CHUẨN BỊ DỮ LIỆU TRONG QUÁ TRÌNH HUẤN LUYỆN MÔ HÌNH HỌC

I. GIỚI THIỆU

Trong quá trình phát triển mô hình học máy, chất lượng dữ liệu và cách xử lý dữ liệu đóng vai trò quan trọng quyết định hiệu quả của mô hình. Một quy trình chuẩn bị dữ liệu chính xác giúp cải thiện hiệu suất, giảm thiểu sai số và tránh các vấn đề liên quan đến overfitting hoặc underfitting. Báo cáo này sẽ trình bày chi tiết từng bước trong quy trình chuẩn bị dữ liệu bao gồm khám phá dữ liệu, phân tích dữ liệu khám phá, tiền xử lý, thao tác dữ liệu và lựa chọn/trích xuất tính năng.

II. CHUẨN BỊ DỮ LIỆU TRONG QUÁ TRÌNH HỌC MÁY

1. KHÁM PHÁ DỮ LIỆU

❖ Định nghĩa

Khám phá dữ liệu là bước đầu tiên trong quy trình chuẩn bị dữ liệu. Đây là quá trình thu thập thông tin tổng quan về dữ liệu, từ đó giúp chúng ta hiểu rõ đặc điểm và cấu trúc của dữ liệu trước khi thực hiện các bước xử lý tiếp theo.

❖ Các bước khám phá dữ liệu

➢ Import thư viện

- **Mục tiêu:**

Mục tiêu của bước này là nhập các thư viện cần thiết để thực hiện việc phân tích và xử lý dữ liệu, cũng như xây dựng mô hình dự đoán. Việc sử dụng các thư viện chuyên dụng giúp cho quá trình này trở nên hiệu quả và thuận tiện hơn.

- **Thư viện sử dụng**

Các thư viện phổ biến thường được sử dụng trong việc phân tích dữ liệu và xây dựng mô hình dự đoán bao gồm:

- pandas: Thư viện cung cấp các công cụ để xử lý dữ liệu, đặc biệt là cho các cấu trúc dữ liệu như DataFrame. Nó cho phép người dùng dễ dàng truy cập, thao tác và phân tích dữ liệu.
- numpy: Thư viện này hỗ trợ các tính toán số học, giúp xử lý mảng và ma trận một cách hiệu quả. NumPy thường được sử dụng để xử lý dữ liệu số và thực hiện các phép toán toán học.
- matplotlib và seaborn: Đây là hai thư viện giúp vẽ đồ thị, trực quan hóa dữ liệu. Chúng cho phép người dùng tạo ra các biểu đồ và hình ảnh giúp minh họa các xu hướng và mô hình trong dữ liệu.
- scikit-learn: Thư viện này cung cấp nhiều công cụ để xây dựng và đánh giá các mô hình học máy. Nó bao gồm nhiều thuật toán học máy khác nhau, từ hồi quy đến phân loại, và cũng hỗ trợ các kỹ thuật như chuẩn hóa và phân chia dữ liệu
- Còn nhiều thư viện khác tùy vào bài toán v.v…

- **Cách thực hiện**

```
import math
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression, Ridge,
```

```
Lasso, ElasticNet
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import r2_score, mean_squared_error,
mean_absolute_error
from sklearn.model_selection import train_test_split
import joblib
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display
from statsmodels.formula import api
from sklearn.feature_selection import RFE
from sklearn.preprocessing import StandardScaler,
PolynomialFeatures
from statsmodels.stats.outliers_influence import
variance_inflation_factor
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Ridge
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [10,6]

import warnings
warnings.filterwarnings('ignore')
```

- **Kết quả**

Sau khi thực hiện đoạn mã trên, tất cả các thư viện cần thiết đã được nhập vào môi trường làm việc, giúp cho người dùng có thể sử dụng các chức năng và công

cụ mà mỗi thư viện cung cấp. Điều này tạo điều kiện thuận lợi cho quá trình phân tích, tiền xử lý dữ liệu và xây dựng mô hình dự đoán

- **Kết luận**

Bước nhập các thư viện là rất quan trọng trong quy trình chuẩn bị dữ liệu và xây dựng mô hình. Các thư viện đã được nhập vào cung cấp các công cụ cần thiết để thực hiện các phân tích và mô hình hóa, từ đó giúp nâng cao hiệu suất và độ chính xác của mô hình dự đoán.

➢ Chuẩn Bị Dữ Liệu

- **Mục tiêu**

Mục tiêu của bước này là tải dữ liệu từ một nguồn cụ thể, định dạng và lưu trữ nó trong một cấu trúc dễ quản lý, thường là DataFrame. Việc này giúp cho quá trình phân tích và xử lý dữ liệu diễn ra thuận lợi hơn, từ đó hỗ trợ xây dựng mô hình dự đoán chính xác và hiệu quả.

- **Cách thực hiện**

```
df =
pd.read_csv('/content/drive/MyDrive/sale/Data/Advertising-B
udget-and-Sales.csv', index_col=0,
names=['TV','Radio','Newspaper','Sales'], skiprows=1)
# Hiển thị 5 dòng đầu tiên
display(df.head())
# Xác định nhãn và đặc trưng
target = 'Sales'
features = [i for i in df.columns if i not in [target]]
# Sao chép data
original_df = df.copy(deep=True)
print('\n\033[1mInference:\033[0m The Dataset consists of
{} features & {} samples.'.format(df.shape[1],
df.shape[0]))
```

- **Kết quả**

Sau khi thực hiện đoạn mã trên, một DataFrame sẽ được tạo ra, lưu trữ tất cả các thông tin từ tệp CSV. DataFrame này sẽ cho phép chúng ta dễ dàng thực hiện các thao tác như truy xuất, phân tích và tiền xử lý dữ liệu

	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

Inference: The Datset consists of 4 features & 200 samples.

- **Kết luận**

Chuẩn bị dữ liệu là một bước thiết yếu trong quy trình xây dựng mô hình dự đoán. Các bước từ khám phá dữ liệu đến tiền xử lý và lựa chọn tính năng đều có tác động lớn đến hiệu suất cuối cùng của mô hình. Bằng cách thực hiện các bước này một cách cẩn thận, chúng ta có thể xây dựng một mô hình dự đoán chính xác và đáng tin cậy.

➢ Kiểm tra các kiểu dữ liệu của tất cả các cột

- **Mục tiêu**

Mục tiêu của bước này là xác định và kiểm tra kiểu dữ liệu của từng cột trong DataFrame. Việc này rất quan trọng vì nó giúp người phân tích hiểu rõ cấu trúc của dữ liệu, từ đó có thể thực hiện các thao tác xử lý và phân tích phù hợp. Kiểu

dữ liệu cũng ảnh hưởng đến các thuật toán học máy có thể được sử dụng trong mô hình.

- **Khái niệm**

Kiểm tra kiểu dữ liệu của các cột trong tập dữ liệu là quá trình xác định xem mỗi cột chứa loại dữ liệu gì (số nguyên, số thực, chuỗi ký tự, dữ liệu phân loại, ngày tháng, hoặc boolean).

- **Cách thực hiện**

```
df.info()
```

- **Kết quả**

Sau khi thực hiện, người dùng sẽ nhận được một danh sách hiển thị kiểu dữ liệu của tất cả các cột trong DataFrame.

```
<class 'pandas.core.frame.DataFrame'>
Index: 200 entries, 1 to 200
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 7.8 KB
```

- **Kết luận**

Việc kiểm tra kiểu dữ liệu của các cột trong DataFrame là bước quan trọng để đảm bảo rằng dữ liệu được hiểu đúng và phù hợp cho các bước xử lý tiếp theo. Nếu có cột nào có kiểu dữ liệu không phù hợp, người dùng cần thực hiện chuyển

đổi kiểu dữ liệu cho phù hợp, từ đó hỗ trợ tốt hơn cho quá trình phân tích và xây dựng mô hình dự đoán.

➢ Kiểm tra số lượng giá trị duy nhất

- **Mục tiêu**

Mục tiêu của bước này là xác định số lượng giá trị duy nhất có trong mỗi cột của DataFrame. Việc này rất hữu ích để hiểu sự phân bố và tính đa dạng của dữ liệu, từ đó giúp trong việc phân tích và lựa chọn mô hình phù hợp. Số lượng giá trị duy nhất cũng có thể ảnh hưởng đến cách mà các biến được sử dụng trong mô hình học máy, đặc biệt là đối với các biến phân loại.

- **Khái niệm**

Giá trị duy nhất trong một cột là những giá trị mà không có sự lặp lại. Ví dụ, trong một cột chứa thông tin về giới tính, số lượng giá trị duy nhất có thể là 2 (nam, nữ)

- **Cách thực hiện**

```
df.nunique().sort_values()
```

- **Kết quả**

Sau khi thực hiện đoạn mã trên, người dùng sẽ nhận được một danh sách hiển thị số lượng giá trị duy nhất trong mỗi cột.

	0
Sales	121
Radio	167
Newspaper	172
TV	190

dtype: int64

- **Kết luận**

Việc kiểm tra số lượng giá trị duy nhất trong mỗi cột là bước quan trọng để hiểu rõ hơn về dữ liệu. Thông tin này giúp xác định các cột nào có thể là biến phân loại (categorical) và có thể cần phải mã hóa để sử dụng trong các mô hình học máy. Nếu một cột có quá nhiều giá trị duy nhất, có thể cần xem xét việc nhóm các giá trị tương tự lại với nhau để giảm độ phức tạp của mô hình.

➢ Phân loại các cột trong bộ dữ liệu thành hai nhóm

- **Mục tiêu**

Bước này nhằm mục đích phân loại các cột trong bộ dữ liệu thành hai nhóm: đặc trưng số học (numerical features) và đặc trưng phân loại (categorical features). Việc phân loại này giúp người phân tích hiểu rõ hơn về tính chất của các biến trong dữ liệu, từ đó có thể áp dụng các phương pháp xử lý và mô hình hóa phù hợp.

- **Khái niệm**

- Đặc trưng số học (Numerical Features): Là các biến có thể biểu diễn bằng số và có thể tính toán các thống kê như trung bình, độ lệch chuẩn, v.v.
- Đặc trưng phân loại (Categorical Features): Là các biến đại diện cho các loại hoặc nhóm, không có ý nghĩa về số lượng và không thể tính toán các thống kê thông thường.

- **Cách thực hiện**

```
# Đếm số lượng giá trị duy nhất trong các cột đặc trưng, sau đó sắp xếp theo thứ tự tăng dần
nu = df[features].nunique().sort_values()

# Khởi tạo danh sách để lưu các đặc trưng số học và phân loại
nf = []  # Danh sách lưu các đặc trưng số học
cf = []  # Danh sách lưu các đặc trưng phân loại
```

```
nnf = 0
ncf = 0
for i in range(df[features].shape[1]):
    # Nếu cột có ≤ 16 giá trị duy nhất, coi nó là đặc
trưng phân loại
    if nu.values[i] <= 16:
        cf.append(nu.index[i])
    # Nếu cột có > 16 giá trị duy nhất, coi nó là đặc
trưng số học
    else:
        nf.append(nu.index[i])
print('\n\033[1mKết luận:\033[0m Bộ dữ liệu có {} đặc
trưng số học và {} đặc trưng phân loại.'.format(len(nf),
len(cf)))
```

- **Kết quả**

  Kết quả của quá trình phân loại sẽ được in ra trong console, với thông tin cụ thể về số lượng các cột trong từng nhóm

```
Kết luận: Bộ dữ liệu có 3 đặc trưng số học và 0 đặc trưng phân loại.
```

- **Kết luận**

  Việc phân loại các cột trong bộ dữ liệu thành đặc trưng số học và phân loại giúp người phân tích dễ dàng lựa chọn các phương pháp xử lý và mô hình hóa phù hợp cho từng loại biến. Điều này cũng góp phần vào việc tối ưu hóa quy trình phân tích và dự đoán trong học máy.

➢ Tổng hợp và hiển thị các thống kê mô tả

- **Mục tiêu**

Mục tiêu của bước này là tổng hợp và hiển thị các thống kê mô tả của bộ dữ liệu. Các thống kê mô tả cung cấp cái nhìn tổng quan về dữ liệu, giúp người phân tích hiểu rõ hơn về phân phối, xu hướng và tính chất của các biến trong bộ dữ liệu.

- **Khái niệm**

Thống kê mô tả: Là các giá trị số cho biết thông tin cơ bản về dữ liệu, bao gồm các thông tin như giá trị trung bình, độ lệch chuẩn, giá trị tối thiểu, giá trị tối đa, và các phần trăm (percentiles). Các thống kê mô tả giúp tóm tắt và mô tả tính chất của dữ liệu mà không cần phải trực tiếp nhìn vào từng quan sát.

- **Cách thực hiện**

```
display(df.describe())
```

- **Kết quả**

Kết quả của quá trình tổng hợp thống kê mô tả

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	14.022500
std	85.854236	14.846809	21.778621	5.217457
min	0.700000	0.000000	0.300000	1.600000
25%	74.375000	9.975000	12.750000	10.375000
50%	149.750000	22.900000	25.750000	12.900000
75%	218.825000	36.525000	45.100000	17.400000
max	296.400000	49.600000	114.000000	27.000000

- **Kết luận**

Việc tổng hợp và hiển thị các thống kê mô tả là một bước quan trọng trong quá trình khám phá dữ liệu. Nó giúp người phân tích nhanh chóng nhận diện các đặc điểm nổi bật của dữ liệu, như sự phân phối và tính chất của các biến, từ đó đưa ra quyết định phù hợp cho các bước xử lý tiếp theo trong quy trình phân tích và xây dựng mô hình học máy.

2. PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

❖ Định Nghĩa

Phân Tích Dữ Liệu Khám Phá (Exploratory Data Analysis - EDA) là quá trình phân tích bộ dữ liệu để tóm tắt các đặc điểm chính, phát hiện các mô hình, xu hướng, và mối quan hệ trong dữ liệu. Mục tiêu của EDA là hiểu rõ hơn về dữ liệu, từ đó đưa ra quyết định sáng suốt cho các bước tiếp theo trong quy trình phân tích, như tiền xử lý dữ liệu và xây dựng mô hình.

❖ Các bước phân tích dữ liệu khám phá

➢ Biểu đồ phân phối của biến mục tiêu

- **Mục tiêu**

Mục tiêu của bước này là trực quan hóa phân phối của biến mục tiêu trong bộ dữ liệu. Việc hiểu rõ phân phối của biến mục tiêu giúp xác định các đặc điểm quan trọng, nhận diện các vấn đề tiềm ẩn và đưa ra quyết định hợp lý trong quá trình phân tích và xây dựng mô hình.

- **Khái niệm**

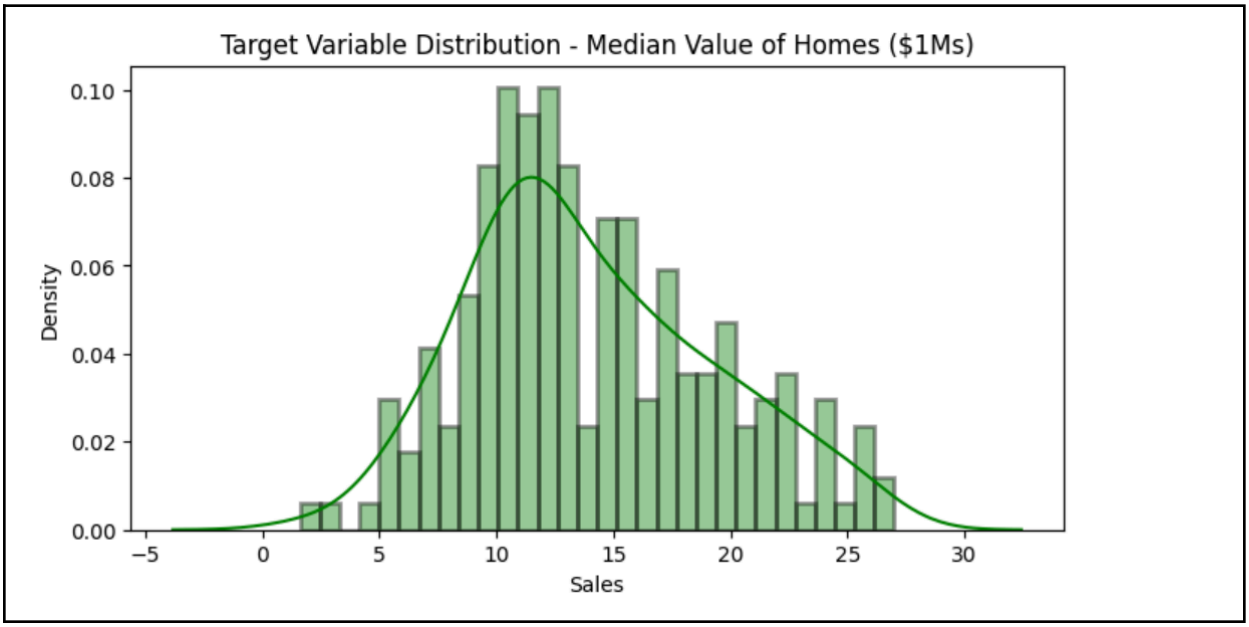
Biểu đồ phân phối: Là một loại biểu đồ giúp thể hiện sự phân bố của một biến số. Nó cho phép nhà phân tích dễ dàng nhận diện các đặc điểm như độ lệch, độ phân tán và các giá trị ngoại lai trong dữ liệu.

- **Cách thực hiện**

```
plt.figure(figsize=[8,4])
sns.distplot(df[target],
color='g',hist_kws=dict(edgecolor="black", linewidth=2),
bins=30)
plt.title('Target Variable Distribution - Median Value of
Homes ($1Ms)')
plt.show()
```

- **Kết quả**

Kết quả của quá trình vẽ biểu đồ phân phối sẽ cho thấy hình dạng và đặc điểm của biến mục tiêu.



- **Kết luận**

Việc tạo biểu đồ phân phối cho biến mục tiêu là một bước quan trọng trong phân tích dữ liệu khám phá. Nó giúp nhà phân tích hiểu rõ hơn về cách mà biến

mục tiêu phân bố trong bộ dữ liệu, từ đó xác định các xu hướng và các vấn đề tiềm ẩn có thể ảnh hưởng đến việc xây dựng mô hình.

➢ Trực quan hóa các đặc điểm số

- **Mục tiêu**

Mục tiêu của bước này là trực quan hóa phân bố của các đặc điểm số trong bộ dữ liệu. Việc này không chỉ giúp nhận diện các xu hướng mà còn hỗ trợ phát hiện các giá trị bất thường (outliers) có thể ảnh hưởng đến việc xây dựng mô hình.

- **Khái niệm**

- Trực quan hóa phân bố: Là quá trình sử dụng các biểu đồ để thể hiện sự phân bố của các đặc điểm số, giúp người phân tích dễ dàng nhận diện các đặc điểm, xu hướng và vấn đề trong dữ liệu.
- Boxplot: Là một loại biểu đồ giúp hiển thị các giá trị bất thường và phân phối của dữ liệu, bao gồm các thông tin như giá trị tối thiểu, giá trị tối đa, và các phần tư (quartiles).

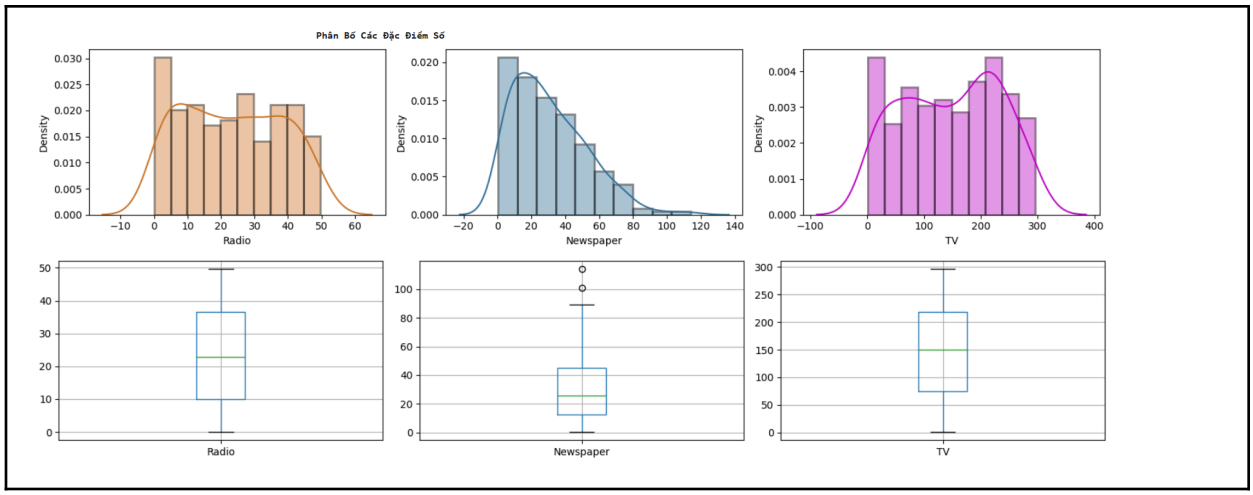
- **Cách thực hiện**

```
print('\033[1mPhân Bố Các Đặc Điểm Số'.center(130))
n = 3
plt.figure(figsize=[15, 3 * math.ceil(len(nf) / n)])
for i in range(len(nf)):
    plt.subplot(math.ceil(len(nf) / 3), n, i + 1)
    sns.distplot(df[nf[i]],
hist_kws=dict(edgecolor="black", linewidth=2), bins=10,
color=list(np.random.randint([255,255,255])/255))
plt.tight_layout()
plt.show()
plt.figure(figsize=[15, 3 * math.ceil(len(nf) / n)])
for i in range(len(nf)):
    plt.subplot(math.ceil(len(nf) / 3), n, i + 1)  #
```

```
    df.boxplot(nf[i])
plt.tight_layout()
plt.show()
```

- **Kết quả**

Kết quả của việc trực quan hóa sẽ cho thấy sự phân bố của các đặc điểm số cũng như các giá trị bất thường trong bộ dữ liệu. Những thông tin này rất hữu ích để hiểu rõ hơn về dữ liệu và chuẩn bị cho các bước xử lý tiếp theo.



- **Kết luận**

Việc trực quan hóa các đặc điểm số giúp nhà phân tích nhận diện rõ hơn về cấu trúc của dữ liệu, từ đó đưa ra các quyết định thông minh hơn trong việc xây dựng mô hình. Cả hai phương pháp, biểu đồ phân bố và boxplot, đều cung cấp thông tin quý giá về đặc điểm và các vấn đề có thể gặp phải trong dữ liệu.

3. TIỀN XỬ LÝ DỮ LIỆU

❖ Định nghĩa

Tiền xử lý dữ liệu là giai đoạn quan trọng trong quy trình phân tích dữ liệu và xây dựng mô hình học máy. Giai đoạn này bao gồm các bước chuẩn bị dữ liệu

thô để đảm bảo rằng dữ liệu có chất lượng cao, sẵn sàng để sử dụng trong các thuật toán học máy. Mục tiêu của tiền xử lý dữ liệu là cải thiện độ chính xác của mô hình và giảm thiểu các vấn đề có thể phát sinh từ dữ liệu không sạch hoặc không phù hợp.

❖ Các bước tiền xử lý dữ liệu

➢ Xóa Bất Kỳ Hàng Nào Trùng Lặp

- **Mục Tiêu**

Mục tiêu của việc xóa hàng trùng lặp trong quá trình tiền xử lý dữ liệu là:

- Cải thiện chất lượng dữ liệu: Đảm bảo rằng tập dữ liệu không chứa các bản sao không cần thiết, từ đó nâng cao tính chính xác của các phân tích và dự đoán.
- Tăng cường độ chính xác của mô hình: Bằng cách loại bỏ các hàng trùng lặp, mô hình học máy có thể học được từ những mẫu dữ liệu chính xác và đại diện, giảm thiểu rủi ro học sai lệch.
- Giảm thiểu thời gian và tài nguyên tính toán: Một tập dữ liệu nhỏ hơn sẽ làm giảm thời gian huấn luyện mô hình và tiêu tốn ít tài nguyên hơn trong quá trình xử lý.

- **Khái Niệm**

Hàng trùng lặp là những hàng có cùng giá trị cho tất cả các cột trong một DataFrame. Chúng có thể xuất hiện do lỗi trong quá trình thu thập dữ liệu hoặc trong quá trình kết hợp nhiều nguồn dữ liệu. Việc xác định và loại bỏ các hàng trùng lặp là cần thiết để cải thiện tính chính xác và độ tin cậy của phân tích.

- **Cách thực hiện**

```
counter = 0
# Lưu kích thước ban đầu của DataFrame
rs, cs = original_df.shape
# Xóa các hàng trùng lặp
```

```
df.drop_duplicates(inplace=True)
# Kiểm tra xem kích thước của DataFrame có thay đổi hay không
if df.shape == (rs, cs):
    print('\n\033[1mInference:\033[0m Bộ dữ liệu không có bất kỳ bản sao nào')
else:
    print(f'\n\033[1mInference:\033[0m Số lượng bản sao đã loại bỏ/sửa chữa ---> {rs - df.shape[0]}')
```

- **Kết quả**

Khi thực hiện đoạn mã trên, ta sẽ nhận được thông tin về số lượng hàng trong DataFrame sau khi xóa các hàng trùng lặp

```
Inference: Bộ dữ liệu không có bất kỳ bản sao nào
```

- **Kết Luận**

Xóa hàng trùng lặp là một bước không thể thiếu trong quy trình tiền xử lý dữ liệu. Việc thực hiện bước này không chỉ giúp nâng cao chất lượng dữ liệu mà còn giúp tối ưu hóa hiệu suất của mô hình học máy. Các nhà phân tích dữ liệu và nhà khoa học dữ liệu cần chú ý đến bước này để đảm bảo rằng mô hình của họ hoạt động ở mức độ tối ưu.

➢ Kiểm tra và thống kê các giá trị null

- **Mục Tiêu**

Mục tiêu của việc kiểm tra và thống kê các giá trị null trong dữ liệu là:

- Xác định mức độ thiếu hụt dữ liệu: Nhận diện các cột hoặc hàng có giá trị thiếu, từ đó có thể đưa ra các biện pháp xử lý phù hợp.
- Đánh giá ảnh hưởng của dữ liệu thiếu: Hiểu rõ tác động của dữ liệu thiếu đối với kết quả phân tích và mô hình học máy.
- Chuẩn bị cho các bước xử lý tiếp theo: Dữ liệu thiếu có thể được xử lý bằng nhiều phương pháp khác nhau, như loại bỏ, thay thế, hoặc sử dụng các thuật toán để điền giá trị.

- **Khái Niệm**

Giá trị null (hoặc NaN - Not a Number) là một giá trị không xác định hoặc không có trong tập dữ liệu. Sự xuất hiện của các giá trị null có thể do nhiều nguyên nhân, bao gồm lỗi trong quá trình thu thập dữ liệu, các câu hỏi không được trả lời trong khảo sát, hoặc dữ liệu không có sẵn từ các nguồn bên ngoài.

- **Cách thực hiện**

```
# Kiểm tra các phần tử null
nvc = pd.DataFrame(df.isnull().sum().sort_values(),
columns=['Tổng số giá trị Null'])
nvc['Tỷ lệ phần trăm'] = round(nvc['Tổng số giá trị Null']
/ df.shape[0], 3) * 100
print(nvc)
```

- **Kết quả**

Khi thực hiện đoạn mã trên, chúng ta sẽ nhận được một DataFrame mới chứa số lượng và tỷ lệ phần trăm của các giá trị null trong từng cột.

	Tổng số giá trị Null	Tỷ lệ phần trăm
TV	0	0.0
Radio	0	0.0
Newspaper	0	0.0
Sales	0	0.0

- **Kết Luận**

  Kiểm tra và thống kê các giá trị null là bước quan trọng trong quy trình tiền xử lý dữ liệu. Điều này không chỉ giúp nhận diện vấn đề trong tập dữ liệu mà còn giúp đưa ra các phương pháp xử lý phù hợp, đảm bảo rằng mô hình học máy hoạt động hiệu quả và chính xác.

➢ Loại bỏ các giá trị ngoại lai

- **Mục Tiêu**

  Mục tiêu của việc loại bỏ các giá trị ngoại lai trong tập dữ liệu là:

  - Cải thiện chất lượng dữ liệu: Giá trị ngoại lai có thể gây ra sự thiên lệch trong các phân tích và kết quả mô hình.
  - Tăng cường độ chính xác của mô hình: Bằng cách loại bỏ các giá trị không đại diện cho tập dữ liệu, mô hình có thể học được các mẫu chính xác hơn.
  - Đảm bảo tính ổn định: Giúp mô hình trở nên ổn định hơn và giảm thiểu khả năng xảy ra overfitting.

- **Khái Niệm**

  Giá trị ngoại lai (outlier) là các điểm dữ liệu nằm xa so với các điểm dữ liệu khác trong cùng một tập hợp. Những giá trị này có thể xuất phát từ nhiều nguyên nhân, như lỗi trong quá trình thu thập dữ liệu hoặc sự biến động tự nhiên trong dữ liệu. Việc không xử lý các giá trị ngoại lai có thể dẫn đến kết quả phân tích không chính xác.

- **Cách thực hiện**

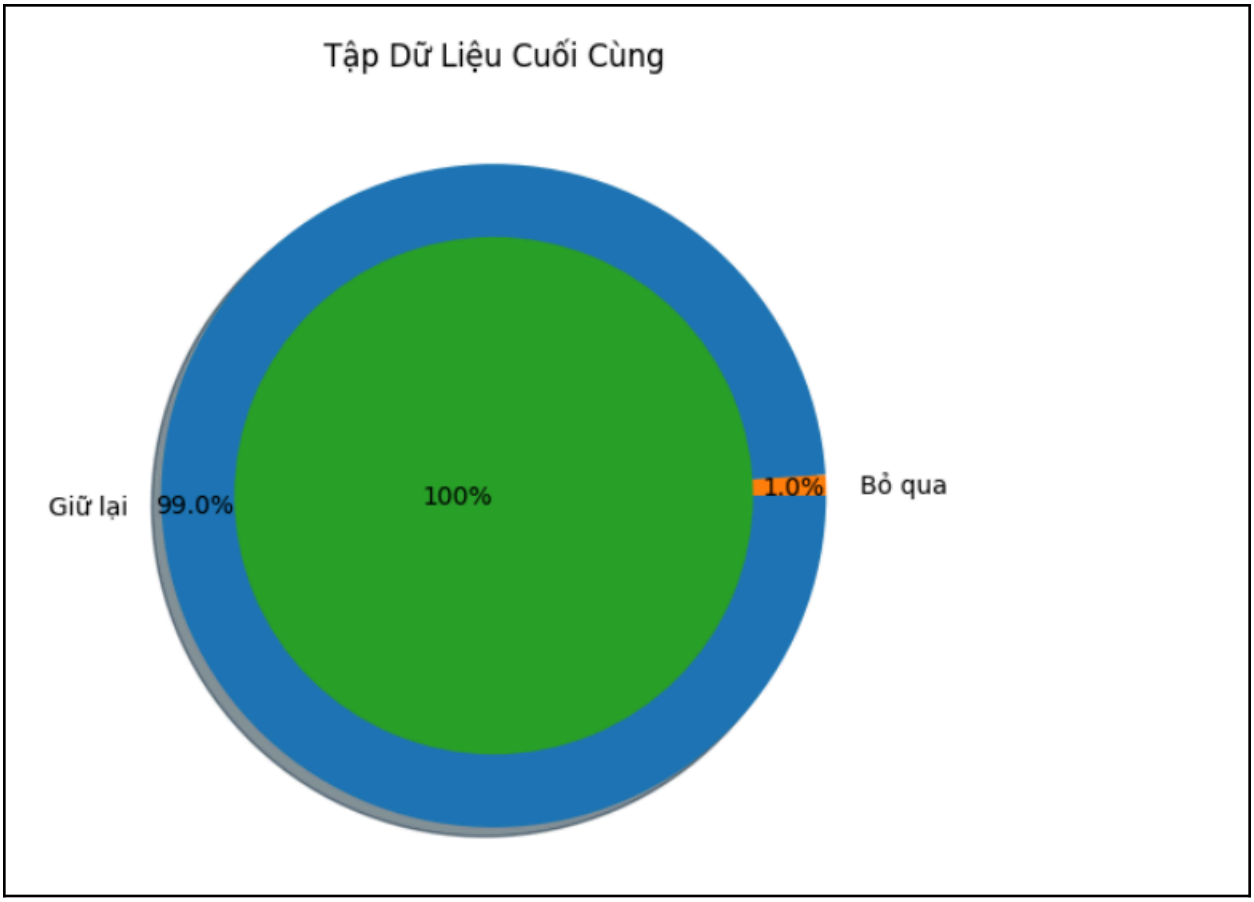
```
df1 = df.copy()
df3 = df1.copy()
df1 = df.copy()
# features1 chứa các đặc trưng số
# nf là danh sách các cột số trong dữ liệu
features1 = nf
for i in features1:
    Q1 = df1[i].quantile(0.25)
    Q3 = df1[i].quantile(0.75)
    IQR = Q3 - Q1
    # Lọc các giá trị không vượt quá ngưỡng trên
    df1 = df1[df1[i] <= (Q3 + (1.5 * IQR))]
    # Lọc các giá trị không vượt quá ngưỡng dưới
    df1 = df1[df1[i] >= (Q1 - (1.5 * IQR))]
    # Đặt lại chỉ số cho DataFrame sau khi lọc
    df1 = df1.reset_index(drop=True)
display(df1.head())
print('\n\033[1mInference:\033[0m\nTrước khi loại bỏ các ngoại lệ, Bộ dữ liệu có {} mẫu.'.format(df3.shape[0]))
print('Sau khi loại bỏ các ngoại lệ, tập dữ liệu hiện có {} mẫu.'.format(df1.shape[0]))
```

- **Kết quả**

Sau khi thực hiện loại bỏ các giá trị ngoại lai, ta sẽ có một DataFrame mới df_cleaned chứa dữ liệu đã được làm sạch.

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

Inference:
Trước khi loại bỏ các ngoại lệ, Bộ dữ liệu có 200 mẫu.
Sau khi loại bỏ các ngoại lệ, tập dữ liệu hiện có 198 mẫu.



- **Kết Luận**

Loại bỏ các giá trị ngoại lai là một bước quan trọng trong quy trình tiền xử lý dữ liệu. Điều này không chỉ cải thiện chất lượng của dữ liệu mà còn tăng cường hiệu suất và độ chính xác của các mô hình học máy. Bằng cách sử dụng các phương pháp như IQR, chúng ta có thể xác định và loại bỏ các giá trị không đại diện một cách hiệu quả.

4. LỰA CHỌN VÀ CHIẾT SUẤT TÍNH NĂNG

❖ Mục Tiêu

Mục tiêu của việc lựa chọn và trích xuất tính năng là:

- Tối ưu hóa mô hình: Giảm số lượng đặc trưng không cần thiết, giúp cải thiện hiệu suất và tốc độ huấn luyện của mô hình.
- Giảm thiểu overfitting: Bằng cách loại bỏ các đặc trưng không quan trọng, chúng ta có thể giảm thiểu nguy cơ mô hình học quá nhiều từ dữ liệu huấn luyện.
- Cải thiện khả năng giải thích: Giúp dễ dàng hơn trong việc giải thích kết quả của mô hình bằng cách sử dụng các đặc trưng có ý nghĩa.

❖ Khái Niệm

Lựa chọn tính năng (feature selection) là quá trình xác định các đặc trưng quan trọng nhất để sử dụng trong mô hình học máy. Ma trận tương quan (correlation matrix) là một công cụ hữu ích để đánh giá mối quan hệ giữa các đặc trưng và biến mục tiêu, từ đó xác định các đặc trưng có ảnh hưởng mạnh nhất đến kết quả.

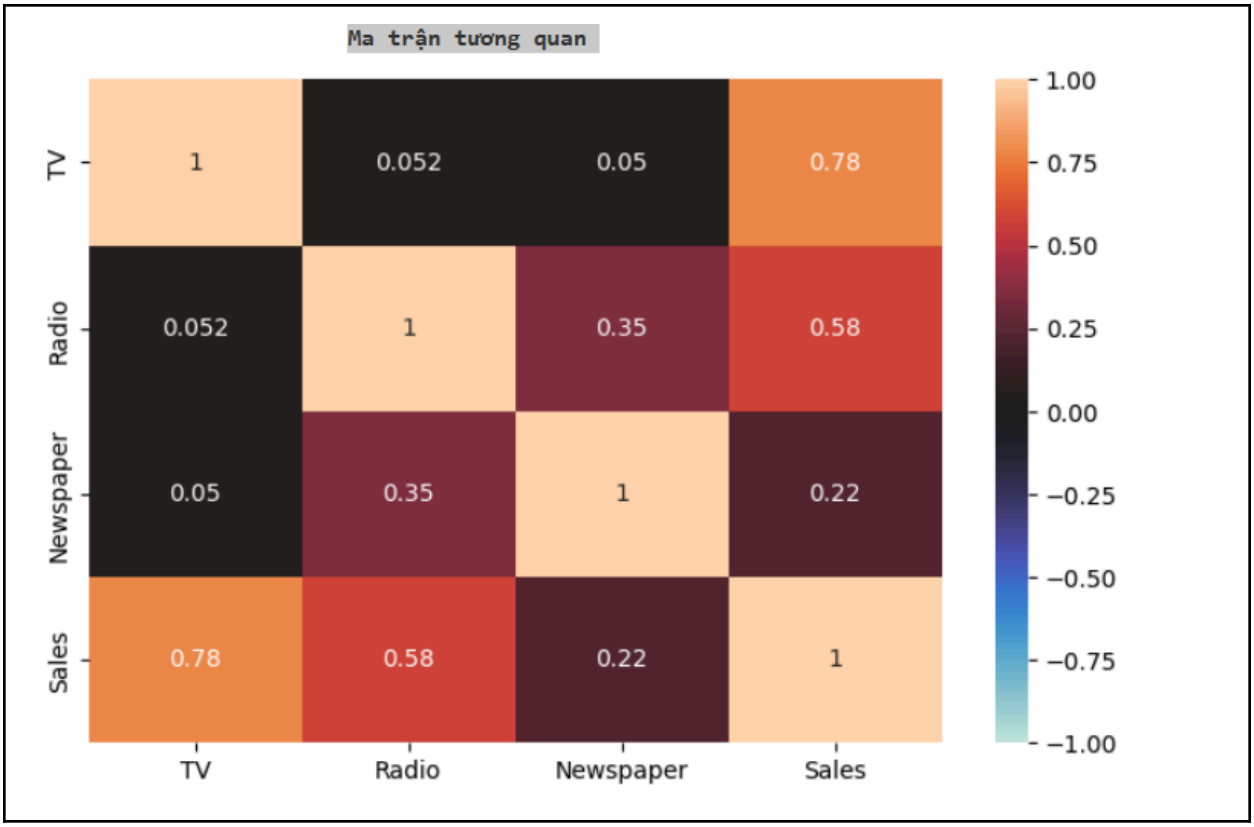
❖ Cách thực hiện

```
print('\033[1mMa trận tương quan'.center(70))
plt.figure(figsize=[8,5])
```

```
sns.heatmap(df.corr(), annot=True, vmin=-1, vmax=1,
center=0)
plt.show()
```

❖ Kết quả

Khi chạy mã trên, ta sẽ nhận được một ma trận tương quan được hiển thị dưới dạng biểu đồ nhiệt (heatmap) và danh sách các đặc trưng có tương quan cao với biến mục tiêu.



❖ Kết Luận

Việc lựa chọn và trích xuất tính năng là một bước quan trọng trong quy trình xây dựng mô hình học máy. Sử dụng ma trận tương quan là một phương pháp hiệu

quả để xác định các đặc trưng có ảnh hưởng mạnh nhất đến biến mục tiêu, từ đó giúp tối ưu hóa mô hình và cải thiện độ chính xác.

5. THAO TÁC DỮ LIỆU

❖ Định nghĩa

Thao tác dữ liệu là quá trình xử lý và biến đổi dữ liệu trước khi áp dụng vào các mô hình học máy. Các thao tác này giúp đảm bảo dữ liệu sẵn sàng để mô hình có thể học hiệu quả và chính xác nhất

❖ Các bước tiền xử lý dữ liệu

➢ Chia Dữ Liệu Thành Các Tập Huấn Luyện, Xác Thực, Và Kiểm Tra

- **Mục tiêu**

Chia dữ liệu thành các tập khác nhau để mô hình có thể học từ tập huấn luyện, tối ưu qua tập xác thực, và kiểm tra độ chính xác trên tập kiểm tra. Việc này giúp mô hình tránh hiện tượng overfitting và đánh giá hiệu quả trên dữ liệu chưa từng thấy.

- **Khái niệm**

Quá trình chia dữ liệu thành các tập riêng biệt cho huấn luyện, xác thực và kiểm tra là một bước quan trọng trong học máy.

- **Cách thực hiện**

```
target = 'Sales'
X = df1.drop([target], axis=1)
Y = df1[target]
Train_X, Temp_X, Train_Y, Temp_Y = train_test_split(X, Y,
train_size=0.65,
test_size=0.35,shuffle=True,random_state=100 )
# Phân chia tập còn lại thành tập validation (50%) và tập
```

```
kiểm tra (50%)
Validation_X, Test_X, Validation_Y, Test_Y =
train_test_split(Temp_X, Temp_Y, train_size=0.5,
test_size=0.5, shuffle=True,random_state=100)
# Hiển thị kích thước của các tập dữ liệu
print('Original set  ---> ', X.shape, Y.shape)
print('Training set  ---> ', Train_X.shape, Train_Y.shape)
print('Validation set ---> ', Validation_X.shape,
Validation_Y.shape)
print('Testing set   ---> ', Test_X.shape, Test_Y.shape)
```

- **Kết quả**

  Chia tệp thành công

```
Original set  --->  (198, 3) (198,)
Training set  --->  (128, 3) (128,)
Validation set --->  (35, 3) (35,)
Testing set   --->  (35, 3) (35,)
```

- **Kết luận:**

  Việc chia dữ liệu thành các tập riêng biệt giúp đảm bảo rằng mô hình có khả năng học từ dữ liệu một cách hiệu quả, đồng thời có thể được đánh giá khách quan trên dữ liệu chưa từng thấy. Điều này giúp giảm thiểu nguy cơ overfitting và cải thiện khả năng tổng quát hóa của mô hình trong thực tế.

➢ Chuẩn Hóa Dữ Liệu

- **Mục tiêu**

  Mục tiêu của chuẩn hóa là đưa tất cả các đặc trưng về cùng một thang đo, đảm bảo rằng chúng không có ảnh hưởng không đáng có đến mô hình chỉ vì sự khác

biệt về đơn vị hoặc kích thước. Thông thường, chuẩn hóa dữ liệu sẽ đưa các đặc trưng về giá trị trung bình bằng 0 và phương sai bằng 1.

- **Khái niệm**

  Chuẩn hóa dữ liệu (data normalization hoặc data standardization) là một bước tiền xử lý quan trọng trong quá trình xây dựng mô hình học máy

- **Cách thực hiện**

```
std = StandardScaler()
print('\033[1mStandardardization on Training
set'.center(120))
 # Chuẩn hóa dữ liệu huấn luyện
Train_X_std = std.fit_transform(Train_X)
 # Chuyển đổi về DataFrame
Train_X_std = pd.DataFrame(Train_X_std,
columns=Train_X.columns)
# Hiển thị thông tin thống kê
display(Train_X_std.describe())
print('\n', '\033[1mStandardardization on Validation
set'.center(120))
# Chuẩn hóa dữ liệu validation
Validation_X_std = std.transform(Validation_X)
# Chuyển đổi về DataFrame
Validation_X_std = pd.DataFrame(Validation_X_std,
columns=Validation_X.columns)
# Hiển thị thông tin thống kê
display(Validation_X_std.describe())
print('\n', '\033[1mStandardardization on Testing
set'.center(120))
# Chuẩn hóa dữ liệu Test
```

```
Test_X_std = std.transform(Test_X)
# Chuyển đổi về DataFrame
Test_X_std = pd.DataFrame(Test_X_std,
columns=Test_X.columns)
display(Test_X_std.describe())
```

- **Kết quả**

  Dữ liệu trong các tập huấn luyện, xác thực và kiểm tra đã được chuẩn hóa bằng StandardScaler

Standardardization on Training set

	TV	Radio	Newspaper
count	1.280000e+02	1.280000e+02	1.280000e+02
mean	1.387779e-16	-1.040834e-16	-6.938894e-17
std	1.003929e+00	1.003929e+00	1.003929e+00
min	-1.751747e+00	-1.495246e+00	-1.488095e+00
25%	-7.851696e-01	-8.953203e-01	-7.869650e-01
50%	4.652318e-02	-3.367144e-02	-2.334411e-01
75%	8.552442e-01	8.211794e-01	6.927891e-01
max	1.586804e+00	1.856178e+00	2.635043e+00

Standardardization on Validation set

	TV	Radio	Newspaper
count	35.000000	35.000000	35.000000
mean	-0.215522	0.095685	-0.110014
std	1.016136	1.115297	1.071448
min	-1.638656	-1.515640	-1.522537
25%	-1.028434	-0.825641	-0.988694
50%	-0.421746	-0.088056	-0.410569
75%	0.604323	1.298740	0.750602
max	1.612721	1.733813	2.861373

Standardardization on Testing set

	TV	Radio	Newspaper
count	35.000000	35.000000	35.000000
mean	-0.405085	0.225624	-0.298388
std	0.962493	0.928868	0.936699
min	-1.829498	-1.413669	-1.448734
25%	-1.322354	-0.448351	-1.094478
50%	-0.286272	0.156673	-0.528654
75%	0.314526	1.132189	0.342224
max	1.620967	1.815390	2.359511

- **Kết luận**

Chuẩn hóa dữ liệu giúp mô hình học hiệu quả hơn bằng cách đưa các đặc trưng về cùng một thang đo. Điều này đặc biệt hữu ích với các thuật toán dựa trên khoảng cách như hồi quy tuyến tính, mạng nơ-ron, hay các mô hình liên quan đến gradient descent. Việc chuẩn hóa trước huấn luyện giúp tránh trường hợp một đặc trưng có ảnh hưởng không đáng có đến kết quả học.

III. KẾT LUẬN

- Từ những kết luận và lợi ích trên thì ta thấy chuẩn bị dữ liệu là bước quan trọng và không thể thiếu trong quá trình xây dựng mô hình học máy. Nó đóng vai trò quyết định đến hiệu quả và độ chính xác của mô hình. Một quy trình chuẩn bị dữ liệu tốt đảm bảo rằng dữ liệu đầu vào là chất lượng, đúng định dạng, và phản ánh đầy đủ các yếu tố cần thiết cho quá trình huấn luyện mô hình.

Từ khám phá, tiền xử lý, thao tác dữ liệu, đến lựa chọn và trích xuất tính năng, mỗi bước đều góp phần giúp mô hình học máy hoạt động hiệu quả hơn, hạn chế sai lệch và tăng khả năng dự đoán. Kết quả cuối cùng là mô hình có độ chính xác cao, ổn định và khả năng tổng quát hóa tốt trên dữ liệu thực tế.

Vì vậy nên dành thời gian để chuẩn bị dữ liệu kỹ lưỡng là một yếu tố không thể bỏ qua trong quá trình phát triển bất kỳ mô hình học máy nào.