Rapport UE Introduction à l'Apprentissage Automatique

Riazi Ibrahim Licence Informatique Luminy

Ait Kettout Younes Licence informatique Luminy

Zidani Fahed Imed Licence informatique Luminy

Les Avengers

Meilleure score estimé par validation croisée :**88 %**

Meilleur score obtenu sur images tests de mi-parcours :**83 %**

Table of Contents

1. Introduction

Ce projet a été porté sur la classification d'images qui indique la présence ou l'absence de la mer dans une image donnée en entrée, en s'appuyant sur différents classificateurs et une data-set qui contient 414 éléments(images). L'objectif est d'avoir le meilleur score possible sur un échantillon d'images de test.

Afin de répondre à cette attente, il faudra tout d'abord choisir la meilleure représentation d'image, puis par la suite le classifieur qui donnera le meilleur score pour le test.

Notre équipe s'est vus en difficultés sur plusieurs reprises notamment sur la représentation des données, étape importante avant le passage à la sélection du classificateur, chaque membre du groupe Fahd, Ibrahim, Younes s'est dispersé de son côté pour récolter le maximum d'informations et de sources afin de se retrouver en fin de semaine dans une séance de meeting de deux heure sur discord pour échanger sur les idées ,scores obtenus et de fixer les prochaines taches pour la semaine à avenir .

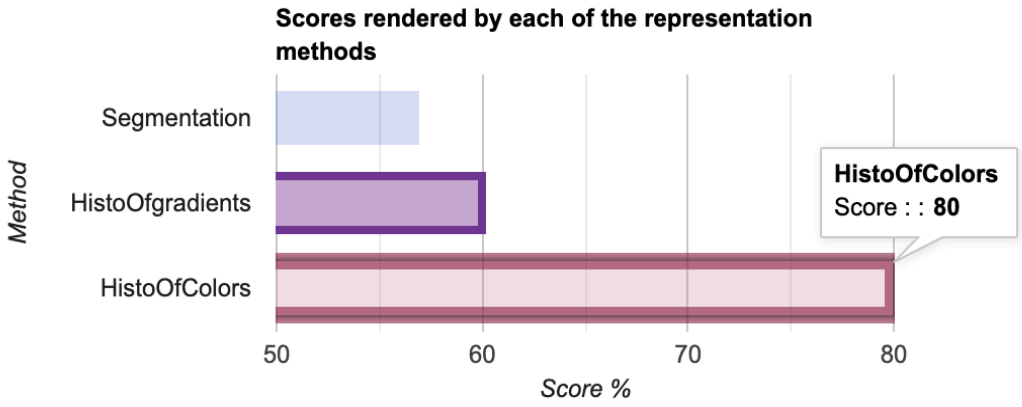
2. Prétraitement des données

2.1. Représentation des données :

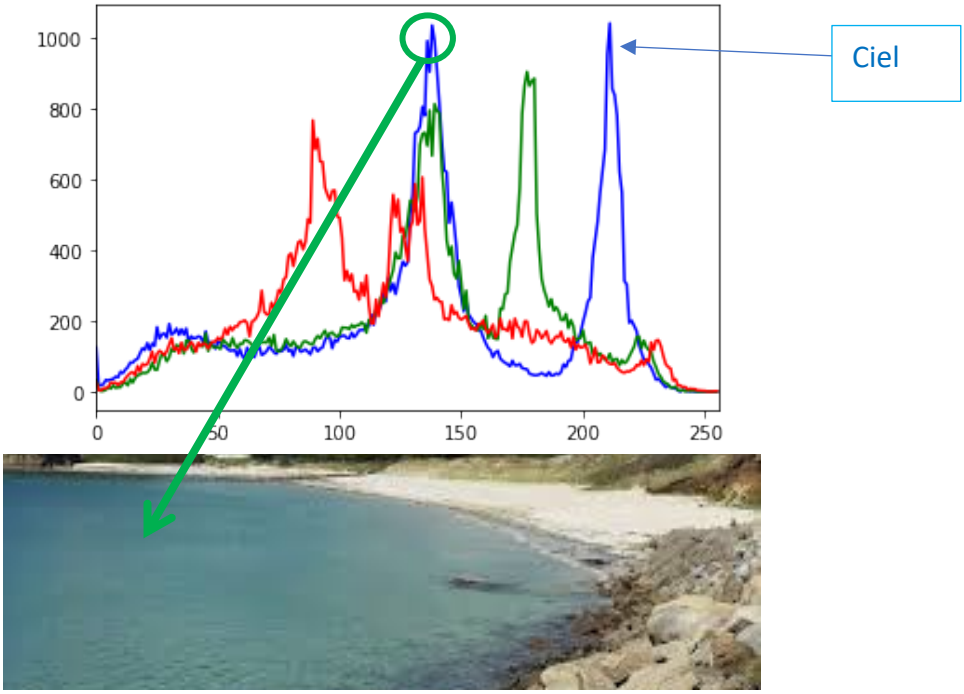
La résolution du problème de la classification d'images selon la présence ou non de la mer, revient à étudier les couleurs constituant l'image.

En se basant sur un calcul du niveau de la couleur bleu dans les images, la présence de la mer dans une image signifie un niveau de bleu élevé, et son absence signifie un niveau beaucoup moins élevé à l'exception des images avec un ciel ou un paysage bleu où les niveaux de bleu va différer.

Après avoir testé plusieurs représentation (histogramme des gradients, segmentation des images, ...) Et vu les résultats qu'on a eu L'histogramme des couleurs est le mode de description des données images le plus adaptée pour ce problème.



L'histogramme des couleurs décrit bien les canaux des couleurs d'images RGB (Rouge Vert Bleu). En creusant le long de cette piste on a été emmener à avoir d'excellents résultats.



- Niveau de la couleur bleu dans un échantillon de notre data-set -

Pour arriver en arriver, faut d'abord lire une image à partir d'un chemin de répertoire, puis la transformer en histogramme grâce aux 3 canaux et la fonction ***calchist*** **()** déjà prédéfinie dans la bibliothèque ***cv2*** de python.

Les processus de redimensionner, remettre et l'échelle et recadrer des images ont été ignoré, vu que ça nous mener à voir des résultats moins importants.

D'autres représentations d'images ont été utilisée mais sans suite, puisqu'ils n'ont pas donné un résultat meilleur que l'histogramme, on cite la repré**senta**tion d'images en ***nparray.***

2.2. Augmentation des données :

L'équipe des Avengers s'est contenté du jeu de données d'apprentissage initial, avec un data set de ***414 images*** [***207*** images avec présence de la mer *(Mer)* et ***207*** images avec absence de la mer *(Ailleurs)*].

L'estimant assez suffisant pour la résolution du problème.

Mer : Dossier dans le projet ***Data/Mer*** contenant les images avec présence de la mer.

Ailleurs : Dossier dans le projet ***Data/Ailleurs*** contenant les images avec absence de la mer.

3. Algorithme(s) d'apprentissage considérés

3.1. Algorithme retenu :

L'algorithme retenu est ***Bagging Bootstrap*** implémentée à notre manière.

Ce choix revient à l'excellent résultat obtenu de ***88%*** supérieur à tous les autres résultats des classificateur, ce dernier est plus adapté pour notre data-set où il concatène le résultat de plusieurs classificateurs, qui ont un taux de prédiction sur notre donnée test de *[70% - 79%].*

3.2. Explication de(s) algorithme(s) retenus :

Notre algorithme de *Bagging Bootstrapp* repose sur les algorithmes suivants :

- ***Régression logistique***.
- ***Classificateur naïf de bayes***.
- ***Support Vector Machine (Les séparateurs à vastes marges)***.
- ***Les arbres de décisions***

Où chaque classifieur de ces algorithme s'entraîne sur un échantillon de données différents, puis chacun des modèles prédit l'échantillon S, et avec un vote majoritaire sur chaque entrée de l'échantillon on la classifie.

Le paramétrage n'a pas été spécifié sur l'ensemble des classificateurs à l'exception du **SVC** et **régression logistique** où on précise qu'ils sont linéaires.

Les classificateurs sur lesquelles reposent notre algorithme de Bagging ont des taux de prédiction de *80 % aux maximum* et cela même avec les changements des paramètres, cela ne remet absolument pas en doutes la puissance de ses algorithmes mais plutôt la taille de notre data-set, où il est difficile de trancher pour un seul classifieur et dire qu'il est le plus approprié pour ce problème.

Naive Bayes Classifier (classificateur naïf de bayes) :

Les classificateurs naïfs de Bayes sont basés sur le théorème de Bayes. Une hypothèse retenue est celle des fortes hypothèses d'indépendance entre les caractéristiques. Ces classificateurs supposent que la valeur d'une caractéristique particulière est indépendante de la valeur de toute autre caractéristique.

Dans une situation d'apprentissage supervisé, les classificateurs Naive Bayes sont entraînés très efficacement. Les classificateurs Naive Bayes ont besoin de petites données d'entraînement pour estimer les paramètres nécessaires à la classification.

Les classificateurs Naive Bayes ont une conception et une mise en œuvre simples et peuvent être appliqués à de nombreuses situations réelles.

La régression logistique :

La régression logistique est une méthode d'analyse statistique qui consiste à prédire une valeur de données d'après les observations réelles d'un jeu de données.

La régression logistique est devenue un outil important dans la discipline de l'apprentissage automatique.

Cette approche permet d'utiliser un algorithme dans l'application d'apprentissage automatique pour classer les données entrantes en fonction des données historiques. Plus il y a de données pertinentes en entrée, plus l'algorithme est en mesure de prédire des classifications au sein des jeux de données.

La régression logistique est l'un des algorithmes d'apprentissage autiomatique les plus couramment utilisés pour les problèmes de classification binaire, lesquels ont deux valeurs par classe, comprenant des prédictions telles que "ceci ou cela", "oui ou non" et "A ou B".

Le but de la régression logistique est d'estimer les probabilités des événements et de déterminer une relation entre les caractéristiques et les probabilités de résultats particuliers.

- ***Support Vector Machine (Les séparateurs à vastes marges)*** :

Les séparateurs à vastes marges (SVM) sont un ensemble de méthodes d'apprentissage supervisé utilisées pour la classification, la régression et la détection des valeurs aberrantes.

L'objectif de l'algorithme SVM est de créer la meilleure ligne ou limite de décision capable de séparer l'espace à n dimensions en classes afin que nous puissions facilement placer le nouveau point de données dans la bonne catégorie à l'avenir. Cette frontière de meilleure décision est appelée un hyperplan.

SVM choisit les points/vecteurs extrêmes qui aident à créer l'hyperplan. Ces cas extrêmes sont appelés vecteurs de support et, par conséquent, l'algorithme est appelé machine à vecteur de support.

Les avantages des machines à vecteurs de support sont :

- Efficace dans les espaces de grande dimension.
- Toujours efficace dans les cas où le nombre de dimensions est supérieur au nombre d'échantillons.
- Utilise un sous-ensemble de points d'apprentissage dans la fonction de décision (appelés vecteurs de support), il est donc également efficace en mémoire.
- Polyvalent : différentes fonctions du noyau peuvent être spécifiées pour la fonction de décision. Des noyaux communs sont fournis, mais il est également possible de spécifier des noyaux personnalisés. Les machines à vecteurs de support (SVM) sont un ensemble de méthodes d'apprentissage supervisé utilisées pour la classification, la régression et la détection des valeurs aberrantes.

Les arbres de décision :

L'arbre de décision est une technique d'apprentissage supervisé qui peut être utilisée à la fois pour les problèmes de classification et de régression, mais elle est généralement préférée pour résoudre les problèmes de classification.

Il s'agit d'un classificateur arborescent, où les nœuds internes représentent les caractéristiques d'un ensemble de données, les branches représentent les règles de décision et chaque nœud feuille représente le résultat.

Dans un arbre de décision, il y a deux nœuds, qui sont le nœud de décision et le nœud feuille. Les nœuds de décision sont utilisés pour prendre n'importe quelle décision et ont plusieurs branches, tandis que les nœuds feuilles sont la sortie de ces décisions et ne contiennent pas d'autres branches.

Les décisions ou le test sont effectués sur la base des caractéristiques de l'ensemble de données donné.

Il s'agit d'une représentation graphique permettant d'obtenir toutes les solutions possibles à un problème/une décision en fonction de conditions données.

C'est ce qu'on appelle un arbre de décision car, semblable à un arbre, il commence par le nœud racine, qui se développe sur d'autres branches et construit une structure arborescente.

Pour construire un arbre, nous utilisons l'algorithme CART, qui signifie Classification and Regression Tree algorithm.

Un arbre de décision pose simplement une question et, en fonction de la réponse (Oui/Non), il divise ensuite l'arbre en sous arbres.

4. Évaluation des performances des classifieurs

4.1. Protocole d'estimation des performances :

Les critères de performances ont été choisi selon le fait que les différentes représentations des images donnaient des scores différents et ces scores diffère d'un classifieur a un autre.

Les estimations ont été obtenues ont croisant les résultats de plusieurs classificateurs, qui ont un taux de prédiction sur notre données tests, variant entre [70% - 79%]

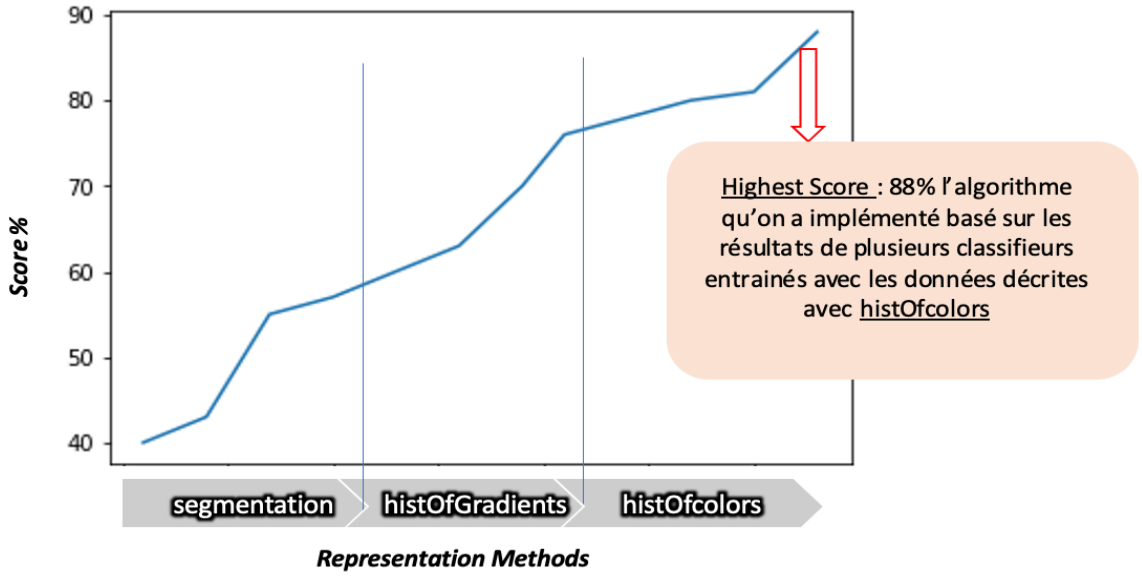
4.2. Performances obtenues :

Au regard du protocole qu'on a expliqué dans la section précédente, les performances qu'on a obtenues par la meilleure approche qu'on a pu implémenter celle du bagging, nous a emmené à un score de *88%.*

Sachant que sur le résultat de mi-parcours on a été sur un score de *79% a 80%*.

- La différence est dû au fait qu'on croisée les résultats de tous les classifieurs qu'on tester notre data-set avec, et on fait un score total mais sur le dernier algorithme qu'on a implémenté ont croisé les résultats de tous les classifieurs qui donnent une note entre *70 %* et *79%*. Ce qui nous permis d'augmenter considérablement le score a *88%.*

4.3. Éventuelles courbes de performance :



5. Conclusion :

Ce projet s'est révélé très enrichissant dans la mesure où il a consisté en une approche concrète de découverte de l'apprentissage automatique à travers une expérience de conception de classification supervisée binaire.

Malgré le temps de travail qui été très serré et la petite data-set qu'on avait à disposition on a pu réaliser une classification donnant un taux élevé.
La prise d'initiative, le respect des délais et le travail en équipe était des aspects essentiels de notre travail.

Les principaux problèmes, que nous avons rencontrés, concernaient la représentation des données, et l'implémentation de ces dernières selon le type ce qu'accepte chacun des classifieurs.

Toutefois, si un tel classifieurs devait être implémenter de manière parfaite sur une durée d'un mois de plus en temps plein , il serait nécessaire de récolter plus de données et d'adapter une meilleur stratégie de représentation .

Références :

- https://iq.opengenus.org/gaussian-naive-bayes/#:~:text=Gaussian%20Naive%20Bayes%20supports%20continuous,(independent%20dimensions)%20between%20dimensions.

- https://whatis.techtarget.com/fr/definition/Regression-logistique#:~:text=La%20r%C3%A9gression%20logistique%20est%20l,et%20%22A%20ou%20B%22.

- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

- https://scikit-learn.org/stable/modules/svm.html

- https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm