn+fp))

print("Precision=",tp/(tp+fp))
f1score= 2 *(recall*precision)/(precision+recall)
print("f1 score=", f1score)
```

```
Recall = Sensitivity =  0.7375
Specificity = 0.7962962962962963
Precision= 0.8428571428571429
f1 score= 0.7866666666666667
```

```
from sklearn.metrics import roc_curve
from  sklearn.metrics import auc
fpr, tpr, thresholds = roc_curve(Y_test,predicted_entropy)
aucP=auc(fpr, tpr)

print('auc of ROC curve = ', aucP)
```

```python

from sklearn.model_selection import cross_val_score

scores = cross_val_score(df, X_train, Y_train,cv=10)

scores
```

```
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
array([0.7962963 , 0.7037037 , 0.79245283, 0.75471698, 0.64150943,
       0.75471698, 0.69811321, 0.81132075, 0.79245283, 0.83018868])
```

```python
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), 3* scores.std() ))
```

```
Accuracy: 0.76 (+/- 0.17)
```

```python
#Support Vector Machines
```

```
(...)
```

```python
from sklearn.svm import SVC
clf_entropy = SVC(kernel='sigmoid')
```

```python
clf_entropy.fit(X_train,Y_train)
```

```
E:\Anaconda\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was e
  y = column_or_1d(y, warn=True)
SVC(kernel='sigmoid')
```

```python
train_entropy = clf_entropy.predict(X_train)
from sklearn.metrics import accuracy_score
acc = accuracy_score(Y_train,train_entropy)
print("Train Accuracy = " ,acc)
```

```
Train Accuracy =  0.6597744360902256
```

```python
from  sklearn.metrics import confusion_matrix
cM = confusion_matrix(Y_test,predicted_entropy)
print(cM)
```

```
[[43 21]
 [11 59]]
```

```python
tn, fn, fp, tp = cM.ravel()
print("Accuracy=",(tp+tn)/(tp+tn+fp+fn))
```

```
Accuracy= 0.7611940298507462
```

```python
recall = tp/(tp+fn)
precision=tp/(tp+fp)
```

```
print("Recall = Sensitivity = ",tp/(tp+fn))
print("Specificity =", tn/(tn+fp))

print("Precision=",tp/(tp+fp))
f1score= 2 *(recall*precision)/(precision+recall)
print("f1 score=", f1score)
```

```
Recall = Sensitivity =  0.7375
Specificity = 0.7962962962962963
Precision= 0.8428571428571429
f1 score= 0.7866666666666667
```

```
from sklearn.metrics import roc_curve
from  sklearn.metrics import auc
fpr, tpr, thresholds = roc_curve(Y_test,predicted_entropy)
aucP=auc(fpr, tpr)

print('auc of ROC curve = ', aucP)
```

```
from sklearn.model_selection import cross_val_score

scores = cross_val_score(df, X_train, Y_train,cv=10)

scores
```

```
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
E:\Anaconda\lib\site-packages\sklearn\model_selection\_validation.py:680: DataConversionWarning: A column-vector y was passed when a 1d
  estimator.fit(X_train, y_train, **fit_params)
array([0.81481481, 0.7037037 , 0.81132075, 0.69811321, 0.67924528,
       0.73584906, 0.79245283, 0.73584906, 0.77358491, 0.79245283])
```

```
print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), 3* scores.std() ))
```

```
Accuracy: 0.75 (+/- 0.14)
```

```
232 if pos_label is None and (
(...)
242     classes_repr = ", ".join(repr(c) for c in classes)
245         specified: either make y_true take value in {0, 1} or
246 elif pos_label is None:
pos_label explicitly
```

SEARCH STACK OVERFLOW