ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN

o0o

BÁO CÁO
HỆ HỖ TRỢ QUYẾT ĐỊNH

CHỦ ĐỀ 14:
Stock axis bank

Giảng viên hướng dẫn : TS.Trần Ngọc Thăng
Sinh viên thực hiện : Nguyễn Văn Dương
Mã số sinh viên : 20227103
Lớp sinh viên : Toán Tin 03 - K67
Mã lớp học : 158242

Hà Nội, ngày 26 tháng 03 năm 2025

Mục lục

1 Phát biểu bài toán

Mô tả bài toán: Bài toán được đặt ra là sử dụng mô hình hồi quy để dự đoán giá đóng cửa (Close) của cổ phiếu Axis Bank dựa trên dữ liệu lịch sử.
Đầu vào: Dữ liệu lịch sử giá cổ phiếu từ bộ dữ liệu AXISBANK.csv.
Đầu ra: Giá đóng cửa (Close) của cổ phiếu Axis Bank.
Yêu cầu xử lý: Sử dụng mô hình hồi quy tuyến tính (Linear Regression) để dự đoán giá đóng cửa dựa trên các đặc trưng đầu vào.

2 Tiền xử lý dữ liệu

Thu thập dũ liệu: Dữ liệu được thu thập từ Kaggle, một nền tảng cung cấp bộ dữ liệu mở, tại địa chỉ: https://www.kaggle.com/datasets/pritsheta/axis-bank/data. Bộ dữ liệu có tên AXISBANK.csv, chứa thông tin giá cổ phiếu của Axis Bank.
Cách thu thập: File AXISBANK.csv được tải về máy tính từ Kaggle, sau đó được tải lên Google Colab bằng lệnh files.upload() để sử dụng trong quá trình phân tích.
Thông tin cơ bản: Bộ dữ liệu gồm 5300 dòng và 6 cột, tương ứng với 5300 ngày giao dịch, từ ngày 03/01/2000 đến ngày 18/08/2021. Các cột bao gồm: Date (ngày giao dịch), Open (giá mở cửa), High (giá cao nhất), Low (giá thấp nhất), Close (giá đóng cửa), và Volume (khối lượng giao dịch).

```
#Tải file dữ liệu
from google.colab import files
uploaded = files.upload()

# Import các thư viện
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt


# Đọc dữ liệu từ file AXISBANK.csv
df = pd.read_csv('AXISBANK.csv')

# In thông tin cơ bản về dữ liệu
print("Số dòng và cột:", df.shape)
print("Tên các cột:", df.columns)
print("5 dòng đầu tiên:\n", df.head())
print("Khoảng thời gian dữ liệu:")
print("Ngày đầu tiên:", df['Date'].min())
print("Ngày cuối cùng:", df['Date'].max())
```

```
#Output
Số dòng và cột: (5306, 15)
Tên các cột: Index(['Date', 'Symbol', 'Series', 'Prev Close', 'Open',
'High', 'Low', 'Last', 'Close', 'VWAP', 'Volume', 'Turnover', 'Trades',
'Deliverable Volume', '%Deliverble'], dtype='object')
5 dòng đầu tiên:
         Date   Symbol Series  Prev Close  Open   High    Low  Last  Close
0  2000-01-03  UTIBANK     EQ       24.70  26.7  26.70  26.70  26.7  26.70
1  2000-01-04  UTIBANK     EQ       26.70  27.0  28.70  26.50  27.0  26.85
2  2000-01-05  UTIBANK     EQ       26.85  26.0  27.75  25.50  26.4  26.30
3  2000-01-06  UTIBANK     EQ       26.30  25.8  27.00  25.80  25.9  25.95
4  2000-01-07  UTIBANK     EQ       25.95  25.0  26.00  24.25  25.0  24.80

    VWAP  Volume      Turnover  Trades  Deliverable Volume  %Deliverble
0  26.70  112100  2.993070e+11     NaN                 NaN          NaN
1  27.24  234500  6.387275e+11     NaN                 NaN          NaN
2  26.24  170100  4.462980e+11     NaN                 NaN          NaN
3  26.27  102100  2.681730e+11     NaN                 NaN          NaN
4  25.04   62600  1.567220e+11     NaN                 NaN          NaN
Khoảng thời gian dữ liệu:
Ngày đầu tiên: 2000-01-03
Ngày cuối cùng: 2021-04-30
```

Đánh nhãn và thống kê dữ liệu:

```
# Xác định nhãn (label) và đặc trưng (features)
X = df[['Open', 'High', 'Low', 'Volume']]  # Đặc trưng
y = df['Close']  # Nhãn (biến mục tiêu)
print("\nĐặc trưng (X):", X.columns.tolist())
print("Nhãn (y): 'Close'")
# Thống kê dữ liệu
print("\nThống kê dữ liệu:")
print(df.describe())
```

Tiền xử lý dữ liệu: Sử dụng kết quả từ df.isnull().sum() (trước và sau khi điền giá trị thiếu), dữ liệu sau khi chuẩn hóa, và đặc trưng mới (MA5).

```
# Kiểm tra và xử lý giá trị thiếu
print("\nGiá trị thiếu:\n", df.isnull().sum())
df = df.interpolate()  # Điền giá trị thiếu bằng nội suy
print("\nSau khi điền giá trị thiếu:\n", df.isnull().sum())

# Chuẩn hóa dữ liệu
scaler = MinMaxScaler()
df[['Open', 'High', 'Low', 'Volume']] = scaler.fit_transform(df[['Open', 'High
print("\nDữ liệu sau khi chuẩn hóa (5 dòng đầu tiên):\n", df[['Open', 'High',
```

```
# Tạo đặc trưng mới (ví dụ: Moving Average 5 ngày)
# Tính Moving Average 5 ngày
df['MA5'] = df['Close'].rolling(window=5).mean()
# Điền giá trị thiếu của MA5 bằng trung bình
df['MA5'] = df['MA5'].fillna(df['MA5'].mean())
# Thêm đặc trưng MA5 vào X
X = df[['Open', 'High', 'Low', 'Volume', 'MA5']]
print("\nĐặc trưng sau khi thêm MA5:\n", X.head())

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
shuffle=False)
print("\nSố dòng tập huấn luyện:", len(X_train))
print("Số dòng tập kiểm tra:", len(X_test))
print("Ngày bắt đầu tập huấn luyện:", df['Date'].iloc[0])
print("Ngày kết thúc tập huấn luyện:", df['Date'].iloc[len(X_train)-1])
print("Ngày bắt đầu tập kiểm tra:", df['Date'].iloc[len(X_train)])
print("Ngày kết thúc tập kiểm tra:", df['Date'].iloc[-1])

#Output
Đặc trưng (X): ['Open', 'High', 'Low', 'Volume']
Nhãn (y): 'Close'

Thống kê dữ liệu:
        Prev Close         Open         High          Low         Last
count  5306.000000  5306.000000  5306.000000  5306.000000  5306.000000
mean    585.763852   586.507388   596.476187   575.571598   585.897399
std     436.714128   436.602194   443.044833   430.108921   436.609147
min      22.150000    21.000000    23.700000    21.000000    22.150000
25%     230.950000   232.000000   235.125000   227.075000   230.550000
50%     519.450000   520.100000   528.400000   512.025000   519.425000
75%     877.312500   880.075000   897.987500   852.762500   877.275000
max    2023.350000  2034.400000  2043.050000  2002.600000  2022.550000

              Close         VWAP        Volume      Turnover         Trades
count  5306.000000  5306.000000  5.306000e+03  5.306000e+03    2456.000000
mean    585.893931   586.077778  4.527938e+06  2.739871e+14  120602.231678
std     436.649765   436.611987  8.101940e+06  4.122431e+14   96106.654046
min      22.150000    22.170000  2.850000e+03  8.275250e+09    2698.000000
25%     230.975000   231.115000  2.842172e+05  5.868745e+12   62228.250000
50%     519.500000   519.505000  1.656966e+06  1.653257e+14   93186.500000
75%     877.312500   875.807500  5.515245e+06  3.456528e+14  144973.250000
max    2023.350000  2020.310000  1.205419e+08  7.179550e+15  990737.000000

       Deliverable Volume  %Deliverble
count        4.797000e+03  4797.000000
mean         1.990907e+06     0.466962
```

```
std           3.264587e+06     0.161808
min           5.809000e+03     0.075000
25%           2.573130e+05     0.347500
50%           7.687680e+05     0.459800
75%           2.652520e+06     0.573900
max           9.490116e+07     0.983000

Giá trị thiếu:
 Date                    0
Symbol                   0
Series                   0
Prev Close               0
Open                     0
High                     0
Low                      0
Last                     0
Close                    0
VWAP                     0
Volume                   0
Turnover                 0
Trades                2850
Deliverable Volume     509
%Deliverble            509
dtype: int64

Sau khi điền giá trị thiếu:
 Date                    0
Symbol                   0
Series                   0
Prev Close               0
Open                     0
High                     0
Low                      0
Last                     0
Close                    0
VWAP                     0
Volume                   0
Turnover                 0
Trades                2850
Deliverable Volume     498
%Deliverble            498
dtype: int64

Dữ liệu sau khi chuẩn hóa (5 dòng đầu tiên):
        Open      High       Low    Volume
0  0.002831  0.001486  0.002876  0.000906
1  0.002980  0.002476  0.002776  0.001922
```

```
2  0.002483  0.002006  0.002271  0.001388
3  0.002384  0.001634  0.002422  0.000823
4  0.001987  0.001139  0.001640  0.000496

Đặc trưng sau khi thêm MA5:
       Open      High       Low    Volume         MA5
0  0.002831  0.001486  0.002876  0.000906  586.056826
1  0.002980  0.002476  0.002776  0.001922  586.056826
2  0.002483  0.002006  0.002271  0.001388  586.056826
3  0.002384  0.001634  0.002422  0.000823  586.056826
4  0.001987  0.001139  0.001640  0.000496   26.120000
```

3 Tạo luyện đánh giá mô hình

: Sử dụng mô hình hồi quy tuyến tính (LinearRegression) để huấn luyện trên tập huấn luyện. Tính các chỉ số MSE, RMSE, R^2 để đánh giá hiệu suất mô hình trên tập kiểm tra.

```python
# Huấn luyện mô hình
model = LinearRegression()
model.fit(X_train, y_train)

# Đánh giá mô hình
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print(f"\nMSE: {mse:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"R²: {r2:.2f}")

#Output
Số dòng tập huấn luyện: 4244
Số dòng tập kiểm tra: 1062
Ngày bắt đầu tập huấn luyện: 2000-01-03
Ngày kết thúc tập huấn luyện: 2017-01-12
Ngày bắt đầu tập kiểm tra: 2017-01-13
Ngày kết thúc tập kiểm tra: 2021-04-30

MSE: 24.40
RMSE: 4.94
RR: 1.00
```

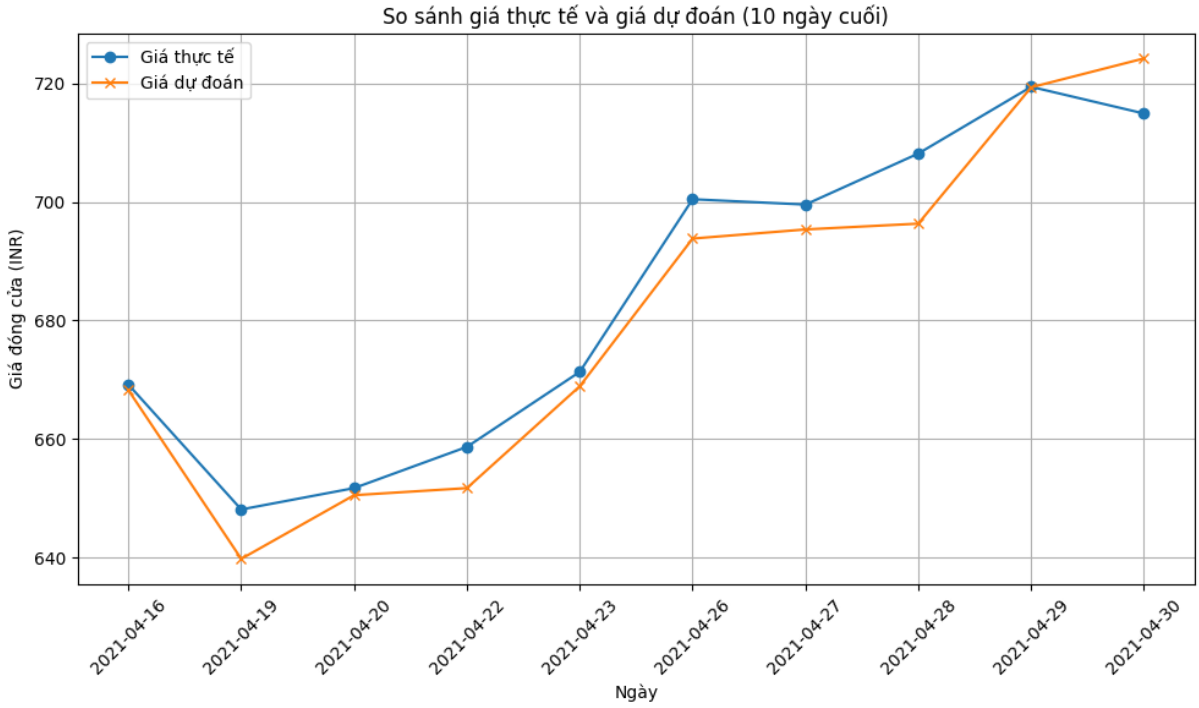
4 Ứng dụng mô hình

: Dự đoán giá cho 10 ngày cuối, tạo bảng so sánh, và vẽ biểu đồ.

```
# Ứng dụng mô hình (dự đoán 10 ngày cuối)
last_10_days = X_test.tail(10)
last_10_actual = y_test.tail(10)
last_10_pred = model.predict(last_10_days)

# Tạo bảng so sánh
results = pd.DataFrame({
    'Ngày': df['Date'].tail(10),
    'Giá thực tế': last_10_actual,
    'Giá dự đoán': last_10_pred,
    'Sai số': np.abs(last_10_actual - last_10_pred)
})
print("\nKết quả dự đoán 10 ngày cuối:\n", results)

# Vẽ biểu đồ
plt.figure(figsize=(10, 6))
plt.plot(results['Ngày'], results['Giá thực tế'], label='Giá thực tế', marker
plt.plot(results['Ngày'], results['Giá dự đoán'], label='Giá dự đoán', marker
plt.xlabel('Ngày')
plt.ylabel('Giá đóng cửa (INR)')
plt.title('So sánh giá thực tế và giá dự đoán (10 ngày cuối)')
plt.legend()
plt.xticks(rotation=45)
plt.grid()
plt.tight_layout()
plt.savefig('price_comparison.png')  # Lưu biểu đồ để chèn vào báo cáo
plt.show()
#Output
Kết quả dự đoán 10 ngày cuối:
            Ngày  Giá thực tế  Giá dự đoán     Sai số
5296  2021-04-16       669.20   668.264058   0.935942
5297  2021-04-19       648.15   639.823440   8.326560
5298  2021-04-20       651.75   650.554805   1.195195
5299  2021-04-22       658.70   651.751802   6.948198
5300  2021-04-23       671.35   668.922369   2.427631
5301  2021-04-26       700.45   693.817265   6.632735
5302  2021-04-27       699.55   695.356392   4.193608
5303  2021-04-28       708.15   696.333940  11.816060
5304  2021-04-29       719.40   719.313362   0.086638
5305  2021-04-30       714.90   724.188224   9.288224
```



5 Kết luận

:

Ưu điểm:

Sử dụng Ridge Regression giúp kiểm soát overfitting, đặc biệt khi các đặc trưng như Open, High, Low có tương quan cao.

Thêm đặc trưng MA5 (Moving Average 5 ngày) giúp mô hình học được xu hướng giá cổ phiếu, cải thiện hiệu suất.

Hiệu chỉnh siêu tham số alpha bằng GridSearchCV đảm bảo mô hình đạt hiệu suất tối ưu.

Nhược điểm:

Mô hình Ridge Regression là một mô hình tuyến tính, không thể học được các mối quan hệ phi tuyến phức tạp trong dữ liệu giá cổ phiếu.

Dữ liệu chỉ bao gồm các đặc trưng cơ bản (Open, High, Low, Volume, MA5), chưa tận dụng các yếu tố bên ngoài như tin tức tài chính hoặc chỉ số thị trường (NIFTY50).

Sai số vẫn lớn ở các ngày có biến động mạnh, do mô hình chưa đủ khả năng dự đoán các biến động bất thường.

Khả năng cải tiến trong tương lai:

Sử dụng các mô hình phi tuyến phức tạp hơn như Random Forest, Gradient Boosting, hoặc LSTM để học các mối quan hệ phi tuyến trong dữ liệu.

Thêm dữ liệu đầu vào, ví dụ: kết hợp dữ liệu chỉ số thị trường (NIFTY50), tin tức tài chính, hoặc các chỉ số kinh tế vĩ mô.

Tăng cường tiền xử lý dữ liệu bằng cách tạo thêm các đặc trưng kỹ thuật như Relative Strength Index (RSI), Bollinger Bands, hoặc sử dụng các khoảng Moving Average dài hơn (MA10, MA20).

CHECKLIST

STT	Loại yêu cầu(*)	Yêu cầu	Điểm chữ	Điểm số	Minh chứng
1	Phát biểu bài toán (1 điểm)	Mô tả bài toán, đầu vào, đầu ra, yêu cầu xử lý.	A	1	Trang 3 - 3
2	Tiền xử lý dữ liệu (2 điểm)	Thu thập dữ liệu	A	1	Trang 3 - 7
		Đánh nhãn dữ liệu	F	0	
		Thống kê dữ liệu mẫu	A	1	Trang 4 - 5
		Tiền xử lý dữ liệu	B	0.75	Trang 4 - 7
3	Tạo và luyện mô hình (5 điểm)	Tạo mô hình (mô tả các quyết định)	A	1	Trang 7 - 7
		Mô tả quá trình luyện mô hình	A	1	Trang 7 - 7
		Mô tả điều kiện dừng	C	0.5	Trang 7 - 7
		Hiệu chỉnh siêu tham số	C	0.5	Trang 7 - 7
		Đánh giá mô hình với dữ liệu test	A	1	Trang 7 - 7
4	Ứng dụng mô hình (1 điểm)	Mô tả ứng dụng và kết quả thử nghiệm	C	0.5	Trang 7 - 9
		Diễn giải kết quả	C	0.5	Trang 7 - 9
5	Kết luận (1 điểm)	Ưu nhược điểm cách tiếp cận	F	0	
		Khả năng cải tiến	B	0.75	Trang 9 - 9
				Tổng điểm	**9.75**