

REPORT

Project 2 – Triển khai mô hình phân loại bệnh ung thư vú

Thành viên:

Nguyễn Khắc Vỹ - 19127637

Nguyễn Phan Vũ - 19127633

1. Chuẩn bị dữ liệu:

Làm sạch dữ liệu (data cleaning):

Tập dữ liệu không bị trùng lặp sau khi kiểm tra.

Dữ liệu có các dòng bị lặp không?

```
if data.duplicated().any() == True:
    have_duplicated_rows = True
else:
    have_duplicated_rows = False
```

[6] ✓ 0.1s

```
assert have_duplicated_rows == False
```

[7] ✓ 0.1s

- Dữ liệu không có các dòng bị trùng lặp.

Kiểm tra giá trị thiếu trên tất cả các cột:

Chỉ có cột **Unnamed: 32** bị thiếu và thiếu tất cả 569 giá trị vì vậy cần xóa ở bước tiền xử lý.

Open in Notebook Editor

1	id	0
2	diagnosis	0
3	radius_mean	0
4	texture_mean	0
5	perimeter_mean	0
6	area_mean	0
7	smoothness_mean	0
8	compactness_mean	0
9	concavity_mean	0
10	concave points_mean	0
11	symmetry_mean	0
12	fractal_dimension_mean	0
13	radius_se	0
14	texture_se	0
15	perimeter_se	0
16	area_se	0
17	smoothness_se	0
18	compactness_se	0
19	concavity_se	0
20	concave points_se	0
21	symmetry_se	0
22	fractal_dimension_se	0
23	radius_worst	0
24	texture_worst	0
25	perimeter_worst	0
26	area_worst	0
27	smoothness_worst	0
28	compactness_worst	0
29	concavity_worst	0
30	concave points_worst	0
31	symmetry_worst	0
32	fractal_dimension_worst	0
33	Unnamed: 32	569
34	dtype: int64	

Tiền xử lý dữ liệu:

Xóa các cột:

- Cột **id** vì không có tác dụng trong việc phân loại.
- Cột **Unnamed: 32** vì chứa toàn biến rỗng.
- Cột **diagnosis** vì nó là class label, là kết quả đầu ra.

Thực hiện xóa và tách tập dữ liệu:

```

y = data.diagnosis
x = data.drop(['Unnamed: 32', 'id', 'diagnosis'], axis = 1 )
x.head()

```

✓ 0.2s

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809

5 rows × 30 columns

Xử lý dữ liệu categorical, văn bản...?

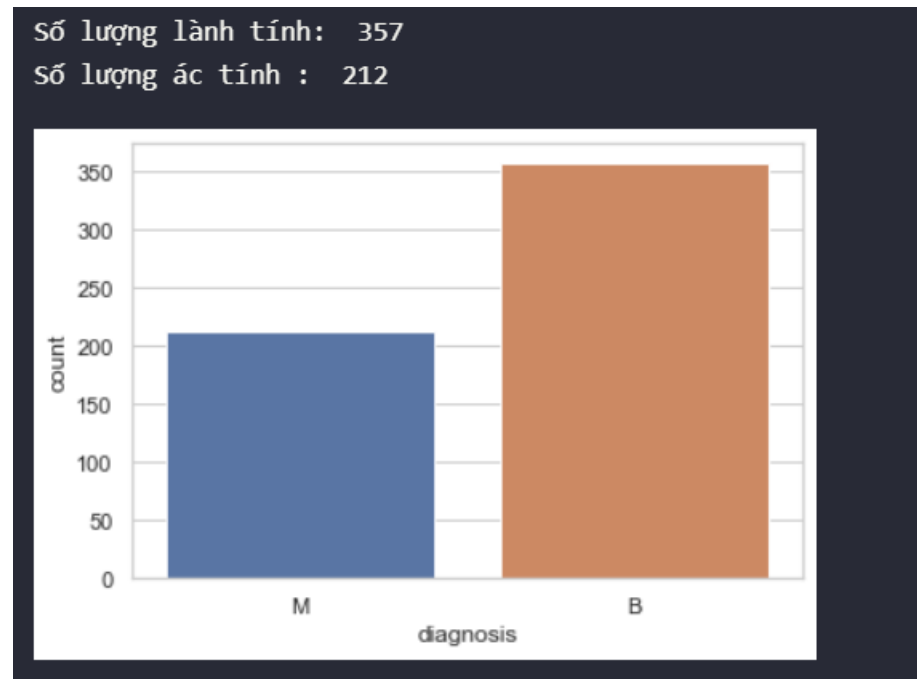
Các kiểu dữ liệu đều ở dạng số.

Open in Notebook Editor

1	radius_mean	float64
2	texture_mean	float64
3	perimeter_mean	float64
4	area_mean	float64
5	smoothness_mean	float64
6	compactness_mean	float64
7	concavity_mean	float64
8	concave points_mean	float64
9	symmetry_mean	float64
10	fractal_dimension_mean	float64
11	radius_se	float64
12	texture_se	float64
13	perimeter_se	float64
14	area_se	float64
15	smoothness_se	float64
16	compactness_se	float64
17	concavity_se	float64
18	concave points_se	float64
19	symmetry_se	float64
20	fractal_dimension_se	float64
21	radius_worst	float64
22	texture_worst	float64
23	perimeter_worst	float64
24	area_worst	float64
25	smoothness_worst	float64
26	compactness_worst	float64
27	concavity_worst	float64
28	concave points_worst	float64
29	symmetry_worst	float64
30	fractal_dimension_worst	float64
31	dtype: object	

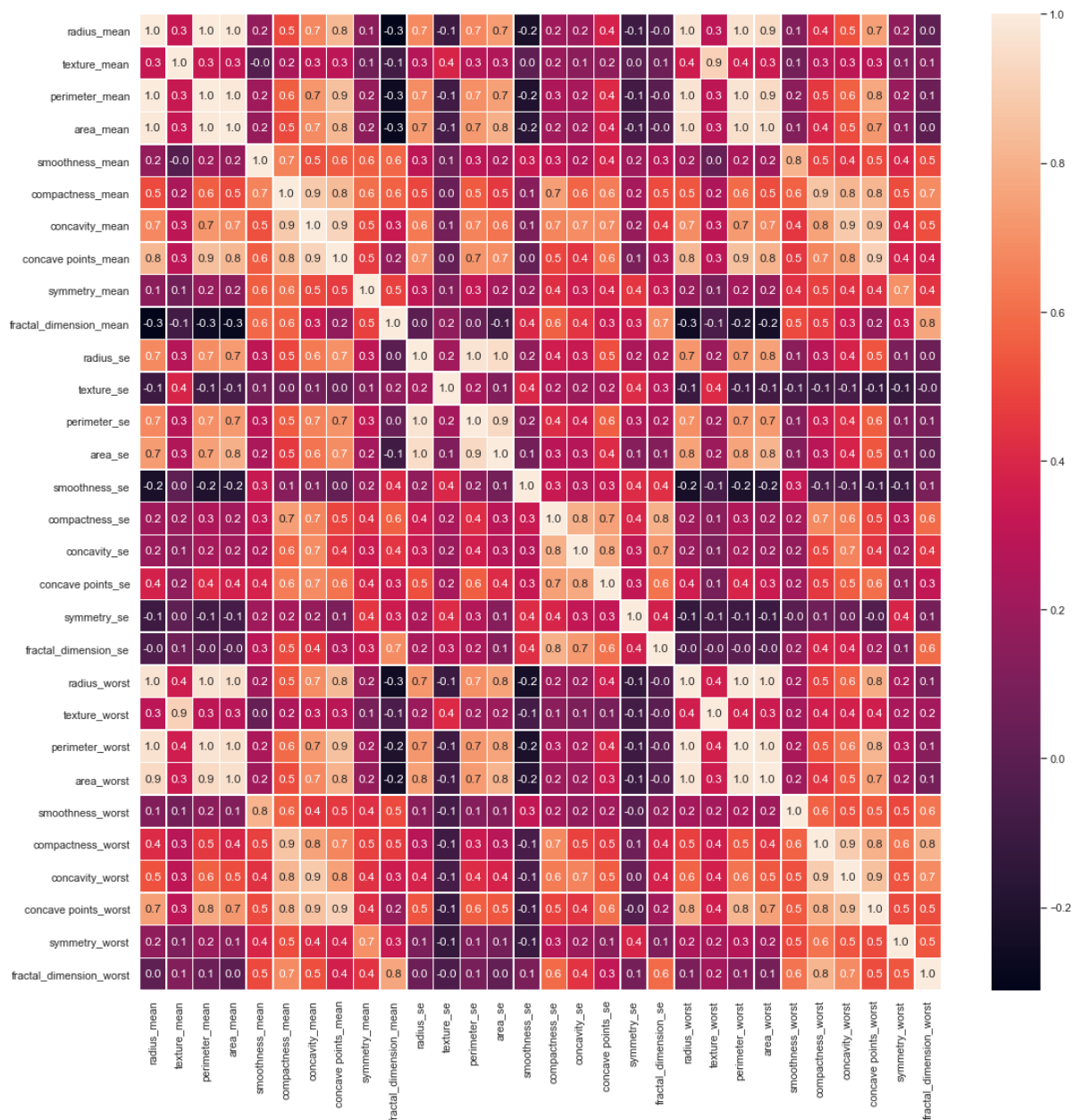
Xem xét liệu có sự bất thường ở output ?

Đối với biến đầu ra **diagnosis** tỉ lệ output không có gì bất hợp lý.



Đối với việc huấn luyện mô hình cần phải tiền xử lý thêm để lấy ra tập dữ liệu mới có độ chính xác cao hơn.

Dưới đây là heat map dùng để xét các hệ số tương quan giữa các biến trong tập dữ liệu.



Từ đó có những nhận định như sau:

- Như có thể thấy trong bản đồ nhiệt **radius_mean**, **perimeter_mean** và **area_mean** có tương quan với nhau nên chúng ta sẽ chỉ sử dụng **area_mean**.

- **Compactness_mean**, **concavity_mean** và **concave points_mean** có tương quan với nhau nên chỉ chọn **concavity_mean**.

- **Radius_se**, **perimeter_se** và **area_se** có tương quan nên chỉ sử dụng **area_se**.

- **Radius_worst**, **perimeter_worst** và **area_worst** có tương quan nên sử dụng **area_worst**.

- **Compactness_worst**, **concavity_worst** và **concave points_worst** tương quan nên sử dụng **concavity_worst**.

- **Compactness_se**, **concavity_se** và **concave points_se** tương quan nhau nên sử dụng **concavity_se**.

- **texture_mean** và **texture_worst** có tương quan và em sử dụng **texture_mean**.
- **area_worst** và **area_mean** có tương quan với nhau, sử dụng **area_mean**.

Tập dữ liệu lúc này còn lại các cột:

```
x_1.columns.values
✓ 0.9s

array(['texture_mean', 'area_mean', 'smoothness_mean', 'concavity_mean',
       'symmetry_mean', 'fractal_dimension_mean', 'texture_se', 'area_se',
       'smoothness_se', 'concavity_se', 'symmetry_se',
       'fractal_dimension_se', 'smoothness_worst', 'concavity_worst',
       'symmetry_worst', 'fractal_dimension_worst'], dtype=object)
```

Tìm độ chính xác với những cột đã chọn:

Ở đây em dùng RandomForestClassifier.

```
# Tách data train 70 % và test 30 %
x_train, x_test, y_train, y_test = train_test_split(x_1, y, test_size=0.3, random_state=42)

#random forest classifier với n_estimators=10 (default)
clf_rf = RandomForestClassifier(random_state=43)
clr_rf = clf_rf.fit(x_train,y_train)

ac = accuracy_score(y_test,clf_rf.predict(x_test))
print('Accuracy is: ',ac)
cm = confusion_matrix(y_test,clf_rf.predict(x_test))
sns.heatmap(cm,annot=True,fmt="d")
✓ 0.4s

Accuracy is: 0.9649122807017544

<AxesSubplot:>
```

	Actual 0	Actual 1
Predicted 0	106	2
Predicted 1	4	59

Nhận thấy được độ chính xác là hơn 96%.

Chọn tập tính năng mới để tìm ra kết quả tốt hơn.

Sử dụng random forest để chọn ra 5 tính năng tốt nhất.

```
Sử dụng random forest để chọn ra 5 tính năng tốt nhất.

# Tạo đối tượng RFE và xếp hạng từng pixel
clf_rf_3 = RandomForestClassifier()
rfe = RFE(estimator=clf_rf_3, n_features_to_select=5, step=1)
rfe = rfe.fit(x_train, y_train)
good_fea = x_train.columns[rfe.support_].values
print('5 tính năng tốt nhất được chọn bởi rfe:', good_fea)
✓ 1.7s

5 tính năng tốt nhất được chọn bởi rfe: ['area_mean' 'concavity_mean' 'area_se' 'concavity_worst' 'symmetry_worst']
```

Chọn ra 5 tính năng tốt nhất gồm : **area_mean, concavity_mean, area_se, concavity_worst, symmetry_worst.**

Giữa các lần chạy khác nhau sẽ cho ra 5 best features khác nhau. Ở lần chạy này là kết quả này và em sẽ dùng nó làm demo.

2. Huấn luyện mô hình:

Mô tả

Mặc định khi tới bước huấn luyện mô hình, hoặc chức năng predict thì tập dữ liệu đã được tiền xử lý chỉ còn 5 feature quan trọng nhất và dữ liệu ta input vào để predict cũng yêu cầu 5 feature đó mà thôi

Visualization Prediction

1. Predict: Enter all the data you need to predict on the left then select the model and configuration on the right, finally click Predict

2. Evaluate: Select the model and configuration on the right, finally click Evaluate Model

Input Data

area mean

concavity mean

area se

concavity worst

symmetry worst

Config

Choose model

Ratio(test) Method

Predict Evaluate Model

Chia tập dữ liệu huấn luyện, kiểm thử:

App cho phép người dùng tùy chỉnh ratio (≥ 0 và ≤ 1) của tập kiểm thử, vì vậy không ràng buộc ở 1 tỉ lệ nào cả

Input Data	Config
area mean <input type="text"/>	Choose model <input type="text" value="Decision tree"/>
concavity mean <input type="text"/>	Ratio(test) <input type="text"/> Method <input type="text" value="entropy"/>
area se <input type="text"/>	
concavity worst <input type="text"/>	
symmetry worst <input type="text"/>	

Thiết lập/ tinh chỉnh các tham số mô hình:

App áp dụng 3 mô hình cơ bản đã được học cho người dùng có thể tùy chọn, và tinh chỉnh các tham số cơ bản đã được học. Để biết được mô hình có thể tùy chỉnh cách tham số nào hãy dùng **combobox (drop down list)** để chọn mô hình, tùy theo mô hình mà sẽ xuất hiện các tùy chỉnh riêng

kNN: Tùy chỉnh k

Config	
Choose model	<input type="text" value="kNN"/>
Ratio(test)	<input type="text"/> K <input type="text"/>

Decision tree: Tùy chỉnh kiểu đo lường theo entropy hoặc gini index

Config	
Choose model	<input type="text" value="Decision tree"/>
Ratio(test)	<input type="text"/> Method <input type="text" value="gini"/>

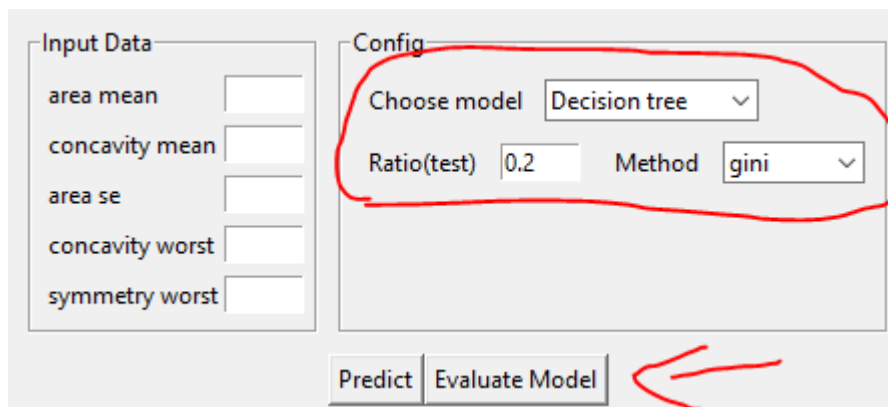
Config	
Choose model	<input type="text" value="Decision tree"/>
Ratio(test)	<input type="text"/> Method <input type="text" value="entropy"/>

Naïve Bayes: Không có

3. Báo cáo kết quả độ chính xác:

Ngoài Predict app còn có chức năng Evaluate model thông qua confuse Matrix và các measurement đã được học như: Accuracy, Precision, Error rate,...

Chức năng Evaluate cũng áp dụng trên tùy chọn mô hình, tùy chọn ratio và các tham số vì vậy người dùng có thể cấu hình model theo ý muốn và nhấn Evaluate để sử dụng



Input Data

area mean

concavity mean

area se

concavity worst

symmetry worst

Config

Choose model

Ratio(test) Method

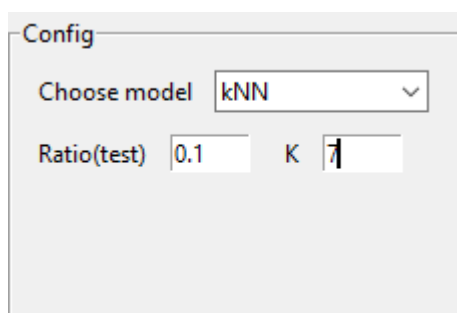
Predict Evaluate Model

Cấu hình ví dụ Model dùng là cây quyết định test set có ratio là 0.2 dùng phương thức đo là gini index. Cuối cùng là bấm Evaluate Model

Ta được kết quả như sau:

Decision tree (ratio 0.2)				
Evaluation				
Confuse matrix			Measurements	
	predict B	predict M		
Actual B	64	3	Accuracy: 0.939	Error rate: 0.061
Actual M	4	43	Sensitivity: 0.955	Specificity: 0.915
			Precision: 0.941	Recall: 0.955

Có thể áp dụng với mô hình và cấu hình khác như sau:



Config

Choose model

Ratio(test) K

kNN (ratio 0.1)

Evaluation

Confuse matrix

	predict B	predict M
Actual B	34	1
Actual M	2	20

Measurements

Accuracy: 0.947 Error rate: 0.053

Sensitivity: 0.971 Specificity: 0.909

Precision: 0.944 Recall: 0.971

4. Video demo:

Link demo trên youtube: <https://youtu.be/KFL8FdLry5o>

Lời kết

Cảm ơn thầy đã xem qua đồ án của chúng em.