UEH UNIVERSITY
Trường Kinh Doanh
KHOA TÀI CHÍNH

Tiểu luận kết thúc học phần

Khoa học dữ liệu tài chính

ỨNG DỤNG PYTHON DỰ BÁO NGUY CƠ VỠ NỢ CỦA DOANH NGHIỆP BẰNG PHƯƠNG PHÁP HỒI QUY LOGISTIC

Học viên thực hiện

Nguyễn Công Hiếu (524102110660)
hieunguyen.524102110660@st.ueh.edu.vn

Giảng viên hướng dẫn

PGS.TS. Phùng Đức Nam | TS. Trần Hoài Nam

TP. Hồ Chí Minh, Tháng 11 Năm 2025

Trường
Kinh Doanh

Khoa Tài chính

B1.902, 279 Nguyễn Tri Phương,
Phường Diên Hồng, TP. Hồ Chí Minh

Tel: +84-28 3526 5830
Email: sof@ueh.edu.vn

Ứng dụng Python dự báo nguy cơ vỡ nợ của doanh nghiệp bằng phương pháp hồi quy logistic

Tháng 11 Năm 2025

Tóm tắt Tiểu luận này mô tả quá trình huấn luyện và dự báo nguy cơ vỡ nợ của các doanh nghiệp dựa trên hồi quy logistic. Mẫu dữ liệu được sử dụng được thu thập từ báo cáo tài chính hợp nhất kiểm toán của các doanh nghiệp niêm yết tại Việt Nam trong giai đoạn 2010 - 2024. Kết quả huấn luyện và dự báo chỉ ra rằng hồi quy logistic là một phương pháp khả thi để dự báo nguy cơ vỡ nợ của doanh nghiệp trên thực tế, đồng thời việc sử dụng dữ liệu lớn hơn và nhiều độ trễ dự báo hơn có thể cải thiện được hiệu quả dự báo. Đồng thời qua nghiên cứu, tác giả cho rằng các biện pháp xử lý dữ liệu đầu vào mới, các biến dự báo mới có thể được sử dụng thêm trong tương lai nhằm cải thiện hiệu quả của phương pháp dự báo.

Mục lục

1 Giới thiệu

1.1 Lý do chọn chủ đề

Vỡ nợ là hiện tượng một doanh nghiệp không thể hoàn thành được các nghĩa vụ nợ của bản thân. Trên thực tế, nhà phân tích quan tâm đến rủi ro vỡ nợ bởi nhiều nguyên nhân. Thứ nhất, rủi ro vỡ nợ ảnh hưởng đến trực tiếp đến định giá thông qua phần bù rủi ro vỡ nợ hoặc chênh lệch tín dụng (Reilly & Brown, 2011, Bodie và c.s. (2018), Hull (2023)). Thứ hai, rủi ro vỡ nợ ảnh hưởng đến giá trị của cổ phiếu, tức là sự sống còn của cổ đông (Ross và c.s., 2019, Brealey và c.s. (2019)). Bên cạnh đó, rủi ro vỡ nợ còn là chỉ báo cho sức khỏe dòng tiền của doanh nghiệp, hoặc là một dấu hiệu dự báo cho các sự kiện kinh tế vĩ mô trong tương lai (chẳng hạn như các cuộc khủng hoảng kinh tế).

Trên thực tế, có nhiều cách để đánh giá nguy cơ vỡ nợ của một doanh nghiệp, ví dụ như ước lượng dựa trên xếp hạng tín dụng của các tổ chức xếp hạng như Standard and Poors, Moodys, Fitch; sử dụng tỷ lệ vỡ nợ (hazard rate), ước lượng dựa trên chênh lệch tính dụng (credit speards), sử dụng định giá cổ phiếu dựa trên mô hình của Merton (1974)... Trong tiểu luận này, tác giả sử dụng hồi quy logistic dựa trên nhị phân hóa chỉ số Altman's Z-Score để dự báo nguy cơ vỡ nợ của doanh nghiệp. Mục tiêu của tiểu luận này là tìm kiếm một phương pháp khả thi để dự báo nguy cơ vỡ nợ, đồng thời đề xuất các hướng phát triển tiếp theo trong việc xác định mô hình dự báo vỡ nợ nói riêng và dự báo các hiện tượng tài chính nói chung.

1.2 Mô tả bài toán

1.2.a Yêu cầu

Tác giả đặt ra yêu cầu sử dụng mô hình logistic (trích dẫn) để dự báo về nguy cơ vỡ nợ của doanh nghiệp dựa trên các yếu tố tài chính trong lịch sử, bao gồm: Biên lợi nhuận ròng (NPM), Lợi nhuận trên tổng tài sản (ROA), Tỷ số thanh toán hiện hành (CR), Tỷ số nợ trên vốn chủ sở hữu (DTE), Vòng quay khoản phải thu (RTR), Vòng quay tổng tài sản (ATR), Quy mô doanh nghiệp (SIZE) và mức độ thâm dụng vốn (FAR), theo đề xuất của Altman (1968), Ohlson (1980), Lupu & Onofrei (2014); Xu & Zhang (2009). Nguy cơ vỡ nợ được đại diện bởi chỉ số Z-Score, được đề xuất bởi Altman (1968).

1.2.b Mô hình dự báo

Hồi quy logistic là một loại mô hình phân loại thống kê xác suất. Nó được sử dụng như một mô hình nhị phân để dự báo một phản hồi nhị phân - cụ thể là kết quả của một biến phụ thuộc dạng phân loại dựa trên một hoặc nhiều biến độc lập và/hoặc biến dự báo

Dạng tổng quát của hồi quy logistic được trình bày dưới đây

$$\Pr(Y = 1 \mid X) = \frac{1}{1 + e^{-\lambda X}}$$

Trong đó λX là một tổ hợp tuyến tính của k biến độc lập $X_1, X_2, ..., X_k$

$$\lambda X = \lambda_0 + \lambda_1 X_1 + \lambda_2 X_2 + ... + \lambda_k X_k$$

Với $\Pr = \Pr(Y = 1 \mid X)$, dưới đây là một dạng viết khác của hồi quy logistic dưới dạng log-odds,

$$\ln\left(\frac{\Pr}{1-\Pr}\right) = \lambda_0 + \lambda_1 X_1 + \lambda_2 X_2 + ... + \lambda_k X_k$$

Trong tiểu luận này, tác giả áp dụng hồi quy logistic dựa trên phương trình dưới đây:

$$\Pr(Z = 1) = \frac{1}{1 + e^{-BX}}$$

Trong mô hình, $\Pr(Z = 1)$ là xác suất công ty đối mặt với nguy cơ vỡ nợ, tức là giá trị Z-Score < 1.8 (Altman, 1968). Bên cạnh đó, B là ma trận các hệ số ước lượng, và X là ma trận các biến dự báo, bao gồm: Biên lợi nhuận ròng (NPM, Lợi nhuận sau thuế/Doanh thu), Lợi nhuận trên tổng tài sản (ROA, Lợi nhuận sau thuế/Tổng tài sản), Tỷ số thanh toán hiện hành (CR, Tài sản hiện hành/Nợ ngắn hạn), Tỷ số nợ trên vốn chủ sở hữu (DTE, Tổng nợ/Vốn chủ sở hữu), Vòng quay khoản phải thu (RTR, Doanh thu bán chậm/Khoản phải thu bình quân), Vòng quay tổng tài sản (ATR, Doanh thu thuần/Tổng tài sản bình quân), Quy mô doanh nghiệp (SIZE, logarit tự nhiên vốn hóa doanh nghiệp) và mức độ thâm dụng vốn (FAR, tổng giá trị tài sản hữu hình bình quân trên tổng tài sản).

Theo Altman (1968), Z-Score được tính như sau:

$$Z = 1.2X_1 + 1.4X_2 + 3.3X_3 + 0.6X_4 + 1.0X_5$$

Trong đó, X_1 bằng vốn lưu động ròng trên tổng tài sản, X_2 bằng lợi nhuận giữ lại trên tổng tài sản, X_3 bằng lợi nhuận trước thuế và lãi vay (EBIT) trên tổng tài sản, X_4 bằng vốn hóa thị trường trên tổng nợ phải trả, và X_5 bằng doanh thu trên tổng tài sản.

1.2.c Dữ liệu

Tác giả sử dụng dữ liệu từ báo cáo tài chính hợp nhất kiểm toán của tất cả các công ty đang niêm yết tại Sở Giao dịch Chứng khoán (GDCK) TP. Hồ Chí Minh (HoSE), Sở GDCK Hà Nội (HNX) và các công ty hiện đang được niêm yết tại sàn UpCOM. Mẫu dữ liệu gốc ban đầu được truy xuất từ nền tảng dữ liệu FiinProX, bao gồm tổng cộng 1647 doanh nghiệp với 24705 quan sát. Tác giả sử dụng Microsoft Visual Basic for Application (VBA) để chuyển dữ liệu thô truy xuất từ nền tảng về dạng dữ liệu bảng (panel data) trước khi thực hiện các tính toán tiếp theo.

2 Thực thi mô hình dự báo

2.1 Một số thiết lập ban đầu

Bước đầu tiên trong thực hiện dự báo là đọc dữ liệu và tính toán các biến để phục vụ cho việc tính toán trong các bước tiếp theo. Quá trình này cần sử dụng hai thư viện là `numpy` và `pandas`, được khai báo như dưới đây

```
import numpy as np
import pandas as pd
```

Để tiện cho việc trình bày tiểu luận, tác giả đã tắt các cảnh báo `FutureWarning` và `RuntimeWarning`, cũng như thiết lập hiển thị làm tròn đến 3 chữ số thập phân đối với các giá trị là số thực (`float`) trong đối tượng `pd.DataFrame`. Các thiết lập ấy được trình bày dưới đây

```
# Turn off some warnings
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.simplefilter(action='ignore', category=RuntimeWarning)
```

```
# Set up display float format
pd.options.display.float_format = '{:.3f}'.format
```

2.2 Đọc dữ liệu và tính toán giá trị của các biến

2.2.a Tính chỉ số Z-Score

Dữ liệu đầu vào để tính Altman's Z-Score được lưu trong tệp `z.xlsx`. Dưới đây là quá trình đọc và thiết lập dữ liệu dạng bảng này trong `python`.

```
# Import dataset of Z-score factors
z = pd.read_excel('data/main/z.xlsx')
z.set_index(['Firm', 'Year'], inplace=True)
z.sort_index(inplace=True)
```

Bước tiếp theo, tác giả thực hiện tính chỉ số Z-Score thông qua hai bước: (1) Tính các thành phần X_1, X_2, X_3, X_4, X_5, và (2) Tính chỉ số Z-Score. Sau đó, tác giả tạo biến nhị phân Z, trong đó $Z = 1$ khi công ty đối mặt với nguy cơ vỡ nợ (tức là Z-Score < 1.8, và $Z = 0$ trong trường hợp còn lại)

```
# Calculate Z-Score components
z_factors = pd.DataFrame({
    'Exchange': z['Exchange'],
    'X1': (z['CurrentAssets'] - z['CurrentLiabilities']) /
z['TotalAssets'],
    'X2': z['RetainedEarnings'] / z['TotalAssets'],
    'X3': (z['EarningBeforeTax'] - z['InterestExpense']) /
z['TotalAssets'],
```

```
    'X4': z['MarketCapitalization'] / z['TotalLiabilities'],
    'X5': z['Sales'] / z['TotalAssets']
})

# Calculate Z-Score and Z classification
z_factors['Z-Score'] = (1.2 * z_factors['X1'] +
                        1.4 * z_factors['X2'] +
                        3.3 * z_factors['X3'] +
                        0.6 * z_factors['X4'] +
                        1.0 * z_factors['X5'])

# Z classification: 1 if Z-Score < 1.8 else 0
z_factors['Z'] = np.where(
    z_factors['Z-Score'].isna(),
    np.nan,
    np.where(z_factors['Z-Score'] >= 1.8, 0, 1)
)
```

2.2.b Tính các biến dự báo

Dữ liệu đầu vào cho các biến dự báo được lưu trong tệp `factors.xlsx`. Quy trình đọc và thiết lập dữ liệu dạng bảng tương tự đối với tệp `z.xlsx` ở trên.

```
# Import dataset of forecast features
features = pd.read_excel('data/main/factors.xlsx')
features.set_index(['Firm', 'Year'], inplace=True)
features.sort_index(inplace=True)
```

Quá trình tính các biến dự báo được tác giả trình bày dưới đây. Trong quá trình này, một số giá trị có thể nhận kết quả là âm vô cực hoặc dương vô cực. Điều này xảy ra bởi một số phép toán như lấy logarit tự nhiên của số không (0) hoặc một phép chia cho không (0) có thể được thực hiện. Những giá trị này cần được loại bỏ tương tự với `NaN` trong các tính toán tiếp theo, vì vậy tác giả quyết định thay thế các giá trị vô cực này bằng `NaN` trước khi thực hiện các tính toán tiếp theo.

```
# Calculate forecast features
# Calculate forecast features
forecast_features = pd.DataFrame({
    'NPM':  features['EarningAfterTax'] / features['Sales'],
    'ROE':  features['EarningAfterTax'] / features['TotalEquity'],
    'QR':   (features['CurrentAssets'] - features['Inventory']) /
features['CurrentLiabilities'],
    'DTE':  (features['TotalAssets'] - features['TotalEquity']) /
features['TotalEquity'],
    'RTR':  features['Sales'] / ((features['ShortTermReceivable'] +
features['LongTermReceivable'] + features['ShortTermReceivable'].shift(1) +
features['LongTermReceivable'].shift(1)) / 2),
    'ATR':  features['Sales'] / ((features['TotalAssets'] +
features['TotalAssets'].shift(1)) / 2),
```

```
    'SIZE': np.log(features['MarketCapitalization']),
    'FAR':  features['PPE'] / features['TotalAssets']
})

# Clean infinite values
forecast_features.replace([np.inf, -np.inf], np.nan, inplace=True)
```

2.2.c Kết nối các bảng dữ liệu

Tác giả kết hợp các cột cần thiết từ 2 đối tượng `pd.DataFrame` ở trên bằng hàm `pd.merge()`. Chỉ các dòng cùng tồn tại ở hai đối tượng được giữ lại sau khi thực hiện ghép nối. Bên cạnh đó, bất kỳ dòng nào chứa giá trị `NaN` đều bị loại bỏ trước khi thực hiện các bước tiếp theo.

```
# Merge Z-Score and forecast features
df = pd.merge(
    z_factors[['Exchange', 'Z-Score', 'Z']],
    forecast_features,
    left_index=True,
    right_index=True,
    how='inner'
)
# Drop NaN value
df.dropna(inplace=True)
```

2.3 Mô tả dữ liệu

Trước khi thực hiện dự báo, tác giả thực hiện một số thống kê mô tả nhằm hiểu rõ hơn về dữ liệu được sử dụng trong huấn luyện và dự báo. Trước hết, mẫu dữ liệu sau khi được xử lý có khoảng 1370 công ty (tùy từng năm), với mẫu ở năm gần nhất bao gồm 351 công ty niêm yết tại HoSE, 291 công ty niêm yết tại HNX và 855 công ty niêm yết tại UpCOM.

```
# Count number of firms by Exchange based on the latest year available
## First, retrieve the latest year for each firm
df_latest =
df.reset_index().sort_values('Year').drop_duplicates(subset=['Firm'],
keep='last')

# Then, count the number of firms by Exchange
count_latest = df_latest['Exchange'].value_counts()

# Display the counts
print(count_latest)
```

```
Exchange
UPCoM    855
HOSE     351
HNX      291
Name: count, dtype: int64
```

Thống kê dưới đây cho thấy số lượng công ty đối mặt với nguy cơ vỡ nợ (Z-Score < 1.8), có rủi ro vỡ nợ thấp (Z-Score $>= 3$), hoặc nằm trong vùng xám ($1.8 <= Z$-Score < 3). Thống kê này chỉ ra rằng trong giai đoạn 2010-2012, có một tỷ lệ tương đối lớn các công ty đối mặt với nguy cơ vỡ nợ. Điều này phù hợp với bối cảnh lịch sử kinh tế Việt Nam vốn đối mặt với nhiều thách thức lớn trong cùng thời kỳ.

```
# Count number of firms by range of Z-Score for each year
## Define bins and labels
bins = [-np.inf, 1.8, 3, np.inf]
labels = ['High Risk', 'Grey Area', 'Low Risk']

# Classify Z-Score into ranges
df['Z-Score Range'] = pd.cut(df['Z-Score'], bins=bins, labels=labels)
count_by_year = df.groupby(['Year', 'Z-Score
Range']).size().unstack(fill_value=0)
count_by_year['Total'] = count_by_year.sum(axis=1)

# Display the counts
print(count_by_year)
```

```
Z-Score Range  High Risk  Grey Area  Low Risk  Total
Year
2010                 189        160       187    536
2011                 290        156       196    642
2012                 319        138       205    662
2013                 296        163       218    677
2014                 302        182       284    768
2015                 365        208       349    922
2016                 451        281       408   1140
2017                 494        307       416   1217
2018                 502        343       432   1277
2019                 541        325       437   1303
2020                 486        321       471   1278
2021                 472        343       546   1361
2022                 523        312       528   1363
2023                 540        286       542   1368
2024                 489        317       561   1367
```

Phần tiếp theo trình bày thống kê mô tả của bảng dữ liêu. Tại đây, chúng ta nhận thấy rằng mẫu được sử dụng bao gồm 15881 quan sát trải dài suốt 15 năm nghiên cứu. Tại đây, tác giả chú ý đến các thống kê mô tả của Z-Score, với giá trị trung bình và trung vị lần lượt là 3.521 và 2.261. Giá trị trung bình này cao hơn 1.8 nhưng nhỏ hơn trung vị chỉ ra rằng phân phối của Z-Score có thể là một phân phối lệch phải, hàm ý đa số công ty trong mẫu nghiên cứu ít phải đối mặt với nguy cơ vỡ nợ nghiêm trọng và có xác suất cao tìm thấy một công ty có rủi ro vỡ nợ rất thấp. Điều này phù hợp với giá trị trung bình của Z là 0.394, hàm ý rằng chỉ 39.4% số quan sát doanh nghiệp-năm có dấu hiệu phải đối mặt với nguy cơ vỡ nợ nghiêm trọng. Bên cạnh đó, tác giả cũng nhận thấy rằng tồn tại sự chênh lệch lớn về mặt giá trị giữa các biến dự

báo, và do đó tác giả cần thực hiện chuẩn hóa các biến này trước khi thực hiện huấn luyện và dự báo trong bước tiếp theo.

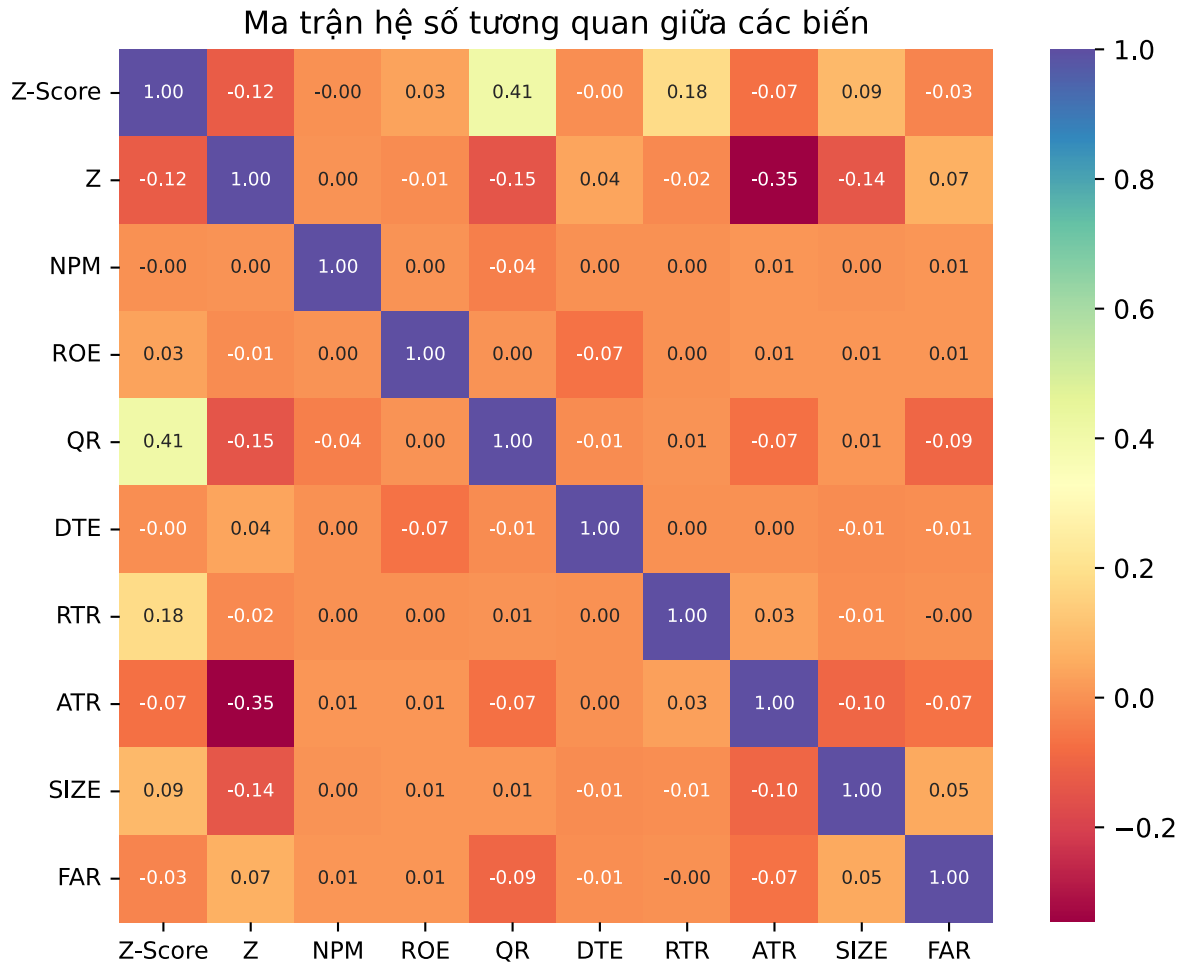
```
# Descriptive statistics
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Z-Score	15881.000	3.521	24.664	−1398.710	1.255	2.261	3.930	1357.368
Z	15881.000	0.394	0.489	0.000	0.000	0.000	1.000	1.000
NPM	15881.000	−3.375	413.961	−51766.712	0.009	0.038	0.101	4922.179
ROE	15881.000	−0.050	15.314	−1908.428	0.025	0.088	0.163	186.272
QR	15881.000	2.067	6.223	0.001	0.610	0.990	1.791	384.841
DTE	15881.000	1.934	29.622	−1634.473	0.380	0.937	1.996	1891.871
RTR	15881.000	12.729	378.274	−0.981	1.765	4.122	8.671	44397.609
ATR	15881.000	1.160	1.397	−0.909	0.369	0.831	1.472	42.208
SIZE	15881.000	26.108	1.867	19.249	24.837	25.919	27.212	33.725
FAR	15881.000	0.222	0.225	0.000	0.048	0.143	0.320	0.986

Trong phần tiếp theo, tác giả trình bày ma trận hệ số tượng quan giữa các biến. Tác giả nhận thấy rằng không có một cặp biến dự báo nào có tương quan lớn với nhau, và do đó tác giả tin rằng vấn đề đa cộng tuyến, vốn làm cho kết quả ước lượng trở nên nhạy cảm, có ít khả năng xảy ra.

```
# Correlation matrix
# Calculate correlation matrix for numeric columns
corrmat = df.select_dtypes(include=[np.number]).corr()

# Use seaborn to create a heatmap
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))
sns.heatmap(corrmat, annot=True, fmt=".2f", cmap='Spectral', square=True,
annot_kws={"size": 7})
plt.xticks(fontsize=9)
plt.yticks(fontsize=9, rotation=0)
plt.title('Ma trận hệ số tương quan giữa các biến')
plt.show()
```



2.4 Thiết lập huấn luyện và kiểm thử

Bước đầu tiên, tác giả tạo một đối tượng `pd.DataFrame` mới để lưu trữ dữ liệu cần thiết cho việc huấn luyện và dự báo. Các cột `Exchange`, `Z-Score` và `Z-Score Range` không còn cần thiết trong các bước tiếp theo, và được tác giả loại ra khỏi đối tượng `pd.DataFrame` mới.

```
data = df.drop(['Exchange','Z-Score','Z-Score Range'], axis=1)
```

Tiếp theo, tác giả định nghĩa hàm `split_train_test()` với mục tiêu chia dữ liệu thành hai phần: tập huấn luyện (`train`) và tập kiểm thử (`test`). Hàm này bao gồm 4 tham số: (1) `data` (`DataFrame`) là mẫu đầu vào, (2) `cutoff` (`int`) là điểm thời gian cắt mẫu, (3) `cols_to_lag` (`list`) là danh sách các biến lấy độ trễ, và (4) `lags` (`int`): số độ trễ sử dụng. Tại đây, tác giả không dùng thuật toán lấy mẫu ngẫu nhiên đơn giản (trong `python` là hàm `train_test_split` thuộc thư viện `scikit-learn`). Lý do của vấn đề này là việc lấy mẫu ngẫu nhiên trên có thể gây ra vấn đề quá khớp nghiêm trọng, do trong tập `train` có thể chứa dữ liệu tương lai, trong khi tập `test` lại chứa dữ liệu quá khứ (tức là, tương lai đã được biết trước). Vì vậy, tác giả sử dụng ý tưởng sử dụng điểm cắt thời gian (time-cutoff) để giải quyết vấn đề này.

```
def split_train_test(data, cutoff, cols_to_lag, lag=1):
    """
```

```
    Split data into training and testing sets.

    Parameters:
    data (DataFrame): Input DataFrame to be split.
    cutoff (int): Year to split the data.
    cols_to_lag (list): List of column names to create lagged features for.
    lags (int): Number of lagged periods to create.

    Returns:
    DataFrame, DataFrame: Two DataFrames representing the training and
testing sets.
    """
    # Create lagged features
    all_lags = [data[['Z']]]

    # Calculate lags using loop
    for i in range(1, lag+1):
        data_lag = data.groupby(level='Firm')[cols_to_lag].shift(i)
        data_lag = data_lag.add_suffix(f'_L{i}')
        all_lags.append(data_lag)

    # Concatenate all lagged features
    data = pd.concat(all_lags, axis=1)
    data.dropna(inplace=True)

    # Split the data
    train = data[data.index.get_level_values('Year') <= cutoff].copy()
    test = data[data.index.get_level_values('Year') > cutoff].copy()
    return train, test
```

Tiếp theo, tác giả thực hiện thiết lập mô hình huấn luyện và kiểm thử. Bước đầu tiên của quá trình này là khai báo các hàm liên quan, bao gồm hàm hồi quy logistic, hàm chuẩn hóa dữ liệu và các hàm đánh giá hiệu quả dự báo.

```
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix,
roc_auc_score, accuracy_score
```

Bước tiếp theo, tác giả định nghĩa hàm `train_and_test_model()` với 4 tham số đầu vào để thực hiện huấn luyện và kiểm thử dựa trên hồi quy logistic. Các tham số đầu vào của hàm bao gồm: (1) `data` (`DataFrame`): dữ liệu đầu vào, bao gồm biến mục tiêu và các biến dự báo, (2) `target` (`str`): biến mục tiêu (mặc định là `Z`), (3) `cutoff` (`int`): điểm cắt thời gian (mặc định là 2023), và (4) `lag` (`int`): số độ trễ được sử dụng (mặc định là 1). Hàm này in ra kết quả huấn luyện mô hình bao gồm độ chính xác tổng thể, độ chính xác trên tập huấn luyện, ROC-AUC Score và ma trận nhầm lẫn (confusion matrix).

```
def train_and_test_model(data, target = 'Z', cutoff=2023, lag=1):
    """
    Train and test a logistic regression model.

    Parameters:
    data (DataFrame): Input DataFrame containing features and target
variable.
    target (str): Name of the target variable column.
    cutoff (int): Year to split the data into training and testing sets.
    lag (int): Number of lagged periods to create.

    Returns:
    None
    """
    # Split data into training and testing sets
    col_names = data.columns.tolist()
    vars = [col for col in col_names if col != target and col !=
'Exchange']
    train, test = split_train_test(data, cutoff=cutoff, cols_to_lag=vars,
lag=lag)

    # Set up features variables
    features = [col for col in train.columns if col != target]

    # Set up X and y for training and testing
    X_train = train[features]
    y_train = train[target]
    X_test = test[features]
    y_test = test[target]

    # Standardize features
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)

    # Initialize and train the model
    model = LogisticRegression(
        class_weight='balanced',
        max_iter=1000,
        random_state=42
    )

    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    y_proba = model.predict_proba(X_test)[:, 1]

    # Evaluate the model
    print('\n')
    print('============== KẾT QUẢ HUẤN LUYỆN MÔ HÌNH ==============')
    print(f'Số lượng quan sát trong tập huấn luyện: {X_train.shape[0]}')
```

```
    print(f'Số lượng quan sát trong tập kiểm tra: {X_test.shape[0]}')
    print(f'Điểm cắt năm: {cutoff}')
    print(f'Số độ trễ được sử dụng: {lag}')
    print(f'Số lượng biến giải thích trong mô hình: {X_train.shape[1]}')

    print('\n==================== ĐÁNH GIÁ MÔ HÌNH ====================')
    print(f'Độ chính xác tổng thể (Accuracy): {accuracy_score(y_test,
y_pred):.3f}')
    print(f'Độ chính xác trên tập huấn luyện (Train Accuracy):
{model.score(X_train, y_train):.3f}')
    print(f'ROC-AUC Score: {roc_auc_score(y_test, y_proba):.3f}')

    print('\n======= BÁO CÁO PHÂN LOẠI (Classification Report) =======')
    print(classification_report(y_test, y_pred))

    print('========== MA TRẬN NHẦM LẪN (Confusion Matrix) ==========')
    conf_matrix = pd.DataFrame(
        confusion_matrix(y_test, y_pred),
        index = ['Thực tế = 0', 'Thực tế = 1'],
        columns = ['Dự báo = 0', 'Dự báo = 1']
    )
    print(conf_matrix)
    print('\n')

    return None
```

2.5 Kết quả huấn luyện

Tại đây, tác giả thực hiện 12 lần huấn luyện và kiểm thử, là kết hợp của các tổ hợp điểm cắt năm (2020, 2021, 2022 và 2023) và số độ trễ được sử dụng (1, 2 và 3). Kết quả chỉ rằng độ chính xác tổng thể dự báo trong cả 12 trường hợp gần tương đương nhau và dao động quanh mức 76-78%, đồng thời độ chính xác trên tập huấn luyện gần tương đồng với độ chính xác tổng thể cũng chỉ ra rằng mô hình trên không gặp vấn đề quá khớp (overfitting). Bên cạnh đó, chỉ số ROC-AUC Score nhận giá trị trên 0.85 trong mọi trường hợp chỉ ra rằng đây là một mô hình tốt và phân loại tương đối chính xác. Ngoài ra, quan sát ma trận nhầm lẫn, tác giả nhận thấy rằng số trong những dự báo sai, số lượng sai lầm loại II diễn ra luôn dưới 30% trong mọi trường hợp, chỉ ra rằng đây là một mô hình dự báo rất tốt.

Quan sát kỹ hơn kết quả, tác giả nhận thấy rằng việc sử dụng dữ liệu lớn hơn để kiểm tra, cũng như đào sâu hơn các độ trễ có thể khiến mô hình trở nên chính xác hơn. Song sự cải thiện ấy trong trường hợp này là không nhiều.

Chi tiết hơn về kết quả được trình bày trong phần dưới

```
cutoff = [2020, 2021, 2022, 2023]
lags = [1, 2, 3]

for year in cutoff:
```

```
    for lag in lags:
        train_and_test_model(data, target='Z', cutoff=year, lag=lag)
```

```
============== KẾT QUẢ HUẤN LUYỆN MÔ HÌNH ================
Số lượng quan sát trong tập huấn luyện: 8995
Số lượng quan sát trong tập kiểm tra: 5389
Điểm cắt năm: 2020
Số độ trễ được sử dụng: 1
Số lượng biến giải thích trong mô hình: 8

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể (Accuracy): 0.771
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.792
ROC-AUC Score: 0.859

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support

         0.0       0.87      0.74      0.80      3389
         1.0       0.65      0.82      0.73      2000

    accuracy                           0.77      5389
   macro avg       0.76      0.78      0.76      5389
weighted avg       0.79      0.77      0.77      5389

========== MA TRẬN NHẦM LẪN (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tế = 0        2522         867
Thực tế = 1         366        1634
```

```
============== KẾT QUẢ HUẤN LUYỆN MÔ HÌNH ================
Số lượng quan sát trong tập huấn luyện: 7635
Số lượng quan sát trong tập kiểm tra: 5283
Điểm cắt năm: 2020
Số độ trễ được sử dụng: 2
Số lượng biến giải thích trong mô hình: 16

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể (Accuracy): 0.769
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.805
ROC-AUC Score: 0.860

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support
```

```
         0.0       0.87      0.75      0.80      3311
         1.0       0.66      0.80      0.72      1972

    accuracy                           0.77      5283
   macro avg       0.76      0.78      0.76      5283
weighted avg       0.79      0.77      0.77      5283

========== MA TRẬN NHẦM LẪN (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tế = 0        2479         832
Thực tế = 1         386        1586




============== KẾT QUẢ HUẤN LUYỆN MÔ HÌNH ================
Số lượng quan sát trong tập huấn luyện: 6315
Số lượng quan sát trong tập kiểm tra: 5165
Điểm cắt năm: 2020
Số độ trễ được sử dụng: 3
Số lượng biến giải thích trong mô hình: 24

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể (Accuracy): 0.771
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.803
ROC-AUC Score: 0.861

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support

         0.0       0.86      0.75      0.80      3232
         1.0       0.66      0.80      0.72      1933

    accuracy                           0.77      5165
   macro avg       0.76      0.78      0.76      5165
weighted avg       0.79      0.77      0.77      5165

========== MA TRẬN NHẦM LẪN (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tế = 0        2432         800
Thực tế = 1         385        1548




============== KẾT QUẢ HUẤN LUYỆN MÔ HÌNH ================
Số lượng quan sát trong tập huấn luyện: 10332
Số lượng quan sát trong tập kiểm tra: 4052
Điểm cắt năm: 2021
Số độ trễ được sử dụng: 1
```

```
Sô´lượng biến giải thích trong mô hình: 8

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể̉(Accuracy): 0.786
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.788
ROC-AUC Score: 0.865

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support

         0.0       0.87      0.77      0.82      2515
         1.0       0.68      0.81      0.74      1537

    accuracy                           0.79      4052
   macro avg       0.78      0.79      0.78      4052
weighted avg       0.80      0.79      0.79      4052

========== MA TRẬN NHÂM LÂN̂ (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tê´= 0        1942         573
Thực tê´= 1         295        1242




============== KÊT QUẢ HUÂN̂ LUYỆN MÔ HÌNH ===============
Sô´lượng quan sát trong tập huấn luyện: 8920
Sô´lượng quan sát trong tập kiểm tra: 3998
Điểm căt năm: 2021
Sô´độ trễđược sửdụng: 2
Sô´lượng biến giải thích trong mô hình: 16

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể̉(Accuracy): 0.778
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.801
ROC-AUC Score: 0.866

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support

         0.0       0.87      0.76      0.81      2478
         1.0       0.67      0.81      0.74      1520

    accuracy                           0.78      3998
   macro avg       0.77      0.78      0.77      3998
weighted avg       0.79      0.78      0.78      3998

========== MA TRẬN NHÂM LÂN̂ (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tê´= 0        1883         595
```

```
Thực tế́= 1         291        1229



============== KẾT QUẢ HUẤN LUYỆN MÔ HÌNH ================
Số́lượng quan sát trong tập huấn luyện: 7567
Số́lượng quan sát trong tập kiểm tra: 3913
Điểm cắt năm: 2021
Số́độ trễ̃được sử̉dụng: 3
Số́lượng biến giải thích trong mô hình: 24

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể̉(Accuracy): 0.780
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.797
ROC-AUC Score: 0.867

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support

         0.0       0.86      0.76      0.81      2421
         1.0       0.68      0.81      0.74      1492

    accuracy                           0.78      3913
   macro avg       0.77      0.79      0.77      3913
weighted avg       0.79      0.78      0.78      3913

========== MA TRẬN NHẦM LẪN (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tế́= 0        1849         572
Thực tế́= 1         289        1203




============== KẾT QUẢ HUẤN LUYỆN MÔ HÌNH ================
Số́lượng quan sát trong tập huấn luyện: 11682
Số́lượng quan sát trong tập kiểm tra: 2702
Điểm cắt năm: 2022
Số́độ trễ̃được sử̉dụng: 1
Số́lượng biến giải thích trong mô hình: 8

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể̉(Accuracy): 0.788
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.786
ROC-AUC Score: 0.869

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support
```

```
         0.0       0.87      0.77      0.82      1683
         1.0       0.68      0.82      0.74      1019

    accuracy                           0.79      2702
   macro avg       0.78      0.79      0.78      2702
weighted avg       0.80      0.79      0.79      2702

========== MA TRẬN NHẦM LẪN (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tê´= 0        1296         387
Thực tê´= 1         187         832




============== KẾT QUẢ HUẤN LUYỆN MÔ HÌNH ================
Sô´lượng quan sát trong tập huấn luyện: 10247
Sô´lượng quan sát trong tập kiểm tra: 2671
Điểm cắt năm: 2022
Sô´độ trễ được sử dụng: 2
Sô´lượng biến giải thích trong mô hình: 16

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể (Accuracy): 0.783
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.797
ROC-AUC Score: 0.872

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support

         0.0       0.88      0.75      0.81      1662
         1.0       0.67      0.83      0.74      1009

    accuracy                           0.78      2671
   macro avg       0.78      0.79      0.78      2671
weighted avg       0.80      0.78      0.79      2671

========== MA TRẬN NHẦM LẪN (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tê´= 0        1254         408
Thực tê´= 1         172         837




============== KẾT QUẢ HUẤN LUYỆN MÔ HÌNH ================
Sô´lượng quan sát trong tập huấn luyện: 8844
Sô´lượng quan sát trong tập kiểm tra: 2636
Điểm cắt năm: 2022
Sô´độ trễ được sử dụng: 3
```

```
Sô´lượng biến giải thích trong mô hình: 24

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể̉(Accuracy): 0.781
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.795
ROC-AUC Score: 0.873

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support

         0.0       0.88      0.75      0.81      1639
         1.0       0.67      0.83      0.74       997

    accuracy                           0.78      2636
   macro avg       0.77      0.79      0.78      2636
weighted avg       0.80      0.78      0.78      2636

========== MA TRẬN NHÂM LÂN̂ (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tê´= 0        1234         405
Thực tê´= 1         171         826




============== KÊT QUẢ HUÂN̂ LUYỆN MÔ HÌNH ===============
Sô´lượng quan sát trong tập huấn luyện: 13033
Sô´lượng quan sát trong tập kiểm tra: 1351
Điểm căt năm: 2023
Sô´độ trễđược sửdụng: 1
Sô´lượng biến giải thích trong mô hình: 8

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể̉(Accuracy): 0.779
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.782
ROC-AUC Score: 0.879

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support

         0.0       0.90      0.73      0.81       866
         1.0       0.64      0.86      0.74       485

    accuracy                           0.78      1351
   macro avg       0.77      0.80      0.77      1351
weighted avg       0.81      0.78      0.78      1351

========== MA TRẬN NHÂM LÂN̂ (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tê´= 0         636         230
```

```
Thực tê´= 1          69         416



============== KÊT QUẢ HUẤN LUYỆN MÔ HÌNH ================
Sô´lượng quan sát trong tập huấn luyện: 11586
Sô´lượng quan sát trong tập kiểm tra: 1332
Điểm cắt năm: 2023
Sô´độ trễ được sử dụng: 2
Sô´lượng biến giải thích trong mô hình: 16

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể(Accuracy): 0.784
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.791
ROC-AUC Score: 0.881

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support

         0.0       0.90      0.75      0.82       851
         1.0       0.65      0.85      0.74       481

    accuracy                           0.78      1332
   macro avg       0.78      0.80      0.78      1332
weighted avg       0.81      0.78      0.79      1332

========== MA TRẬN NHẦM LẪN (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tê´= 0         636         215
Thực tê´= 1          73         408



============== KÊT QUẢ HUẤN LUYỆN MÔ HÌNH ================
Sô´lượng quan sát trong tập huấn luyện: 10159
Sô´lượng quan sát trong tập kiểm tra: 1321
Điểm cắt năm: 2023
Sô´độ trễ được sử dụng: 3
Sô´lượng biến giải thích trong mô hình: 24

==================== ĐÁNH GIÁ MÔ HÌNH ====================
Độ chính xác tổng thể(Accuracy): 0.781
Độ chính xác trên tập huấn luyện (Train Accuracy): 0.789
ROC-AUC Score: 0.884

======= BÁO CÁO PHÂN LOẠI (Classification Report) =======
              precision    recall  f1-score   support
```

```
         0.0       0.90      0.74      0.81       844
         1.0       0.65      0.85      0.74       477

    accuracy                           0.78      1321
   macro avg       0.77      0.80      0.77      1321
weighted avg       0.81      0.78      0.79      1321

========== MA TRẬN NHẦM LẪN (Confusion Matrix) ==========
             Dự báo = 0  Dự báo = 1
Thực tê´= 0         626         218
Thực tê´= 1          71         406
```

2.6 Thảo luận và đề xuất

Kết quả huấn luyện trên chỉ ra rằng hồi quy logistic là một phương pháp khả thi để dự báo nguy cơ vỡ nợ (với đại diện là chỉ số Z-Score của Altman). Đồng thời, việc mở rộng thêm quy mô nghiên cứu, ví dụ như sử dụng dữ liệu huấn luyện lớn hơn, thêm biến dự báo, hoặc đào sâu hơn vào các độ trễ có thể giúp kết quả dự báo trở nên tốt hơn. Tuy nhiên, nhà phân tích cần thận trọng khi thực hiện các mở rộng này, bởi chúng ta phải đánh đổi phần hiệu quả tăng thêm này với cái giá về hiệu suất hoạt động của mô hình, chi phí thu thập thông tin... và những đánh đổi này có thể không mang lại hiệu quả về kinh tế.

3 Thông tin bổ sung

3.1 Kết quả kiểm tra đạo văn

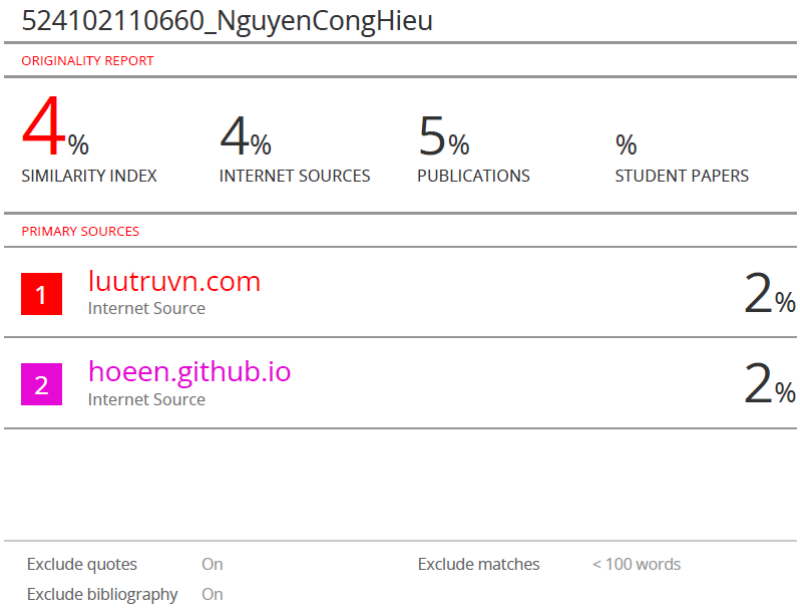


Figure 1: Kết quả kiểm tra đạo văn

3.2 Liên kết ngoài

Repository lưu trữ dự án được truy cập qua liên kết này

Kết quả kiểm tra đạo văn đầy đủ được truy cập qua liên kết này.

Tài liệu tham khảo

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *The Journal of Finance*, *23*(4), 589–609. http://www.jstor.org/stable/2978933

Bodie, Z., Kane, A., & Marcus, A. (2018). *Investments*. McGraw-Hill Education. https://books.google.com.vn/books?id=3QlmvgAACAAJ

Brealey, R., Myers, S., & Allen, F. (2019). *Principles of Corporate Finance*. McGraw-Hill Education. https://books.google.com.vn/books?id=0280wAEACAAJ

Hull, J. (2023). *Risk Management and Financial Institutions*. Wiley. https://books.google.com.vn/books?id=WO6hEAAAQBAJ

Lupu, D., & Onofrei, M. (2014). The Modeling of Forecasting the Bankruptcy Risk in Romania. *Onofrei, M., & Lupu, D.(2014). THE MODELING OF FORECASTING THE BANKRUPTCY RISK IN ROMANIA. Economic Computation & Economic Cybernetics Studies & Research*, *48*(3).

Merton, R. C. (1974). On the Pricing of Corporate Debt: The Risk Structure of Interest Rates. *The Journal of Finance*, *29*(2), 449–470. http://www.jstor.org/stable/2978814

Ohlson, J. A. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research*, *18*(1), 109–131. http://www.jstor.org/stable/2490395

Reilly, F., & Brown, K. (2011). *Investment Analysis and Portfolio Management*. Cengage Learning. https://books.google.com.vn/books?id=ze0JAAAAQBAJ

Ross, S., Westerfield, R., Jaffe, J., & Jordan, B. (2019). *Corporate Finance*. McGraw-Hill Education. https://books.google.com.vn/books?id=LAi_uAEACAAJ

Xu, M., & Zhang, C. (2009). Bankruptcy prediction: the case of Japanese listed companies. *Review of accounting studies*, *14*(4), 534–558.