BÁO CÁO ĐỒ ÁN

Triển khai mô hình Phân loại bệnh ung thư vú

MÔN HỌC:
KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG

GIẢNG VIÊN HƯỚNG DẪN

Thầy Lê Ngọc Thành
Thầy Nguyễn Thái Vũ

Người thực hiện:

1. Ngô Đăng Khoa – 19127444
2. Nguyễn Gia Hân – 19127134
3. Huỳnh Cao Nhật Hiếu - 19127399

Mục lục

1. Xây dựng mô hình phân lớp

i. Khám phá dữ liệu

diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean	radius_se	texture
M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	1.0950	0.9
M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7
M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	0.7456	0.7
M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	0.4956	1.1
M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	0.7572	0.7

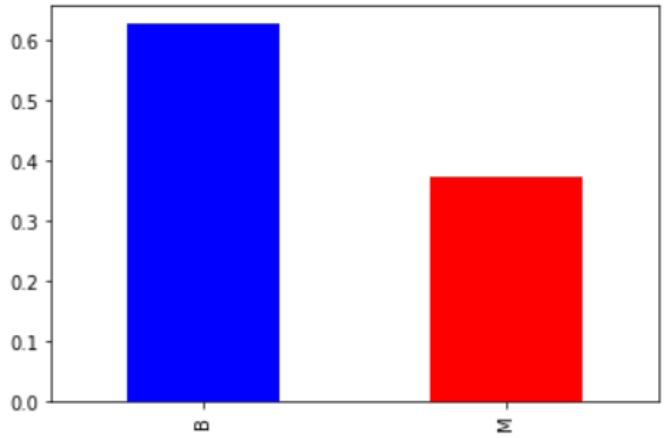
```
kclass 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   id                      569 non-null    int64
 1   diagnosis               569 non-null    object
 2   radius_mean             569 non-null    float64
 3   texture_mean            569 non-null    float64
 4   perimeter_mean          569 non-null    float64
 5   area_mean               569 non-null    float64
 6   smoothness_mean         569 non-null    float64
 7   compactness_mean        569 non-null    float64
 8   concavity_mean          569 non-null    float64
 9   concave points_mean     569 non-null    float64
 10  symmetry_mean           569 non-null    float64
 11  fractal_dimension_mean  569 non-null    float64
 12  radius_se               569 non-null    float64
 13  texture_se              569 non-null    float64
 14  perimeter_se            569 non-null    float64
 15  area_se                 569 non-null    float64
 16  smoothness_se           569 non-null    float64
 17  compactness_se          569 non-null    float64
 18  concavity_se            569 non-null    float64
 19  concave points_se       569 non-null    float64
 20  symmetry_se             569 non-null    float64
 21  fractal_dimension_se    569 non-null    float64
 22  radius_worst            569 non-null    float64
 23  texture_worst           569 non-null    float64
 24  perimeter_worst         569 non-null    float64
 25  area_worst              569 non-null    float64
 26  smoothness_worst        569 non-null    float64
 27  compactness_worst       569 non-null    float64
 28  concavity_worst         569 non-null    float64
 29  concave points_worst    569 non-null    float64
 30  symmetry_worst          569 non-null    float64
 31  fractal_dimension_worst 569 non-null    float64
 32  Unnamed: 32             0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

```
df.shape
```

```
(569, 33)
```

- Dữ liệu trên có 569 dòng và 33 cột.
- Mỗi dòng dữ liệu các đặc điểm của nhân tế bào vú đã được tính toán từ hình ảnh số hóa của một bệnh nhân và chuẩn đoán xem bệnh nhân u lành tính (*benign*) hay ác tính (*malignant*).
- Ngoài cột **Unnamed: 32** ra thì không có cột nào thiếu dữ liệu.
- Không có dòng dữ liệu nào bị trùng lắp.
- Ở tập dữ liệu này ta sẽ có cột output là cột **diagnosis** dùng để chuẩn đoán xem bệnh nhân đó có u lành tính hay ác tính.

```
B    62.741652
M    37.258348
Name: diagnosis, dtype: float64
```



- Ta thấy được tỉ lệ các giá trị của cột đầu ra như sau:
  - **Benign** là 62.7%
  - **Malignant** là 37.25%.

ii. Tiền xử lí dữ liệu

- Đầu tiên ta sẽ bỏ đi một số cột như sau:
  - Bỏ cột **id** đi vì cột này chỉ dùng để định danh cho mỗi dòng dữ liệu, không có ý nghĩa gì trong việc huấn luyện.
  - Bỏ cột **Unnamed**: 32 vì cột này có quá nhiều giá trị thiếu.

- Tiếp theo ta sẽ tách DataFrame trên thành 2 tập input (**X_df**) ứng với các cột giá trị thông số trên hình ảnh và output (**y_sr**) ứng với cột Diagnosis.

```
# Tách X và y
y_sr = df["diagnosis"] # sr là viết tắt của series
X_df = df.drop(["diagnosis", 'id', 'Unnamed: 32'], axis=1)
```

- Ở đây ta sẽ tách tập dữ liệu thành 2 phần như sau:
  - 90% cho dữ liệu **train và validiation.**
  - 10% cho dữ liệu **test** cuối.
  - Stratify = y_sr để giữ cho tỉ lệ các class như với ban đầu

```
# Tách tập train và tập test theo tỉ lệ 90%:10%
train_X_df, test_X_df, train_y_sr, test_y_sr = \
                        train_test_split(X_df, y_sr,
                                         test_size=0.1,
                                         stratify=y_sr,
                                         random_state=0)
```

```
train_X_df.shape, train_y_sr.shape
```

```
((512, 30), (512,))
```

```
test_X_df.shape, test_y_sr.shape
```

```
((57, 30), (57,))
```

iii. Tạo pipeline cho mô hình KNN

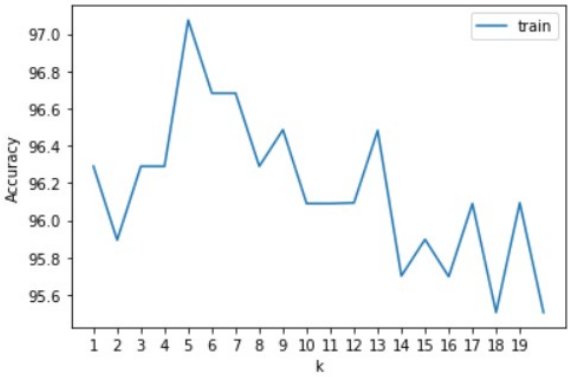
- Nhóm quyết định thực hiện kĩ thuật tạo pipeline cho mô hình *KNN* bằng thư viện *Sklearn*.
- Pipeline cho mô hình KNN gồm có hai bước:
  - Bước 1: Sử dụng *StandardScaler* để chuẩn hóa các cột của mô hình.
  - Bước 2: Áp dụng mô hình *KNN* để huấn luyện và dự đoán sử dụng measurement là **Euclide distance**.

```
cancer_pipeline = Pipeline([('normalize', StandardScaler()), ("KNN", KNeighborsClassifier())])
```

- Để có thể đánh giá được với tập dữ liệu trên chạy tốt nhất với tham số **k** là bao nhiêu, ta sẽ thử lần lượt từng con số k để xem kết quả nào cho ra là tốt nhất thông qua phương pháp K-fold cross-validiation để đánh giá. Với việc sử dụng phương pháp này, ta chỉ có thể thấy được kết quả trung bình trong tập validiation ở mỗi tham số **k.**

```
val_accs = []
param = [item for item in range(1,21)]
best_val_acc = 0
best_k = 0
best_k_test = 0
scores = 0
for k in param:
    cancer_pipeline['KNN'].set_params(n_neighbors=k)
    cv = KFold(n_splits=10, random_state=1, shuffle=True)
    scores = cross_val_score(cancer_pipeline, train_X_df, train_y_sr, scoring='accuracy', cv=cv, n_jobs=-1)
    print(scores)
    val_acc = scores.mean() * 100
    val_accs.append(val_acc)
    if val_acc > best_val_acc:
        best_k = k
        best_val_acc = val_acc
```

iv. Kết quả mô hình.




```
Best val_accuracy:  97.0739064856712
Best k:  5
```

- Như vậy với kết quả trên ta sẽ sử dụng model KNN với siêu tham số k = 5.
- Cuối cùng, ta sẽ sử dụng tham số k tốt nhất đã tìm được và sử dụng để huấn luyện mô hình dự đoán trên tập test đã chia ở phần trước.

```
cancer_pipeline['KNN'].set_params(n_neighbors=5)
cancer_pipeline.fit(train_X_df, train_y_sr)
cancer_pipeline.score(test_X_df, test_y_sr) * 100
```

```
92.98245614035088
```

- Như vậy độ xác của mô hình này là **92,98%**

2. Xây dựng ứng dụng

Nhóm sử dụng thư viện ***tkinter*** để xây dựng ứng dụng cho phép người dùng thực hiện các thao tác liên quan đến khai thác dữ liệu (biểu diễn dữ liệu, thực hiện các thống kê cơ bản) và triển khai mô hình phân lớp (kết quả dự đoán của mô hình).

Ứng dụng này gồm có 3 chức năng chính theo yêu cầu của đề bài, đó là:

- Biểu diễn dữ liệu dưới dạng biểu đồ (boxplot, histogram (với một biến), scatter plot (với 2 biến) và correlogram (với 3 biến trở lên)).
- Thực hiện hiển thị các thống kê cơ bản như max, min, variance, mean, standard deviation cho từng biến.
- Dự đoán kết quả của người dùng dựa trên các giá trị đầu vào mà người dùng nhập.

3. Tham khảo

[1] https://phocode.com/python/python-tkinter-lap-trinh-gui-voi-tkinter/?fbclid=IwAR2HkBrX6U-0yWBdRY3w8g1jSlWwtILatgof6XY_q3BU_gvwAwU3P6KoSg0

[2] https://scikit-learn.org/stable/modules/cross_validation.html?fbclid=IwAR03mk9bOoAax_lSJ39Bfx0Bn1o1JUCrLxIAYyJ0T634uMkNn9B9x1eYMeM

[3] https://www.youtube.com/watch?v=8HSvi2SZm3U&list=LL&index=3&t=11s&ab_channel=duchieuvn

[4] https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html?fbclid=IwAR1VCQZa_G6T1H4Q_g-46JkZxD7RQAdmEx-brWJdjcsgUE1MY5Da7cP1lpM

[5] https://datascience.stackexchange.com/questions/52704/how-to-save-a-knn-model?fbclid=IwAR0dtn7c7vjRxaP4i7Hb_SgE3M9PPfavW1K4GZYHBV5wwQtFnCJNZVRMp0k

4. Link demo

https://drive.google.com/file/d/1w8YkI6u-YeOblRo0skqD87kJUqUtJUI9/view?fbclid=IwAR1DibZ7y86ZabQUxORIfrzdcID02-C-VXV-g2Qlp5h6ehgQbLpuKv0cqGI