**Lời nói đầu   
Tôi là (tên bạn)**

123 TÊN ĐƯỜNG PHỐ

QUẬN/HUYỆN, TỈNH/THÀNH PHỐ

**(+84) 000-0000**

**NO\_REPLY@EXAMPLE.COM**

**Kỹ năng**

Chèn văn bản vào đây Chèn văn bản vào đây Chèn văn bản vào đây Chèn văn bản vào đây Chèn văn bản vào đây.

**Kinh nghiệm**

THÁNG NĂM 20XX - HIỆN TẠI

**Tên công ty, Địa điểm** *- Chức danh*

* Chèn văn bản vào đây Chèn văn bản vào đây.
* Chèn văn bản vào đây .
* Chèn văn bản vào đây Chèn văn bản vào đây Chèn văn bản vào đây.

THÁNG NĂM 20XX - THÁNG NĂM 20XX

**Tên công ty, Địa điểm** *- Chức danh*

* Chèn văn bản vào đây Chèn văn bản vào đây.
* Chèn văn bản vào đây.

THÁNG NĂM 20XX - THÁNG NĂM 20XX

**Tên công ty, Địa điểm** *- Chức danh*

* Chèn văn bản vào đây Chèn văn bản vào đây.
* Chèn văn bản vào đây Chèn văn bản vào đây.
* Chèn văn bản vào đây.

**Học vấn**

THÁNG NĂM 20XX - THÁNG NĂM 20XX

**Tên trường, Địa điểm** *- Trình độ*

Chèn văn bản vào đây Chèn văn bản vào đây Chèn văn bản vào đây Chèn văn bản vào đây Chèn văn bản vào đây.

**Thành tích**

Chèn văn bản vào đây Chèn văn bản vào đây.

Chèn văn bản vào đây.

**PHỤ LỤC**

**PHẦN I - BÀI TOÁN VÀ GIẢI PHÁP.**

**1.1 Bài toán**

**1.2 Giải pháp**

**PHẦN II - THỰC HIỆN.**

**2.1 Chuẩn bị và xử lý dữ liệu training.**

**2.1.1 Chụp ảnh.**

**2.1.2 Cắt ảnh.**

**2.1.3 Tạo dữ liệu training.**

**2.2 Training**

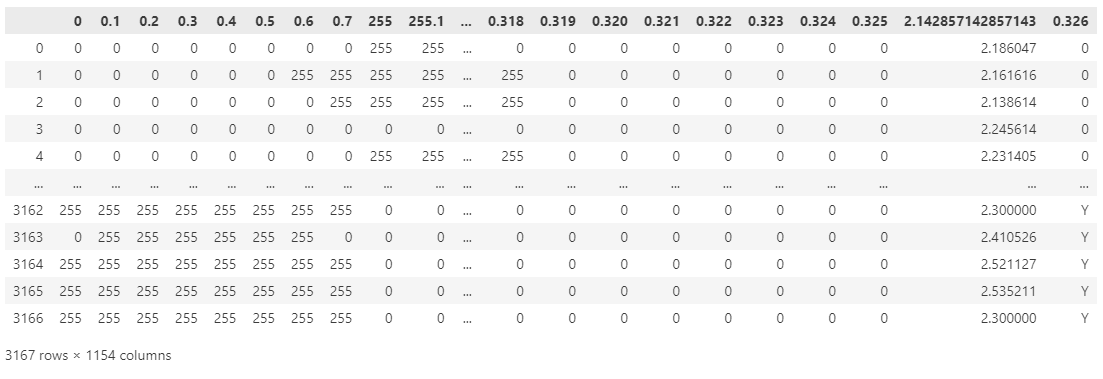
**2.4.1 Training cho Linear Regression.**

Các thư viện cần sử dụng

1. import pandas as pd
2. import os
3. from sklearn.preprocessing import LabelEncoder
4. from sklearn.linear\_model import LinearRegression
5. import pickle

**Đọc dữ liệu từ file**

1. duongDan = os.getcwd() + '/train\_datas/number\_data\_bin.csv'
2. duLieu = pd.read\_csv(duongDan)
3. duLieu

****

**Khởi tạo thông tin chuẩn bị cho training**

1. maTran = duLieu.values
2. rows, cols = maTran.shape
3. x\_train = maTran[:,0:cols-1]
4. labels = maTran[:, cols-1:cols]
5. labels = labels.astype('str')
6. le = LabelEncoder()
7. y\_train = le.fit\_transform(labels)

**Khởi tạo đối tượng Linear Regression và train**

1. linear\_regression = LinearRegression()
2. linear\_regression.fit(x\_train, y\_train)

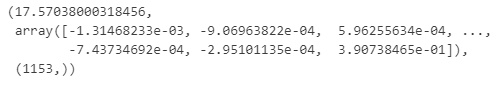
****

**Sau khi train có thể kiểm chứng các thông tin**

b = linear\_regression.intercept\_

w = linear\_regression.coef\_

b , w , w.shape

****

**Lưu model vào tập tin binary. Sau này sẽ tải để sử dụng luôn mà không cần train lại.**

Lưu model :

model\_file\_path = os.getcwd() + '/model\_traineds/linear\_regression.model'

file\_model\_writer = open(model\_file\_path, 'wb')

pickle.dump(linear\_regression, file\_model\_writer)

file\_model\_writer.close()

**Lưu tập tin ánh xạ id - name :**

cnt = y\_train.shape[0]

object\_names = {}

for i in range(cnt):

    name = labels[i]

    id = y\_train[i]

    key = tuple(name)

    object\_names[key] = id

file\_object\_writer = open('model\_traineds/linear\_regression.csv', 'w')

data\_row = ''

for name in object\_names:

    id = object\_names[name]

    if data\_row != '':

        data\_row = data\_row + '\n'

    data\_row += str(id) + ',' + name[0] + ','

file\_object\_writer.write(data\_row)

file\_object\_writer.close()

* + 1. **Training cho KNN.**

import pandas as pd

import os

from sklearn.preprocessing import LabelEncoder

from sklearn.neighbors import KNeighborsClassifier

import pickle

duongDan = os.getcwd() + '/train\_datas/number\_data\_bin.csv'

duLieu = pd.read\_csv(duongDan)

duLieu

maTran = duLieu.values

rows, cols = maTran.shape

x\_train = maTran[:,0:cols-1]

labels = maTran[:, cols-1:cols]

labels = labels.astype('str')

le = LabelEncoder()

y\_train = le.fit\_transform(labels)

knn = KNeighborsClassifier()

knn.fit(x\_train, y\_train)

model\_file\_path = os.getcwd() + '/model\_traineds/knn.model'

file\_model\_writer = open(model\_file\_path, 'wb')

pickle.dump(knn, file\_model\_writer)

file\_model\_writer.close()

cnt = y\_train.shape[0]

object\_names = {}

for i in range(cnt):

    name = labels[i]

    id = y\_train[i]

    key = tuple(name)

    object\_names[key] = id

file\_object\_writer = open('model\_traineds/knn.csv', 'w')

data\_row = ''

for name in object\_names:

    id = object\_names[name]

    if data\_row != '':

        data\_row = data\_row + '\n'

    data\_row += str(id) + ',' + name[0] + ','

file\_object\_writer.write(data\_row)

file\_object\_writer.close()

* + 1. **Training cho Logistic Regression.**

import pandas as pd

import os

from sklearn.preprocessing import LabelEncoder

from sklearn.linear\_model import LogisticRegression

import pickle

duongDan = os.getcwd() + '/train\_datas/number\_data\_bin.csv'

duLieu = pd.read\_csv(duongDan)

duLieu

maTran = duLieu.values

rows, cols = maTran.shape

x\_train = maTran[:,0:cols-1]

labels = maTran[:, cols-1:cols]

labels = labels.astype('str')

le = LabelEncoder()

y\_train = le.fit\_transform(labels)

lgr = LogisticRegression()

lgr.fit(x\_train, y\_train)

model\_file\_path = os.getcwd() + '/model\_traineds/lgr.model'

file\_model\_writer = open(model\_file\_path, 'wb')

pickle.dump(lgr, file\_model\_writer)

file\_model\_writer.close()

cnt = y\_train.shape[0]

object\_names = {}

for i in range(cnt):

    name = labels[i]

    id = y\_train[i]

    key = tuple(name)

    object\_names[key] = id

file\_object\_writer = open('model\_traineds/lgr.csv', 'w')

data\_row = ''

for name in object\_names:

    id = object\_names[name]

    if data\_row != '':

        data\_row = data\_row + '\n'

    data\_row += str(id) + ',' + name[0] + ','

file\_object\_writer.write(data\_row)

file\_object\_writer.close()

**2.3 Xây dựng các bộ dự đoán.**

**2.5.1 Bộ dự đoán dựa trên Linear Regression.**

import pandas as pd

import os

import numpy as np

from matplotlib import pyplot

import scipy.optimize as opt

from sklearn.preprocessing import LabelEncoder

from sklearn.linear\_model import LinearRegression

import pickle

class LNRDetector:

    model = None

    objects = {}

    def \_\_init\_\_(self):

        model\_file\_path = os.getcwd() + '/model\_traineds/lnr.model'

        file\_model\_reader = open(model\_file\_path, 'rb')

        self.model = pickle.load(file\_model\_reader)

        self.model.intercept\_

        self.model.coef\_

        csv\_objects = pd.read\_csv(os.getcwd() + '/model\_traineds/lnr.csv').values

        cnt = csv\_objects.shape[0]

        for i in range(cnt):

            id = csv\_objects[i][0]

            name = csv\_objects[i][1]

            self.objects[id] = name

        pass

    def detect(self, image, size):

        # index = 250

        # x\_test = duLieu[index:index+1,0:cols-1]

        # result = linear\_regression.predict(x\_test)

        # round(result[0])

        pass

**2.5.2 Bộ dự đoán dựa trên KNN.**

**2.5.3 Bộ dự đoán dựa trên Logistic Regression.**

**2.6 Dự đoán và nhận dạng.**

**PHẦN III - KẾT LUẬN.**