TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN – CƠ – TIN HỌC

------------------------



**BÁO CÁO THỰC TẬP**

**MÔN: ĐỒ ÁN PHẦM MỀM**

Họ và Tên: **Bùi Hải Đăng**

Chuyên Ngành: **Máy tính và khoa học thông tin**

Lớp: **K62A4**

Khóa: **2017-2021**

HÀ NỘI – NĂM 2019

**MỤC LỤC**

[**PHẦN I : GIỚI THIỆU 1**](#_Toc17026312)

[**1. Những công việc được giao 1**](#_Toc17026313)

[**2. Những công cụ sử dụng giải quyết công việc 1**](#_Toc17026314)

[**PHẦN II : CÔNG VIỆC TRIỂN KHAI 2**](#_Toc17026315)

[**1. Học Python và triển khai một số thuật toán bằng ngôn ngữ này 2**](#_Toc17026316)

[**2. Tìm hiểu một số thư viện của Python 4**](#_Toc17026317)

[**3.Xây dựng Sentiment Analysis System 5**](#_Toc17026318)

[**4. Deploy trên Web 10**](#_Toc17026319)

[**PHẦN III : KẾT QUẢ ĐẠT ĐƯỢC 13**](#_Toc17026320)

[**PHẦN IV : PHỤ LỤC 16**](#_Toc17026321)

# PHẦN I : GIỚI THIỆU

## 1. Những công việc được giao

- Học và làm quen với ngôn ngữ lập trình Python:

* Học cú pháp
* Class
* Học cấu trúc dữ liệu và giải thuật : Áp dụng vào viết một số giải thuật cơ bản trong Python như : Sắp xếp và tìm kiếm
* Tìm hiểu một số thư viện trong Python để áp dụng cho Meachine Learning
  + Numpy
  + Sklearn
  + Gensim
  + Pandas
* Xây dựng Sentiment analysis system và Deploy trên Web:

Cơ bản về Sentiment analysis system: Là hệ thông phân tích tình cảm đề cập đến việc sử dụng xử lý ngôn ngữ tự nhiên, phân tích văn bản, ngôn ngữ học tính toán và sinh trắc học để xác định một cách có hệ thống, trích xuất, định lượng và nghiên cứu các trạng thái tình cảm và thông tin chủ quan.

## 2. Những công cụ sử dụng giải quyết công việc

- IDE Pycharm :

* Ưu điểm: PyCharm có giao diện dễ nhìn, xử lý nhanh ,cung cấp nhiều tính năng thông minh như bộ code completion, dễ dàng điều hướng và kiểm tra lỗi. IDE này có thể tự động thụt lề, phát hiện văn bản trùng lặp và kiểm tra lỗi.
* Nhược điểm : Muốn thêm những tính năng vượt trội thì phải trả phí
* Hệ điều hành Ubuntu:
  + - Ưu điểm :
      1. HĐH với giao diện khá đơn giản, dễ tiếp cận
      2. Dung lượng nhẹ, khởi động nhanh , xử lý tác vụ nhanh
      3. Mạnh mẽ trong lập trình, có cộng đồng lớn
      4. HĐH mã nguồn mở
* Nhược điểm:

1. Thỉnh thoảng hay bị crash
2. Nhiều phần mềm không trợ

# PHẦN II : CÔNG VIỆC TRIỂN KHAI

## 1. Học Python và triển khai một số thuật toán bằng ngôn ngữ này

Em bắt đầu tìm hiểu về Python từ những thứ cơ bản nhất . Bắt đầu từ việc tìm một IDE thích hợp để code, em chọn PyCharm.

Tiếp theo, em bắt đầu học Cú pháp của Python :

* + Biến số
  + Toán tử số học
  + Boolean và toán tử logic : Trong đó có toán tử in/not in để kiểm tra phần tử có tồn tại hay không
  + Cấu trúc điều khiển:

Python hộ trợ một số cấu trúc điểu khiển thông dụng. Hầu hết các cấu trúc điều khiển đều dựa vào thụt đầu dòng (indention) để tạo thành một block xử lý, thay vì sử dụng { ... } như JAVA

- If...elif...else

- Switch ... case

- For ... in

- While

* + Hàm:

Python khai bào hàm theo cú pháp :

**def functioname(param,param2,...):**

**#code**

* + Xử lý chuỗi :

- Khai báo chuỗi

- Nối chuỗi

- Trích xuất chuỗi con

- Lấy đồ dài chuỗi

.....

* + List :

List trong Python là cấu trúc mảng và các phần tử có index có thứ tự.

Em học một số tính năng của List như :

- Kiểm tra theo giá trị

- Trích xuất mảng con

- Thêm, xóa phần tử

.....

* + Tuple
  + Set
  + Dictionary

Sau khi học về Cú pháp, em tìm hiểu về cấu trúc thư của một package trong Python. Cụ thể :

- Một Package sẽ có cấu trúc như sau:

**|**--demopackage

| |-- \_\_init\_\_.py

| |-- mymodule.py

| |-- mymodule2.py

**-** Khi sử dụng một module thuộc một package thì các lệnh trong file \_\_init\_\_.py sẽ được thực hiện trước.Thông thường thì file này sẽ rỗng.

Đã nắm được cơ bản về Python, em được giao nhiệm vụ đọc về Cấu trúc dữ liệu và Giải thuật

Tham khảo trên trang web: <https://vietjack.com/cau-truc-du-lieu-va-giai-thuat/> và kiến thức đã học được ở môn Cấu trúc dữ liệu và Giải thuật trên trường, em đã code trên Python một số giải thuật sau:

**-** Giải thuật sắp xếp trộn Merge sort

**def** mergeSort(arr):  
 **if** len(arr) >1:  
 mid = len(arr)//2  
 L = arr[:mid]  
 R = arr[mid:]  
 mergeSort(L)  
 mergeSort(R)  
 i = j = k = 0  
 **while** i < len(L) **and** j < len(R):  
 **if** L[i] < R[j]:  
 arr[k] = L[i]  
 i += 1  
 **else**:  
 arr[k] = R[j]  
 j += 1  
 k += 1  
 **while** i<len(L):  
 arr[k] = L[i]  
 i+= 1  
 k+= 1  
 **while** j < len(R):  
 arr[k] = R[j]  
 j += 1  
 k += 1

**-** Giải thuật sắp xếp trộn Quick sort

**def** partition(arr, low, high):  
 i = low -1  
 pivot = arr[high]  
 **for** j **in** range (low,high):  
 **if** arr[j] <= pivot:  
 i = i + 1  
 arr[i],arr[j] = arr[j],arr[i]  
  
 arr[i+1], arr[high] = arr[high],arr[i+1]  
 **return** (i+1)  
**def** quickSort(arr,low,high):  
 **if** low<high:  
 pi = partition(arr,low,high)  
 quickSort(arr,low,pi-1)  
 quickSort(arr,pi+1, high)

**-** Giải thuật Tìm kiếm nhị phân Binary Search

**def** binarySearch (arr, l,r,x):  
 **if** r >= 1:  
 mid = l +(r-1)//2  
 **if** arr[mid] == x:  
 **return** mid  
 **elif** arr[mid] >x:  
 **return** binarySearch(arr,l, mid-1,x)  
 **else** :  
 **return** binarySearch(arr, mid +1,r,x)  
 **else**:  
 **return** -1

## 2. Tìm hiểu một số thư viện của Python

Những thư viện em đã được giao để tìm hiểu đó là : Numpy, Sklearn, Gensim, Pandas

- Numpy:

Numpy là một thư viện lõi phục vụ cho khoa học máy tính của Python, hỗ trợ cho việc tính toán các mảng nhiều chiều, có kích thước lớn với các hàm đã được tối ưu áp dụng lên các mảng nhiều chiều đó. Numpy đặc biệt hữu ích khi thực hiện các hàm liên quan tới Đại Số Tuyến Tính.

- Sklearn:

Là thư viện phổ biến nhất trong Python. Nó viết sẵn các thuật toán phức tạp, ta chỉ cần import, đưa dữ liệu vào để lấy kết quả.

Em đã sử dụng SVM (Support Vector Machine) trong thư viện này để phân chia dữ liệu.

- Gensim:

Thư viện xử lý ngôn ngữ tự nhiên chuyên biệt về topic model

- Pandas:

Thư viện xử lý mạnh mẽ dữ liệu lớn. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu Dataframe.

Là một công cụ cho phép đọc/ ghi dữ liệu giữa bộ nhớ và nhiều định dạng file: csv,text,excel,sql database, hdf5.

Có thể thêm, xóa các cột dữ liệu.

Tối ưu về hiệu năng.

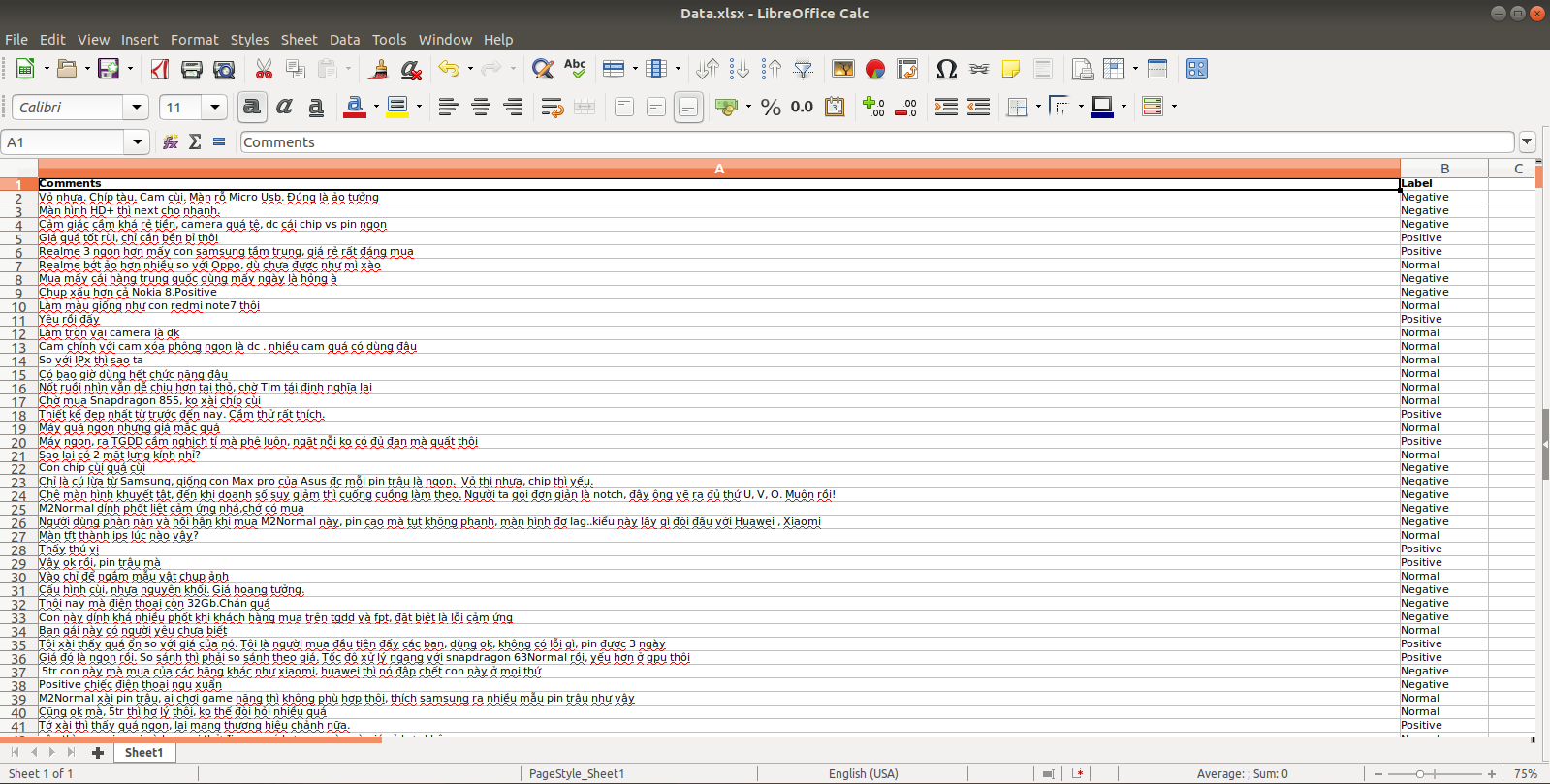
## 3.Xây dựng Sentiment Analysis System

* 1. **Thu thập dữ liệu trên Web:**

**-** Thu thập khoảng 1000 bình luận của người dùng đánh giá các sản phẩm công nghệ trên các trang : fpt.com, thegioididong.com, vnreview.com.

**-** Gán nhãn cho các bình luận với 3 trạng thái : tiêu cực ( negative ), bình thường (normal), tích cực (positvie).

**-** Lưu dữ liệu này trên excel : data.xlsx



* 1. **Tiền xử lý dữ liệu**
* Import dữ liệu vào Python:

- Sử dụng thư viện Pandas để import dữ liệu dạng xlsx vào Python:

**import** pandas **as** pd

**import** xlrd

col = [0]  
 data = pd.read\_excel(**r'/home/dang/Downloads/Data.xlsx'**,usecols=col)

- Lưu data dưới dạng List:

dataList = data[**'Comments'**].values.tolist()

* Tách từng từ trong một câu để xử lý:

texts=[]

**for** data **in** dataList:  
 comment = data.lower().split(**' '**)  
 texts.append(comment)  
 print(texts)

#Tách từng từ trong câu

text = [item **for** e **in** texts **for** item **in** e]  
 print(text)

* Loại bỏ các ký hiệu đặc biệt, khoảng trắng và chữ số: g

Các ký tự đặc biệt như “[!,@,#,$,%,...](mailto:!@#$%25,...)” , khoảng trắng và các chữ số có thể gây ra sai sót khi traning data

**for** e **in** text[:]:  
 text.remove(e)  
 e = e.translate(str.maketrans(**''**, **''**, string.punctuation))  
 e = re.sub(**r'\d+'**,**''**,e)  
 e.strip()  
 text.append(e)

* Chuẩn hóa các từ viết tắt :

VD : - “đk, đc, dc, dk” sẽ chuyển thành từ “được”

- “k, ko, hok” sẽ chuyển thành từ “không”

**-** “vs” sẽ chuyển thành “với”

Code trên Python:

wrongWords1= [**'dc'**,**'đk'**,**'dk'**]  
wrongWords2= [**'k'**,**'ko'**,**'hok'**]  
**for** i **in** range(0,len(text)):  
 **if** text[i] == **'vs'**:  
 text[i] = **'với'**

**for** j **in** range(0,len(wrongWords1)):  
 **if** text[i]==wrongWords1[j]:  
 text[i] = **'được'  
 for** h **in** range(0,len(wrongWords2)):  
 **if** text[i]== wrongWords2[h]:  
 text[i] = **'không'**

* Tách từ :

Trong tiếng Việt, dấu cách (space) không được sử dụng như 1 kí hiệu phân tách từ, nó chỉ có ý nghĩa phân tách các âm tiết với nhau. Vì thế, để xử lý tiếng Việt, công đoạn tách từ (word segmentation) là 1 trong những bài toán cơ bản và quan trọng bậc nhất.

Trong bài toán này em sử dụng thư viện “underthesea”

Code trên Python:

**from** underthesea **import** word\_tokenize

**for** i **in** range(0,len(dataList)):  
 radars\_string = **'\n'**.join(s **for** s **in** dataList[i])  
 word\_tk.append(word\_tokenize(radars\_string))

KQ: ['màn hình', 'hd', 'thì', 'next', 'cho', 'nhanh'], ['cảm giác', 'cầm', 'khá', 'rẻ tiền', 'camera', 'quá', 'tệ', 'được', 'cái', 'chip', 'với', 'pin', 'ngon'], ['giá', 'quá', 'tốt', 'rùi', 'chỉ', 'cần', 'bền bỉ', 'thôi'], ...

* Loại bỏ StopWords:

Những từ xuất hiện nhiều lần nhưng lại không mang nhiều ý nghĩa. Em sử dụng một set chứa 20 từ :

thì

để

này

cũng

...

Code trên Python :

stopwords = set(**'thì để này cũng bởi chưa cùng đã đang do đó nên nếu cho là được cái quá rất chỉ th'**.split(**' '**))

new\_data = [[word **for** word **in** document **if** word **not in** stopwords] **for** document **in** word\_tk]

KQ: ['màn hình', 'hd', 'next', 'nhanh'], ['cảm giác', 'cầm', 'khá', 'rẻ tiền', 'camera', 'tệ', 'chip', 'với', 'pin', 'ngon'], ['giá', 'tốt', 'rùi', 'cần', 'bền bỉ'], ....

-Sau khi đã hoàn thành công đoạn trên, xuất dữ liệu dạng file Excel (.xlsx) để tiếp tục bước tiếp theo:

Code trên Python :

#đưa từng từ đơn thành câu ban đầu

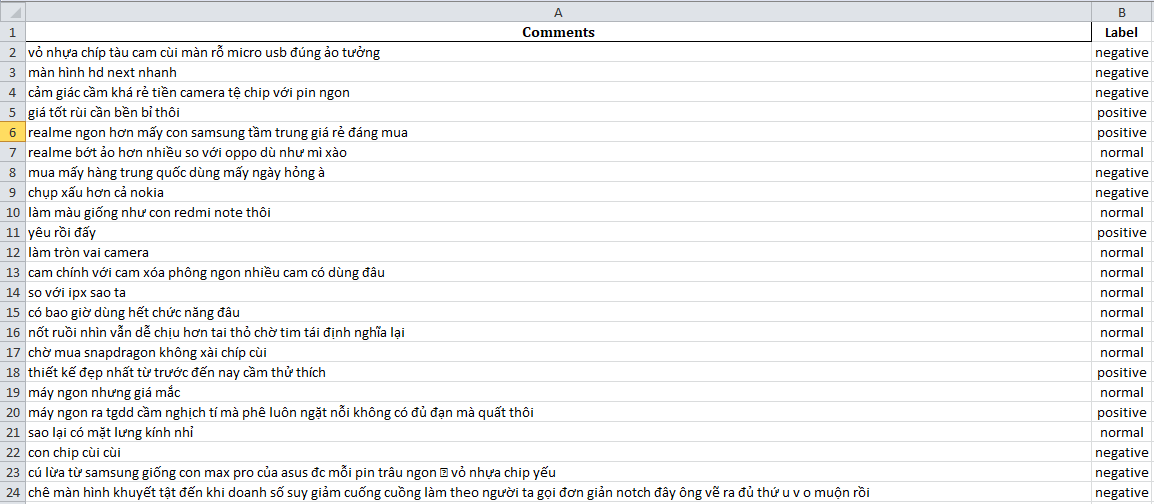
large = []

**for** i **in** range(0,len(texts)):  
 large.append(len(texts[i]))

old= []  
e = 0  
**for** i **in** range(0,len(large)):  
 old.append(text[e:large[i]+e])  
 e = large[i]+e  
new = [**' '**.join(l) **for** l **in** old]  
print(new)

dataNew = pd.DataFrame(new)  
export\_excel = dataNew.to\_excel(**r'/home/dang/Desktop/DEMO.xlsx'**)

- Dữ liệu sau khi chỉnh sửa :



**3.3 Chuẩn bị bộ Test dữ liệu và Train dữ liệu**

- Dữ liệu sẽ được chia làm 2 bộ, Training và Test. Tập Training data sẽ được sử dụng để phù hợp với model và dự đoán sẽ được thực hiện trên tập dữ liệu thử nghiệm.

Code trên Python:

**import** pandas **as** pd  
**from** sklearn **import** model\_selection, naive\_bayes, svm  
**from** sklearn.metrics **import** accuracy\_score

Corpus = pd.read\_excel(**r'/home/dang/Desktop/DEMO.xlsx'**,)  
Train\_X, Test\_X, Train\_Y, Test\_Y = model\_selection.train\_test\_split(Corpus[**'Comments'**],Corpus[**'Label'**])

- Vector hóa từ:

Đây là một quá trình để đưa một tập văn bản thành nhưng vector đặc trưng. Ở đây, em sử dụng phương thức ***TF-IDF***

Chuyển đổi Train\_X và Test\_X thành vector TF-IDF. Chúng sẽ chứa mỗi hàng một List số nguyên duy nhất.

Code trên Python:

**from** sklearn.feature\_extraction.text **import** TfidfVectorizer  
  
Tfidf\_vect = TfidfVectorizer()  
Tfidf\_vect.fit(Corpus[**'Comments'**])  
  
Train\_X\_Tfidf = Tfidf\_vect.transform(Train\_X)  
  
Test\_X\_Tfidf = Tfidf\_vect.transform(Test\_X)

Từ điển được học từ dữ liệu:

print(Tfidf\_vect.vocabulary\_)

{'vỏ': 1758, 'nhựa': 1060, 'chíp': 170, 'tàu': 1601, 'cam': 119, 'cùi': 260, 'màn': 843, 'rỗ': 1292, 'micro': 816, 'usb': 1689, 'đúng': 1912, 'ảo': 1989, 'tưởng': 1637, 'hình': 522, 'hd': 472, 'next': 917, 'nhanh': 982,...}

- Sử dụng thuật toán Machine Learning để dự đoán kết quả:

Sử dụng ***SVM – Support Vector Machine***

Code trên Python:

SVM = svm.SVC(kernel=**'linear'**)  
SVM.fit(Train\_X\_Tfidf,Train\_Y)  
  
predictions\_SVM = SVM.predict(Test\_X\_Tfidf)

**def** testComment(comment):  
  
 p=comment  
 Test\_p =Tfidf\_vect.transform([p])  
 **return** SVM.predict(Test\_p)

\*vectorized data:

(0, 134) 0.4704786153091223

(0, 186) 0.38733870146827654

(0, 778) 0.45061283210267533

(0, 834) 0.1612977138347135

(0, 951) 0.23484709715831104

(0, 1040) 0.2748320753945878

(0, 1308) 0.2708850214537861

(0, 1394) 0.23484709715831104

(0, 1521) 0.31147442056743146

(0, 1719) 0.20812431061478448

(1, 71) 0.2273187770579956

(1, 85) 0.2159468583096959

(1, 245) 0.2600039080524031

(1, 328) 0.28858604572205615

(1, 531) 0.17770770638392158

(1, 668) 0.28131712029851086

(1, 999) 0.16948612883448674

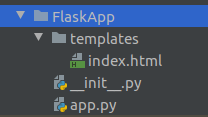
(1, 1016) 0.2453786527754752

...

## 4. Deploy trên Web

Sử dụng **Flask**, một framework Python cho ứng dụng web.

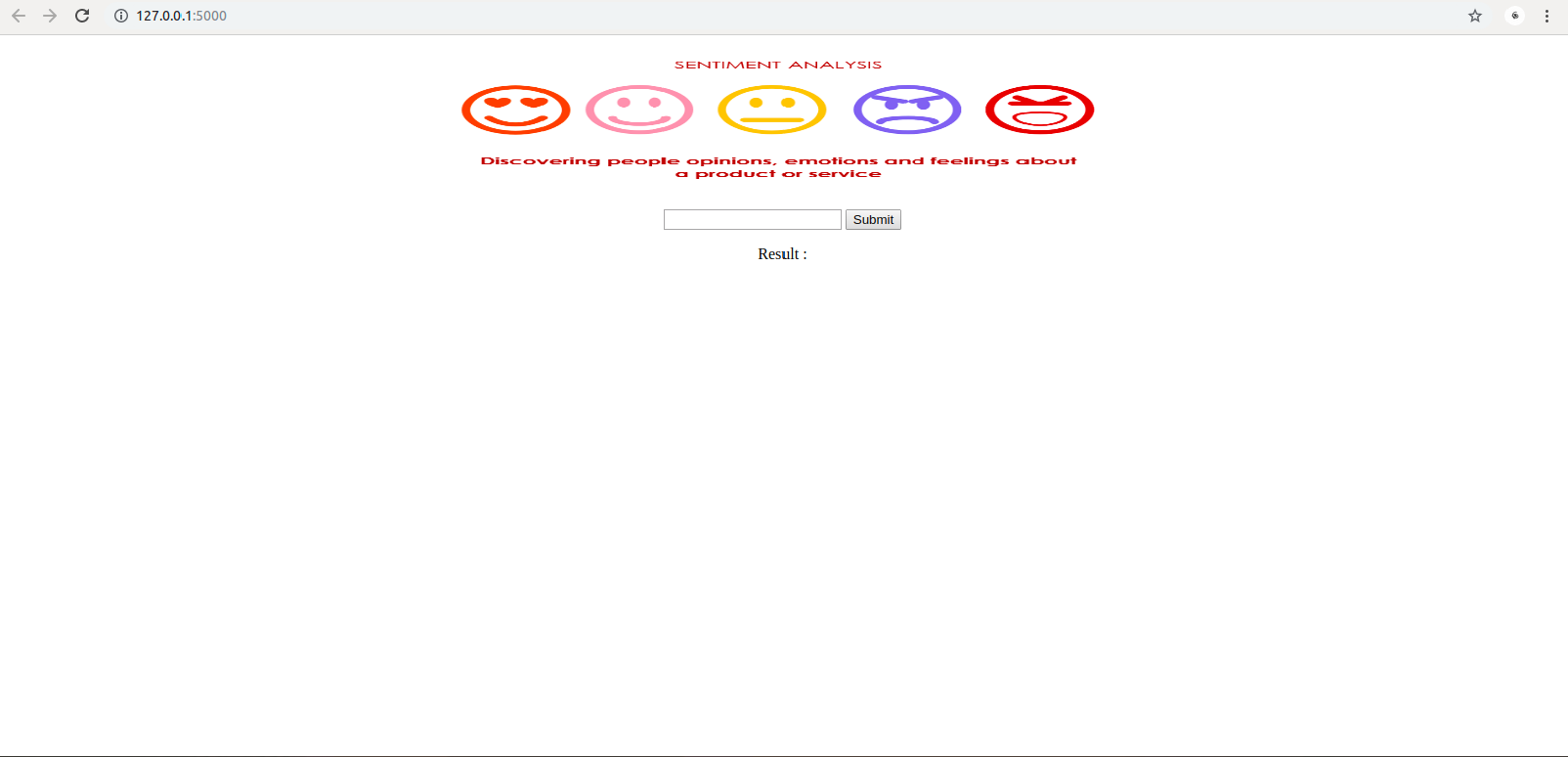
Tạo một package FlaskApp:



File index.html:

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>Sentiment Analysis</**title**>  
</**head**>  
<**body**>  
<**p align="center"**>  
 <**img src="https://media.licdn.com/dms/image/C4D12AQHPAZFZZxBtng/article-cover\_image-shrink\_600\_2000/0?e=1568246400&v=beta&t=p1FJXS-uNne98PGDRmpY9EkARqatJOAqFzSx948KBxM"  
 width="700" height="142"**>  
</**p**>  
<**div align="center"**>  
  
  
 <**form action="/app" method="post" id="form1"**>  
 <**input type="text" size="80" id="keyword"**>  
 <**button type="submit"**>Submit</**button**>  
 </**form**>  
  
 <**p id="result"**>Result : </**p**>  
</**div**>  
</**body**>  
<**script  
 src="https://code.jquery.com/jquery-3.4.1.min.js"  
 integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo="  
 crossorigin="anonymous"**></**script**>  
<**script**>  
 $("#form1").submit(function(event){  
 $.ajax({  
 url: '/app',  
 data: {keyword : $('#keyword').val()},  
 type: 'POST',  
 dataType: "text",  
 success: function(response){  
 $('#result').text('Result: ' + response);  
 },  
 error: function (error) {  
 alert(error.data);  
 }  
 });  
 event.preventDefault();  
 });  
  
  
</**script**>  
</**html**>

Bố cục trang web sẽ như sau :



- file app.py :

**from** flask **import** Flask, render\_template, request, redirect, url\_for, views  
**import** re  
**import** Test2  
  
app = Flask(\_\_name\_\_)  
  
@app.route(**"/"**, methods=[**'GET'**])  
**def** test():  
 **return** render\_template(**'index.html'**)  
  
  
@app.route(**"/app"**, methods=[**'POST'**])  
**def** appreciate():

keyword = request.form[**'keyword'**]  
*#xử lý ngoại lệ*match = re.search(**r'(.\*) không thể không thích'**, keyword)  
**if** match != **None**:  
 result = **'positive'  
else**:  
 result = str(Test2.testComment(keyword))  
**return** result

**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 app.run()

# PHẦN III : KẾT QUẢ ĐẠT ĐƯỢC

-“Tôi rất thích sản phẩm này”



- “Sản phẩm này rất tệ”



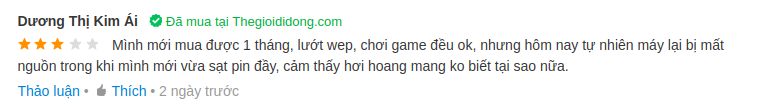
- “Sản phẩm này cũng tạm ổn”

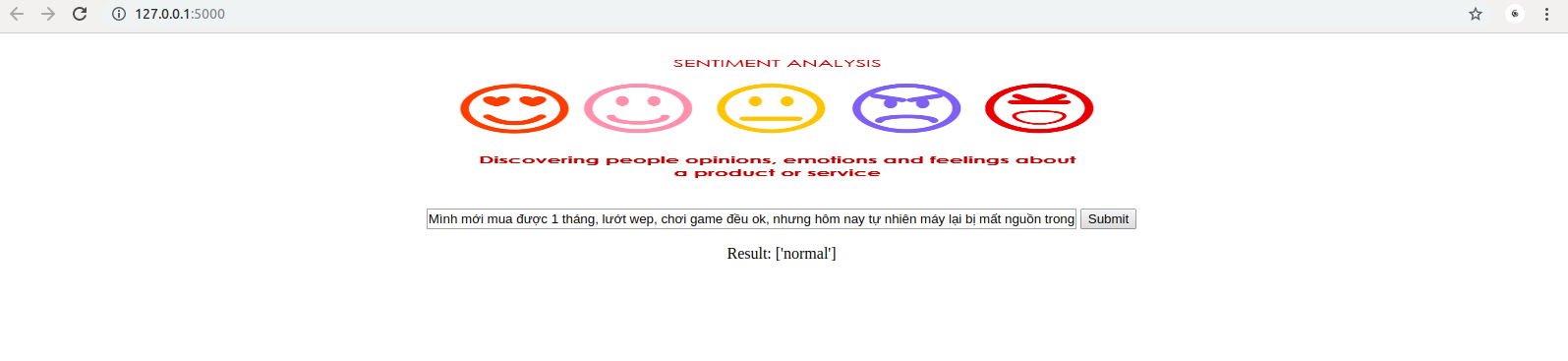


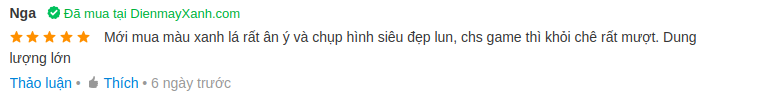
- “Tôi không thích màn hình, nhưng tôi ***không thể không thích*** chiếc điện thoại này”

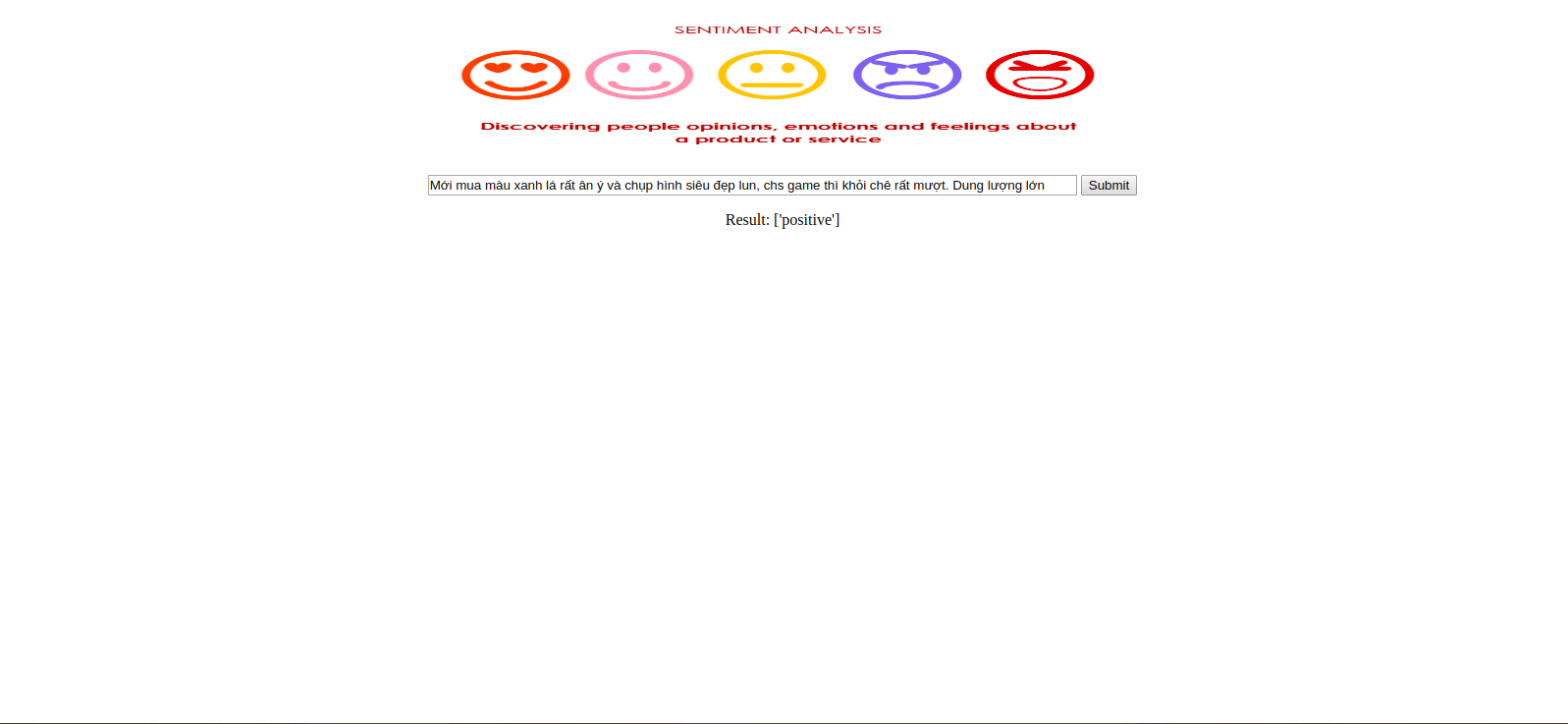


- Một vài bình luận trên trạng Thegioididong.com









# PHẦN IV : PHỤ LỤC

**\* Các tài liệu tham khảo:**

1. [https://vietjack.com](https://vietjack.com/)
2. [http://stackoverflow.com](http://stackoverflow.com/)
3. <https://github.com/>
4. <https://www.tutorialspoint.com/python/python_reg_expressions>
5. <https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python>
6. <https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34>
7. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
8. <https://www.w3schools.com/html/>
9. Sách : Python Cơ Bản ( tác giả: Võ Duy Tuấn )