**BÁO CÁO BÀI TẬP LỚN MÔN HỌC**

**Học phần: Nhập môn Học máy và Khai phá dữ liệu**

**ĐỀ TÀI: DỰ ĐOÁN GIÁ BITCOIN THEO GIỜ**

**I. TỔNG QUAN ĐỀ TÀI**

Trong bối cảnh thị trường tiền điện tử phát triển mạnh mẽ với sự biến động phức tạp của giá Bitcoin, việc dự đoán giá trở nên cần thiết hơn bao giờ hết cho các nhà đầu tư và nhà phân tích thị trường. Dự án này nhằm xây dựng một hệ thống dự đoán giá Bitcoin theo giờ, giúp nắm bắt được biến động ngắn hạn của thị trường.

Làm như nào để thực hiện mong muốn trên?

* Đề xuất phương pháp sử dụng các thuật toán học máy từ bài giảng môn học
* Thử nghiệm nhiều mô hình khác nhau và so sánh hiệu suất
* Tiến hành dự đoán giá Bitcoin cho 12 giờ tiếp theo

Lợi ích:

1. Tự động hóa: hệ thống sẽ tự động dự đoán giá Bitcoin, tiết kiệm thời gian và công sức so với phân tích thủ công
2. Độ chính xác và hiệu quả: sử dụng các mô hình học máy và kỹ thuật tối ưu giúp gia tăng độ chính xác trong dự đoán
3. Nâng cao quyết định đầu tư: nhà đầu tư có thể tham khảo dự đoán để đưa ra quyết định thông minh hơn
4. Sự cập nhật về dữ liệu: hệ thống có thể liên tục học và cập nhật từ dữ liệu mới

**II. PHÂN TÍCH ĐỀ TÀI**

**1. THU THẬP DỮ LIỆU**

Thu thập dữ liệu lớn chứa thông tin giá Bitcoin theo từng phút từ file btcusd\_1-min\_data.csv với các thông tin:

* Timestamp: Mốc thời gian
* Open: Giá mở cửa
* High: Giá cao nhất
* Low: Giá thấp nhất
* Close: Giá đóng cửa
* Volume: Khối lượng giao dịch

**2. TIỀN XỬ LÝ DỮ LIỆU**

Bao gồm:

* Chuyển đổi dữ liệu từng phút sang dữ liệu theo giờ
* Làm sạch và xử lý dữ liệu bị thiếu
* Thêm các chỉ báo kỹ thuật như MA (Moving Average), RSI, MACD...
* Chuyển đổi dữ liệu thành định dạng phù hợp cho các mô hình học máy

**3. XÂY DỰNG MÔ HÌNH HỌC MÁY**

Các mô hình được thử nghiệm từ bài giảng:

* Linear Regression (Hồi quy tuyến tính) - Lecture 3
* Ridge Regression (Hồi quy Ridge) - Lecture 3
* Decision Tree (Cây quyết định) - Lecture 8
* Random Forest (Rừng ngẫu nhiên) - Lecture 8
* KNN (K-Nearest Neighbors) - Lecture 7
* SVM (Support Vector Machine) - Lecture 11

Tối ưu hoá mô hình bằng các kỹ thuật tìm kiếm tham số tối ưu như Grid Search và RandomizedSearchCV. Hiệu suất chương trình được đánh giá bằng RMSE, MAE, R² và MAPE.

**III. CÁC MÔ HÌNH SỬ DỤNG**

**3.1. LINEAR REGRESSION (HỒI QUY TUYẾN TÍNH)**

Mô hình cơ bản nhất, dựa trên việc tìm một hàm tuyến tính tốt nhất để khớp với dữ liệu. Công thức có dạng: y = w₀ + w₁x₁ + w₂x₂ + ... + wₙxₙ Trong đó y là giá Bitcoin cần dự đoán, x₁, x₂, ..., xₙ là các đặc trưng, và w₀, w₁, ..., wₙ là các trọng số cần học.

**3.2. RIDGE REGRESSION (HỒI QUY RIDGE)**

Một biến thể của hồi quy tuyến tính với thêm thành phần điều chuẩn L2 để ngăn overfitting. Có dạng: Cực tiểu hoá RSS(w) + λ∑wᵢ² Trong đó λ là tham số điều chỉnh mức độ phạt các trọng số lớn.

**3.3. DECISION TREE (CÂY QUYẾT ĐỊNH)**

Cây quyết định được xây dựng dựa trên dữ liệu được huấn luyện, bao gồm nút gốc, nút trong (biểu diễn một điều kiện kiểm tra), nút lá; dữ liệu sẽ được phân chia từ nút gốc cho tới khi đạt điều kiện dừng.

**3.4. RANDOM FOREST (RỪNG NGẪU NHIÊN)**

Rừng ngẫu nhiên được tạo ra bằng cách huấn luyện nhiều cây quyết định độc lập trên các tập dữ liệu con được lấy mẫu ngẫu nhiên từ tập dữ liệu huấn luyện, giúp tăng tính đa dạng và tính chia sẻ kiến thức.

**3.5. KNN (K-NEAREST NEIGHBORS)**

KNN là thuật toán dựa trên láng giềng gần nhất, dự đoán giá trị của một điểm dữ liệu mới dựa trên k điểm dữ liệu gần nhất với nó trong không gian đặc trưng.

**3.6. SVM (SUPPORT VECTOR MACHINE)**

SVM là một thuật toán học máy mạnh mẽ được sử dụng cho các bài toán phân loại và hồi quy. Cơ sở lý thuyết của SVM dựa trên việc tìm kiếm một siêu phẳng (hyperplane) tốt nhất để phân chia các dữ liệu trong không gian nhiều chiều.

**IV. CÔNG NGHỆ SỬ DỤNG**

Ngôn ngữ lập trình: Python

Các framework sử dụng:

1. NumPy và pandas được sử dụng để làm việc với dữ liệu ma trận và dataframe
2. Scikit-learn cung cấp các mô hình học máy: LinearRegression, Ridge, DecisionTreeRegressor, RandomForestRegressor, KNeighborsRegressor, SVR
3. Các công cụ đánh giá: mean\_squared\_error, mean\_absolute\_error, r2\_score
4. Công cụ chia dữ liệu: train\_test\_split
5. Công cụ tìm kiếm tham số: GridSearchCV, RandomizedSearchCV
6. Matplotlib để tạo và hiển thị các biểu đồ

**V. XÂY DỰNG CHƯƠNG TRÌNH VÀ CÀI ĐẶT**

**1. XÂY DỰNG DỮ LIỆU VÀ TIỀN XỬ LÝ DỮ LIỆU**

Các bước chính:

1. Đọc dữ liệu từ file btcusd\_1-min\_data.csv và chuyển đổi trường timestamp thành định dạng datetime
2. Kiểm tra và xử lý dữ liệu bị thiếu (isnull().sum())
3. Tổng hợp dữ liệu từ mức phút lên mức giờ sử dụng phương thức resample('H')
4. Thêm các chỉ báo kỹ thuật như:
   * Đường trung bình động (MA6, MA12, MA24)
   * Chỉ số sức mạnh tương đối (RSI)
   * MACD, Bollinger Bands, và các chỉ báo khác
5. Chia dữ liệu thành tập huấn luyện (80%) và kiểm tra (20%)
6. Chuẩn hóa dữ liệu bằng MinMaxScaler

**2. XÂY DỰNG MÔ HÌNH VÀ HUẤN LUYỆN**

Các mô hình được huấn luyện và tối ưu hóa:

1. **Linear Regression:**

lr\_model = LinearRegression()

lr\_results = evaluate\_model(lr\_model, X\_train, X\_test, y\_train, y\_test, scaler\_y)

1. **Ridge Regression:**

ridge\_model = Ridge(alpha=1.0)

ridge\_results = evaluate\_model(ridge\_model, X\_train, X\_test, y\_train, y\_test, scaler\_y)

1. **Decision Tree:**

dt\_model = DecisionTreeRegressor(random\_state=42)

param\_grid = {

'max\_depth': [None, 3, 5, 10],

'min\_samples\_split': np.arange(2, 20, 2),

'min\_samples\_leaf': np.arange(1, 20, 2),

'max\_features': [None, 'sqrt', 'log2']

}

dt\_results = evaluate\_model(dt\_model, X\_train, X\_test, y\_train, y\_test, scaler\_y)

1. **Random Forest:**

rf\_model = RandomForestRegressor(n\_estimators=100, random\_state=42)

param\_grid = {

'n\_estimators': [100, 200],

'max\_depth': [None, 5, 10],

'min\_samples\_split': [2, 5],

'min\_samples\_leaf': [1, 2],

'bootstrap': [True]

}

rf\_results = evaluate\_model(rf\_model, X\_train, X\_test, y\_train, y\_test, scaler\_y)

1. **KNN Regressor:**

knn\_model = KNeighborsRegressor(n\_neighbors=5)

knn\_results = evaluate\_model(knn\_model, X\_train, X\_test, y\_train, y\_test, scaler\_y)

1. **SVM Regressor:**

svr\_model = SVR(kernel='rbf', C=1.0)

svm\_grid = {

'C': [0.1, 1, 10, 100, 1000],

'gamma': [1, 0.1, 0.01, 0.001, 0.0001],

'kernel': ['rbf']

}

svr\_results = evaluate\_model(svr\_model, X\_train, X\_test, y\_train, y\_test, scaler\_y)

Dự đoán giá cho 12 giờ tiếp theo:

future\_prices = predict\_next\_hours(best\_model, last\_data, scaler\_y, hours=12)

**3. KẾT QUẢ VÀ SO SÁNH CÁC MÔ HÌNH**

Từ kết quả chạy mô hình trong log, ta thấy:

1. Linear Regression:
   * RMSE: $321.63
   * MAE: $157.52
   * R²: 0.9998
   * MAPE: 0.3705%
2. Ridge Regression:
   * RMSE: $338.98
   * MAE: $170.83
   * R²: 0.9998
   * MAPE: 0.3998%
3. Decision Tree:
   * RMSE: $7371.42
   * MAE: $2365.37
   * R²: 0.8869
   * MAPE: 3.5769%
4. Random Forest:
   * RMSE: $7400.27
   * MAE: $2284.47
   * R²: 0.8860
   * MAPE: 3.3339%
5. KNN Regression:
   * RMSE: $8285.57
   * MAE: $4057.83
   * R²: 0.8571
   * MAPE: 8.7585%
6. SVM (Support Vector Regression):
   * RMSE: $11978.55
   * MAE: $5660.74
   * R²: 0.7012
   * MAPE: 9.8339%

Linear Regression cho kết quả tốt nhất với RMSE thấp nhất ($321.63) và R² cao nhất (0.9998).

Dự đoán giá Bitcoin cho 12 giờ tiếp theo: 2025-03-14 13:00: $83822.75 2025-03-14 14:00: $84336.12 2025-03-14 15:00: $82414.73 2025-03-14 16:00: $83154.16 2025-03-14 17:00: $84637.99 2025-03-14 18:00: $84978.45 2025-03-14 19:00: $85641.84 2025-03-14 20:00: $86240.69 2025-03-14 21:00: $86754.22 2025-03-14 22:00: $87272.07 2025-03-14 23:00: $86309.88 2025-03-15 00:00: $86806.17

**VI. KẾT LUẬN**

Từ kết quả thực nghiệm, có thể rút ra một số kết luận:

1. Mô hình Linear Regression đã được chọn là mô hình tốt nhất cho dự đoán giá Bitcoin theo giờ với RMSE thấp nhất ($321.63) và R² cao nhất (0.9998).
2. So sánh các mô hình từ slide môn học:
   * Linear Regression và Ridge Regression (Lecture 3) cho kết quả tốt nhất trong bài toán này
   * Decision Tree và Random Forest (Lecture 8) có hiệu suất trung bình
   * KNN (Lecture 7) và SVM (Lecture 11) cho kết quả kém hơn
3. Đặc điểm quan trọng của dự đoán:
   * Dự đoán giá Bitcoin theo giờ thay vì theo ngày giúp nắm bắt tốt hơn biến động ngắn hạn
   * Các chỉ báo kỹ thuật cung cấp thông tin quan trọng để dự đoán sự biến động của giá
4. Lưu ý quan trọng: Thị trường tiền điện tử rất biến động và chịu ảnh hưởng của nhiều yếu tố không lường trước được. Dự đoán này chỉ nên được sử dụng như một tham khảo, không nên dùng làm cơ sở duy nhất cho quyết định đầu tư.

**Trong tương lai, hệ thống có thể được cải thiện bằng cách:**

* Bổ sung thêm các đặc trưng từ phân tích cảm xúc trên mạng xã hội
* Thử nghiệm các mô hình học sâu như LSTM cho dự đoán chuỗi thời gian
* Phát triển hệ thống cảnh báo khi giá đạt ngưỡng nhất định
* Tích hợp dữ liệu từ nhiều sàn giao dịch để có cái nhìn toàn diện hơn