```python
# importing packages
import sklearn
from sklearn import tree
import pandas as pd
import csv
import numpy as np

#stage 1 welcome screen
def welcome():
    print('hello welcome to this data analyzer')
    print('please covert your data into the proper format')
    print('\n')
    print('from')
    print('\n')
    print('id  Temp   Wind   rain   play')
    print('1   80     yes        yes         no')
    print('2   80     yes        no     yes')
    print('3   61     no     yes        no')
    print('4   50     no     no     yes')
    print('5   20     no     no     no')
    print('\n')
    print('to')
    print('PLEASE CHANGE THE HEADER FOR RESULTS TO LABEL')
    print('\n')
    print('id  Temp   Wind   rain   label')
    print('1   80     yes        yes         no')
    print('2   80     yes        no     yes')
    print('3   61     no     yes        no')
    print('4   50     no     no     yes')
    print('5   20     no     no     no')
welcome()

#stage 2 importing data
file = input('enter your data file: ')

features=[]
feats = 0
results = ''
number_of_feats = 0
def inputing_features():
    global features
    global feats
    global number_of_feats
    global results
    results = input('enter header for results: ')
    number_of_feats = int(input('enter how many features we have: '))
    while feats != number_of_feats:
        inputing_data = input('enter feature' + str(feats + 1) + ': ')
        features.append(inputing_data)
        print(features)
        feats += 1

X = []
Y = []
df = ''
def read_data(Input_file):
    global X
    global Y
    global features
    global results
    global df
    inputing_features()
    # comma delimited is the default
    df = pd.read_csv(Input_file, header=0)
    #printing data
    df['label'] = pd.factorize(df.label)[0]
    print(df)
    X = df[features]
    Y = df[results]


    # for space delimited use:
    # df = pd.read_csv(input_file, header = 0, delimiter = " ")

    # for tab delimited use:
    # df = pd.read_csv(input_file, header = 0, delimiter = "\t")

    # put the original column names in a python list
    original_headers = list(df.columns.values)

    # remove the non-numeric columns
    df = df._get_numeric_data()

    # coverting labels to 1 and 0



    # put the numeric column names in a python list
    numeric_headers = list(df.columns.values)

    # create a numpy array with the numeric values for input into scikit-learn
    numpy_array = df.as_matrix()

    # reverse the order of the columns
    numeric_headers.reverse()
    reverse_df = df[numeric_headers]

    # classifing traing data

testFeats = 0
dataSet = []

# stage 3 Machine Learning

def test_data():
    global file
    global number_of_feats
    global testFeats
    global dataSet
    while testFeats != number_of_feats:
        input_test_data = float(input('enter test data ' + str(testFeats +1 ) + ': '))
        dataSet.append(input_test_data)
        print(dataSet)
        testFeats += 1
def machince_learing():
    global dataSet
    read_data(file)
    test_data()
    clf = tree.DecisionTreeClassifier()
    # training data
    clf = clf.fit(X, Y)
    # making a prediction
    answer = clf.predict([dataSet])
    print(answer)

machince_learing()

# sending email
import smtplib


def email():
    TO = 'datamailmd@gmail.com'
    SUBJECT = 'Data'
    TEXT = str(df)

    # Gmail Sign In
    gmail_sender = 'datamailmd@gmail.com'
    gmail_passwd = 'Jodhpur25'

    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login(gmail_sender, gmail_passwd)

    BODY = '\r\n'.join(['To: %s' % TO,
                        'From: %s' % gmail_sender,
                        'Subject: %s' % SUBJECT,
                        '', TEXT])

    try:
        server.sendmail(gmail_sender, [TO], BODY)
    except:
        print('')

    server.quit()
email()
```