TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

BÁO CÁO CUỐI KÌ MÔN HỌC
Nhập Môn Xử Lí Dữ Liệu Lớn

BÀI TẬP TỔNG HỢP MÔN DỮ LIỆU LỚN

Giáo viên hướng dẫn: ThS. Nguyễn Thành An
Sinh viên thực hiện: Dương Trọng Chí - MSSV: 52000742
Trần Khánh Duy - MSSV: 52000042
Trần Gia Hoàng - MSSV: 52000759

Hồ Chí Minh - 2022

Mục lục

Bảng phân chia công việc

MSSV	Họ Tên	Email	Phân công công việc	Mức độ hoàn thành
52000759	Trần Gia Hoàng	52000759@student.tdtu.edu.vn	- Tìm hiểu câu 1, 2, 3, 4. - Demo code câu 3, 4	100%
52000042	Trần Khánh Duy	52000042@student.tdtu.edu.vn	- Tìm hiểu câu 1, 2, 3, 4 - Demo code câu 2, 4	100%
52000742	Dương Trọng Chí	52000742@student.tdtu.edu.vn	- Tìm hiểu câu 1, 2, 3, 4 - Dem code câu 1, 4	100%

Chương 1

Bài làm

1.1 Câu 1

Để hiển thị hình ảnh của các loài động vật ta sẽ định nghĩa một phương thức có tên là `plot_image_grid()`. Phương thức này sẽ gọi phương thức `imgshow()` để vẽ hình các động vật. Đồng thời, ứng với mỗi loài vật phương thức `plot_image_grid()` sẽ gán các label tương ứng với loài vật đó.

Để thực hiện giảm số chiều của ma trận bằng thuật toán SVD. Đầu tiên, ta dùng phương thức computeSVD() của thư viện numpy để tách ma trận ban đầu lần lượt thành 3 ma trận con bao gồm U, s, V. Sau đó, ta sẽ chuyển các ma trận này thành dạng mảng numpy để thực hiện việc tính toán trên các ma trận, ta chuyển ma tra trận U, s, V về dạng numpy.array, ta sẽ tiến hành dùng phương thức U.rows.collect() cho ma trận U để tiến hành làm phẳng, tiến hành hoán vị đối với ma trận V cuối cùng là nhân 3 ma trận này lại. Kết quả là ta sẽ có một ma trận mới với số chiều đã được giảm.

Để chuyển ma trận này về thành ma trận có kích thước 128x128x3 ta tiến hành dùng hàm `zip(), map()` để chuyển các phần tử từ dạng `((title,row),index)` về dạng index, title, row và lưu kết quả vào biến có tên là `pet3_r100_train`. Sau đó, ta tiến hành dùng hàm map để chuyển từng phần tử trong biến `pet3_r100_train` về dạng ma trận có kích thước 128x128x3.

1.2 Câu 2

Chuyển đổi kiểu dữ liệu cho các cột thông qua RDD với row, create dataframe sử dụng order by "user" để dữ liệu được xắp sếp tăng dần thông qua user id.

Sử dụng tập rating2k như tập training để test với tập test là user có id > 70 (4 user còn lại). Sử dụng mô hình ALS với UserBlocks là 70 để dữ đoán cho 4 user cuối cùng. Evaluate bằng MSE và so sánh độ tương đồng bằng Pearson Correlation Coefficient.

1.3 Câu 3

Có 3 bước cần làm ở bài dự đoán giá cổ phiếu này:

- Bước 1: Xử lý phân chia bộ dữ liệu gốc ra thành hai tập train và test thông hàm splitData(rdd). Sau đó thông qua hàm `convertDataForModel(data)` để chuyển đổi dữ liệu ở hai tập train và test sang dạng (Features, Nextday). Features chứa các đặc trưng cho mô hình (ở bài này là giá cổ phiếu của 5 ngày kế tiếp nhau và Nextday là giá cổ phiếu của ngày kế tiếp). Tạo Dataframe cho hai tập dữ liệu train và test. Dùng VectorAssembler để chuyển các cột thuộc Features của tập train và test thành một vector và lưu trong cột "Independent Features" của Dataframe.

- Bước 2: Sử dụng mô hình Linear regression và cho mô hình học dữ liệu train thông qua hàm `fit()`. Dùng hàm `evaluate()` và truyền vào các tập train, test cho mô hình dự đoán ra kết quả.

- Bước 3: Tính đoán độ đo Mean Squared Error trên cả tập train và test để biết sai số của mô hình.

1.4 Câu 4

Để tiến hành huấn luyện mô hình:

- Bước 1: Ta tiến hành tách dữ liệu thành 2 cột là label và features bằng phương thức `map()`. Cột label sẽ chứa nhãn của các loài động vật và cột features là vector chứa các đặc trưng của bức ảnh. Sau đó, ta tiến hành chuyển chúng thành dataFrame bởi phương thức `toDF()`.

- Bước 2: Ta tiến hành gọi mô hình LogisticRegression để tiến hành phân loại các lớp và dùng `OneVsRest()` để tiến hành phân loại được nhiều label. Sau đó, ta tiến hành dùng phương thức `fit()` để tiến hành huấn luyện mô hình.

Ta tiến hành đánh giá độ chính xác accuracy của mô hình ta dùng phương thức `evaluate` bằng cách gọi class `MulticlassClassificationEvaluator()`.