TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO HỆ THỐNG THỜI GIAN THỰC**

ĐỀ TÀI DỰ ĐOÁN PHIM

*Giảng Viên Hướng Dẫn*: TS TRẦN THẾ VŨ

SINH VIÊN THỰC HIỆN:

* TRƯƠNG VĂN KIÊN – 102140075 – 14T2
* TRẦN ĐỨC LONG – 102140078 – 14T2

Đà Nẵng, 11/2018

**Ý KIẾN ĐÁNH GIÁ CỦA GIẢNG VIÊN**

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

……………………………………………………………………………

MỤC LỤC

|  |  |  |
| --- | --- | --- |
| **DANH SÁCH HÌNH ẢNH** | | |
| Hình |  | Trang |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |

# **LỜI NÓI ĐẦU**

Có thể nói Công Nghệ Thông Tin ở nước ta đang có những bước chuyển mình đáng kinh ngạc. Ngày nay Công Nghệ Thông Tin đang đóng vai trò quan trọng thúc đẩy sự phát triển chung của cả nước. Trước xu hướng đó nhu cầu về việc tạo ra nguồn nhân lực Công Nghệ Thông Tin chất lượng cao ở nước ta ngày càng lớn, đặc biệt là trong thời đại công nghệ 4.0 hiện nay. Để làm được điều này các trường Đại Học ở nước ta đã đưa vào giảng dạy rất nhiều học phần rất cần thiết và thiết thực nhằm củng cố và nâng cao kiến thức cho sinh viên về những lĩnh vực đang nổi trổi như AI, BigData, Internet Of Things... Và dự án “Dự Đoán Phim” trong học phần Hệ Thống Thời Gian Thực là một trong những học phần như vậy. Dự án đã giúp rất nhiều sinh viên có cơ hội tìm hiểu và phát triển về lĩnh vực AI, Data Processing và ứng dụng vào bài toán thực tiễn trong cuộc sống. Có thể nói qua dự án, chúng em đã học được rất nhiều kiến thức bổ ích.

Chúng em xin được bày tỏ lòng biết ơn chân thành tới thầy giáo TS. Trần Thế Vũ - giảng viên khoa CNTT trường ĐH Bách Khoa Đà Nẵng, người thầy đã trực tiếp hướng dẫn và tận tình giúp đỡ, chỉ bảo chúng trong xuốt thời gian thực hiện dự tài. Cảm ơn Thầy đã tập tình hướng dẫn giúp đỡ chúng em trong suốt quá trình thực hiện đề tài.

Cuối cùng trong quá trình thực hiện và làm đề tài chắc chắn chúng em đã mắc không ít những thiết sót và kết quả chưa hoàn hảo, mong Thầy có thể thông cảm!

Chúng em xin chân thành cảm ơn!

# **PHẦN I: CƠ SỞ LÝ THUYẾT**

## **1/ Dự án Dự Đoán Phim**

### **1.1/ Đặt vấn đề**

Với đời sống ngày càng phát triển như hiện nay, việc tận hưởng cuộc sống và sử dụng các dịch vụ giải trí như đi xem phim, đi ăn uống… đang rất được ưa chuộng trong cuộc sống hiện tại.

Tuy nhiên, đối với dịch vụ xem phim giải trí, nhận ra ngành dịch vụ này đang được rất nhiều người ưa chuộng nên có rất nhiều phim được sản xuất ra để đáp ứng. Nhưng tỉ lệ phim không hay, phim chỉ mang tính thương mại với nội dung kém đang khiến cho người dùng bỏ tiền để xem phim nhưng nhận lại nhiều thước phim không đặc sắc và bỏ phí thời gian để có những phút giây không thoải mái.

Nhận ra được điều đó, chúng em làm đề tài dự đoán phim để cho ra được cái nhìn tổng quan về phim sắp chiếu sẽ được dự đoán điểm bao nhiêu dựa trên nhà sản xuất phim và diễn viên. Từ đó có thể cho người dùng lựa chọn những bộ phim tốt nhất và có nội dung hay để thưởng thức những giờ phút xem phim thật hay và ý nghĩa.

### **1.2/ Yêu cầu dự án:**

Với đề tài dự đoán phim, ta có một số yêu cầu của dự án như sau:

1. Cần số lượng dữ liệu phim lớn
2. Cần có một thuật toán tốt để đào tạo học máy
3. Cần có các công cụ và thư viện hỗ trợ chúng ta tốt khi làm dự án

### **1.3/ Giải pháp:**

Với những yêu cầu như trên thì ta sẽ có hướng đi như sau:

1. Dữ liệu sẽ được lấy từ trang web: [1]
2. Dùng thư viện ‘urllib.request’ để truy cập với địa chỉ website
3. Dùng thư viện ‘bs4 - BeautifulSoup’ để parse file ‘html’ để truy cập dữ liệu
4. Dùng thư viện ‘csv’ để lưu dữ liệu xuống file csv
5. Dùng FireBase để lưu trữ dữ liệu
6. Dùng thư viện Tensorflow và thuật toán LinearRegression để train dữ liệu có sẵn và dự đoán kết quả dựa trên dữ liệu đã train
7. Tất cả thông tin phim sẽ được hiển trị trên ứng dụng Android để người dùng có thể xem trực quan

## **2/ Thuật Toán LinearRegression (Hồi quy tuyến tính)**

## **2.1/ Giới Thiệu:**

"Hồi quy tuyến tính" là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví dụ, dự đoán giao thông ở một cửa hàng bán lẻ, dự đoán thời gian người dùng dừng lại một trang nào đó hoặc số trang đã truy cập vào một website nào đó v.v...

## **2.2/ Phân Tích Toán Học**

### **2.2.1/ Dạng của Linear Regression**

y≈f(x)=^y

y≈f(x)=y^f(x)=w1x1+w2x2+w3x3+w0    (1)

Trong phương trình ( 1 ) phía trên, nếu chúng ta đặt w = [ w 0 , w 1 , w 2 , w 3 ] T = là vector (cột) hệ số cần phải tối ưu và ¯ x = [ 1 , x 1 , x 2 , x 3 ] (đọc là x bar trong tiếng Anh) là vector (hàng) dữ liệu đầu vào mở rộng. Số 1 ở đầu được thêm vào để phép tính đơn giản hơn và thuận tiện cho việc tính toán. Khi đó, phương trình (1) có thể được viết lại dưới dạng:

y ≈ ¯ x w = ^ y

+Với ¯ x là một vector hàng.

### **2.2.2/ Sai số dự đoán**

Chúng ta mong muốn rằng sự sai khác e giữa giá trị thực y và giá trị dự đoán ^ y (đọc là y hat trong tiếng Anh) là nhỏ nhất. Nói cách khác, chúng ta muốn giá trị sau đây càng nhỏ càng tốt:

1/2 \* e^2 = 1/2 ( y – ^y ) = 1/2 ( y − ¯ x \* w )^2

+ trong đó hệ số 1 2 (lại) là để thuận tiện cho việc tính toán (khi tính đạo hàm thì số 1 2 sẽ bị triệt tiêu). Chúng ta cần e 2 vì e = y − ^ y có thể là một số âm, việc nói e nhỏ nhất sẽ không đúng vì khi e = − ∞ là rất nhỏ nhưng sự sai lệch là rất lớn

### **2.2.3/ Hàm mất mát**

Điều tương tự xảy ra với tất cả các cặp (input, outcome) ( x i , y i ) , i = 1 , 2 , … , N , với N là số lượng dữ liệu quan sát được. Điều chúng ta muốn, tổng sai số là nhỏ nhất, tương đương với việc tìm w để hàm số sau đạt giá trị nhỏ nhất:

L ( w ) = 1 /2 \* ∑(N)(i = 1) ( yi − ¯ xi \* w )^2 ( 2 )

Hàm số L ( w ) được gọi là hàm mất mát (loss function) của bài toán Linear Regression. Chúng ta luôn mong muốn rằng sự mất mát (sai số) là nhỏ nhất, điều đó đồng nghĩa với việc tìm vector hệ số w sao cho giá trị của hàm mất mát này càng nhỏ càng tốt. Giá trị của w làm cho hàm mất mát đạt giá trị nhỏ nhất được gọi là điểm tối ưu (optimal point), ký hiệu: w ∗ = arg min w L ( w )

Trước khi đi tìm lời giải, chúng ta đơn giản hóa phép toán trong phương trình hàm mất mát ( 2 ) . Đặt y = [ y 1 ; y 2 ; … ; y N ] là một vector cột chứa tất cả các output của training data; ¯ X = [ ¯ x 1 ; ¯ x 2 ; … ; ¯ x N ] là ma trận dữ liệu đầu vào (mở rộng) mà mỗi hàng của nó là một điểm dữ liệu. Khi đó hàm số mất mát L ( w ) được viết dưới dạng ma trận đơn giản hơn:

L ( w ) = 1/2 \* ∑(N)(i= 1) ( yi − ¯ xi \* w )^2

= 1/2 ∥ y − ¯ X\* w ∥ ( 3 )

với ∥ z ∥^2 là Euclidean norm (chuẩn Euclid, hay khoảng cách Euclid), nói cách khác ∥ z ∥ 2 2 là tổng của bình phương mỗi phần tử của vector z . Tới đây, ta đã có một dạng đơn giản của hàm mất mát được viết như phương trình ( 3 ) . 2.4. Nghiệm cho bài toán Linear Regression Cách phổ biến nhất để tìm nghiệm cho một bài toán tối ưu (chúng ta đã biết từ khi học cấp 3) là giải phương trình đạo hàm (gradient) bằng 0! Tất nhiên đó là khi việc tính đạo hàm và việc giải phương trình đạo hàm bằng 0 không quá phức tạp. Thật may mắn, với các mô hình tuyến tính, hai việc này là khả thi. Đạo hàm theo w của hàm mất mát là:

∂\*L(w) / ∂\*w = ¯X^T ( ¯X \* w − y )

Phương trình đạo hàm bằng 0 tương đương với:

¯X^T \* ¯X \* w = ¯X^T \* y ≜ b (4)

(ký hiệu ¯ X T y ≜ b nghĩa là đặt ¯ X T y bằng b ).

Nếu ma trận vuông A ≜ ¯ X T ¯ X khả nghịch (non-singular hay invertible) thì phương trình ( 4 ) có nghiệm duy nhất: w = A − 1 b . Vậy nếu ma trận A không khả nghịch (có định thức bằng 0) thì sao? Nếu các bạn vẫn nhớ các kiến thức về hệ phương trình tuyến tính, trong trường hợp này thì hoặc phương trinh ( 4 ) vô nghiệm, hoặc là nó có vô số nghiệm. Khi đó, chúng ta sử dụng khái niệm giả nghịch đảo A † (đọc là A dagger trong tiếng Anh). (Giả nghịch đảo (pseudo inverse) là trường hợp tổng quát của nghịch đảo khi ma trận không khả nghịch hoặc thậm chí không vuông.

# **II: PHÂN TÍCH VÀ THỰC HIỆN**

## **1/ Sơ đồ khối hệ thống**

Với những cơ sở lý thuyết của hệ thống, chúng em đã phân tích để cho ra sơ đồ khối của hệ thống như sau:

Thu thập dữ liệu từ train web

Phân loại dữ liệu train (phim đang chiếu), dữ liệu test (phim sắp chiếu)

Biến đổi dữ liệu thành dữ liệu có thể train được

Tiến hành training

Tiến hành dự đoán điểm phim bằng dữ liệu phim sắp chiếu

Cho ra điểm phim dự đoán được

Đưa tất cả dữ liệu trên Firebase

Xuất kết quả ra ứng dụng android

## **3/ Các bước thực hiện**

## **3.1/ Thu nhập dữ liệu từ trang web**

Truy cập vào trang web : [1]

Ta sẽ lấy dữ liệu theo 2 tiêu chí: Phim Đang Chiếu và Phim Sắp Chiếu

Đầu tiên, ta phải đọc được liên kết của trang web và parse nó ra data

import bs4

from urllib.request import urlopen

import csv

import dateutil.parser as dparser

def readUrl(url):

web\_data = urlopen(url)

html = web\_data.read()

web\_data.close()

data = bs4.BeautifulSoup(html, 'html.parser')

return data

url = "https://moveek.com/phim-viet-nam/"

soup = readUrl(url)

results = soup.findAll('div','panel panel-post panel-movie-grid')

Đối với phim đang chiếu, ta sẽ lấy các thuộc tính của phim: ‘Tên Phim’, ‘Điểm IMDB’, ‘Nhà sản xuất’, ‘Thông Tin Phim’, ‘Diễn Viên’, ‘Thời lượng’, ‘Poster’, ‘Ngày’ và ‘Trailer’:

rows = [];

rows.append(['Film', 'Imdb','Director','Description','Star', 'Time', 'Poster', 'Date', 'Trailer'])

**for** result **in** results:

data = result.find('a')

*#Get link film*

link = 'https://moveek.com'+data.get('href')

*#Get data from link*

soup = readUrl(link)

resultFilms = soup.find('div','movie-header-wrapper')

*#1- Get Namme Film*

srcName = resultFilms.find('img')

name = srcName.get('alt')

*#2 get poster*

srcPoster = resultFilms.find('img')

poster = srcPoster.get('data-srcset')

**if** poster == **None**:

linkPoster = ''

**else**:

poster = poster.split(", ")

linkPoster = poster[1].replace("2x","")

*#2 - Get percents*

percent = resultFilms.find('div','action chart-pie chart-pie-small pull-left')

**if**(percent == **None**) :

percent = '5'

**else** :

percent = percent.getText().replace("%","")

percent = int(percent)/10

*#get trailer*

trailer = resultFilms.find('a', 'action action-trailer btn btn-rounded btn-lg btn-default')

**if** trailer **is** **None**:

trailer = ""

**else**:

trailer = 'https://moveek.com'+trailer.get('href')

trailer = readUrl(trailer)

trailer = trailer.find('div', 'js-video youtube widescreen')

trailer = trailer.find('iframe')

trailer = trailer.get('src')

trailer = trailer.replace("//", "")

print(trailer)

*# get date*

srcDate = resultFilms.findAll('p','text-white')

date = srcDate[-1].getText().split("**\n**")[1].strip()

date = dparser.parse(date,fuzzy=**True**)

date = str(date.date())

*#3 - Get director*

srcDirector = resultFilms.findAll('p','text-white')

director = srcDirector[len(srcDirector)-1].find('a');

**if**(director == **None**):

director = ''

**else**:

director = director.getText()

*#4 - Get description*

description = soup.find('div','synopsis m-t m-b-xs')

**if**(description == **None**):

description=''

**else** :

description = description.getText()

*#5 Get Time*

time = resultFilms.find('a','action action-imdb btn btn-rounded btn-lg btn-icon btn-default')

**if**(time == **None**):

time = '100phút'

**else** :

time = time.getText()

**if** "ph" **not** **in** time:

time = '100phút'

*#6 - Get Actors*

resultActorFilms = soup.find('div','movie-actors panel wrapper m-t')

listActor=[];

**if**(resultActorFilms == **None**) :

listActor= 'No actor'

**else** :

actors = resultActorFilms.findAll('img')

**for** actor **in** actors:

listActor.append(actor.get('alt'))

rows.append([name,percent, director,''+description.strip(),listActor,time, linkPoster, date, trailer])

- Đối với phim sắp chiếu, ta sẽ lấy các thuộc tính của phim: ‘Tên Phim’, ‘Nhà sản xuất’, ‘Thông Tin Phim’, ‘Diễn Viên’, ‘Ngày Khởi Chiếu, ‘Poster’.

rows = [];

rows.append(['Film','Director','Description','Date Create', 'Star', 'Poster'])

**for** result **in** results:

data = result.find('a')

*#Get link film*

link = 'https://moveek.com'+data.get('href')

*#Get data from link*

soup = readUrl(link)

resultFilms = soup.find('div','movie-header-wrapper')

*#1- Get Namme Film*

srcName = resultFilms.find('img')

name = srcName.get('alt')

*#2 get poster*

srcPoster = resultFilms.find('img')

poster = srcPoster.get('data-srcset')

**if** poster == **None**:

linkPoster = 'https://i.moveek.com/bundles/ornweb/img/no-poster.png?v=7206'

**else**:

poster = poster.split(", ")

linkPoster = poster[1].replace("2x","")

*#2 - Get director*

srcDirector = resultFilms.findAll('p','text-white')

director = srcDirector[len(srcDirector)-1].find('a');

**if**(director == **None**):

director = ''

**else**:

director = director.getText()

*#3 - Get Description*

description = soup.find('div','synopsis m-t m-b-xs')

**if**(description == **None**):

description = ''

**else**:

description = description.getText()

*#4 - Get Date Create*

srcDate = resultFilms.findAll('p','text-white')

date = srcDate[-1].getText().split("**\n**")[1].strip()

date = date.replace("Khởi chiếu: ", "").replace(" tại Việt Nam", "")

print(date)

*#5 - Get Actors*

soup3 = readUrl(link)

resultActorFilms = soup.find('div','movie-actors panel wrapper m-t')

listActor=[];

**if**(resultActorFilms == **None**) :

listActor='No Actor';

**else** :

actors = resultActorFilms.findAll('img')

**for** actor **in** actors:

listActor.append(actor.get('alt'))

rows.append([name,director,''+description.strip(),date,listActor, linkPoster])

- Sau đó ta dùng thư viện ‘csv’ để lưu dữ liệu xuống file csv:

**with** open('FullInfoFilmTrain.csv','w', newline='',encoding="utf-8-sig") **as** f\_output:

csv\_output = csv.writer(f\_output)

csv\_output.writerows(rows)

## **3.2/ Xử lý dữ liệu**

- Ta sẽ lấy 3 thuộc tính của phim đang chiếu ra để làm feature cho dữ liệu train: ‘Imdb’, ‘Nhà sản xuất’ và ‘Diễn viên’

directors = []

actors = []

point = []

for row in rows[1:]:

directors.append(row[1])

actors.append(row[2])

point.append(row[0])

- Sau đó, ta chuyển dữ liệu về chữ thường để tiện cho việc train và sắp xếp dữ liệu cũng như bỏ những dữ liệu bị trùng đi

directors = sorted(list(set(directors)))

actors = sorted(list(set(actors)))

point = sorted(list(set(point)))

## **3.3/ Train dữ liệu**

## **3.4/ Đưa dữ liệu lên Firebase**

Sau khi ta đã có tất cả dữ liệu, ta kết nối với module firebase:

from firebase import firebase

from google.cloud import storage

import pyrebase

import pandas as pd

import numpy as np

Tiếp theo, khởi tạo firebase với các thông tin cấu hình:

config = {

"apiKey": "AIzaSyCaPLWEXT3iXLwixzt8RjnEfJUAp7GSp88",

"authDomain": "mooview-b78bd.firebaseapp.com",

"databaseURL": "https://mooview-b78bd.firebaseio.com",

"projectId": "mooview-b78bd",

"storageBucket": "mooview-b78bd.appspot.com",

"messagingSenderId": "614552655393",

"serviceAccount": "service.json"

}

firebase = pyrebase.initialize\_app(config)

database = firebase.database()

Cuối cùng, ta đưa kết quả lên firebase với dữ liệu đọc từ file CSV mà ta đã lưu trước đó:

data = pd.read\_csv("FullInfoFilmTrain.csv", encoding='utf-8-sig')

name = data['Film'].tolist()

name = np.asarray(name)

director = data['Director'].tolist()

director = np.asarray(director)

actors = data['Star'].tolist()

actors = np.asarray(actors)

des = data['Description'].tolist()

des = np.asarray(des)

imdb = data['Imdb'].tolist()

imdb = np.asarray(imdb)

time = data['Time'].tolist()

time = np.asarray(time)

poster = data['Poster'].tolist()

poster = np.asarray(poster)

date = data['Date'].tolist()

date = np.asarray(date)

new\_actors = []

for actor in actors:

list\_actor = actor.replace("[", "").replace("]", "").replace("'", "")

new\_actors.append(list\_actor)

for i in range(len(name)):

autoID = database.child("CurrentFilm").generate\_key()

info = {

"name": name[i],

"director": director[i],

"actor": new\_actors[i],

"description": des[i],

"time": time[i],

"point": imdb[i],

"poster": poster[i],

"date": date[i]

}

database.child(autoID).set(info)

## **3.5/ Hiển thị kết quả lên ứng dụng android**

< ảnh >

# **PHẦN III: KẾT QUẢ VÀ ĐÁNH GIÁ**

## **1/ Kết quả:**

Ảnh

## **2/ Đánh giá:**

- Theo như kết quả thu được thì bài toán đưa ra kết quả đúng khoảng ? %.

- Các nguyên nhân gây ra việc đếm không chính xác:

- Hạn chế của bài toán:

# **TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | M. COMPANY, “Phim Việt Nam,” 11 11 2018. [Trực tuyến]. Available: https://moveek.com/phim-viet-nam/. |