Description des choix : modélisation, estimation, logiciels

Table des matières

Partie 1 : Présentation

1.1 Introduction et objectif

Dans le cadre de notre projet, notre commanditaire a exprimé le besoin de disposer d'un pipeline généralisé, capable de traiter n'importe quel jeu de données. L'objectif est de permettre à ce pipeline de classifier, ou plutôt de créer des clusters de données en fonction d'une variable spécifique choisie par l'utilisateur. Ainsi, nous avons été chargés de concevoir et développer un tel pipeline en utilisant les technologies prescrites et l'assistance de notre professeur Maximilien.

1.2 Technologies et assistance

Pour mener à bien ce projet, nous avons utilisé les technologies qui nous ont été recommandées et nous avons bénéficié de l'aide précieuse de Maximilien. Son expertise nous a permis de mieux comprendre les enjeux et les défis liés à la réalisation d'un pipeline généralisé, ainsi que les meilleures pratiques pour y parvenir. Grâce à cela, nous avons pu optimiser nos choix techniques et méthodologiques pour aboutir à une solution performante et adaptée aux besoins de notre commanditaire.

1.3 Entraînement du pipeline et base de données utilisée

Bien que le pipeline développé soit conçu pour être général et s'adapter à divers jeux de données, il est essentiel de le tester et de l'entraîner. Pour ce faire, nous avons utilisé la base de données WikiArt, qui recense plusieurs dizaines de milliers de tableaux. Cette base de données représente un excellent terrain d'entraînement pour notre modèle, puisqu'elle contient une grande variété de données et permet d'évaluer l'efficacité de notre pipeline dans la création de clusters pertinents.

En exploitant cette base de données, nous avons pu affiner les performances de notre pipeline et nous assurer qu'il est capable de gérer efficacement différents types de données. Cela nous a également permis d'identifier d'éventuelles améliorations et d'adapter notre solution aux exigences spécifiques de notre commanditaire.

1.4 Conclusion de la présentation

En somme, le développement de ce pipeline généralisé a été mené avec succès grâce à l'utilisation de technologies appropriées et à l'aide apportée par Maximilien. La base de données WikiArt a servi de support pour l'entraînement de notre modèle, nous permettant d'optimiser ses performances et de le préparer pour une utilisation par notre commanditaire. Nous sommes convaincus que ce pipeline répondra aux besoins et aux attentes, tout en offrant une solution polyvalente et efficace pour la classification et la création de clusters de données.

Partie 2 : Outils

2.1 Langage Python

Pour mener à bien notre projet, nous avons opté pour l'utilisation du langage de programmation Python. Ce choix s'explique par la popularité et la polyvalence de Python dans le domaine du traitement de données et de l'apprentissage automatique. En effet, Python est un langage de haut niveau, open-source et multiplateforme qui facilite la réalisation de projets complexes grâce à sa syntaxe claire et concise. De plus, Python dispose d'un vaste écosystème de bibliothèques et de frameworks dédiés au traitement de données, ce qui en fait un outil idéal pour notre pipeline de clustering généralisé.

2.2 Librairie Bokeh

Afin de produire des rendus graphiques de qualité pour la visualisation des clusters générés par notre pipeline, nous avons utilisé la librairie Bokeh. Bokeh est une bibliothèque Python dédiée à la création de visualisations interactives pour le Web. Elle permet de générer facilement des graphiques élégants et dynamiques à partir de données complexes, en offrant une grande flexibilité et un contrôle précis sur les éléments visuels.

Bokeh s'intègre parfaitement avec le langage Python et d'autres bibliothèques de traitement de données, ce qui facilite son utilisation dans le cadre de notre projet. Grâce à Bokeh, nous avons pu créer des visualisations attrayantes et informatives pour présenter les résultats de notre pipeline de manière claire et intuitive.

En résumé, l'association du langage Python et de la librairie Bokeh nous a permis de concevoir et de développer un pipeline de clustering généralisé performant et convivial. Les capacités de Python en matière de traitement de données et d'apprentissage automatique, combinées à la puissance de Bokeh pour la création de visualisations interactives, ont été essentielles pour répondre aux exigences de notre commanditaire et offrir une solution complète et efficace.

Partie 3 : Répartition des tâches et méthodologie

3.1 Organisation du travail

Afin d'optimiser notre travail et de progresser efficacement sur les différents aspects du développement de ce pipeline, nous avons choisi de nous répartir les tâches. Cette organisation nous a permis de travailler simultanément sur les divers éléments nécessaires à la réalisation de notre projet et d'assurer une meilleure coordination au sein de notre équipe.

3.2 Entraînement du modèle et extraction du tensor

L'un des membres de notre équipe a été chargé de l'entraînement de notre modèle de Convolutional Neural Network (CNN) sur le jeu de données WikiArt. Son objectif principal était de fournir le meilleur modèle possible et d'en extraire un tensor. Pour ce faire, il s'est appuyé sur des modèles préexistants qu'il a améliorés en utilisant notre base de données. Cette approche a permis de gagner du temps et d'assurer la qualité et la pertinence du modèle entraîné.

3.3 Réduction de dimension

Pendant ce temps, un autre membre de notre équipe s'est concentré sur la réduction de dimension des données pour permettre une représentation graphique des résultats du modèle. Plusieurs méthodes ont été utilisées pour cette tâche, telles que :

- t-SNE (t-distributed Stochastic Neighbor Embedding) : une technique de réduction de dimension non linéaire qui préserve les relations locales entre les points de données.
- UMAP (Uniform Manifold Approximation and Projection) : une méthode de réduction de dimension qui préserve à la fois les relations locales et globales, offrant des représentations plus cohérentes et interprétables.
- ACP (Analyse en Composantes Principales) : une méthode linéaire de réduction de dimension qui projette les données sur les axes principaux de variation, maximisant la variance tout en minimisant la perte d'information.

Le but de cette étape était de réduire les résultats du modèle à deux dimensions, facilitant ainsi leur représentation graphique et leur interprétation.

3.4 Conclusion sur la répartition des tâches

La répartition des tâches au sein de notre équipe a permis de travailler de manière efficace et synchronisée sur les différentes composantes du pipeline. L'entraînement du modèle et la réduction de dimension ont été menés de front, permettant un développement rapide et cohérent du pipeline. Cette organisation a contribué au succès de notre projet et à la satisfaction de notre commanditaire.

Partie 4 : Hypothèse sur les méthodes de réduction de dimension

4.1 Supposition initiale

Au début de notre projet, nous avons émis l'hypothèse que les méthodes de réduction de dimension UMAP et t-SNE seraient plus performantes pour notre modèle par rapport à l'ACP. Cette supposition reposait sur le fait qu'UMAP et t-SNE sont des techniques non linéaires, tandis que l'ACP est une méthode linéaire.

Les méthodes non linéaires, comme UMAP et t-SNE, sont souvent considérées comme plus adaptées pour traiter des données complexes et de grande dimension, car elles peuvent capturer et préserver les structures locales et globales inhérentes aux données. En revanche, l'ACP, en tant que méthode linéaire, peut parfois être moins efficace pour traiter des données présentant des relations non linéaires.