Đề thi:

R PROGRAMMING LANGUAGE FOR DATA SCIENCE

Thời gian làm bài : từ khi nhận đề đến 23h30, Chủ Nhật ngày 08/05/2022

Đọc kỹ các thông tin dưới đây trước khi làm bài :

- HV tạo một folder là **LDS7_HoVaTen_Cuoi_Ky** (nằm trong folder **LDS7_K274_ONLINE_HoVaTen** đã share trên Google Drive), lưu tất cả bài làm vào để GV chấm điểm.
- Đến deadline, HV gửi mail cho giáo viên kèm link của folder **LDS7_HoVaTen_Cuoi_Ky**, HV không gửi bài thi sẽ không có điểm thi.
- HV được sử dụng tài liệu.
- HV sẽ bị trừ điểm nếu bài làm giống nhau.

Chú ý, với mỗi câu:

- Lần lượt thực hiện các bước làm bài như đã được hướng dẫn làm demo/ bài tập trong lớp.
- Tiền xử lý dữ liệu (nếu cần)
- Mỗi câu là 1 file, các yêu cầu nhận xét kết quả trong từng câu được viết trong cell dưới định dạng Markdown.

1. Monthly champagne sales millions – Time series Analysis (1.0 điểm)

- *Tạo tập tin: **question_1.ipynb** (toàn bộ code của câu 1 sẽ được viết trong file này)*
- Cho dữ liệu **champagne_new.xlsx** là dữ liệu bán champagne theo thời gian từ tháng 01-1964 đến tháng 09-1972
- Yêu cầu:
    1. Đọc dữ liệu
    2. Xem thông tin chung từ dữ liệu: head(), số dòng, số cột, str()…
    3. Chuyển dữ liệu này thành Time Series object => in Time Series object.
    4. Vẽ Time Series object vừa tạo.
    5. Thực hiện việc decompostition, nhận xét.
    6. Thực hiện việc dự báo và vẽ biểu đồ so sánh với thực tiễn.
    7. Dự đoán số tiền champagne bán được của **9 tháng tiếp theo** từ tháng 10-1972 đến tháng 06-1973.

2. Normal – Binormial Distribution (0.5 điểm)

- *Tạo tập tin: **question_2.ipynb** (toàn bộ code của câu 2 sẽ được viết trong file này)*
- Thực hiện các yêu cầu sau:
    1. Giả sử chỉ số IQ thường được phân phối với giá trị trung bình là 100 và độ lệch chuẩn là 15.
        a. Vậy tỷ lệ bao nhiêu phần trăm người có IQ nhỏ hơn 125?
        b. Vậy tỷ lệ bao nhiêu phần trăm người có IQ lớn hơn 110?
        c. Vậy tỷ lệ bao nhiêu phần trăm người có IQ trong khoảng từ 110 và 125 ?
    2. Nếu X được phân phối nhị thức (binomial distribution) với 6 lần thử và xác suất thành công bằng 1/4 ở mỗi lần thử thì xác suất của 4 lần thành công là bao nhiêu?
    3. Xúc xắc có 6 mặt, tìm xác suất để có được 2 lần mặt 4 nút trong 5 lần đổ xúc xắc.

3. Lung Function in 1 to 10 Year Old Children (1.5 điểm)

- *Tạo tập tin: **question_3.ipynb** (toàn bộ code của câu 3 sẽ được viết trong file này)*
- Cho dữ liệu **children_lung.csv**. Bộ dữ liệu có 654 trẻ em từ 1 đến 10 tuổi. Với output **y** là **FEV (forced exhalation volume)**, thước đo lượng không khí mà ai đó có thể buộc phải thở ra từ phổi của họ và inputs **X** là một hoặc nhiều thuộc tính còn lại (Nguồn dữ liệu: Dữ liệu này là một phần của bộ dữ liệu được đưa ra trong Kahn, Michael (2005). "An Exhalent Problem for Teaching Statistics", The Journal of Statistical Education, 13".
- Yêu cầu: Sử dụng **Linenear Regression** để thực hiện việc **dự đoán FEV** dựa trên các thuộc tính **age/ age và ht**.
- Gợi ý các bước thực hiện:
    1. Đọc dữ liệu và gán cho biến data.
    2. Xem thông tin data: head(), số dòng, số cột, summary…
    3. Tiền xử lý dữ liệu (nếu cần).
    4. Vẽ biểu đồ quan sát mối liên hệ giữa FEV và age. Quan sát và nhận xét. Có vấn đề gì đặc biệt từ dữ liệu không? Nếu có thì đó là vấn đề gì?
    5. Chia dữ liệu data thành 2 bộ dữ liệu **data_FEV_less_10** (chứa các mẫu có FEV <10) và **data_FEV_more_10** (chứa các mẫu có FEV >=10)
    6. Với **data_FEV_more_10**:
        a. Thực hiện Simple Linear Regression để **dự đoán FEV từ age**. Xây dựng model. Đánh giá model.
        b. Cho age lần lượt là: [2, 3, 4, 5]. Hãy cho biết FEV lần lượt là bao nhiêu?
        c. Trực quan hóa kết quả.
    7. Với **data_FEV_less_10**:
        a. Thực hiện Multiple Linear Regression để **dự đoán FEV từ age và ht**.
        b. Cho age và ht lần lượt là: age = [5, 6, 7, 8, 9], ht = [49.5, 55, 57, 60, 62] . Hãy cho biết FEV lần lượt là bao nhiêu?
        c. Trực quan hóa kết quả.

4. Mushroom (1.0 điểm)

- *Tạo tập tin: **question_4.ipynb** (toàn bộ code của câu 4 sẽ được viết trong file này)*
- Cho dữ liệu mushroom trong tập tin **mushrooms.csv** chứa thông tin của các mẫu nấm, nấm ăn được và không ăn được.
    - Dữ liệu có thể tham khảo và download tại: https://www.kaggle.com/jnduli/decision-tree-classifier-for-mushroom-dataset/data

Data Infromation : Bộ dữ liệu chứa 23 thuộc tính. Thuộc tính "**class**" là class attribute (output).

Attribute Information:

- **class: edible=e, poisonous=p**
- cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
- cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
- cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
- bruises: bruises=t, no=f

- odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
- gill-attachment: attached=a, descending=d, free=f, notched=n
- gill-spacing: close=c, crowded=w, distant=d
- gill-size: broad=b,narrow=n
- gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
- stalk-shape: enlarging=e, tapering=t
- stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
- stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s
- stalk-surface-below-ring: fibrous=f, scaly=y, silky=k, smooth=s
- stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
- stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
- veil-type: partial=p,universal=u
- veil-color: brown=n, orange=o, white=w, yellow=y
- ring-number: none=n, one=o,two=t
- ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
- spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
- population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
- habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

- Yêu cầu: Sử dụng **cả Logistic Regression và Decision Tree** để thực hiện việc xác định một mẫu nấm là **nấm ăn được** hay **nấm độc** dựa vào các thông tin còn lại. Trong hai thuật toán trên thì thuật toán nào phù hợp hơn cho bộ dữ liệu này? Vì sao ?
- Gợi ý các bước thực hiện cho từng thuật toán :
  1. Đọc dữ liệu và gán cho biến data.
  2. Xem thông tin data: head(), số dòng, số cột, summary…
  3. Tiền xử lý dữ liệu (nếu cần).
  4. Tạo train và test từ dữ liệu data.
  5. Xây dựng model với train.
  6. In summary của model.
  7. Dự đoán y_pred từ test => so sánh với y_test.
  8. Đánh giá model.
  9. Trực quan hóa model.

5. Clustering (1.0 điểm)

- *Tạo tập tin: **question_5.ipynb** (toàn bộ code của câu 5 sẽ được viết trong file này)*
- Cho dữ liệu **clustering.csv**

- Yêu cầu: Đọc dữ liệu, chuẩn hóa dữ liệu (nếu cần) và sử dụng KMeans để thực hiện việc **phân cụm** khách hàng dựa trên cột **LoanAmount** (đơn vị tính : in Thousands) và cột **ApplicantIncome** trong dữ liệu được cung cấp..
- Gợi ý các bước thực hiện:
    1. Đọc dữ liệu và gán cho biến data.
    2. Xem thông tin data: head(), số dòng, số cột, summary.
    3. Tiền xử lý dữ liệu (nếu cần).
    4. Vẽ hình để xem mối liên hệ giữa privileges và learning. Cho nhận xét dựa trên biểu đồ.
    5. Xây dựng model từ dữ liệu privileges và learning.
    6. Tìm kết quả => có bao nhiêu cụm => mẫu nào thuộc cụm nào?
    7. Vẽ hình (với mỗi cụm là một màu) => xem kết quả.
    8. Đưa ra một số nhận xét dựa trên kết quả.

6. Groceries dataset (cộng 0.5 điểm nếu làm đúng)

- *Tạo tập tin: **question_6.ipynb** (toàn bộ code của câu 6 sẽ được viết trong file này)*
- Cho dữ liệu **ItemList.xlsx**
- Yêu cầu: Áp dụng thuật toán Apriori để tính toán mức độ kết hợp giữa các item.
    1. Đọc và chuẩn hóa dữ liệu
    2. Áp dụng Apriori, tìm kết quả. (Chú ý : tự lựa chọn các tham số phù hợp cho thuật toán, lưu ý với số lượng transaction càng nhiều thì các ngưỡng càng nhỏ)
    3. Trực quan hóa dữ liệu: Vẽ biểu đồ thể hiện 15 sản phẩm được mua nhiều nhất.
    4. Tìm kiếm thông tin từ kết quả: nếu mua 'sausage' thì được gợi ý mua gì?

--- ☺ Chúc các bạn làm bài tốt ☺ ---