

Website dự đoán doanh số bán hàng

1. Giới thiệu

- Đây là một bản demo giúp các bạn có thể làm và hình dung việc sử dụng Machine Learning để dự đoán doanh số bán hàng trong tháng thứ ba thông qua lãi suất và doanh số bán hàng của hai tháng trước đó.
- Trong mô hình demo này ta sẽ sử dụng phương pháp hồi quy tuyến tính (*Linear Regression*) để dự đoán.
- Về phần Website, ta sẽ dùng Flask để xây dựng.
- Mình sẽ sử dụng sublime để code mô hình này.

2. Hồi quy tuyến tính (*Linear Regression*):

- Hồi quy tuyến tính được xây dựng với mục đích tìm ra mối quan hệ giữa một hay nhiều đặc trưng của biến đầu vào và biến mục tiêu liên tục *y* (*continuous target variable*)
- Ta có thể biểu diễn về hồi quy tuyến tính dưới phương trình sau.

$$Y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

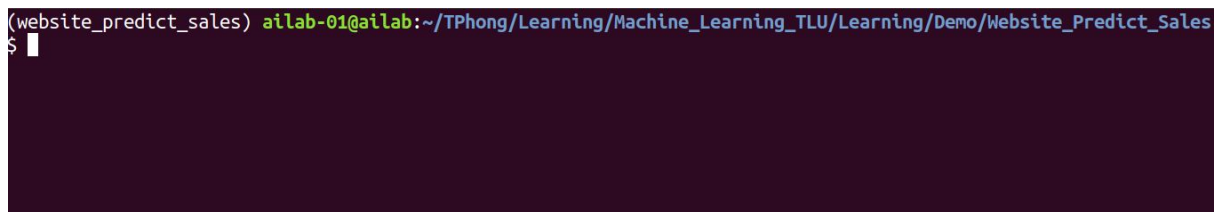
- + Y : Giá trị dự đoán
- + θ_0 : Độ lệch, biểu thị sự chênh lệch giữa giá trị trung bình mà mô hình dự đoán và giá trị thực tế của dữ liệu (*bias*)
- + $\theta_1 .. \theta_n$: thông số để mô hình tính toán ra Y (*model parameters*)
- + $x_1 .. x_n$: đầu vào của bài toán, thường là các đặc trưng của dữ liệu có tác động đến bài toán như độ ẩm, nhiệt độ, v.v.. (*feature values*)

3. Flask:

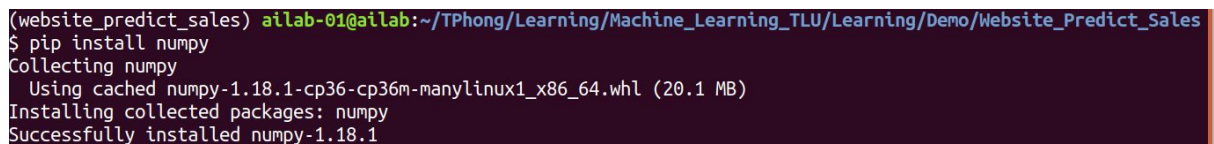
- Flask là một micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.
- Lý do sử dụng Flask (Tham khảo):
 - Dễ dàng sử dụng
 - Tích hợp hỗ trợ khả năng testing
 - Điều phối tốt yêu cầu RESTful
 - Tài liệu hướng dẫn hỗ trợ mở
 - Dễ dàng trong việc xây dựng sever và sửa lỗi.
- Một số link bạn có thể tham khảo về Flask sẽ được gắn ở dưới phần nguồn tham khảo.

4. Cài đặt:

- Về môi trường ta có thể sử dụng môi trường tại máy và cài đặt các thư viện dưới đây. Ngoài ra bạn cũng có thể cách tạo môi trường khác để làm việc (link hướng dẫn: <https://bom.to/B1tRdY>)
 - numpy
 - pandas
 - scikit-learn
 - flask
- Cách cài đặt: pip install tên thư viện cần tải



Hình 1. Mình đang sử dụng môi trường mới tạo website_predict_sales

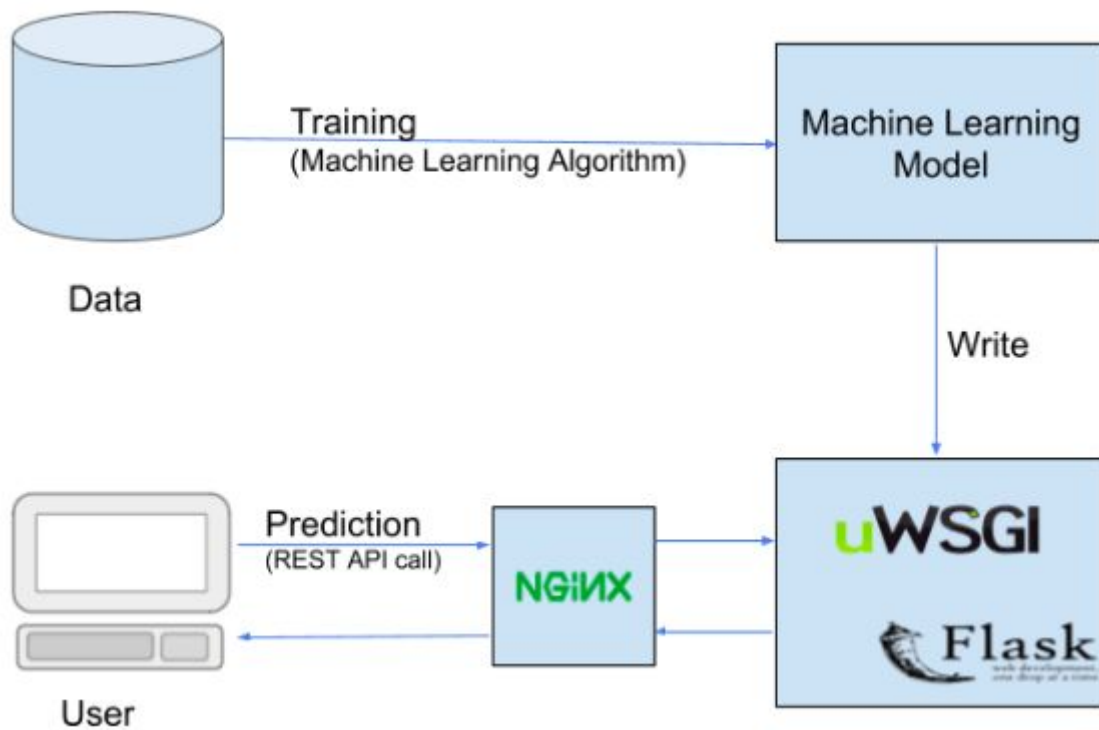


Hình 2. Ví dụ về cài đặt thư viện numpy

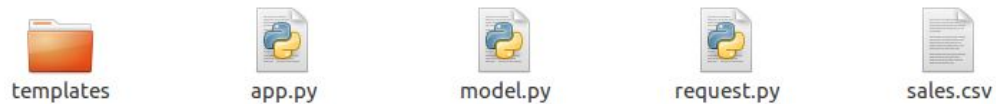
```
(website_predict_sales) aillab-01@aillab:~/TPhong/Learning/Machine_Learning_TLU/Learning/Demo/Website_Predict_Sales$ conda list
# packages in environment at /home/aillab-01/anaconda3/envs/website_predict_sales:
#
# Name                    Version            Build                Channel
#-----
_libgcc_mutex             0.1                main
ca-certificates           2020.1.1           0
certifi                   2019.11.28         py36_0
click                     7.0                pypi_0              pypi
flask                     1.1.1              pypi_0              pypi
itsdangerous              1.1.0              pypi_0              pypi
jinja2                    2.11.1             pypi_0              pypi
joblib                    0.14.1             pypi_0              pypi
ld_impl_linux-64          2.33.1             h53a641e_7
libedit                   3.1.20181209       hc058e9b_0
libffi                    3.2.1              hd88cf55_4
libgcc-ng                 9.1.0              hdf63c60_0
libstdcxx-ng              9.1.0              hdf63c60_0
markupsafe                 1.1.1              pypi_0              pypi
ncurses                    6.2                he6710b0_0
numpy                     1.18.1             pypi_0              pypi
openssl                   1.1.1d             h7b6447c_4
pandas                    1.0.1              pypi_0              pypi
pip                       20.0.2             py36_1
python                    3.6.10             h0371630_0
python-dateutil           2.8.1              pypi_0              pypi
pytz                      2019.3             pypi_0              pypi
readline                  7.0                h7b6447c_5
scikit-learn              0.22.2             pypi_0              pypi
scipy                     1.4.1              pypi_0              pypi
setuptools                45.2.0             py36_0
six                       1.14.0             pypi_0              pypi
sqlite                    3.31.1             h7b6447c_0
tk                         8.6.8              hbc83047_0
werkzeug                  1.0.0              pypi_0              pypi
wheel                     0.34.2             py36_0
```

Hình 3. Hiện thị các packages đã cài đặt ở thư viện

- Cấu trúc dự án gồm bốn phần:
 - **model.py** : Chứa code về mô hình machine learning để luyện tập và dự đoán doanh số bán hàng trong tháng thứ ba dựa trên hai tháng trước đó.
 - **app.py** : Chứa flask APIs nhận thông tin chi tiết bán hàng thông qua lệnh GUI hoặc API, tính toán giá trị dự đoán dựa trên mô hình ở trên và trả về dự đoán.
 - **request.py** : Sử dụng mô đun yêu cầu gọi APIs được xác định trong file app.py và hiện thị giá trị trả về.
 - **HTML/CSS** : Chứa mẫu HTML và CSS để người dùng nhập chi tiết bán hàng và hiện thị ra doanh số dự đoán của tháng thứ ba (Front End).
 - **data.csv** : Chứa dữ liệu huấn luyện cho mô hình.



Hình 4. Sơ đồ mô tả cách hoạt động của dự án



Hình 5. Các file có trong dự án



Hình 6. file trong tệp tin templates

5. Lập trình:

- Mình sẽ lướt qua các file có trong dự án và sẽ chỉ tập giải thích về file model.py để huấn luyện dữ liệu.
- **index.html**
 - Phần code trong file HTML các bạn có thể thiết kế lại để có giao diện đẹp và tốt hơn. Ở đây mình sẽ chỉ xây dựng một giao diện đơn giản.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Deployment Tutorial 1</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body style="background: #FFF;">
  <div class="login">
    <h1>Website Predict Sales</h1>

    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict') }}" method="post">
      <input type="text" name="rate" placeholder="tỉ lệ" required="required" />
      <input type="text" name="sales in first month" placeholder="Doanh số tháng thứ nhất" required="required" />
      <input type="text" name="sales in second month" placeholder="Doanh số tháng thứ hai" required="required" />
      <button type="submit" class="btn btn-primary btn-block btn-large">Predict sales in third month</button>
    </form>

    <br>
    <br>
    {{ prediction_text }}
  </div>
</body>
</html>
```

Hình 7. file code index.html

- **style.css**

- Phần code hỗ trợ xây dựng giao diện trên css

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);

html { width: 100%; height:100%; overflow:hidden; }

body {
  width: 100%;
  height:100%;
  font-family: 'Helvetica';
  background: #000;
  color: #fff;
  font-size: 24px;
  text-align:center;
  letter-spacing:1.4px;
}

.login {
  position: absolute;
  top: 40%;
  left: 50%;
  margin: -150px 0 0 -150px;
  width:400px;
  height:400px;
}

.login h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-align:center; }

input {
  width: 100%;
  margin-bottom: 10px;
  background: rgba(0,0,0,0.3);
  border: none;
  outline: none;
  padding: 10px;
  font-size: 13px;
  color: #fff;
  text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
  border: 1px solid rgba(0,0,0,0.3);
  border-radius: 4px;
  box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
  -webkit-transition: box-shadow .5s ease;
  -moz-transition: box-shadow .5s ease;
  -o-transition: box-shadow .5s ease;
  -ms-transition: box-shadow .5s ease;
  transition: box-shadow .5s ease;
}

}
```

Hình 8. file code style.css

- **request.py**

- file request sẽ gửi yêu cầu dự đoán từ trên website về sever để tính toán.

```
import requests

url = 'http://localhost:5000/results'
r = requests.post(url,json={'rate':5, 'sales_in_first_month':200, 'sales_in_second_month':400})

print(r.json())
```

Hình 9. file code request.py

- **app.py**

- Tạo ra một sever để xử lý các thông tin được gửi từ file request. Các thông tin được gửi từ file request sẽ được tính toán dựa trên kết quả huấn luyện mô hình dữ liệu từ file model.py.

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__, template_folder='templates')
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():

    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Sales should be $ {}'.format(output))

@app.route('/results', methods=['POST'])
def results():

    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

if __name__ == "__main__":
    app.run(debug=True)
```

Hình 10. file code app.py

- **model.py**

- Đầu tiên ta cần import các thư viện sử dụng trong dự án.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle
```

Hình 11. Khai báo thư viện

- Đọc dữ liệu từ file data.csv thông qua pandas

```
dataset = pd.read_csv('sales.csv')
dataset['rate'].fillna(0, inplace=True)
dataset['sales_in_first_month'].fillna(dataset['sales_in_first_month'].mean(), inplace=True)
```

Hình 12. Đọc dữ liệu từ data.csv

- Phân chia dữ liệu sau khi đọc từ file dữ liệu vào các biến toàn cục để huấn luyện mô hình

```
X = dataset.iloc[:, :3]
def convert_to_int(word):
    word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
    return word_dict[word]
X['rate'] = X['rate'].apply(lambda x : convert_to_int(x))
y = dataset.iloc[:, -1]
```

Hình 13. Phân chia dữ liệu vào

- Khai báo mô hình hồi quy tuyến tính và huấn luyện mô hình với các biến được phân chia từ dữ liệu

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X, y)
```

Hình 14. Huấn luyện mô hình

- Ghi mô hình đã huấn luyện ra file model.pkl. Khi chạy sever ta sẽ load kết quả này vào để dự đoán

```
pickle.dump(regressor, open('model.pkl', 'wb'))
```

Hình 15. Huấn luyện mô hình

```

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle

dataset = pd.read_csv('sales.csv')

dataset['rate'].fillna(0, inplace=True)

dataset['sales_in_first_month'].fillna(dataset['sales_in_first_month'].mean(), inplace=True)

X = dataset.iloc[:, :3]

def convert_to_int(word):
    word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
    return word_dict[word]

X['rate'] = X['rate'].apply(lambda x : convert_to_int(x))

y = dataset.iloc[:, -1]

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

regressor.fit(X, y)

pickle.dump(regressor, open('model.pkl', 'wb'))

```

Hình 16. Tổng thể toàn bài code

- Các bạn có thể down data và code của dự án bên dưới:
<https://bom.to/V55fXd>

6. Chạy mô hình

- Chạy file code model.py để ta tạo ra model dự đoán

python model.py

- Kiểm tra xem trong thư mục biên dịch file app.py đã có model.pkl rồi tiếp tục biên dịch file app.py. Khi đó hệ thống đã tạo sever

python app.py

- Truy cập vào đường dẫn tại running on

```

* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 258-374-352

```

d. Kết quả

Website Predict Sales

| | | | |
|-------|-------------------------|------------------------|----------------------|
| tỉ lệ | Doanh số tháng thứ nhất | Doanh số tháng thứ hai | Dự đoán tháng thứ ba |
|-------|-------------------------|------------------------|----------------------|

7. Tổng kết:

- Qua bản demo trên sẽ giúp bạn có cái nhìn gần hơn về học máy với một mô hình đơn giản và ít dữ liệu.
- Trên thực tế để dự đoán bạn cần nhiều thứ tác động đến kết quả dự đoán hơn. Ví dụ như bài toán giá nhà có các yếu tố tác động như diện tích, địa điểm, số căn phòng trong nhà, v.v..
- Nếu các bạn có ý định học về web có thể bắt đầu tìm hiểu và sử dụng Flask.
- Chúc các bạn có thể bước tiếp trên con đường phát triển AI

8. Nguồn tham khảo

- Link tiếng anh:
<https://towardsdatascience.com/how-to-easily-deploy-machine-learning-models-using-flask-b95af8fe34d4>
- Hướng sử dụng các thư viện trong Machine Learning:
 - <https://bom.to/yrqNFX>
- Link tài liệu hướng dẫn Flask:
 - <https://flask.palletsprojects.com/en/1.0.x/tutorial/>
 - <https://pythonspot.com/flask-web-app-with-python/>
 - <https://code.visualstudio.com/docs/python/tutorial-flask>
 - <http://exploreflask.com/en/latest/>
 - <https://realpython.com/search?q=flask>
 - <https://buildmedia.readthedocs.org/media/pdf/flask/latest/flask.pdf>

- <https://buildmedia.readthedocs.org/media/pdf/building-web-applications-with-flask/latest/building-web-applications-with-flask.pdf>
- <https://pythonhow.com/wp-content/uploads/2016/01/Building-a-Python-Flask-Website.pdf>
- <https://github.com/pallets/flask>

Mong được sự đóng góp và góp ý đến từ các bạn
Mọi chi tiết liên hệ: dtphong010199@gmail.com