TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

———————— * ————————

BÁO CÁO GIỮA KỲ

ĐỀ TÀI: PHÂN TÍCH VÀ ĐỊNH GIÁ ĐIỆN THOẠI

HỌC PHẦN: KỸ NGHỆ TRI THỨC

Giảng viên hướng dẫn: TS.Nguyễn Nhật Quang

Nhóm sinh viên thực hiện:

1. Nguyễn Thanh Huyền (20184122)
2. Cao Minh Hiếu (2018096)
3. Hà Thị Hạnh (20184191)
4. Nguyễn Thị Hà (20184086)
5. Ngô Bích Trang (20184204)

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

MỤC LỤC

PHẦN 1: GIỚI THIỆU

1.1. Giới thiệu bài toán

Hiện nay có rất nhiều công ty sản xuất điện thoại lớn và mạnh. Cũng như có rất nhiều công ty con phụ thuộc và phân phối sản phẩm. Để một công ty mới bước chân vào thị trường này cần rất nhiều yếu tố phụ thuộc. Tuy nhiên yếu tố tưởng chừng như đơn giản nhưng rất quan trọng, là giá một chiếc điện thoại được bán ra thị trường. Tuy nhiên, giá một sản phẩm không thể giả định vì mỗi sản phẩm có giá trị khác nhau. Để giải quyết vấn đề này, một dữ liệu mở đã thu thập rất nhiều dữ liệu về giá điện thoại của số lượng lớn các công ty khác nhau. Chính vì vậy, chúng em dự định sử dụng dữ liệu này để tìm ra khoảng giá của một chiếc điện thoại dựa trên các thông số của máy như RAM, màu sắc, ROM,... để đưa ra khoảng giá thích hợp nhất cho một chiếc điện thoại.

Khoảng giá ở đây được chia thành 4 loại tương ứng: rẻ, tầm trung, trung bình cao, cao.

1.2. Dữ liệu

1.2.1. Tổng quan về dữ liệu

Dữ liệu về thông số điện thoại được lấy trên danh sách Datasets Kaggle với tên Mobile Price Classification.

Dữ liệu gồm:

- 2 file train.csv và test.csv
- 42 cột: 37 integer, 4 decimal và 1 id

1.2.2. Mô tả trường dữ liệu

Tên trường	Ý nghĩa
battery_power	Dung lượng pin tính theo mAh
blue	Có bluetooth hay không
clock_speed	Tốc độ vi xử lý
dual_sim	Có 2 sim không
fc	Số megapixels camera trước
four_g	Có 4G hay không

three_g	Có 3G hay không
int_memory	Bộ nhớ trong tính bằng GB
m_dep	Độ dày điện thoại tính theo cm
mobile_wt	Trọng lượng điện thoại
n_cores	Số lõi xử lý
pc	Số megapixels camera chính
px_height	Độ phân giải theo chiều dài
px_width	Độ phân giải theo chiều rộng
ram	RAM tính theo MB
sc_h	Chiều dài màn hình tính theo cm
sc_w	Chiều rộng màn hình tính theo cm
talk_time	Thời gian sử dụng dài nhất trong một lần sạc pin
touch_screen	Có touch screen hay không
wifi	Có wifi hay không
price_range	Đây là output của mô hình với 4 giá trị: + 0 (low cost) + 1 (medium cost) + 2 (high cost) + 3 (very high cost)

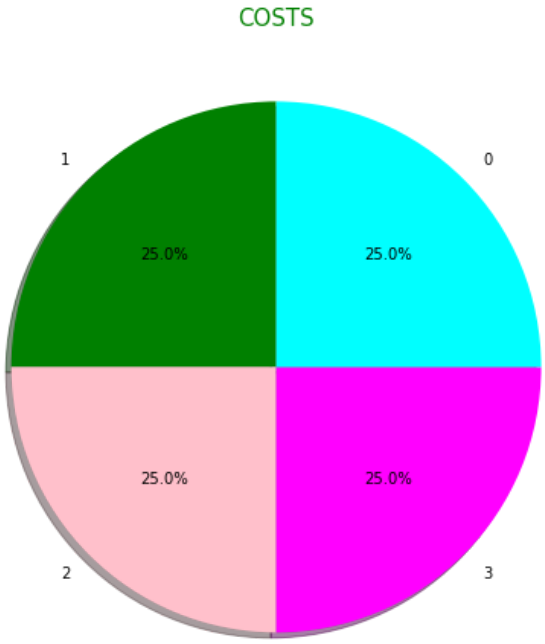
PHẦN 2: PHÂN TÍCH DỮ LIỆU

2.1. Tổng quan về dữ liệu

- Dữ liệu có 2000 bản ghi
- Không có bản ghi nào bị trùng lặp

2.2. Đánh giá nhãn

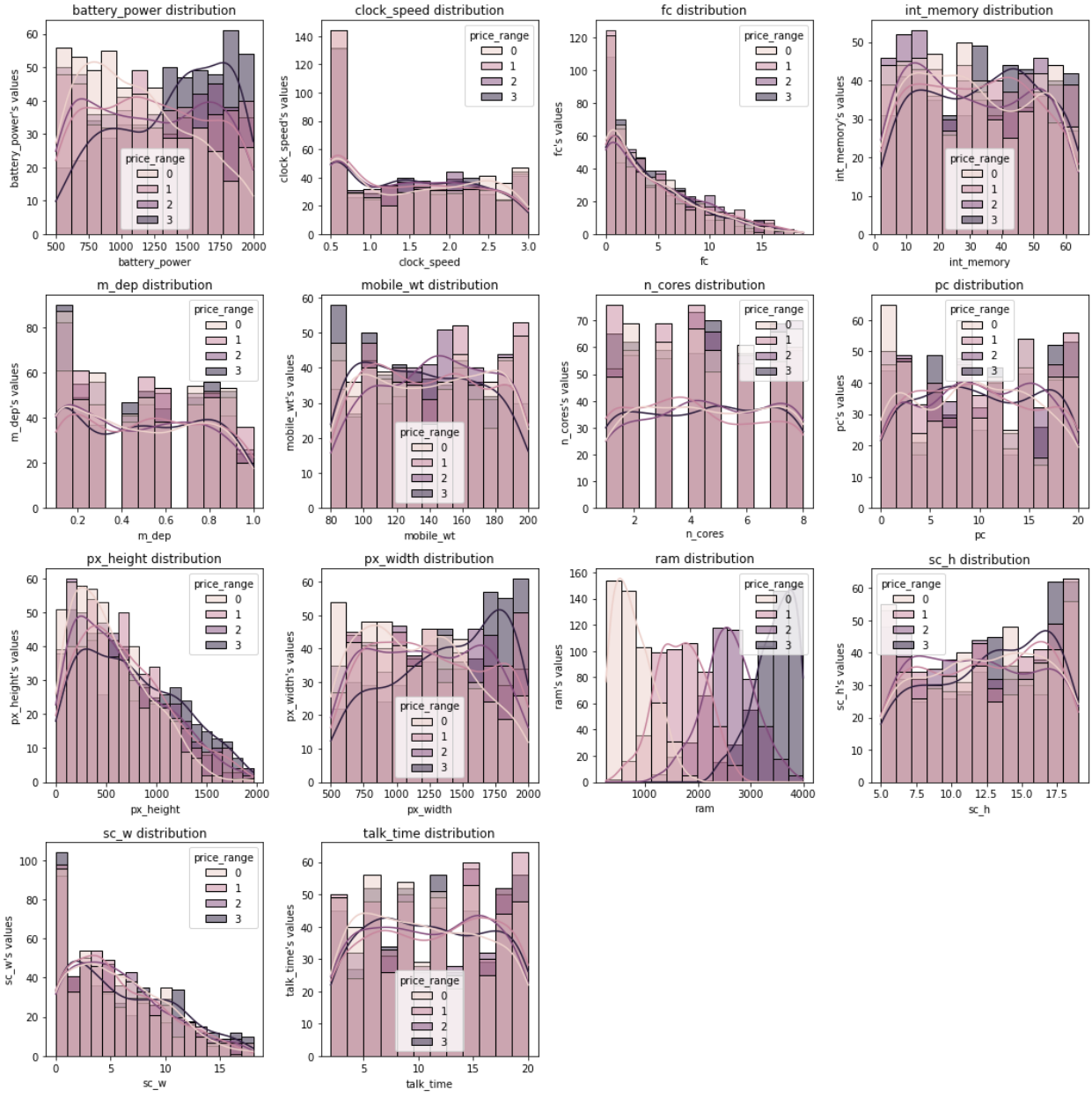
- Dữ liệu cân bằng về số lượng bản ghi từng nhãn: 500 bản ghi cho từng khoảng giá trị
- Nhận định: dữ liệu cân bằng về số lượng nhãn.



Hình 3.2: Phân bố nhãn của dữ liệu

2.3. Đánh giá tương quan các trường dữ liệu tới nhãn

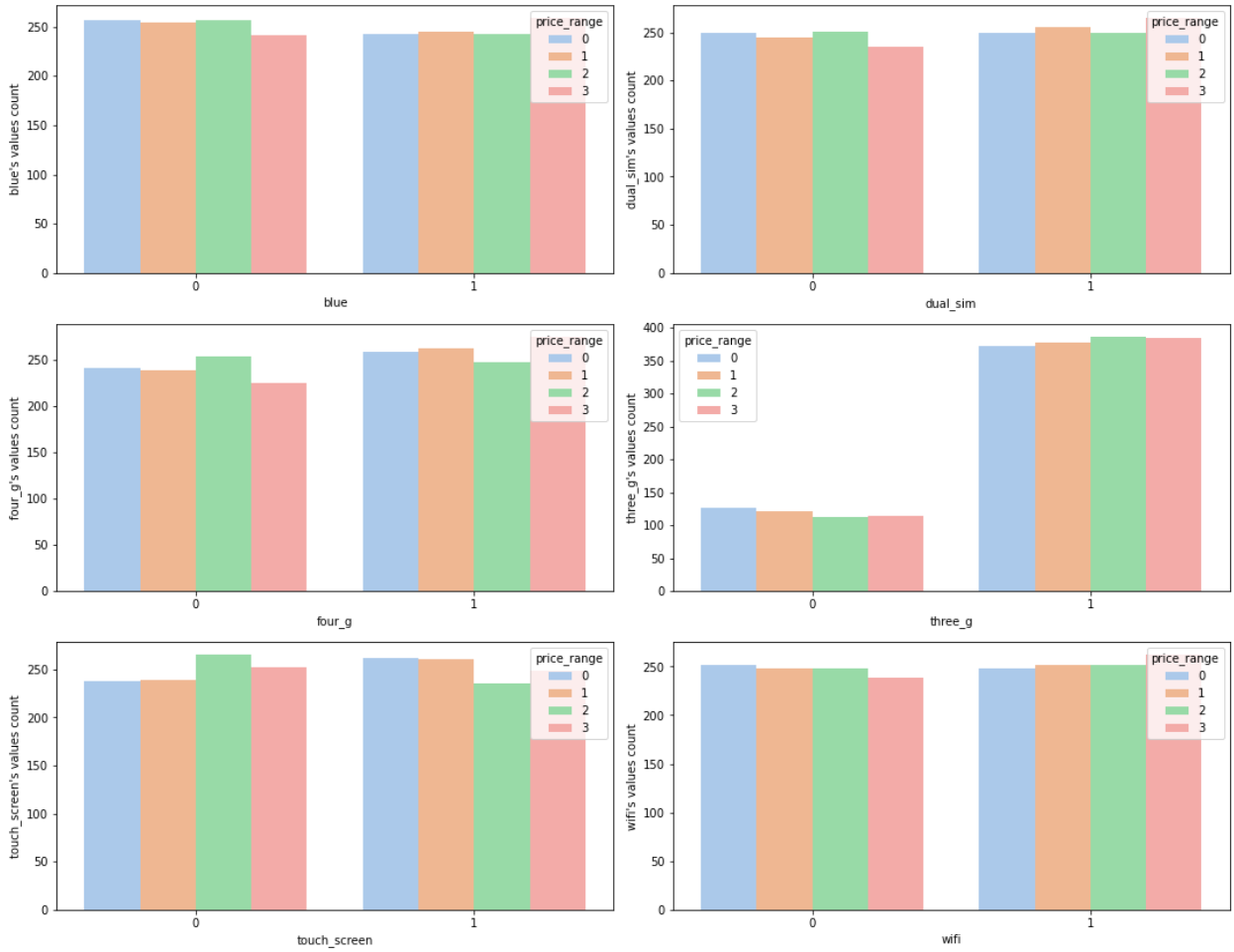
Dữ liệu numerical



Hình 2.1: Biểu đồ phân bố dữ liệu từng trường numerical data với price range

- Qua biểu đồ phân bố dữ liệu từng trường numerical data với price range có thể thấy với từng loại dữ liệu, price range phân bố khá đều, ngoại trừ trường ram - gần với phân phối chuẩn.

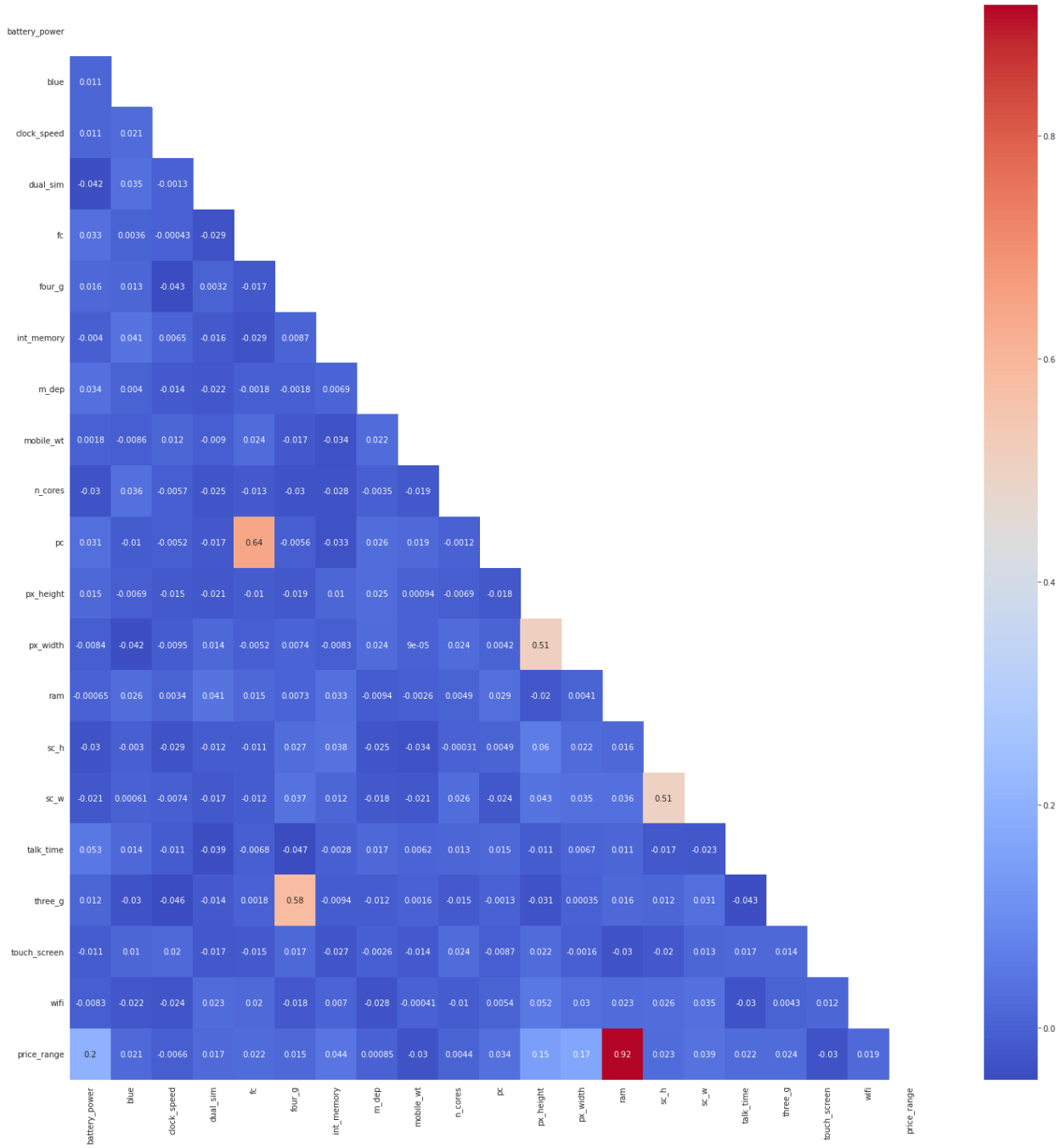
Dữ liệu categorical



Hình 2.2: Biểu đồ số lượng bản ghi từng loại feature theo price range

- Từng biểu đồ nhỏ phía trên thể hiện số lượng dữ liệu theo giá trị tương ứng 0 - không có chức năng và 1 - sở hữu chức năng so theo từng loại price range. Ví dụ: ở trường wifi, có thể thấy số lượng điện thoại có wifi và không có wifi khá tương đồng và giá của điện thoại có và không có wifi cũng khá bằng nhau. Thể hiện cho việc wifi có hay không, chưa chắc ảnh hưởng đến khoảng giá điện thoại.
- Trong những biểu đồ nhỏ trên, có thể thấy số lượng bản ghi từng loại tính năng khá tương đồng nhau, nhiên có tính năng 3G khá đặc biệt. Nhưng theo quan sát, có thể thấy khoảng giá điện thoại lại khá bằng nhau ở từng loại có 3G và không có 3G, nên rất có khả năng 3G ảnh hưởng ít đến giá điện thoại.

Độ tương quan giữa các features



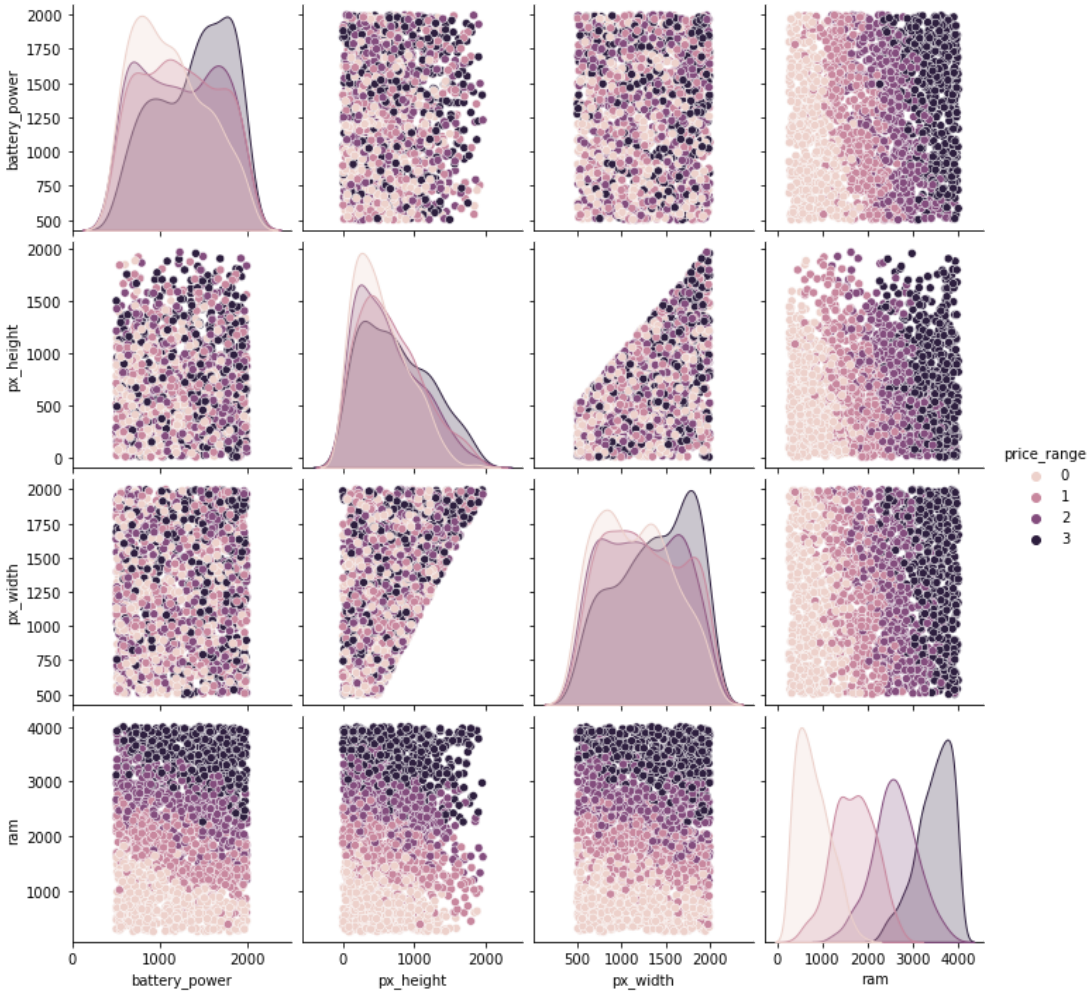
Hình 2.3: Biểu đồ độ tương quan giữa các features

- Từ biểu đồ tương quan giữa các trường dữ liệu ta có thể thấy hệ số tương quan giữa (battery power - price range: 0.2), (px height - price range: 0.15), (px width - price range: 0.17), (ram- price range: 0.92) đạt giá trị cao tức là các cặp trường dữ liệu này ảnh hưởng mạnh đến khoảng giá điện thoại. Trong đó, đặc biệt là trường **ram** ảnh hưởng rất

mạnh đến khoảng giá điện thoại (0.92), rất có thể mang tính quyết định đến kết quả bài toán.

- Ngược lại các hệ số tương quan giữa các trường dữ liệu đạt giá trị thấp (được hiển thị với màu sắc đậm hơn) biểu thị mức độ ảnh hưởng kém giữa các cặp dữ liệu này.
- Dựa vào mức độ ảnh hưởng này chúng ta có thể có được gợi ý về lựa chọn ra các trường dữ liệu có hệ số tương quan với nhãn (price range) lớn để thực hiện đào tạo cho mô hình, ngược là các trường dữ liệu có hệ số tương quan rất thấp có thể được lược bỏ khỏi dữ liệu đào tạo vì nó không ảnh hưởng nhiều đến việc đưa ra quyết định nhãn.

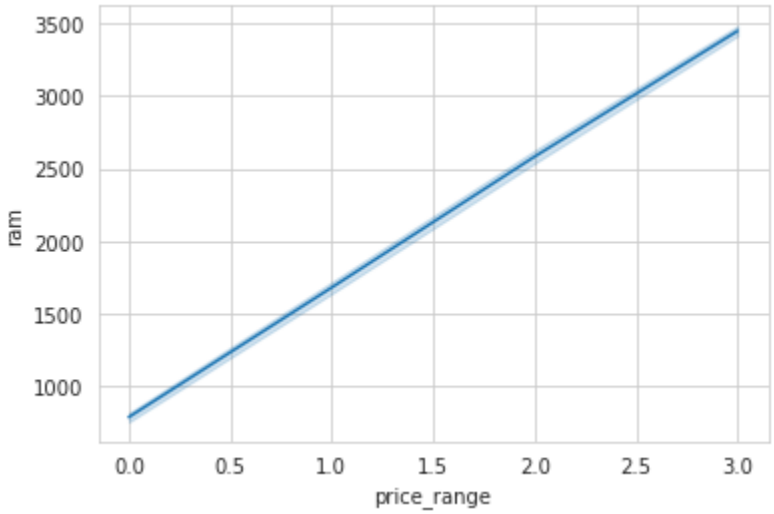
Độ tương quan giữa các features



Hình 2.4: Biểu đồ độ tương quan giữa các features quan trọng với từng loại khoảng giá

- Qua biểu đồ, có thể thấy tổ hợp các cặp tạo bởi (px_width, px_height, battery power) thì khoảng giá phân tán khá hỗn loạn và không có quy tắc. Nhưng khi (px_width, px_height, battery power) ghép cặp với ram thì khoảng giá phân biệt rất rõ ràng.

Ảnh hưởng của ram lên khoảng giá



Hình 2.5: Biểu đồ đường độ tương quan giữa ram với khoảng giá

- Qua biểu đồ có thể thấy khi ram càng cao thì giá điện thoại càng cao, tương quan rất mạnh.

2.4. Kết luận

- Qua những phân tích trên, có thể thấy những trường ảnh hưởng nhiều nhất đến kết quả bài toán là px_width, px_height, battery power và ram.
- Thuộc tính ram ảnh hưởng rất mạnh đến giá trị điện thoại, được thể hiện qua việc ram càng cao thì giá điện thoại càng cao.
- Để giải quyết bài toán, chỉ cần sử dụng dữ liệu các trường px_width, px_height, battery power và ram, những trường dữ liệu khác hoàn toàn không cần xét đến.

PHẦN 3: PHƯƠNG PHÁP BIỂU DIỄN VÀ XÂY DỰNG TRI THỨC

3.1. Giới thiệu về nhóm bài toán phân loại

Một bài toán được gọi là **bài toán phân loại** [1] nếu các nhãn của dữ liệu đầu vào được chia thành một số hữu hạn nhóm. Ví dụ: Gmail xác định xem một email có phải là spam hay không; các hãng tín dụng xác định xem một khách hàng có khả năng thanh toán nợ hay không,...

Do bài toán hướng đến việc phân tính và định giá điện thoại vào 04 mức giá (rẻ, tầm trung, trung bình cao, cao), tức mục tiêu là các giá trị rời rạc, nên có thể kết luận bài toán thuộc nhóm bài toán phân loại hay classification.

Trong bài báo cáo môn học này, thuật toán **Softmax Regression** và **Decision Tree** được sử dụng để giải quyết bài toán.

3.2. Hồi quy Softmax (Softmax Regression)

3.2.1 Khái niệm

Hồi quy Softmax [2] còn được gọi là hồi quy logistic đa thức do hàm giả thuyết mà nó sử dụng, là một thuật toán học tập có thể được sử dụng trong một số vấn đề bao gồm phân loại văn bản. Đây là một mô hình hồi quy khái quát hóa hồi quy logistic đến các vấn đề phân loại trong đó đầu ra có thể nhận nhiều hơn hai giá trị có thể.

3.2.2 Khi nào nên sử dụng hồi quy Softmax (Softmax Regression)

Hồi quy Logistic đa thức đòi hỏi nhiều thời gian hơn vì nó sử dụng một thuật toán lặp để ước tính các tham số của mô hình. Sau khi tính toán các thông số này, hồi quy Softmax có khả năng cạnh tranh về mức tiêu thụ CPU và bộ nhớ. Hồi quy Softmax được ưu tiên hơn khi chúng ta có các đặc trưng thuộc loại khác nhau (liên tục, rời rạc, biến giả, v.v.), tuy nhiên, vì nó là một mô hình hồi quy, nên nó dễ bị các vấn đề đa cộng tuyến hơn vì vậy nên tránh sử dụng khi các đặc trưng có tương quan cao.

3.2.3 Hàm Softmax (Softmax Function)

Hàm softmax tính toán xác suất xảy ra của một sự kiện. Nói một cách khác, hàm softmax sẽ tính khả năng xuất hiện của một class trong tổng số tất cả các class có thể xuất hiện. Sau đó, xác suất này sẽ được sử dụng để xác định class mục tiêu cho các input.

Hàm softmax biến vector k chiều có các giá trị thực bất kỳ thành vector k chiều có giá trị thực có tổng bằng 1. Giá trị nhập có thể dương, âm, bằng 0 hoặc lớn hơn 1, nhưng hàm softmax sẽ luôn biến chúng thành một giá trị nằm trong khoảng (0:1].

Viết ngắn gọn, ta được công thức hàm Softmax:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Trong đó:

- $\vec{z}$: Giá trị vector nhập vào cho hàm Softmax, từ z0 -> zk
- z_i : Tất cả các giá trị z đều là giá trị vector nhập cho hàm softmax. Các giá trị này có thể là bất cứ số thực nào (số dương, số âm hay số 0).
- e^{Z_i} : Hàm lũy thừa tiêu chuẩn được áp dụng cho mỗi giá trị nhập. Nó đưa ra một giá trị dương > 0. Giá trị này rất nhỏ nếu nó là giá trị âm, và rất lớn nếu giá trị dương. Tuy nhiên nó sẽ cố định trong khoảng (0, 1]
- $\sum_{j=1}^{K} e^{z_j}$: Số class trong một phân loại nhiều class , đảm bảo rằng tổng các giá trị sẽ luôn bằng 1 và nằm trong khoảng (0, 1]

3.3. Cây quyết định

3.3.1. Khái niệm

Cây quyết định (decision tree) [3] là một thuật toán học máy có giám sát. Thuật toán sử dụng mô hình cây để mô phỏng lại cách con người tư duy và suy nghĩ bằng cách đặt ra các câu hỏi để đưa ra quyết định cuối cùng.

Thuật toán biểu diễn tri thức bằng một tập luật và thứ tự sử dụng các luật. Một mô hình cây sẽ được xây dựng từ tập luật và thứ tự sử dụng luật đó, mỗi nút của cây sẽ biểu diễn một thuộc tính của tập dữ liệu, mỗi nhánh biểu diễn một luật và mỗi lá biểu diễn một kết quả dự đoán. Việc phân tách nhánh của cây ảnh hưởng lớn đến kết quả dự đoán của thuật toán, ví dụ trong thuật

toán ID3 đánh giá việc phân nhánh sử dụng các chỉ số là entropy và information gain.

3.3.2. Thuật toán CART

Trong đồ án môn học này, nhóm em sử dụng thuật toán CART (Classification and Regression Tree) để xây dựng cây quyết định.

a) Giới thiệu

Thuật toán CART (Cây phân loại và hồi quy) một thuật toán cây quyết định áp dụng cây nhị phân bằng cách sử dụng thuộc tính và ngưỡng mang lại mức tăng thông tin lớn nhất tại mỗi nút. Trong thuật toán này, chỉ số Gini được sử dụng để xác định ngưỡng của các thuộc tính và phân nhánh ở các nút. Chỉ số Gini được tính bằng công thức sau:

$$Gini = 1 - \sum_{y=1}^{C} (p_i)^2$$

Trong đó:

+ C là số lớp cần phân loại
+ $\sum_{i=1}^{C} p_i = 1$ và $p_i = \frac{n_i}{N}$, với n_i là số lượng phần tử ở lớp thứ i
+ $N = \sum_{i=1}^{N} n_i$ là tổng số lượng phần tử ở nút đó

Từ công thức trên, ta nhận thấy:

+ $Gini \geq 0$, dấu bằng xảy ra khi $\exists j$: $p_j = 1$ và $p_k = 0 \ \forall k \neq j$
+ $Gini \leq \frac{C-1}{C}$, dấu bằng xảy ra khi $p_i = \frac{1}{C} \ \forall j$

Ta thấy chỉ số Gini thấp nhất (bằng 0) khi nút chỉ chứa dữ liệu của một lớp duy nhất. chỉ số Gini là cao nhất khi dữ liệu các lớp ở trong nút đó cân bằng. Như vậy, ta sẽ mong muốn một chỉ số gini thấp tại các lớp con.

b) Cách xây dựng cây quyết định

Bước 1: Nhận thông tin tại điểm phân nhánh hiện thời ở mỗi đầu vào (ở mỗi vòng lặp);

Bước 2: Từ điểm phân nhánh ở Bước 1, xác định điểm phân nhánh "tốt nhất" mới theo chỉ số Gini;

Bước 3: Phân nhánh cây theo điểm chia "tốt nhất" ở Bước 2;

Bước 4: Tiếp tục phân nhánh cho đến khi thỏa mãn điều kiện dừng, hoặc không còn phân nhánh nào mong muốn nữa.

c) Cách đưa ra quyết định với dữ liệu mới

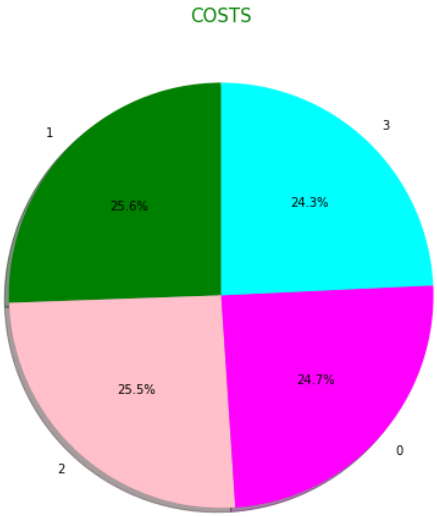
Bắt đầu từ nút gốc, dựa theo thuộc tính được xét tại nút để quyết định chuyển sang nhánh bên trái hay bên phải. Quá trình này được lặp lại cho đến khi đến một nút lá. Tại nút lá, kết quả dự đoán sẽ thuộc về lớp có xác suất lớn nhất trong các lớp.

PHẦN 4: XÂY DỰNG MÔ HÌNH ĐỂ GIẢI QUYẾT BÀI TOÁN

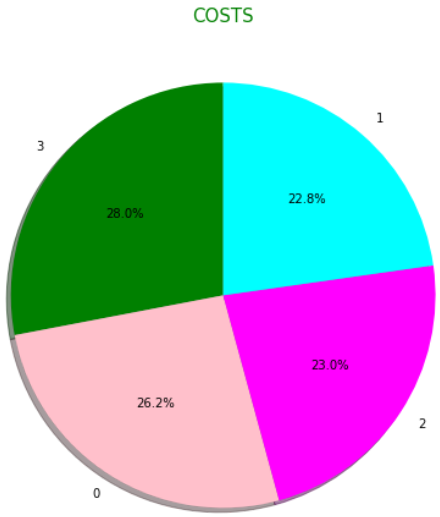
4.1. Chia dữ liệu bài toán

- Bộ dữ liệu bao gồm hai file được lưu dưới định dạng csv, gồm: train.csv và test.csv.
- Số lượng bản ghi tương ứng của các tập dữ liệu trên là:
  + Tập dữ liệu train.csv: 2000 bản ghi. (có nhãn)
  + Tập dữ liệu test.csv: 1000 bản ghi. (không có nhãn)
- Dữ liệu trong file train.csv được chia thành hai bộ: Train và Validation.
  + Tập dữ liệu train: 1600 bản ghi.
  + Tập dữ liệu validation: 400 bản ghi.
- Tập dữ liệu Train, Validation được sử dụng trong quá trình xây dựng và huấn luyện cây nhằm tinh chỉnh các siêu tham số để tìm ra kết quả tốt nhất, tập dữ liệu Test được sử dụng sau quá trình xây dựng và huấn luyện để kiểm thử.

Phân bố nhãn của tập Train:



Phân bố nhãn tập Validation:



4.2. Phương pháp đánh giá

4.2.1. Confusion Matrix

Trong đánh giá bài toán phân loại, một ma trận được xây dựng để so sánh kết quả phân loại thu được với các giá trị thực tế của quan sát đã cho để đánh giá hiệu suất của mô hình phân loại, được gọi là Confusion Matrix. Ví dụ, Confusion Matrix cho bài toán phân lớp nhị phân (Binary Classification) như sau:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Trong đó:

+ True Positive (TP) cho biết mô hình dự đoán kết quả là đúng và quan sát thực tế là đúng.
+ False Positive (FP) cho biết mô hình dự đoán một kết quả đúng, nhưng quan sát thực tế là sai.
+ False Negative (FN) cho biết mô hình dự đoán một kết quả sai, trong khi quan sát thực tế đáng ra phải là đúng.
+ True Negative (TN), cho biết mô hình dự đoán một kết quả sai, trong khi kết quả thực tế cũng sai. (tức mô hình dự đoán đúng)

4.2.2. Độ chính xác (Accuracy)

Độ chính xác là phương pháp đánh giá được sử dụng phổ biến cho bài toán phân loại. Độ chính xác là thang đo được tính trên số lượng kết quả đúng trên tổng số các trường hợp được kiểm tra.

Độ chính xác là một phương pháp đánh giá tốt trong trường hợp bài toán có số lượng nhãn giữa các lớp cân bằng nhau.

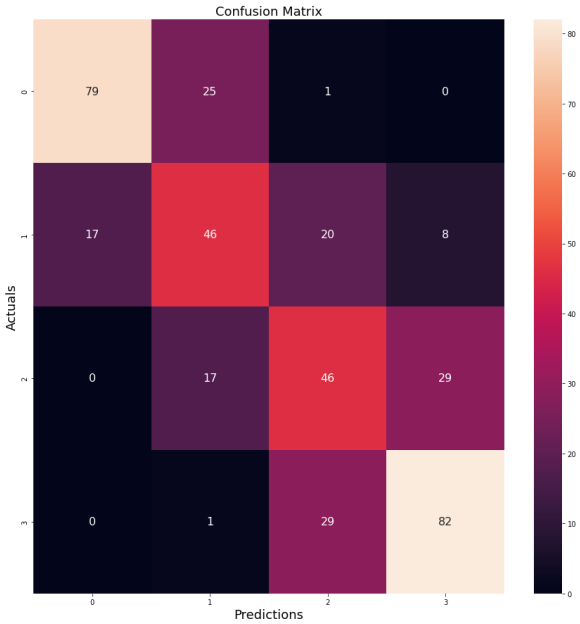
$$\text{Accuracy} = (TP+TN)/(TP+FP+FN+TN)$$

Trong bài toán Định giá điện thoại, số lượng các nhãn là bằng nhau.

4.3. Softmax Regression

Thực hiện huấn luyện dữ liệu với dữ liệu gốc ban đầu, thu được:

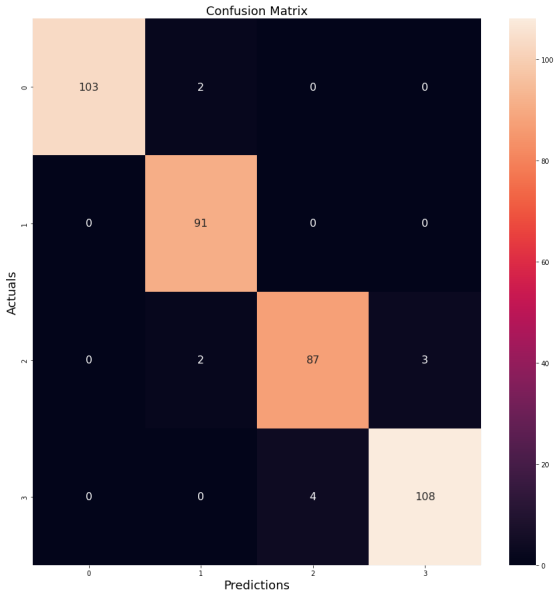
- Accuracy: 0.6325
- Confusion Matrix:



Softmax regression là thuật toán dễ bị ảnh hưởng bởi dữ liệu nhiễu. Sau khi xử lý dữ liệu, loại bỏ các yếu tố nhiễu và tìm ra các đặc trưng của bộ dữ liệu, bao gồm 4 trường có độ tương quan lớn với nhãn nhất: 'ram', 'battery_power', 'px_height', 'px_width'

Kết quả:

- Accuracy: 0.9725
- Confusion Matrix

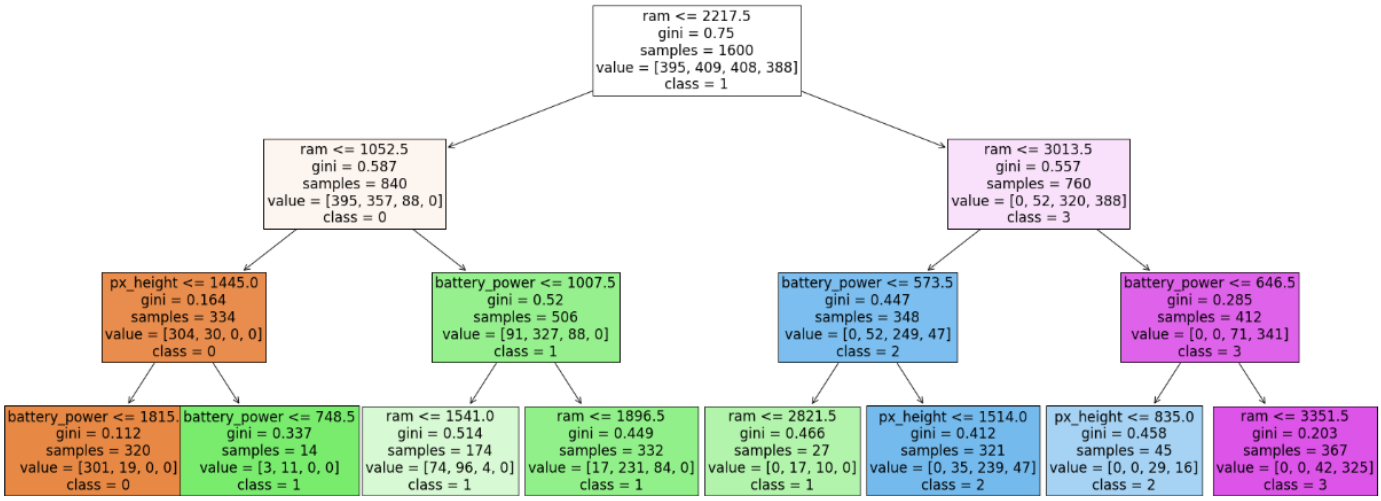


4.3. Cây quyết định (Decision Tree)

4.3.1. Thuật toán

Trong đồ án môn học này, thuật toán **CART (Classification And Regression Tree)** được sử dụng để xây dựng cây quyết định. Chi tiết về thuật toán này đã được trình bày chi tiết trong phần 2.3.

4.3.2. Xây dựng cây quyết định



Hình 4: Mô tả cây được xây dựng bằng thuật toán CART.

Lấy nút gốc làm ví dụ giải thích những thông tin có trong hình:

ram <= 2217.5
gini = 0.75
samples = 1600
value = [395, 409,408,388]
class = 1

Có 5 thông tin trong hình này gồm: ram, gini, samples, value và class

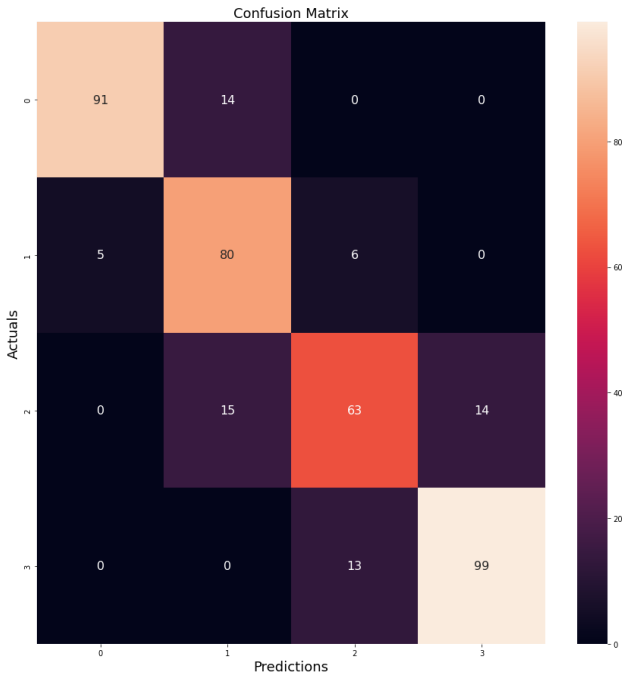
- **ram**: thông tin này cho biết feature nào được dùng để phân tách dữ liệu. Ở đây ram <= 2217.5 có nghĩa là cây phân loại đã phân đôi dữ liệu thành 2 tập thoả mãn điều kiện trên. Tại các nút không nhất thiết là ram, mà còn có thể là các features khác được
- **gini**: Giá trị gini khi sử dụng thuộc tính (ở đây là ram) để phân tách.
- **samples**: số lượng phần tử trong một nút cụ thể. Trong trường hợp của nút gốc, có 1600 giá trị vì ban đầu khi chia tập dữ liệu thành tập train và tập val. Số lượng mẫu giảm dần khi đi xuống vì tại các nút trước đó dữ

liệu đã được phân loại. Càng xuống sâu cây phân loại càng chính xác (lý thuyết).

- **value**: số lượng phần tử được chia tương ứng với các lớp phân loại. Ví dụ ở nút gốc là [395, 409, 408, 388] có nghĩa lớp 0 có 395 bản ghi, lớp 1 có 409 bản ghi, lớp 2 có 408 bản ghi, lớp 3 có 388 bản ghi.
- **class**: lớp giả định cho dữ liệu tại nút đó. Ví dụ nút gốc class=1 có nghĩa 1600 bản ghi đang được mô hình phân loại là lớp 1 - giá tầm trung.

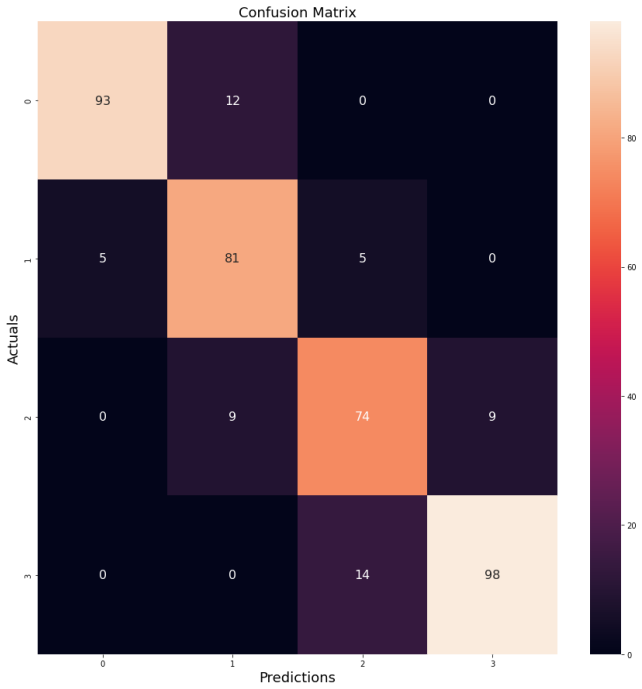
Trước khi xử lý dữ liệu, ta có kết quả:

- Accuracy: 0.8325
- Confusion matrix:



Sau khi chọn những features quan trọng và huấn luyện lại, thu được kết quả:

- Accuracy: 0.865
- Confusion matrix:



4.4. Kết luận

- Qua thực nghiệm có thể tái khẳng định, 4 trường dữ liệu được chọn là px_width, px_height, battery power và ram là những tính năng quan trọng, ảnh hưởng trực tiếp đến kết quả mô hình.
- Khi loại bỏ những tính năng không cần thiết (loại bỏ trị thức dư thừa) thì mô hình học tốt hơn - ở cả 2 mô hình, sau khi loại bỏ các tính năng không cần thiết thì thu được accuracy cao hơn.
- Mô hình tốt nhất cho bài toán này là **softmax regression** với accuracy bằng 0.9725.

PHẦN 5: DEMO ỨNG DỤNG

5.1. Giới thiệu về thư viện Streamlit

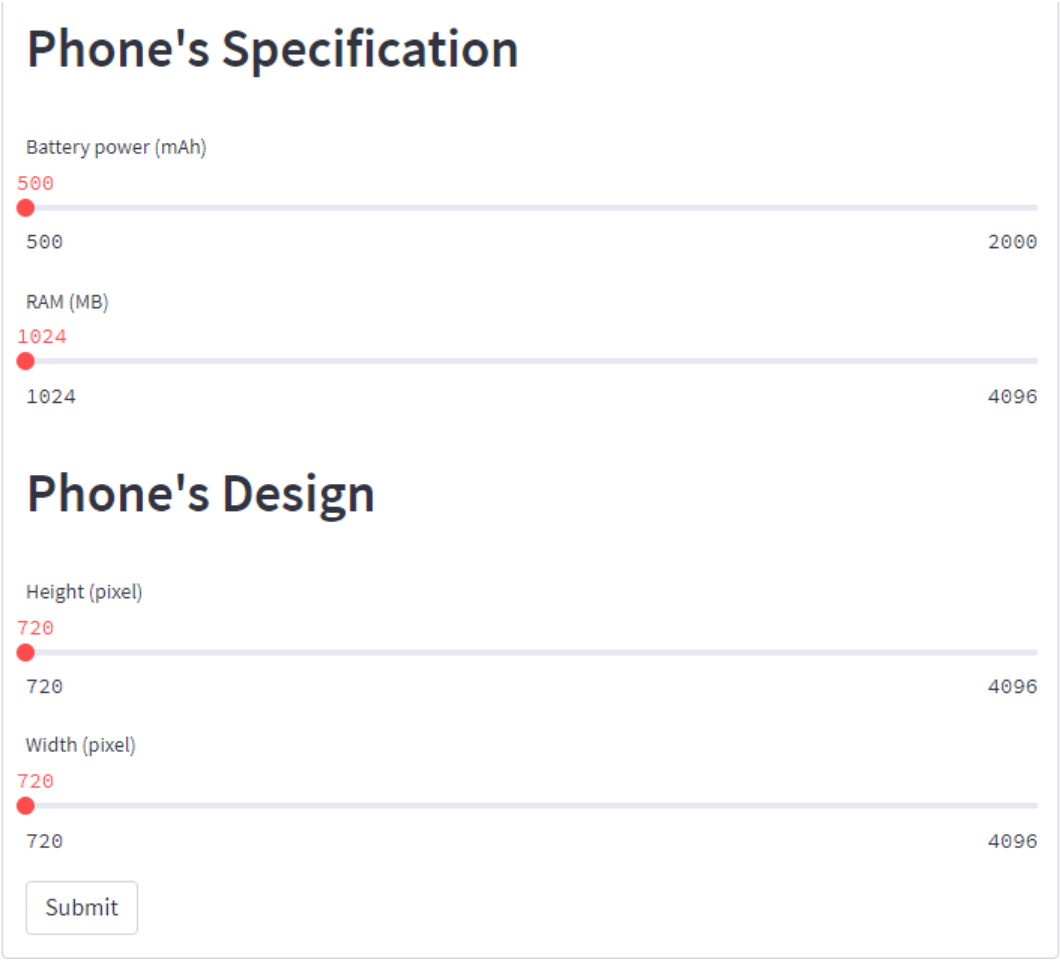
Streamlit là một thư viện mã nguồn mở giúp hỗ trợ xây dựng nhanh sản phẩm Demo các mô hình học máy và học sâu. Thư viện Streamlit xây dựng dựa trên ngôn ngữ Python nên dễ dàng tiếp cận với những người phát triển trong lĩnh vực Học máy và Khoa học dữ liệu mà không cần có nhiều kiến thức về lập trình web. Streamlit tương tự với Jupyter notebook, tuy nhiên ứng dụng sẽ chỉ hiển thị giao diện và kết quả.

Trang chủ của thư viện Streamlit: https://streamlit.io/

5.2. Demo thuật toán Decision Tree và Softmax Regression

Đầu vào của ứng dụng là thông số kỹ thuật của điện thoại, bao gồm 4 đặc trưng đã được chọn ra ở sau phần xử lý dữ liệu. Từ đây chúng ta có thể thử nghiệm các thông số kỹ thuật của chiếc điện thoại để đưa ra chi phí hợp lý.

Dưới đây là kết quả dự đoán được trực quan bằng ứng dụng.



PHẦN 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Những thành tựu đạt được

Thông qua đồ án môn học này, những kết quả thu được như sau:

- Khảo sát, đánh giá và phân tích dữ liệu các thông số kỹ thuật của một chiếc điện thoại để định giá xem chi phí cho một chiếc điện thoại.
- Tìm hiểu về biểu diễn tri thức, xây dựng tri thức và sử dụng tri thức với một dữ liệu mới trên các thuật toán phân loại.
- Tìm hiểu Thuật toán Softmax Regression, một phương pháp được mở rộng để khắc phục hạn chế của các phương pháp phân loại nhị phân.
- Xây dựng và phát triển mô hình cây quyết định theo thuật toán CART. Phân tích các ưu và nhược điểm của cây quyết định.

6.2. Thảo luận và những khó khăn trong quá trình phát triển

- **Vấn đề 1**: về dữ liệu của bài toán.
  Dữ liệu của bài toán là một trong những vấn đề khó khăn. Dữ liệu ban đầu có 21 trường, gồm rất nhiều trường thông tin và phải thông qua nhiều quá trình trích chọn đặc trưng mới thu được các đặc trưng có ảnh hưởng tốt nhất.
- **Vấn đề 2**: về khả năng diễn giải của cây.
  Đối với cây quyết định, khả năng diễn giải đơn giản thông qua mô hình cây được xây dựng từ dữ liệu. Nhìn vào cây đã được xây dựng, ta có thể hiểu được về việc đưa ra quyết định ngay lập tức.

6.3. Hướng phát triển tiếp theo

Kết quả thu được khi sử dụng cây quyết định trong đồ án môn học này đạt đến một độ chính xác có thể chấp nhận được (80%), đồng thời chỉ ra được ảnh hưởng vô cùng lớn của việc xử lý dữ liệu đến thuật toán Softmax Regression để đạt đến độ chính xác 97%. Trong tương lai, nhóm sẽ tiếp tục tìm hiểu và sử dụng những phương pháp và các kỹ thuật xử lý khác nhằm đạt kết quả cao hơn và có nhiều ý nghĩa trong đời sống hơn.

TÀI LIỆU THAM KHẢO

Dữ liệu:

Dữ liệu về thông số điện thoại được lấy trên danh sách Datasets Kaggle với tên Mobile Price Classification.

Package / Source code sử dụng:

- Xây dựng cây quyết định: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- Xây dựng ứng dụng DEMO: https://streamlit.io/

Tài liệu lý thuyết:

[1] Các bài toán phân loại (Classification)

- https://machinelearningcoban.com/2017/02/11/binaryclassifiers/

[2] Softmax Regression

- https://vn.got-it.ai/blog/softmax-function-la-gi-tong-quan-ve-softmax-function
- http://deeplearning.stanford.edu/tutorial/supervised/SoftmaxRegression/
- https://machinelearningcoban.com/2017/02/17/softmax/

[3] Decision Tree

- https://en.wikipedia.org/wiki/Decision_tree
- https://scikit-learn.org/stable/modules/tree.html
- https://machinelearningcoban.com/2018/01/14/id3/
- The Basics of Decision Trees
- Classification in Decision Tree — A Step by Step CART (Classification And Regression Tree)