PTML REPORT

Alexandre Lemonnier - Sarah Gutierez
Victor Simonin - Alexandre Poignant

Promotion 2023

Contents

1 Supervised learning

In this first part, we had to work on a dataset and perform a supervised learning on it. We decided to work on mobile price classification, and try to find some relation between features of a mobile phone (RAM, number of cores, internal memory...) and its selling price. We found a dataset that instead of giving the actual price of each phone, give a price range indicating how high the price is.

1.1 Dataset overview

1.1.1 Features description

The dataset we used for training has 2000 entries and 21 features which are:

Column name	Type	Informations
battery_power	int64	Total energy a battery can store in one time measured in mAh
blue	int64	Has bluetooth or not
clock_speed	float64	Speed at which microprocessor executes instructions
dual_sim	int64	Has dual sim support or not
fc	int64	Front Camera mega pixels
four_g	int64	Has 4G or not
int_memory	int64	Internal Memory in Gigabytes
m_dep	float64	Mobile Depth in cm
mobile_wt	int64	Weight of mobile phone
n_cores	int64	Number of cores of processor
pc	int64	Primary Camera mega pixels
px_height	int64	Pixel Resolution Height
px_width	int64	Pixel Resolution Width
ram	int64	Random Access Memory in Megabytes
sc_h	int64	Screen Height of mobile in cm
sc_w	int64	Screen Width of mobile in cm
talk_time	int64	Longest time that a single battery charge will last when you are

three_g	int64	Has 3G or not
touch_screen	int64	Has touch screen or not
wifi	int64	Has wifi or not
price_range	int64	This is the target variable with value of 0 (low cost), 1 (medium cost), 2 (high cost) and 3 (very high cost).

1.1.2 Analysis

First of all, we checked that there were no missing and duplicated data. We check that there were no outliers too.
Secondly, we did some analysis on each feature to find out how there were represented in the dataset and how there are affected by price. We used the **seaborn** library to make some plot. What we find out is that the RAM, the phones' resolution and the battery power features affected the most the price range of a phone:

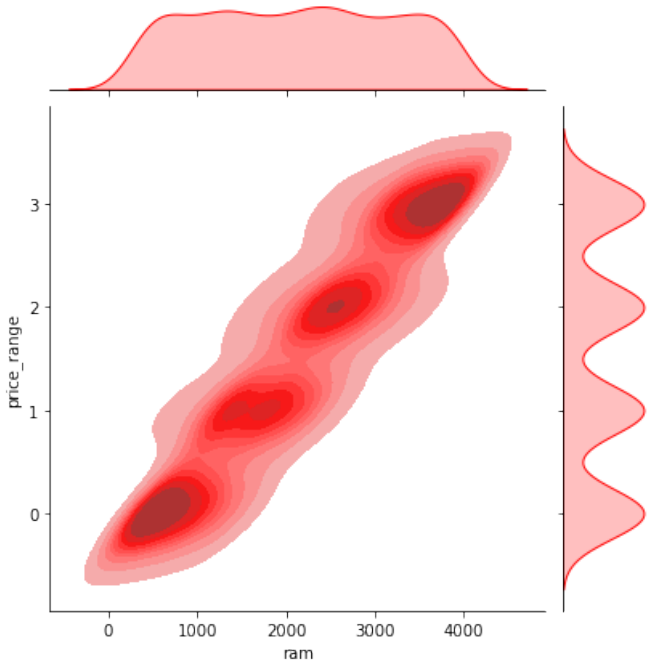

Figure 1.1: Joint plot : RAM repartition according to price range

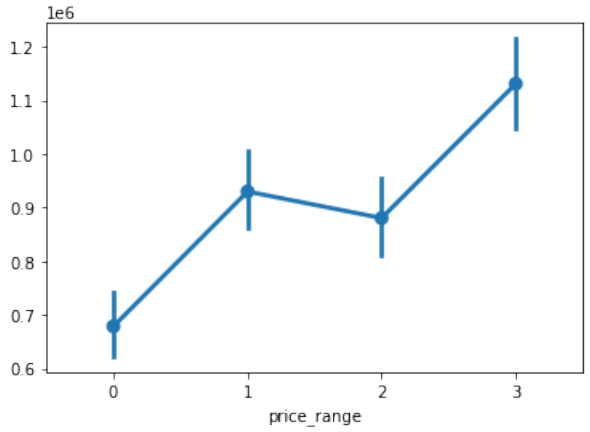

Figure 1.2: Point plot : Resolution confidence interval (95%) according to price range

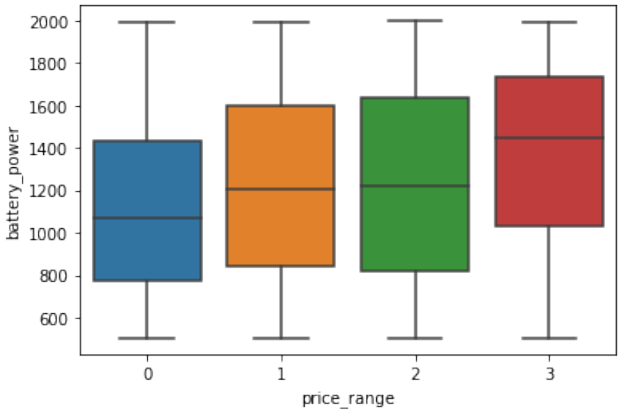

Figure 1.3: Box plot : Battery power according to price range

As we can see on these graphics, higher the RAM, the resolution or the battery power of the phone is, the higher is the price too. We found again, these correlations in the correlation matrix of our dataset, represented as a heatmap here:

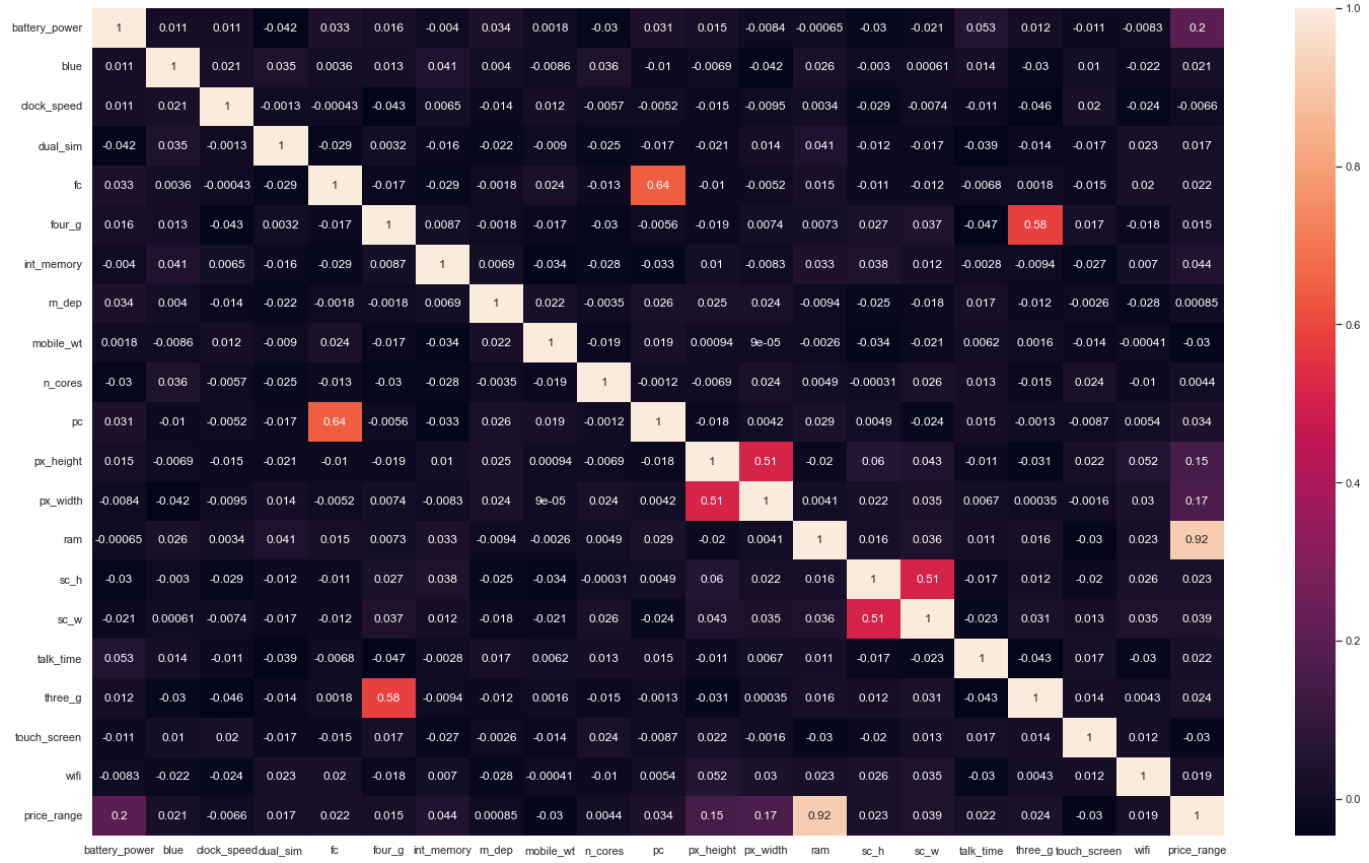

Figure 1.4: Heatmap of the correlation matrix of the training dataset

According to the correlation matrix, the correlation value between the battery power and the price is **0.2**, between resolution (px_height and px_width columns) and price is about **0.16** and between RAM and price is **0.92**, meaning that we can almost deduce the price range of the phones by the RAM it has.

1.1.3 Outliers management

To check if we got some outliers in our dataset, we decided to use the z-scores of the features of our dataset. We used the **scipy** library to use its scoring method. The z-score is a statistical method used to find how much value of the standard deviation is far a feature value from its mean.

$$z = \frac{X - \mu}{\sigma}$$

We define an outlier of our dataset as a value that is more than 3 standard deviations from the mean. This result by no outlier found.

1.2 Machine Learning

For the machine learning part, we were free to implement a regression or a classification. The goal with our dataset, was to classify all the phones with their characteristics in a range of price with value of 0 (low cost), 1 (medium cost), 2 (high cost) and 3 (very high cost). So we have a classification problem.

We decided to test several algorithms with the **sklearn** and **xgboost** library. First, we split our data to obtain a train dataset and a test dataset, then we iterate on the classifier to fit them and test them. The scoring is here a R2 score, which is evaluated by cross validation.

The results are the following :

Regression Algorithms tested	R2_score obtained
Support Vector Classifier	0.960
Perceptron	0.193
LinearRegression	0.916
Lasso	0.916
Ridge	0.916
Standard Gradient Descent	0.421
RandomForest	0.908
DecisionTree	0.864
KnnClassifier	0.943
XGBoost	0.936

We can analyse here that we have a few algorithms with great score, especially the Support Vector Classifier with 96% of success. The Perceptron and the Standard Gradient Descent seems to have more difficulty in this case.

The last part was to verify our results with a classification report containing the precision, recall, F1 score for each class and the macro average (averaging the unweighted mean per label), the weighted average (averaging the support-weighted mean per label). We also included a confusion matrix to check our results. This part has been done for the KNN Classifier and the Support Vector Classifier.

2 Unsupervised learning

2.1 Dataset

2.1.1 Analysis

We chose to work on a dataset containing information about 167 countries. The dataset has 10 columns which are:

Column	Type	Information
country	object	Name of the country
child_mort	float64	Death of children under 5 years of age per 1000 live births
exports	float64	Exports of goods and services per inhabitant. Given as percentage of the GDP per inhabitant
health	float64	Total health spending per inhabitant. Given as percentage of GDP per inhabitant
imports	float64	Imports of goods and services per inhabitant. Given as percentage of the GDP per inhabitant
Income	int64	Net income per person.
Inflation	float64	The measurement of the annual growth rate of the Total GDP
life_expec	float64	The average number of years a new born child would live if the current mortality patterns are to remain the same
total_fer	float64	The number of children that would be born to each woman if the current age-fertility rates remain the same
gdpp	int64	The GDP per inhabitant. Calculated as the Total GDP divided by the total population

After loading the data we checked that there were no missing and duplicated data. And we decided to visualize it with **seaborn** library.

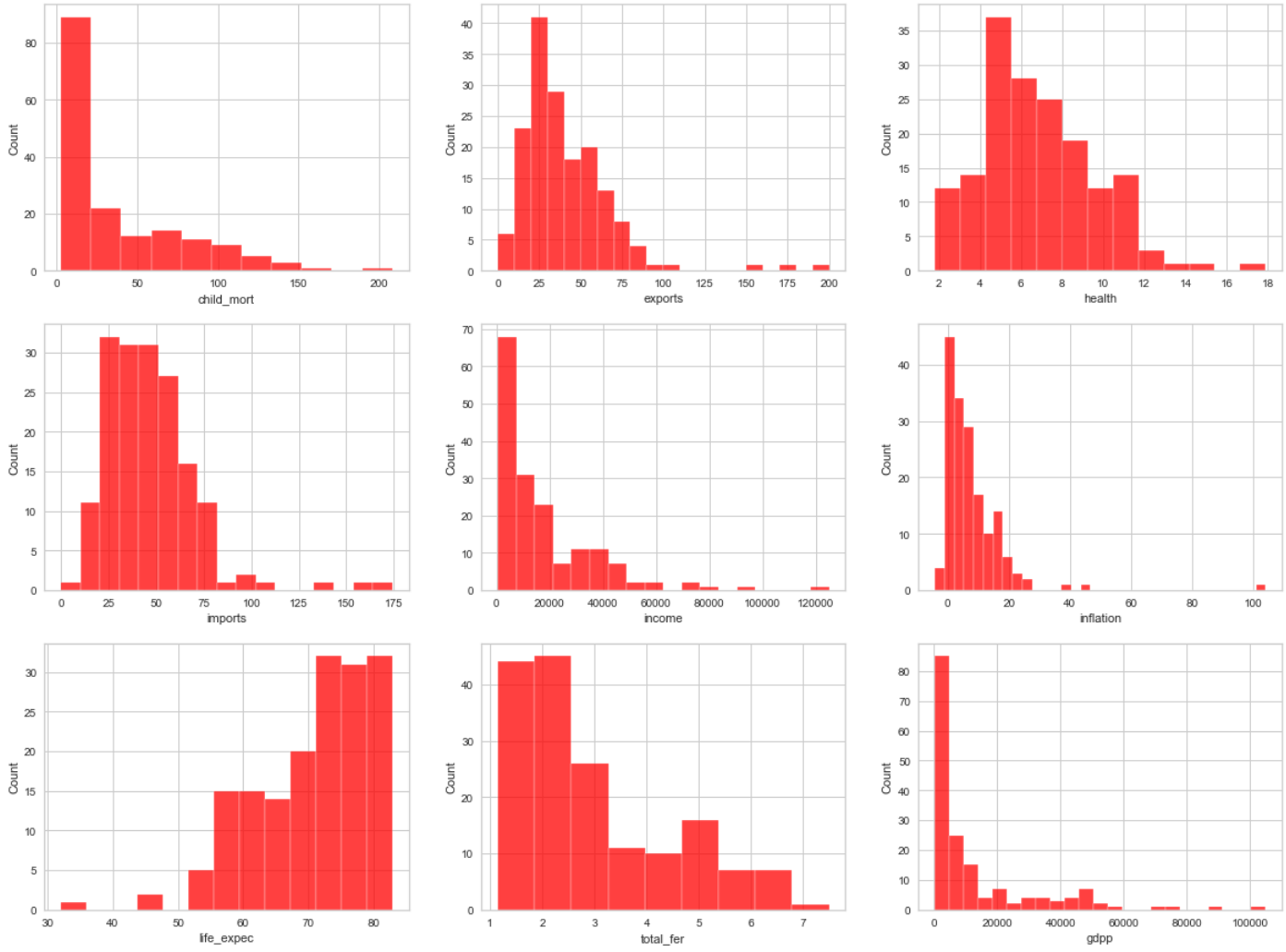

Figure 2.1: Histograms of each feature

In these graphs we can see that there are a few outliers. It is interesting to keep them because they with bring a lot of information for the extreme cases.

2.1.2 The Goal

Our objectives in this unsupervised learning is to decide whether or not a country should receive a Funding for Development Aid.

2.1.3 Evaluation and Correlation of the dataset

To see which columns are linked to each other we decided to compute the correlation matrix. With it we were able to know that some of the features are highly

correlated and that we might need to remove them to perform our unsupervised machine learning. Here are the features that are highly correlated:

- life_expect with child mortality
- total_fertility with child mortality
- income with gdpp
- child_mort with income and gdpp

2.2 Scaling

To make sure every column is scaled properly between each others, we decided to use the MinMacScaler from **sklearn**. We did not use StandardScaler because our data has outliers and the result would not have been balanced.

2.3 Clusters

2.3.1 PCA

As we have used MinMax, and so put our data between 0 and 1, we can now use PCA from the **sklearn** library again. We decide to keep all component witch represent 5% to 100% of the explained variance. It seems to be a good compromise as half of the dimension has gone for not so much information.

2.4 Machine Learning

For this machine learning part, we decided to use the K-Means algorithm from the **sklearn** library. The objective was to apply a clustering technique on our data to classify them and to know which one of them should receive a Funding for Development Aid.

We applied the algorithm on two datasets, the first one which was the result of the PCA, and the second one which was the result of the MinMax Scaling. The most optimized number of cluster is 3, we used the elbow method and the silouhette method from the **sklearn** library to find it.

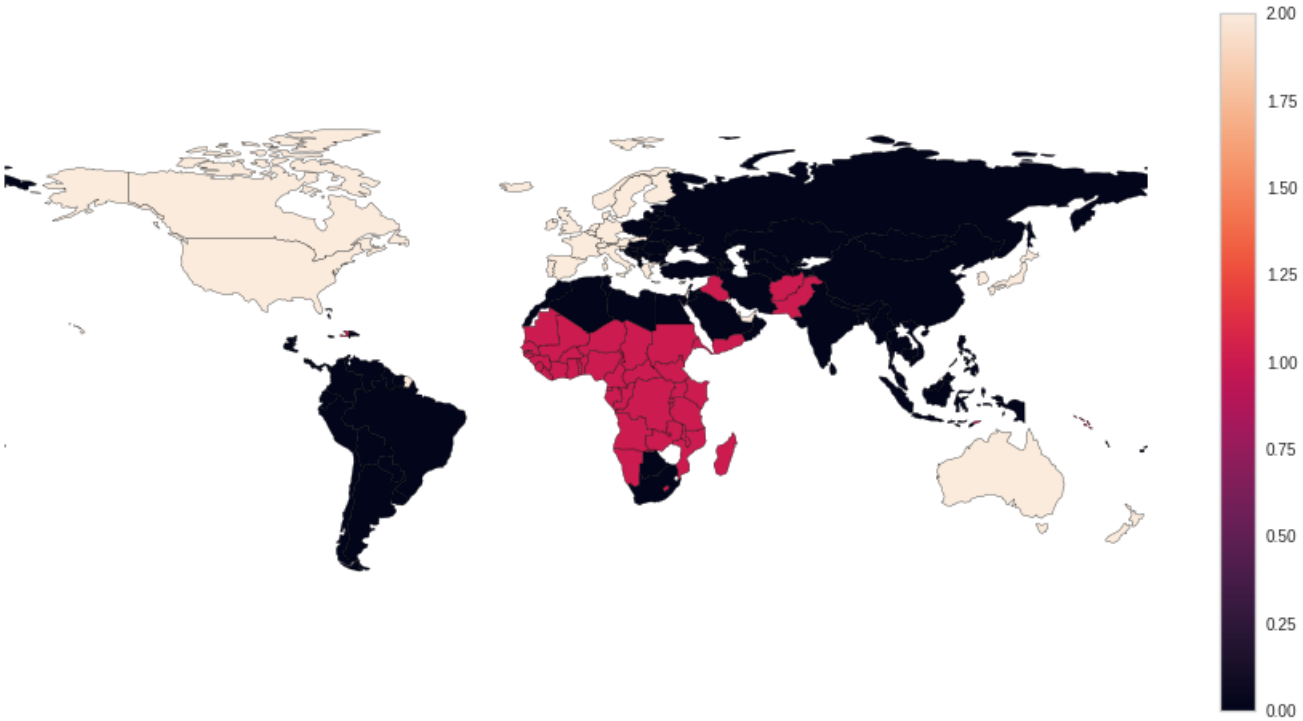

Figure 2.2: Map of the clusters (with **geoplot** library)

Then we obtain 3 clusters for our countries, we can see they tend to be overlapping between clusters. Cluster 2 is more spread out and clusters 0 and 1 tend to overlap.

- Countries in Cluster 2 (characterised by showing really strong or positive values such as good economic development, high life expectancy, low child mortality) are located in North America, Europe, Oceania and a couple in Asia.
- Countries in Cluster 1 (characterised by having the most negative values: high child mortality, lowest economic development) are located across Africa and Asia.
- Countries in Cluster 0 (characterised by showing average values for all features when compared with other clusters) are located across South America, parts of Africa, Europe and Asia. Blank spaces (like Mexico) are of countries with no available data.