from sklearn.feature\_selection import SelectKBest

from sklearn.feature\_selection import chi2

import pandas as pd

import numpy as np

from sklearn.experimental import enable\_iterative\_imputer

from sklearn.linear\_model import Lasso

from sklearn.impute import IterativeImputer

from sklearn.model\_selection import train\_test\_split

from skfeature.function.similarity\_based import fisher\_score

import matplotlib.pyplot as plt

df = pd.read\_csv('originalData.csv')

X=df.drop('num',axis=1)

y=df['num']

print(type(X))

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X,y,test\_size=0.2, stratify=y, random\_state=2)

lr = Lasso()

imp=IterativeImputer(estimator=lr,verbose=2,max\_iter=330,tol=1e-10,imputation\_order='roman')

imp.fit(X)

a=imp.transform(X)

##type(a)

my\_array = np.array(a)

X = pd.DataFrame(my\_array)

aX = X.to\_numpy()

aY = y.to\_numpy()

ranks = fisher\_score.fisher\_score(aX,aY)

feature\_importances=pd.Series(ranks, X.columns[0:len(X.columns)])

print(feature\_importances)

feature\_importances.plot(kind='barh', color='teal')

plt.show()