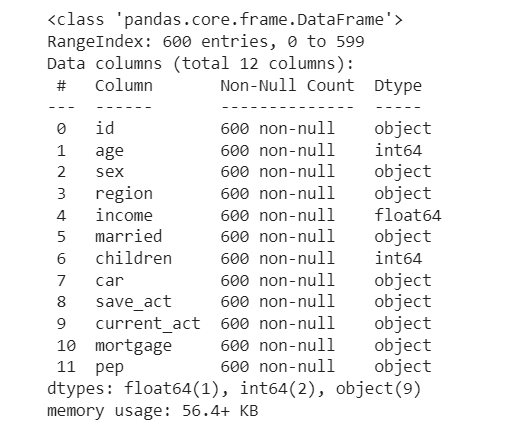
**BÁO CÁO THỰC HÀNH 7**

# Câu 1:

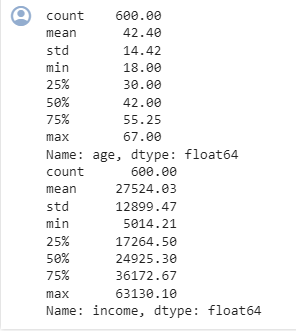
**Tiền xử lý dữ liệu và số dữ liệu của từng cột :**



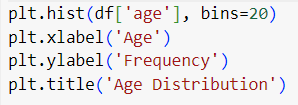
**Tính toán các giá trị thống kê mô tả cho cột tuổi và thu nhập**



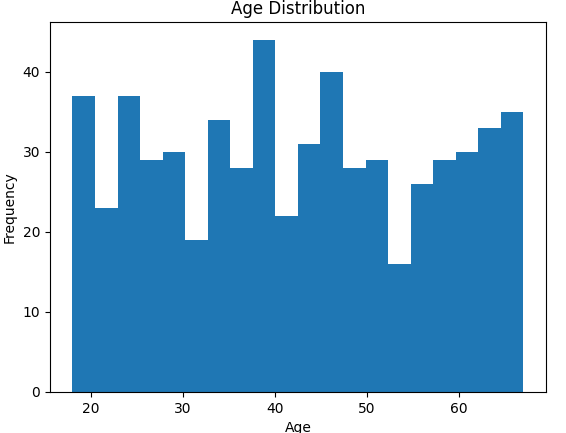
**Xuất ra kết quả**



**Vẽ biểu đồ phân bố tần số với thư viện matplotlib**

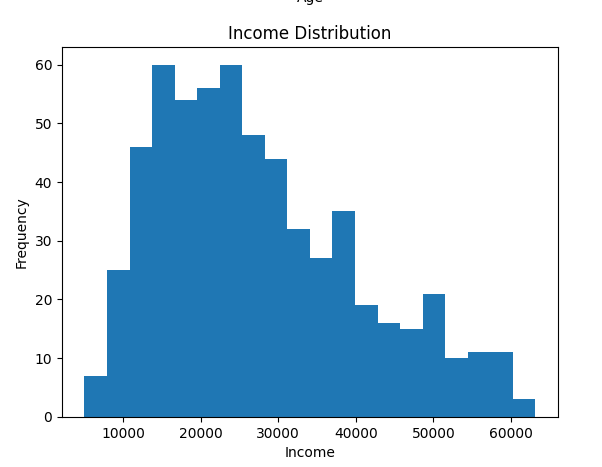
****

Biểu đồ phân bố tần số của tuổi

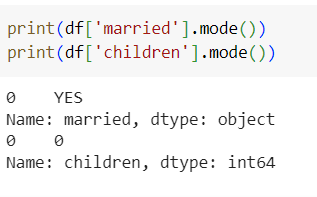


Ta có thể thấy tần số độ tuổi 39, 40 chiếm số lượng lớn rõ rệt, ngoài ra còn 19,23,32 và ngoài 60. Độ tuổi giữa 50 chiếm tấn số thấp nhất so với còn lại

Biểu đồ phân bố tần số của thu nhập



Ta thấy được số lượng thu nhập chiếm ưu thế là giữa 10000 tới hơn 20000 thuộc độ tuổi 50-60 tuổi. Người lớn tuổi đang giữ mức thu nhập trung bình tới khá so với lứa tuổi trẻ hơn như 25-35 tuổi với mức thu nhập từ 40000 đến 50000. Ngoài ra còn số ít tần số trong độ tuổi từ 10-20 có được mức thu nhập với 60000

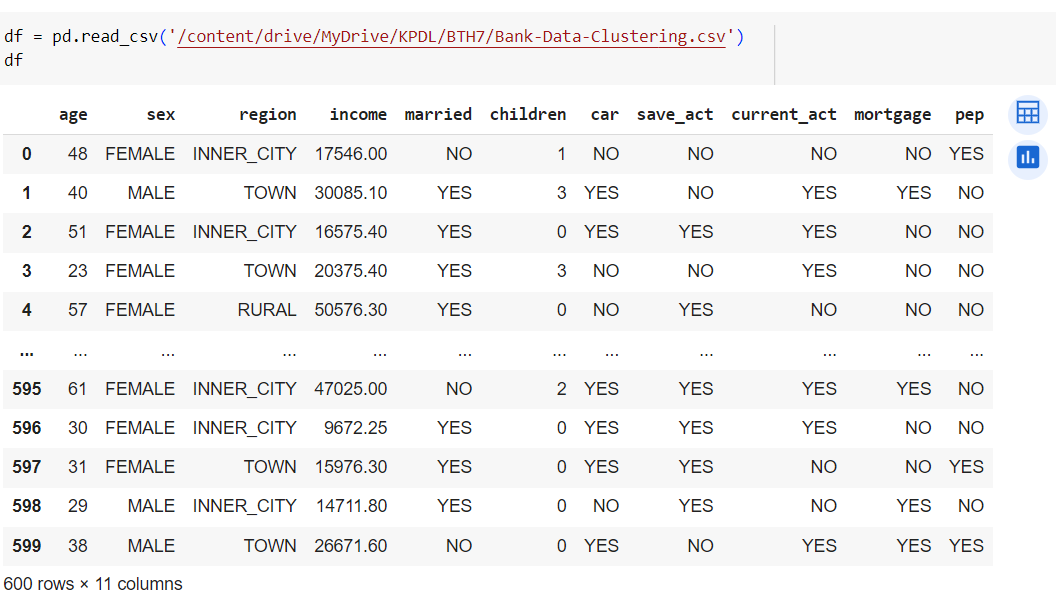
**Tính phần trăm của các cột trình trạng hôn nhân và số con**

Đa số đã kết hôn nhưng chưa có con

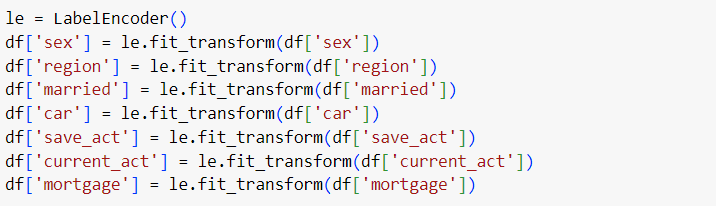
Lưu dữ liệu hiện tại vào tập tin mới có tên là Bank-Data-Clustering



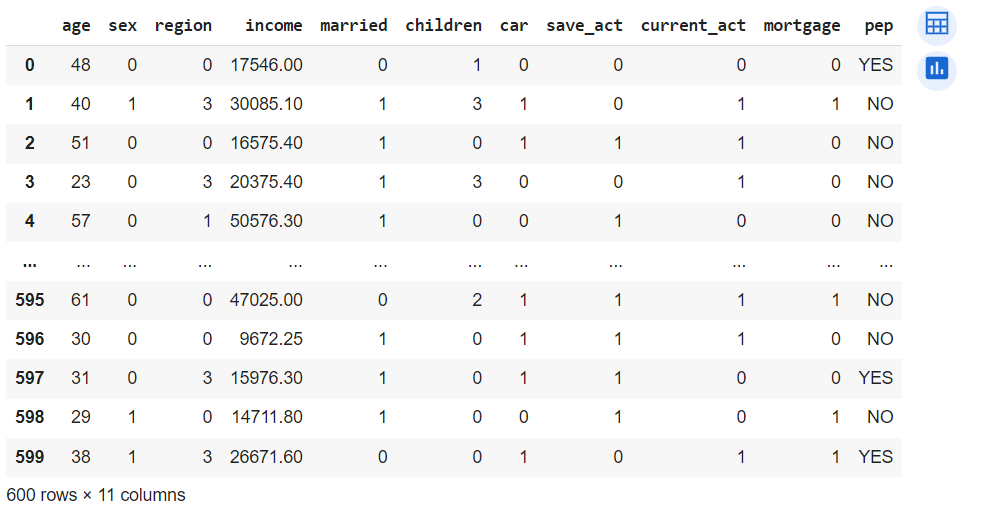
# Câu 2:



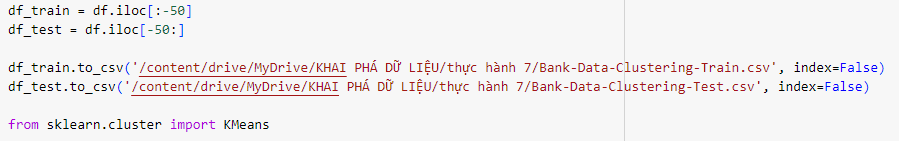
Chuyển đổi dữ liệu categorical thành dạng số nguyên và add thành một file csv mới có tên là: Bank-Data-Clustering\_std.csv



**Kết quả sau khi chuyển hóa**



Tách dữ liệu train - test theo tỷ lệ 550:50 và xuất file Train & Test cho các lần sử dụng sau



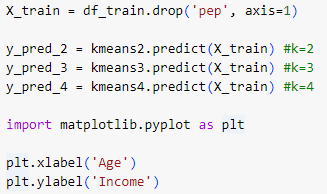
**Thực hiện gom cụm KMean cho tập tin Train với k=2, k=3, và k=4**

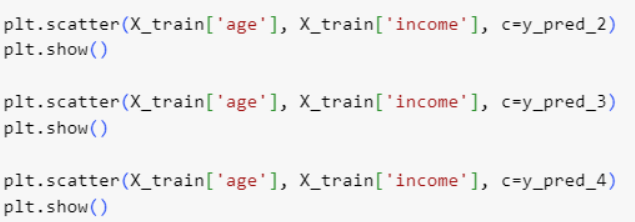
****



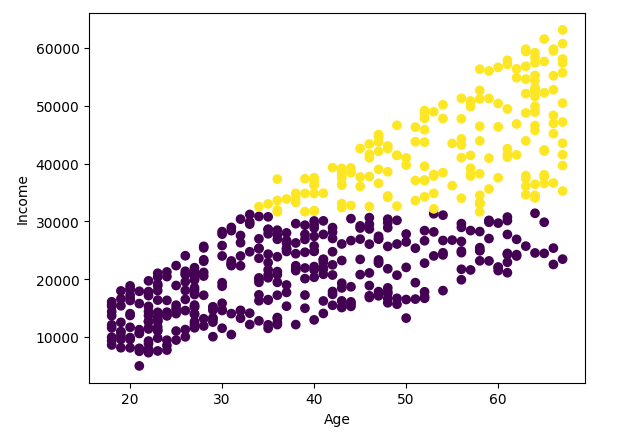
**Chuẩn hóa dữ liệu bằng KMean**

Ta nhận được 3 biến chứa nhãn từ 3 model đã train với các k khác nhau



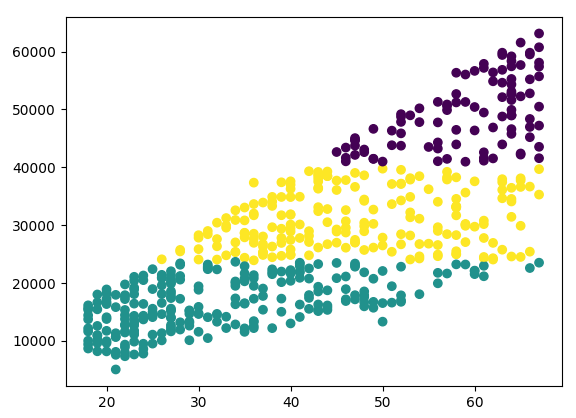


* Bảng thu nhập theo độ tuổi với k=2



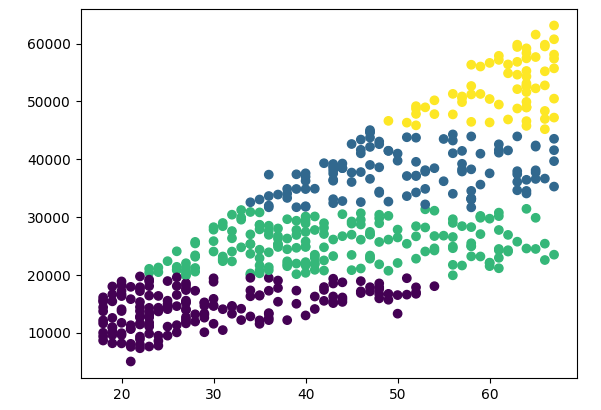
Được chia làm 2 cụm chính, sự phân bố đều. Ta thấy tuổi càng cao thì thu nhập càng cao với độ tuổi 55-66 có được mức thu nhập khoảng 400000 trở lên

* Bảng thu nhập theo độ tuổi với k=3



Được phân bố phổ biến vào thu nhập thấp đến trung bình từ 20000-40000 và rải rác ở mọi lứa tuổi. Có thể nói bình quân thu nhập thấp

* Bảng thu nhập theo độ tuổi với k=4



Được chia thành 4 cụm

Thu nhập thấp chủ yếu ở độ tuổi từ 20 đến 30

Thu nhập trung bình thấp: phân bố giảm dần từ 30 đến 60

Thu nhập trung bình: phân bố nhiều ở độ tuổi 40 - 50, thưa dần từ sau 50

Thu nhập cao: Chủ yếu từ độ tuổi 50 trở đi

=> Chia số cụm càng cao, ta càng thấy rõ sự chênh lệch về tuổi tác và thu nhập

Ở biểu đồ này ta thấy rõ, độ tuổi 20 - 30 hầu như không có được mức thu nhập cao, và ngược lại khoảng tuổi cao như 50 - 60 hầu như không có mức thu nhập thấp.

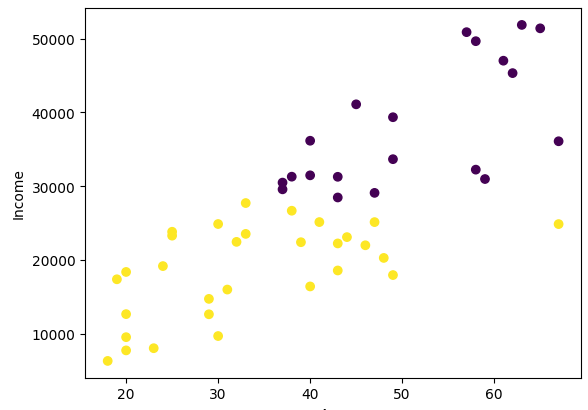
**Thực hiện gom cụm KMean cho tập tin Test với k=2, k=3, và k=4**



**Chuẩn hóa dữ liệu bằng KMean**

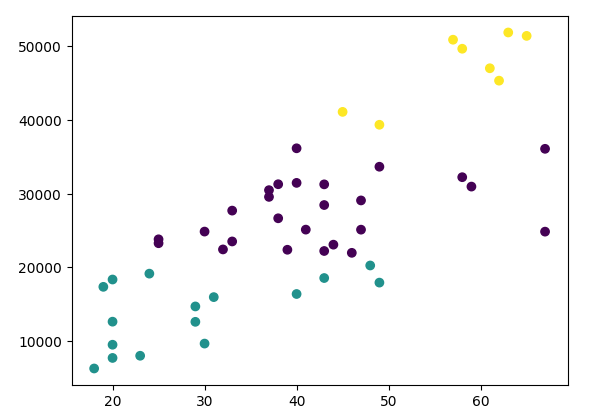


* Bảng thu nhập theo độ tuổi với k=2



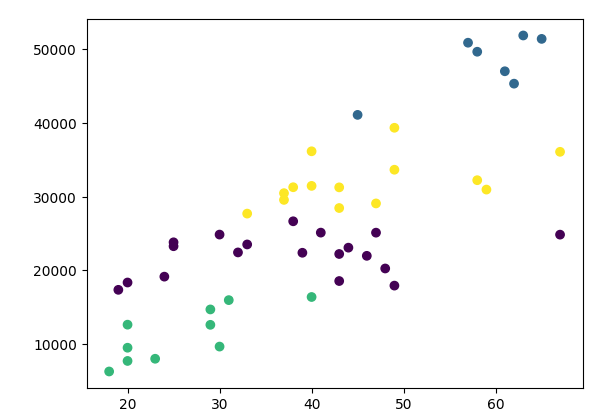
Phân bố rải rác, ta thấy ở tuổi cao thu nhập cũng cao

* Bảng thu nhập theo độ tuổi với k=3



Phân bố nhiều ở thu nhập trung bình, trong độ tuổi từ 35-45 với mức thu nhập 30000 đến 40000 và số lượng thu nhập cao trong độ tuổi 50 trở lên không đáng kể

* Bảng thu nhập theo độ tuổi với k=4



Thu nhập thấp: chủ yếu độ tuổi từ 20 đến 30

Thu nhập trung bình thấp: phân bố giảm dần từ 30 đến 50

Thu nhập trung bình: phân bố nhiều ở độ tuổi 40 - 50, thưa dần từ sau 50

Thu nhập cao: Chủ yếu từ độ tuổi 60 trở đi

=> Chia số cụm càng cao, ta càng thấy rõ sự chênh lệch về tuổi tác và thu nhập

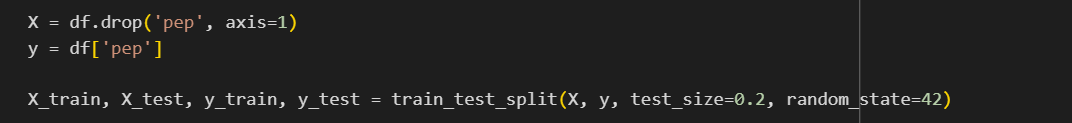
Ở biểu đồ này ta thấy rõ, độ tuổi 20 - 30 hầu như không có được mức thu nhập cao, và khoảng tuổi cao như 50 - 60 hầu như không có mức thu nhập thấp hoặc rất ít

# Câu 3:

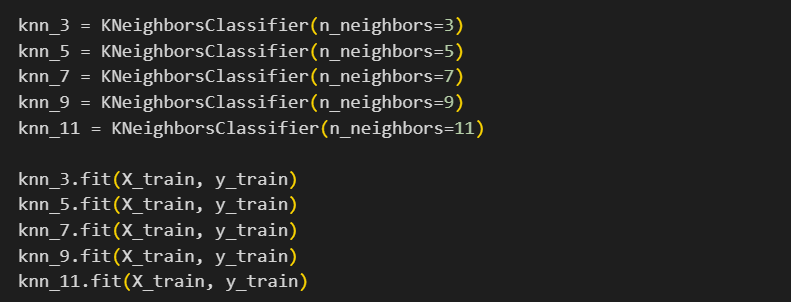
Sử dụng lại file Bank-Data-Clustering\_std:

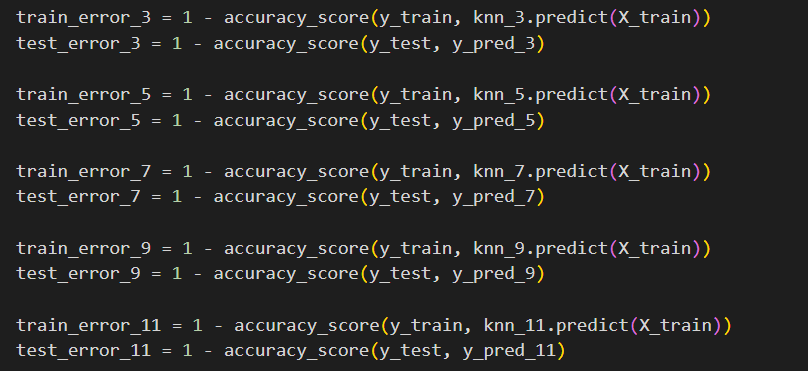


Các đặc trưng lưu vào biến X và nhãn lưu vào biến Y, sau đó chia tỉ lệ train test là 80:20

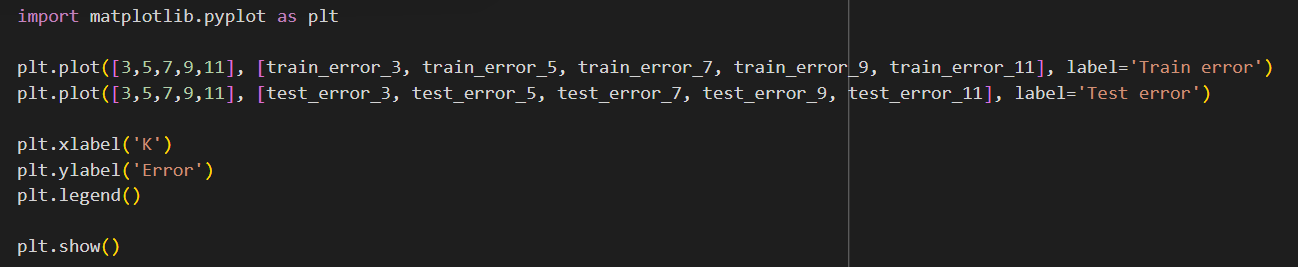


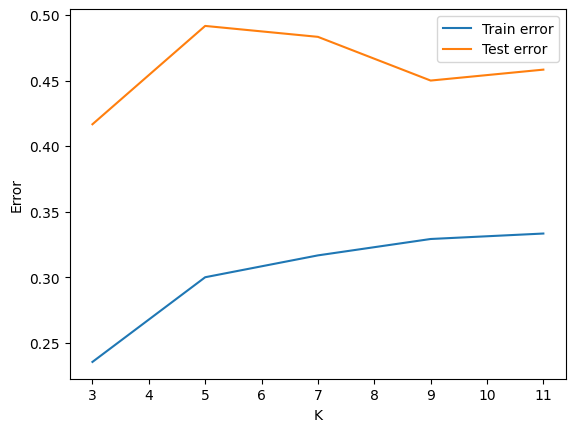
Xây dựng model KNN với n neighbors = [3,5,7,9,11] sau đó train và đánh giá mô hình:



Import hàm **accuracy\_score** tính toán sai số của model:

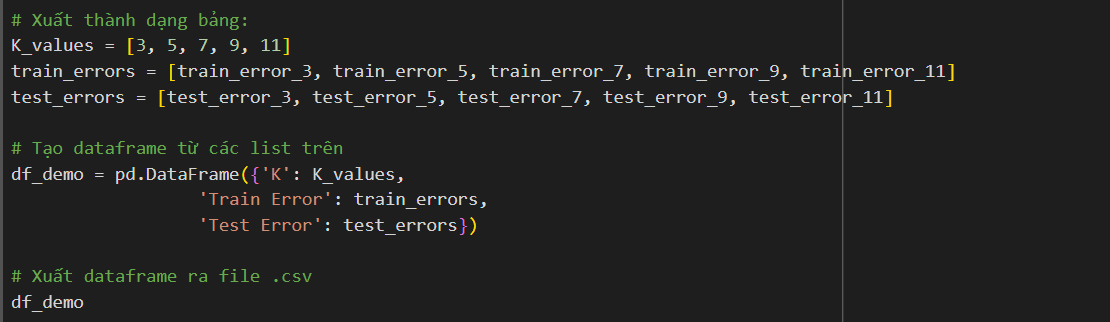
Vẽ biểu đồ đường xem độ sai số theo n\_neighbors tăng dần:

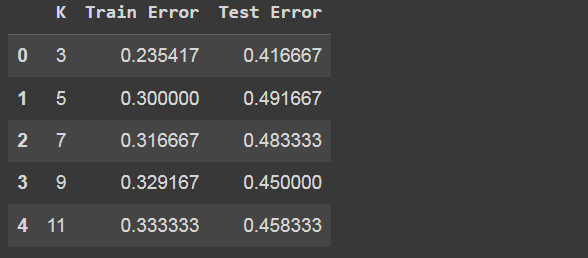




**Kết luận:** Ta thấy rằng tại k = 9 thì tập test có độ sai số nhỏ nhất, từ k = 11 trở đi sai số bắt đầu tăng dần => chọn n\_neighbors = 9 là phù hợp nhất

Ta xuất sai số thành dạng bảng để xem chính xác sai số của mô hình:





# Câu 4:

Khởi tạo 3 mô hình Decision Tree, Random Forest, Extra Tree để phân loại:

