```python
# -*- coding: utf-8 -*-
"""Data_Preprocessing

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1cqiOtupPBH0ZZWkk2pkLA-k84lVnQgSB

<center><h1>Mini Project 1 - Logistic Regression</h1>
<h4>This file is for data preprocessing. Important characteristics of features and the distribution of classes could be found in the result of this file.</h4></center>

<h3>Team Members:</h3>
<center>
Yi Zhu, 260716006<br>
Fei Peng, 260712440<br>
Yukai Zhang, 260710915
</center>
"""

from google.colab import drive
drive.mount('/content/drive')

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import random
from scipy import stats
from google.colab import files

path1 = "/content/drive/My Drive/ECSE_551_F_2020/Mini_Project_01/hepatitis.csv"
path2 = "/content/drive/My Drive/ECSE_551_F_2020/Mini_Project_01/bankrupcy.csv"

hepatitis_data = pd.read_csv(path1)
bankrupcy_data = pd.read_csv(path2)

class Data_Processing:
    def __init__(self, data, name = 'New Data'):
        self.data = data
        self.name = name

    def partition_by_class(self):
        pos, neg = [], []
        data = self.data
        for row in range(1, data.shape[0]):
            if data[row, -1] == 1:
                pos.append(list(data[row]))
            else:
                neg.append(list(data[row]))
        self.pos = pos
        self.neg = neg

    def show_y_dist(self, ydata):
        plt.figure(figsize=(5,4))
        plt.subplot(111), sns.countplot(x='ClassLabel', data=ydata)
        plt.title('Distribution of the two classes {}'.format(self.name))
        plt.savefig("Distribution of the two classes {}.png".format(self.name), dpi = 1200)
        files.download("Distribution of the two classes {}.png".format(self.name))
        plt.show()

    def show_x_dist(self, xdata):
        pos = self.pos
        neg = self.neg
        for i in range(0,len(pos[1])):
            fig, (ax1, ax2) = plt.subplots(1, 2)
            fig.set_size_inches(10,4)
            plt.setp(ax1.xaxis.get_majorticklabels(), rotation=90)
            plt.setp(ax2.xaxis.get_majorticklabels(), rotation=90)
            fig.suptitle('Histogram of {}'.format(xdata.keys()[i]))
            ax1.title.set_text('{} positive class'.format(xdata.keys()[i]))
            ax1.hist([pos[j][i] for j in range(len(pos))])
            ax2.title.set_text('{} negative class'.format(xdata.keys()[i]))
            ax2.hist([neg[j][i] for j in range(len(neg))])

    def find_null_data(self):
        data, pos, neg = self.data, self.pos, self.neg
        num_data = min(len(pos), len(neg))
        num_feature = len(pos[0])
        null_feature_count = np.zeros(num_feature)
        pos = random.sample(pos, k = num_data)
        neg = random.sample(neg, k = num_data)
        for i in range(len(pos[1])):
            posi_list = []
            nega_list = []
            for j in range(len(pos)):
                posi_list.append(pos[j][i])
                nega_list.append(neg[j][i])
            a, b = stats.ks_2samp(posi_list, nega_list)
            if(b > 0.35):
                null_feature_count[i] += 1
            elif ((b > 0.10) and (a > 0.10)):
                null_feature_count[i] += 0.5
        if sum(null_feature_count) > num_feature * 0.1:
            print("cannot remove this many features")
        else:
            print("Here are the features we recomend you to delete")
            for i in range(len(null_feature_count)):
                if null_feature_count[i] > 0:
                    print("{}: {}".format(i, null_feature_count[i]))

"""### Plot the classes and features distribution for Hepatitis data"""

hepatitis_data = pd.read_csv(path1)
data = hepatitis_data
data1 = Data_Processing(data.values, 'hepatitis')
data1.partition_by_class()
data1.show_y_dist(data)
data1.show_x_dist(data)
data1.find_null_data()

"""### Plot the classes and features distribution for Bankruptcy data"""

bankrupcy_data = pd.read_csv(path2)
data = bankrupcy_data
data1 = Data_Processing(data.values, 'bankrupcy')
data1.partition_by_class()
data1.show_y_dist(data)
data1.show_x_dist(data)
data1.find_null_data()

"""After tests, we find that in bankrupcy.csv, if features: 43, 60, (36) are deleted, it may deliver a better result.
```

```
<br>
43, 60 very high p-value in ks test
<br>
36 relative high but steady p-value
<hr>

Hepatitis cannot use this method since we have many features are only in 1 or 0.
"""
```