TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Thái Gia Bảo – 52000014**

**BÁO CÁO CUỐI KÌ**

**XỬ LÝ DỮ LIỆU LỚN**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Thái Gia Bảo – 52000014**

**BÁO CÁO CUỐI KÌ**

**XỬ LÝ DỮ LIỆU LỚN**

Người hướng dẫn

**ThS. Nguyễn Thành An**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

# LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Thầy Nguyễn Thành An – Giảng viên khoa Công nghệ thông tin – Trường đại học Tôn Đức Thắng, đã hỗ trợ và giúp đỡ nhiệt tình trong quá trình thực hiện Dự án này.

Chúng em trân trọng cảm ơn Thầy Cô giảng viên Trường đại học Tôn Đức Thắng nói chung cũng như Thầy Cô giảng viên khoa Công nghệ thông tin nói riêng đã giảng dạy và truyền đạt nhiều kinh nghiệm quý trong suốt quá trình học tập tại trường.

Cuối cùng, xin cám ơn gia đình, bạn bè đã luôn động viên và đồng hành trong quá trình học tập cũng như quá trình thực hiện Dự án này.

Mặc dù rất cẩn thận trong quá trình thực hiện đồ án cũng như viết báo cáo nhưng cũng không tránh khỏi những thiếu sót. Chúng em rất mong nhận đựợc sự góp ý từ các Thầy để đồ án được hoàn thiện hơn.

Xin chân thành cám ơn!

*TP. Hồ Chí Minh, ngày 21 tháng 5 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Thái Gia Bảo*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của ThS. Nguyễn Thành An. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 21 tháng 5 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Thái Gia Bảo*

# TÓM TẮT

Tóm tắt yêu cầu của bài đồ án:

**Câu 1: Phân Cụm Dữ Liệu (2.0 điểm)**

* **Dữ liệu**: mnist\_mini.csv.
* **Nhiệm vụ**: Sử dụng thuật toán k-Means với giá trị k=10. Gán trọng số gấp 100 lần cho các điểm dữ liệu tại các dòng 0, 1, 2, 3, 4, 7, 8, 11, 18, 61.
* **Yêu cầu**: Tính trung bình khoảng cách từ các điểm dữ liệu tới centroid và vẽ biểu đồ cột.

**Câu 2: Giảm Số Chiều với SVD (2.0 điểm)**

* **Dữ liệu**: mnist\_mini.csv.
* **Nhiệm vụ**: Giảm số chiều từ 784 xuống 3 bằng SVD. Chọn ngẫu nhiên 100 điểm dữ liệu và sử dụng kết quả phân cụm từ Câu 1 để vẽ biểu đồ 3D phân bố.

**Câu 3: Khuyến Nghị Sản Phẩm với Collaborative Filtering (2.0 điểm)**

* **Dữ liệu**: ratings2k.csv.
* **Nhiệm vụ**: Chia dữ liệu thành tập training (70%) và test (30%). Sử dụng thuật toán ALS để khảo sát hiệu suất theo độ đo MSE với các giá trị số lượng người dùng tương đồng trong đoạn [10; 20]. Vẽ biểu đồ cột để trực quan hóa MSE.

**Câu 4: Dự Đoán Giá Chứng Khoán (2.0 điểm)**

* **Dữ liệu**: stockHVN2022.csv.
* **Nhiệm vụ**: Tạo cột “fluctuation” chứa biên độ dao động giá cổ phiếu. Dự đoán biên độ ngày tiếp theo dựa trên biên độ 5 ngày liền trước bằng mô hình Linear Regression. Tính Mean Square Error (MSE) trên tập training và test, và vẽ biểu đồ cột.

**Câu 5: Phân Loại Đa Lớp với Pyspark (1.0 điểm)**

* **Dữ liệu**: mnist\_mini.csv.
* **Nhiệm vụ**: Xây dựng mô hình phân loại đa lớp với 3 mô hình: Multi-layer Perceptron, Random Forest, Linear SVM. Đánh giá độ chính xác (Accuracy) và vẽ biểu đồ cột đôi so sánh độ chính xác trên tập training và test.

**MỤC LỤC**

Contents

[LỜI CẢM ƠN iii](#_Toc195882561)

[TÓM TẮT v](#_Toc195882562)

[DANH MỤC CÁC CHỮ VIẾT TẮT viii](#_Toc195882563)

[1. Câu 1: Phân cụm dữ liệu ix](#_Toc195882564)

[1.1. Giới thiệu ix](#_Toc195882565)

[1.2. Công cụ và thư viện sử dụng ix](#_Toc195882566)

[1.3. Quy trình thực hiện ix](#_Toc195882567)

[1.4. Mã giả ix](#_Toc195882568)

[1.5. Kết quả x](#_Toc195882569)

[2. Câu 2: Giảm số chiều với SVD xi](#_Toc195882570)

[2.1. Giới thiệu xi](#_Toc195882571)

[2.2. Quy trình thực hiện xi](#_Toc195882572)

[2.3. Mã giả xiii](#_Toc195882573)

[2.4. Kết quả xiv](#_Toc195882574)

[3. Câu 3: Khuyến nghị sản phẩm với Collabrative Filtering xiv](#_Toc195882575)

[3.1. Giới thiệu xiv](#_Toc195882576)

[3.2. Quy trình thực hiện xv](#_Toc195882577)

[3.3. Mã giả xv](#_Toc195882578)

[3.4. Kết quả xvi](#_Toc195882579)

[**3.4.1.** **MSE** xvi](#_Toc195882580)

[**3.4.2.** **Biểu đồ trực quan** xvii](#_Toc195882581)

[4. Câu 4: Dự đoán giá chứng khoán xvii](#_Toc195882582)

[4.1. Giới thiệu xvii](#_Toc195882583)

[4.2. Phân tích và chuẩn bị dữ liệu xvii](#_Toc195882584)

[4.3. Xây dựng và đánh giá mô hình xviii](#_Toc195882585)

[**4.3.1.** **Xây dựng mô hình** xviii](#_Toc195882586)

[**4.3.2.** **Đánh giá mô hình** xviii](#_Toc195882587)

[4.4. Mã giả xviii](#_Toc195882588)

[4.5. Kết quả xix](#_Toc195882589)

[5. Câu 5: Phân loại đa lớp với pyspark xix](#_Toc195882590)

[5.1. Giới thiệu xix](#_Toc195882591)

[5.2. Quy trình thực hiện xix](#_Toc195882592)

[BẢNG TỰ ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH 2](#_Toc195882593)

[TÀI LIỆU THAM KHẢO 3](#_Toc195882594)

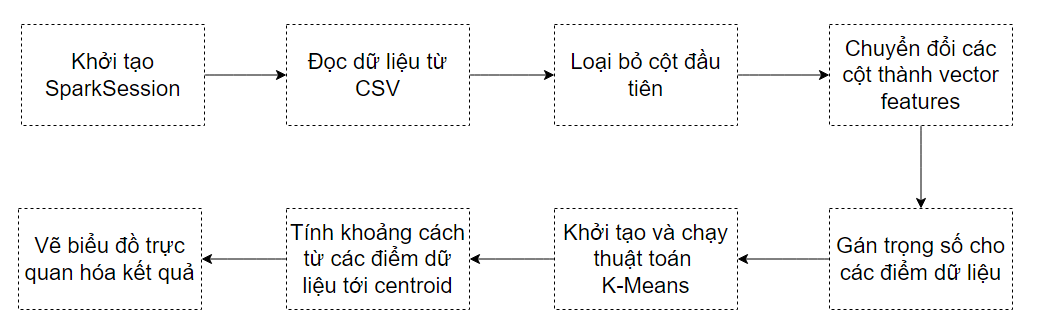
# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| MSE | Mean Squared Error |

1. **Câu 1: Phân cụm dữ liệu**
   1. **Giới thiệu**

Mục tiêu của bài này là áp dụng thuật toán K-Means để phân cụm dữ liệu từ tập dữ liệu MNIST, một tập dữ liệu hình ảnh chữ số viết tay phổ biến, nhằm tìm hiểu về kỹ thuật phân cụm dữ liệu và đánh giá hiệu quả của thuật toán khi áp dụng các trọng số khác nhau cho các điểm dữ liệu cụ thể.

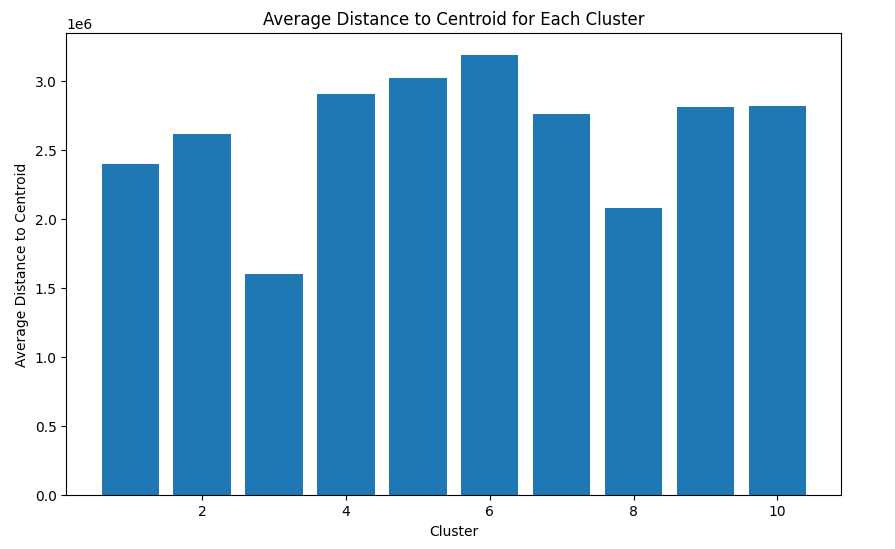
* 1. **Công cụ và thư viện sử dụng**
* **PySpark**: Một công cụ xử lý dữ liệu lớn phân tán, cho phép xử lý và phân tích dữ liệu trên cụm máy tính.
* **Matplotlib**: Thư viện Python dùng để vẽ biểu đồ và trực quan hóa dữ liệu.
* **Pandas**: Thư viện Python dùng để thao tác và phân tích dữ liệu.
* **Scikit-learn**: Thư viện Python cung cấp các công cụ học máy đơn giản và hiệu quả để phân tích dữ liệu.
  1. **Quy trình thực hiện**

****

* 1. **Mã giả**

|  |
| --- |
| **class KMeansAnalysis:**  ***def \_\_init\_\_(self, spark):***  ***def read\_data(self, file\_path):***  # Đọc dữ liệu từ file CSV và trả về DataFrame  ***def preprocess\_data(self, df):***  # Loại bỏ cột đầu tiên (loại ký số)  # Tạo vector từ các cột pixel  # Định nghĩa UDF để nhân trọng số  # Gán trọng số gấp 100 lần cho các điểm dữ liệu được chỉ định  # Nhân trọng số vào các thành phần của vector bằng UDF  ***def run\_kmeans(self, df, k):***  # Khởi tạo và chạy thuật toán KMeans với số cụm k  ***def calculate\_avg\_distance(self, model):***  # Lấy ra centroids từ model  # Sử dụng hàm transform để dự đoán nhóm cho mỗi điểm dữ liệu và tính khoảng cách tới centroid  # Tính khoảng cách từ mỗi điểm tới centroid của nhóm mà nó thuộc với  # Tính trung bình khoảng cách cho mỗi cụm  ***def plot\_avg\_distances(self, cluster\_avg\_distances):***  # Trích xuất cluster và giá trị trung bình khoảng cách  # Vẽ biểu đồ cột |

* 1. **Kết quả**

****

1. **Câu 2: Giảm số chiều với SVD**
   1. **Giới thiệu**

* Trong câu này, chúng em sẽ sử dụng phương pháp Phân rã giá trị suy biến (Singular Value Decomposition - SVD) để giảm số chiều của tập dữ liệu từ 784 xuống còn 3.
* Sau đó, chọn ngẫu nhiên 100 điểm dữ liệu đã được giảm số chiều và vẽ biểu đồ 3D để mô tả phân bố của các điểm này trong không gian 3 chiều.
* Cuối cùng, sử dụng thư viện PySpark để thực hiện giảm số chiều và matplotlib để vẽ biểu đồ.
  1. **Quy trình thực hiện**

Các bước thực hiện tuần tự như sau:

1. Khởi tạo và đọc dữ liệu:

* Khởi tạo SparkSession.
* Đọc dữ liệu MNIST từ tệp "mnist\_mini.csv" vào DataFrame.

1. Tiền xử lý dữ liệu:

* Lưu lại cột đầu tiên (loại ký số) chứa nhãn của ảnh (label).
* Tạo vector từ các cột pixel (bỏ cột đầu tiên).
* Thêm lại cột nhãn (label) vào DataFrame.

1. Giảm số chiều:
   * Thêm cột tạm thời "id" để theo dõi dữ liệu gốc.
   * Chuyển đổi DataFrame thành mảng NumPy để sử dụng SVD.
   * Sử dụng thuật toán SVD để giảm số chiều dữ liệu từ 784 xuống 3.
   * Chuyển đổi kết quả SVD trở lại thành Spark DataFrame.
   * Thêm lại cột nhãn (label) vào DataFrame kết quả.
   * Loại bỏ cột tạm thời "id".
2. Chạy KMeans trên dữ liệu giảm chiều:

* Chạy thuật toán KMeans với k=10 trên dữ liệu đã giảm số chiều.

1. Lấy mẫu dữ liệu:

* Chọn ngẫu nhiên 100 điểm dữ liệu từ DataFrame sau khi giảm số chiều.

1. Lấy nhãn phân cụm:

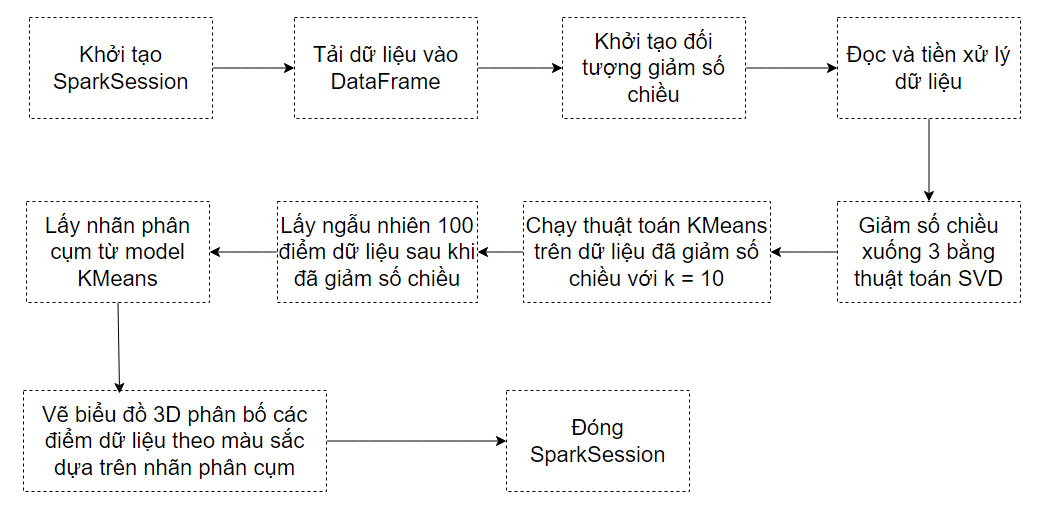
* Sử dụng model KMeans để dự đoán nhãn phân cụm cho các điểm dữ liệu được lấy mẫu.

1. Vẽ biểu đồ 3D:

* Chuyển đổi dữ liệu thành mảng NumPy.
* Vẽ biểu đồ 3D phân bố các điểm dữ liệu theo màu sắc dựa trên nhãn phân cụm.

1. Đóng SparkSession:

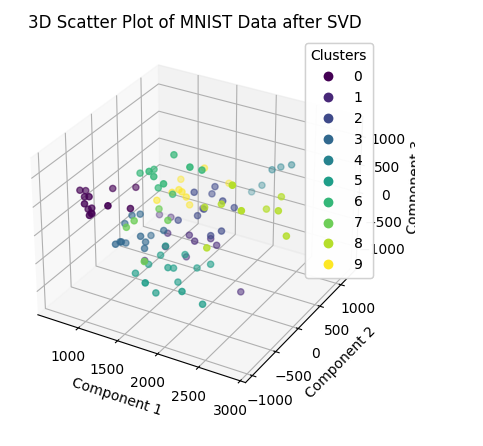
* Đóng SparkSession.

****

* 1. **Mã giả**

|  |
| --- |
| ***class DimensionalityReduction:***  ***def \_\_init\_\_(self, spark):***  ***def read\_data(self, file\_path):***          # Đọc dữ liệu từ file CSV và trả về DataFrame          # Lưu lại cột đầu tiên (loại ký số)          # Tạo vector từ các cột pixel (bỏ cột đầu tiên)          # Thêm lại cột label vào DataFrame          # Thêm cột id để giữ lại thông tin ban đầu            # Chuyển đổi DataFrame thành mảng NumPy để sử dụng SVD          # Giảm số chiều bằng SVD          # Chuyển đổi lại thành DataFrame Spark  ***def sample\_data(self, df, sample\_size):***          # Chọn ngẫu nhiên sample\_size điểm dữ liệu  ***def run\_kmeans\_on\_reduced\_data(self, df, k):***          # Chạy thuật toán KMeans trên dữ liệu đã giảm số chiều  ***def get\_cluster\_labels(self, model, df):***          # Đổi tên cột để phù hợp với mô hình KMeans  ***def plot\_3d\_clusters(self, sampled\_df, cluster\_labels):***          # Chuyển đổi thành mảng NumPy để vẽ biểu đồ          # Vẽ các điểm dữ liệu với màu sắc dựa trên nhãn phân cụm |

* 1. **Kết quả**

****

1. **Câu 3: Khuyến nghị sản phẩm với Collabrative Filtering**
   1. **Giới thiệu**

* Trong thời đại ngày nay, việc khuyến nghị sản phẩm dựa trên dữ liệu người dùng trở nên ngày càng phổ biến. Collaborative Filtering là một trong những phương pháp phổ biến nhất trong lĩnh vực này, giúp dự đoán sở thích của người dùng dựa trên hành vi đánh giá hoặc mua hàng trước đó của họ.
* Trong báo cáo này, chúng em áp dụng Collaborative Filtering trên tập dữ liệu "ratings2k.csv" để xây dựng một mô hình khuyến nghị sản phẩm.
  1. **Quy trình thực hiện**

Các bước thực hiện tuần tự như sau:

1. **Khởi tạo và đọc dữ liệu:**

* Khởi tạo phiên làm việc với Spark và đọc dữ liệu từ tệp CSV.
* Hiển thị schema và dữ liệu.

1. **Phân chia dữ liệu:**

* Phân chia dữ liệu thành tập huấn luyện (70%) và kiểm tra (30%).

1. **Huấn luyện và đánh giá mô hình ALS:**

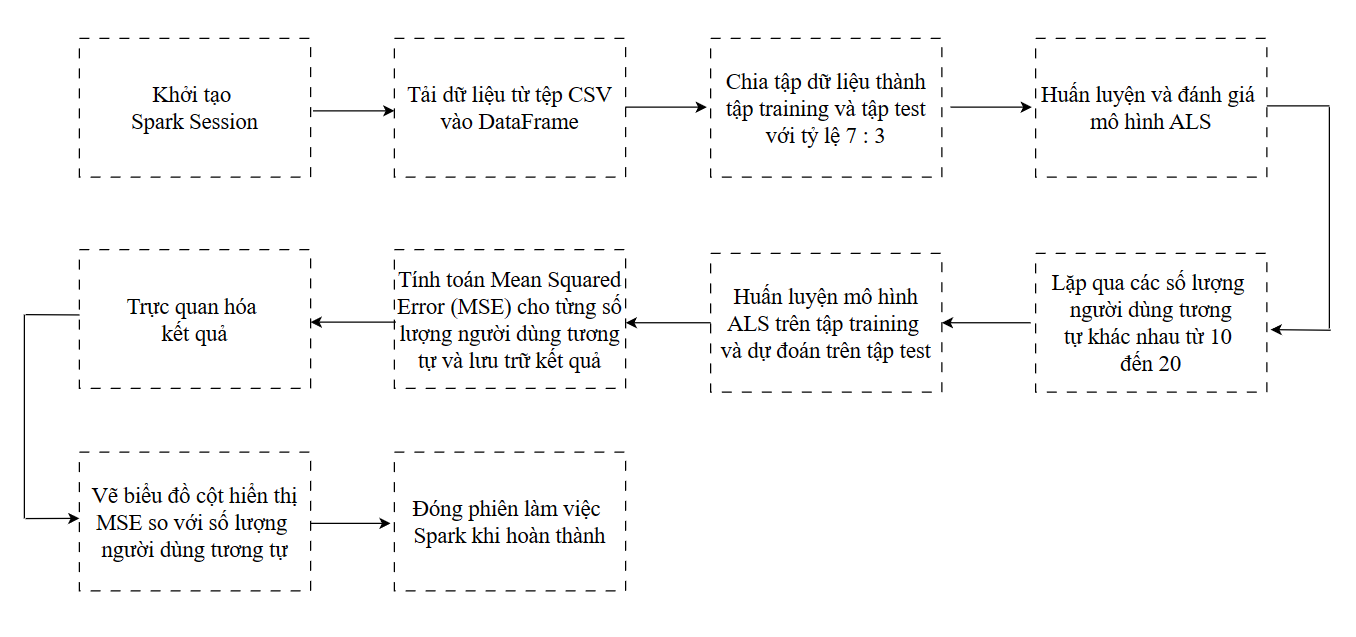
* Lặp qua các số lượng người dùng tương tự khác nhau từ 10 đến 20.
* Huấn luyện mô hình ALS trên tập huấn luyện và dự đoán trên tập kiểm tra.
* Tính toán Mean Squared Error (MSE) cho từng số lượng người dùng tương tự và lưu trữ kết quả.

1. **Trực quan hóa kết quả:**

* Vẽ biểu đồ cột hiển thị MSE so với số lượng người dùng tương tự.

1. **Đóng phiên làm việc Spark:**

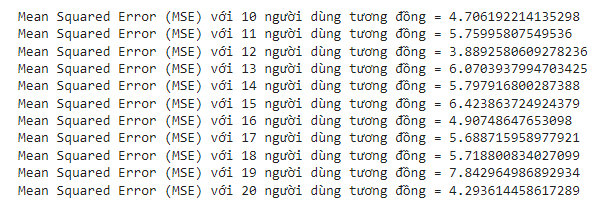
* Đóng phiên làm việc với Spark khi hoàn thành.



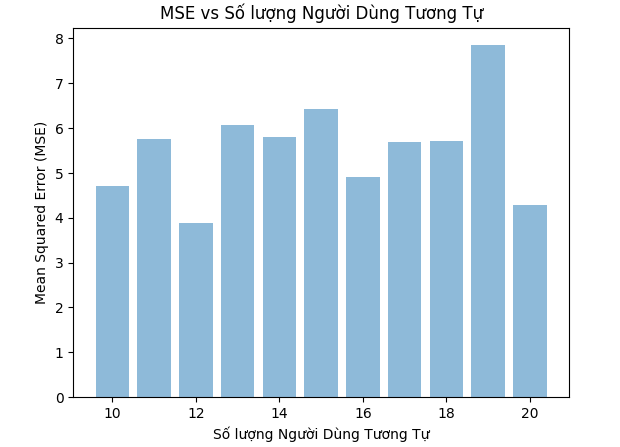
* 1. **Mã giả**

|  |
| --- |
| ***class CollaborativeFilteringModel:***  ***def \_\_init\_\_(self, spark, file\_path):***  # Khởi tạo phiên Spark và đường dẫn tệp  # Khởi tạo dữ liệu, tập huấn luyện và tập kiểm tra  # Khởi tạo danh sách để lưu trữ các giá trị MSE  ***def load\_data(self):***  # Tải dữ liệu từ tệp CSV vào DataFrame  # In schema và hiển thị dữ liệu  # Phân chia dữ liệu thành các tập huấn luyện và kiểm tra  # In dữ liệu huấn luyện và kiểm tra  # Lặp qua các số lượng người dùng tương tự khác nhau  # Huấn luyện mô hình ALS với dữ liệu huấn luyện  # Dự đoán trên dữ liệu kiểm tra  # Đánh giá mô hình sử dụng MSE  # Thêm giá trị MSE vào danh sách  # In MSE cho số lượng người dùng tương tự hiện tại  ***def plot\_mse(self, num\_similar\_users):***  # Vẽ biểu đồ MSE so với số lượng người dùng tương tự |

* 1. **Kết quả**
     1. **MSE**

****

* + 1. **Biểu đồ trực quan**

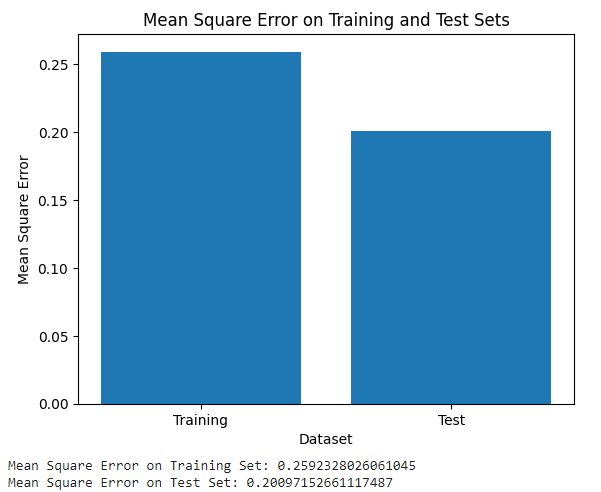
****

1. **Câu 4: Dự đoán giá chứng khoán**
   1. **Giới thiệu**

* **Đề tài**: Dự đoán giá chứng khoán sử dụng mô hình Linear Regression trên Apache Spark.
* **Mục tiêu**: Dự đoán biên độ dao động của giá cổ phiếu HVN dựa trên dữ liệu lịch sử.
  1. **Phân tích và chuẩn bị dữ liệu**
* **Dữ Liệu Sử Dụng**: Tệp stockHVN2022.csv.
* **Biên Độ Dao Động**: Tính toán biên độ dao động của giá cổ phiếu theo công thức:
* **Chia Dữ Liệu**:
  + Sử dụng dữ liệu từ tháng 01 đến hết tháng 06 làm tập train.
  + Từ tháng 07 đến hết cho tập test.
  1. **Xây dựng và đánh giá mô hình** 
     1. **Xây dựng mô hình**
* **Mô Hình**: Sử dụng Linear Regression để dự đoán biên độ dao động.
* **Vector Features**: Sử dụng biên độ 5 ngày trước làm đặc trưng.
* **Huấn Luyện Mô Hình**: Sử dụng tập train để huấn luyện mô hình.
  + 1. **Đánh giá mô hình**
* **Sai Số Mean Square Error (MSE)**: Đánh giá mô hình trên cả tập train và tập test.
* **Kết Quả**: MSE trên tập test để đánh giá hiệu suất dự đoán của mô hình.
  1. **Mã giả**

|  |
| --- |
| ***class StockPredictor:***  ***def \_\_init\_\_(self, spark, file\_path):***  # Khởi tạo các thuộc tính và đọc dữ liệu từ file CSV  ***def read\_data(self):***  # Đọc dữ liệu từ file CSV  ***def calculate\_fluctuation(self):***  # Tính toán biên độ dao động của giá cổ phiếu  ***def split\_data(self):***  # Chia dữ liệu thành tập train và tập test  ***def prepare\_features(self):***        # Tạo vector features từ biên độ 5 ngày trước  ***def train\_model(self):***  # Huấn luyện mô hình Linear Regression  ***def evaluate\_model(self):***  # Đánh giá mô hình trên tập train và tập test  ***def plot\_mse(self):***  # Vẽ biểu đồ MSE trên tập train và tập test |

* 1. **Kết quả**

****

1. **Câu 5: Phân loại đa lớp với pyspark**
   1. **Giới thiệu**

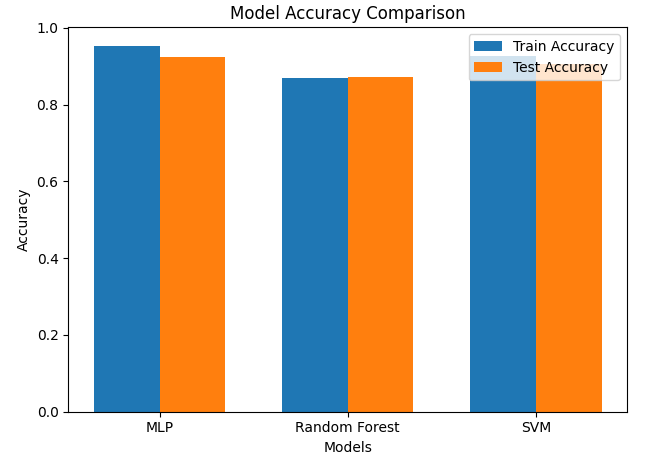
Bài toán này tập trung vào việc xây dựng và đánh giá các mô hình phân loại đa lớp bằng cách sử dụng PySpark.

Ba mô hình được sử dụng là:

* Multilayer Perceptron
* Random Forest
* Linear Support Vector Machine (SVM)
* Mục tiêu của dự án là đánh giá độ chính xác của các mô hình trên tập kiểm tra và so sánh chúng.
  1. **Quy trình thực hiện**
  2. **Mã giả**

|  |
| --- |
| ***class DataProcessor:***          # Chuyển đổi tên cột và tạo DataFrame Spark          # Tạo vectơ đặc trưng và chia dữ liệu thành tập huấn luyện và tập kiểm tra    ***class MNISTClassifier:***  ***def \_\_init\_\_(self):***          # Khởi tạo đối tượng MulticlassClassificationEvaluator    ***def train\_evaluate(self, model, train\_data, test\_data):***          # Huấn luyện mô hình và đánh giá độ chính xác  # Lớp vẽ biểu đồ  ***class Plotter:***  ***def plot\_accuracies(models, train\_accuracies, test\_accuracies):***          # Vẽ biểu đồ so sánh độ chính xác |

* 1. **Kết quả**

****

**THUẬN LỢI VÀ KHÓ KHĂN CỦA ĐỀ TÀI**

# BẢNG TỰ ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Yêu cầu** | **Người thực hiện** | **Mức độ hoàn thành** | | | |
| 0% | 25% | 50% | 100% |
|  |  |  |  |  | 🗸 |
|  |  |  |  |  | 🗸 |
|  |  |  |  |  | 🗸 |
|  |  |  |  |  | 🗸 |

# TÀI LIỆU THAM KHẢO

**Tiếng Việt**

Nguyễn Văn A, Trần Thị B. (20XX). On the Application of the N-Queens Problem in Computer Science. Journal of Computer Science, 10(2), 123–135.

Lê Thị C, Nguyễn Đình D. (20YY). A Survey of N-Queens Problem Solving Approaches. In Proceedings of the International Conference on Artificial Intelligence (ICAI), (pp. 45-56).

**Tiếng Anh**

Smith, J., & Johnson, E. (20XX). A Survey of N-Queens Problem Solving Approaches. International Journal of Artificial Intelligence Research, 5(2), 123–135.