IMT Nord Europe
École Mines-Télécom
IMT-Université de Lille

Génération de suggestions de musiques en fonction d'une image (Concept de Story Instagram)

UV Projet

Encadré par :
PR. WANNOUS Hazem

Réalisé par
HADDOUCH Mohammed Amine
MAHRAOUI Youssef
MEKOUAR Soumaya
SASSIOUI Mohcine

Villeneuve d'Ascq, France

10 mars 2023

Table des matières

1 Introduction Générale

Bienvenue dans la présentation de notre projet de suggestion de musiques basé sur l'analyse d'images et de descriptions de haute qualité grâce à l'utilisation de l'API d'Azure Cognitive Services et de modèles de traitement de langage naturel. L'objectif de notre projet est de proposer une méthode innovante pour suggérer de la musique en se basant sur les émotions dégagées d'une image, tout en tenant compte de la description associée à cette image. Nous avons développé une interface web conviviale pour faciliter l'interaction de l'utilisateur avec notre modèle de suggestion de musique, qui est encapsulé et optimisé pour offrir une expérience rapide et précise. Dans ce projet, nous avons relevé plusieurs défis, notamment la sélection et le traitement de données de haute qualité, ainsi que l'intégration et la mise en œuvre de technologies de pointe telles que l'API d'Azure Cognitive Services et les modèles de traitement de langage naturel. Cependant, nous sommes convaincus que notre projet offre une valeur ajoutée significative en proposant une solution novatrice pour la suggestion de musique personnalisée, qui peut être étendue à des applications mobiles et même intégrée à des réseaux sociaux tels qu'Instagram.

2 Contexte du projet

2.1 Présentation du contexte

Le projet que nous avons réalisé s'inscrit dans le cadre de l'UV Projet et porte sur la génération de suggestions de musique en fonction d'une image. L'idée est née de la problématique que l'on rencontre souvent lors de la publication d'une Story Instagram : trouver une musique qui correspond parfaitement à l'image que l'on souhaite partager. Ce processus peut être fastidieux et prendre beaucoup de temps.

Notre objectif est donc de proposer une solution qui permettrait de générer automatiquement des suggestions de musiques en fonction de l'image sélectionnée. Pour ce faire, nous avons utilisé des modèles de traitement du langage naturel (NLP) tels que Hugging Face ainsi que des API de reconnaissance d'image et de traitement du son proposées par Azure. Nous avons également développé une interface en HTML, CSS, Flask et Python pour permettre à l'utilisateur de télécharger une image et recevoir des suggestions de musiques en retour.

Notre projet se distingue des autres solutions similaires existantes sur le marché par son utilisation de techniques de NLP et de modèles de traitement du langage pour améliorer la pertinence des suggestions de musiques. Nous sommes convaincus que cette solution pourrait trouver un large public et faciliter la vie de nombreuses personnes cherchant à publier des stories Instagram avec une bande sonore parfaitement adaptée à leur image.

2.2 Problématique

Lorsque nous publions une Story sur Instagram, l'image que nous choisissons est souvent accompagnée d'une musique pour créer une ambiance particulière. Cependant, choisir la bonne musique peut s'avérer être un processus long et fastidieux. En effet, il faut parcourir une grande quantité de musiques pour trouver celle qui correspond le mieux à l'image que nous souhaitons publier. Cette tâche peut être particulièrement difficile pour les personnes qui ne connaissent pas bien les différentes options de musique disponibles ou qui n'ont pas le temps de passer des heures à chercher la musique idéale.

C'est pourquoi notre projet de génération de suggestions de musique en fonction d'une image vise à résoudre cette problématique. Nous avons développé un système qui utilise la reconnaissance d'image pour trouver des suggestions de musiques qui correspondent parfaitement à l'image sélectionnée. En utilisant des algorithmes de traitement de langage naturel et de reconnaissance d'image, notre système est capable de déterminer les caractéristiques clés de l'image, puis de générer des suggestions de musique qui correspondent à ces caractéristiques. De cette manière, notre système peut aider les utilisateurs d'Instagram à trouver rapidement et facilement la musique idéale pour accompagner leur image, sans avoir à passer des heures à chercher manuellement.

2.3 Étude de l'existant

Notre projet vise à générer des suggestions de musique en fonction d'une image pour faciliter la publication de Stories Instagram. Afin de mener à bien ce projet, il est important de comprendre les solutions existantes et les approches utilisées pour résoudre des problèmes similaires.

Après une étude de l'existant, nous avons constaté qu'il existe déjà des systèmes de recommandation de musique basés sur l'historique de l'utilisateur, la classification de genre, et même la reconnaissance de la voix. Cependant, il n'y a pas de système de recommandation de musique basé sur l'analyse d'images, ce qui rend notre projet unique.

En résumé, l'étude de l'existant montre que les solutions existantes ne sont pas entièrement satisfaisantes pour notre projet. Notre approche innovante basée sur l'apprentissage automatique, combinée à l'utilisation d'API telles que Azure Computer Vision et Hugging Face, nous permettra de générer des suggestions de musique en fonction de l'image téléchargée pour une expérience utilisateur plus fluide et plus intuitive.

2.4 Objectifs du projet

Les objectifs de notre projet étaient les suivants :

— Développer une application qui génère des suggestions de musique en fonction d'une image téléchargée.
— Utiliser des techniques de traitement de données et d'apprentissage automatique pour analyser les caractéristiques de l'image et recommander des musiques appropriées.

— Intégrer des API de services tiers (comme Azure et Hugging Face) pour améliorer la précision des suggestions.
— Concevoir une interface utilisateur conviviale pour permettre aux utilisateurs de télécharger facilement une image et de recevoir des suggestions de musique.
— Répartir les tâches de manière efficace entre les membres de l'équipe pour respecter les délais du projet.
— Suivre un cycle de vie de projet rigoureux pour garantir que toutes les étapes du projet sont effectuées de manière ordonnée et que les problèmes sont résolus rapidement.

2.5 Démarches et cycle de vie du projet

2.5.1 Cycle de vie du Projet

Le projet a été réalisé par une équipe de 4 personnes : Amine, Youssef, Soumaya et Mohcine. Nous avons commencé le projet le 6 février et avons eu un 4 semaines pour le réaliser. Le cycle de vie du projet a été détaillé comme suit :

1- Analyse des besoins et des fonctionnalités : Cette étape consiste à identifier les besoins et les fonctionnalités attendus pour l'application de génération de suggestions de musique en fonction d'une image. Nous avons commencé par une phase de planification pour définir les différentes tâches et les affecter à chaque membre de l'équipe en fonction de ses compétences.

Nous avons également planifié les différentes étapes de développement, notamment la création du modèle d'apprentissage profond, l'intégration de l'API Web et la création de l'interface utilisateur.

Pendant la phase de développement, nous avons suivi une approche Agile en utilisant des méthodes Scrum. Nous avons travaillé en étroite collaboration pour assurer une communication régulière et pour résoudre rapidement les problèmes qui se posaient.

Lors de cette étape, nous avons aussi procédé à une recherche approfondie de Datasets existants afin de choisir les bons DataSets appropriés.

2- Conception : Cette étape comprend la conception de l'interface utilisateur, le nettoyage et l'ingestion du DataSets, la définition de l'architecture logicielle, ainsi que la planification des tâches à effectuer.

3- Développement : Cette étape inclut la mise en œuvre des différentes fonctionnalités de l'application et leur intégration. Premièrement, la recherche d'une technologie permettant l' "Image Captionning" et son intégration au code existant. Ensuite, l'implementation du modèle de recherche de similarités à l'aide du modèle Hugging Face.

4- Tests et validation : Une fois l'application développée, une série de tests et de validations ont été effectués sur un environnement local pour s'assurer que l'application est opérationnelle et fonctionne correctement.

La phase de test a été réalisée en parallèle avec la phase de développement, et nous avons utilisé différentes techniques de test pour garantir la qualité.

2.5.2 Répartition des tâches :

La répartition des tâches a été affectée comme suit :

Haddouch Mohammed Amine : Participation au développement de la partie back-end de l'application et à la partie Interfaçage.

Mahraoui Youssef : Conception et Intégration du modèle de recherche de similarités et participation dans la partie Ingénierie de données.

Mekouar Soumaya : Intégration de l'API d'Azure et et participation dans la partie Ingénierie de données.

Sassioui Mohcine : Participation à la recherche de DataSets, et participation à la conception de l'interface utilisateur.

Communs entre les différents membres du groupe :

— Tests et validation de l'application.
— Amélioration des performances de l'application.
— Rédaction de la documentation du projet.

2.5.3 Répartition hebdomadaire des tâches :

Semaine 1 : Collecte et pré traitement des données : Recherche de datasets et nettoyage des données Implémentation de l'API d'Azure Cognitive Services pour la génération de descriptions d'images.

Semaine 2 : Mise en place du modèle de suggestion de musique et intégration de l'API d'Azure et optimisation du modèle de Hugging Face.

Semaine 3 : Développement de l'interface Web pour encapsuler le modèle de suggestion de musique Tests de l'interface Web et amélioration de l'expérience utilisateur.

Semaine 4 : Amélioration du modèle utilisant Hugging Face et Intégration de l'interface Web avec le modèle de suggestion de musique et rédaction de la documentation du projet.

3 Réalisation du projet

3.1 Volet fonctionnel

Le projet consiste en le développement d'un modèle d'apprentissage automatique qui a pour objectif la suggestion d'un ensemble de musiques qui s'adaptent à une image donnée. Une application Web à été développée pour en-capsuler le modèle permettant de générer des suggestions de musiques en fonction d'une image fournie par l'utilisateur. L'interface graphique a été développée en utilisant les langages HTML, CSS et JavaScript. Elle permet à l'utilisateur de charger une image depuis son ordinateur. Après avoir soumis

l'image, l'application affiche une liste de suggestions de musiques. Chaque suggestion est représentée par le titre de la chanson et un lien cliquable qui permet sa lecture via une redirection.

3.2 Volet technique

3.2.1 Choix du DataSet

Concevoir un modèle d'apprentissage automatique nécessite un flux de données important. La première partie de notre projet a donc été consacrée à la recherche de données. Deux approches ont été mises en place afin de compléter cette étape :

— Scraping (Récolte de données) : Pour cette première approche, on a développé un algorithme de Scraping qui a pour objectif la récolte des informations via différents réseaux sociaux, le principe était de parcourir un ensemble de comptes de célébrités qui ont des comptes bien gérés à accès publique afin de récolter leurs "Storys", à savoir des images avec leurs musiques correspondantes idéalement préselectionnées. Malheureusement, on a pas pu achever cette approche vu que les différents réseaux sociaux interdisent cette opération.

— Dataset déjà prêt : Cette approche consiste à travailler avec un Dataset qui contient un ensemble de musiques avec des tags correspondants. Nous avons eu du mal à trouver un ensemble de données préexistant contenant des musiques et leurs tags associés. Finalement, nous avons pu trouver un dataset contenant plus de 90 000 musiques avec leurs émotions dégagées. C'est le Dataset avec lequel nous allons travaillé tout au long de notre projet. Cette approche va être détaillée dans la partie suivante.

3.2.2 Ingénierie de données

La phase suivante était d'entamer la partie d'Ingénierie des données. Il était crucial de nettoyer le dataset en éliminant les doublons et en supprimant les entrées erronées pour garantir la qualité des données. Nous avons également effectué une transformation majeure consistant à extraire les différentes combinaisons d'émotions dégagées par les musiques. Cette opération nous a permis d'obtenir plus de 9031 combinaisons, et nous avons eu l'idée de transformer le dataset de telle manière que chaque ligne contienne une combinaison d'émotions et les musiques associées. Ainsi, au lieu d'avoir plus de 90 000 lignes, nous avons obtenu 9031 lignes, chacune correspondant aux musiques associées à chaque set de sentiments (seeds) Cette transformation a amélioré la performance de notre modèle et réduit le temps de réponse.

3.2.3 Traitement de l'Image

Au cours de notre projet, nous avons essayé d'utiliser différentes bibliothèques de Python pour traiter nos images. Nous avons testé Pillow, OpenCV et Scikit-Image, entre

autres. Chacune de ces bibliothèques a ses avantages et inconvénients, et nous avons pu réaliser des opérations de traitement d'image telles que la conversion de format d'image, la détection de contour et l'amélioration de la qualité de l'image. Cependant, nous avons rapidement réalisé que ces bibliothèques n'étaient pas suffisantes pour répondre à tous nos besoins en matière de traitement d'image. Nous avons finalement opté pour l'utilisation de l'API Cognitive Services d'Azure, qui offre une gamme complète de fonctionnalités pour le traitement d'image, y compris la reconnaissance de texte, la détection d'objets, la reconnaissance faciale et la classification d'image. Cette API utilise des algorithmes de deep learning pour fournir des résultats précis et fiables. L'utilisation de l'API Cognitive Services d'Azure nous a permis de disposer de fonctionnalités de traitement d'image plus avancées et plus fiables que les bibliothèques Python que nous avons utilisées auparavant. Cette API a été facile à intégrer à notre projet et nous avons pu l'utiliser pour améliorer considérablement notre workflow de traitement d'image.

3.2.4 Recherche de Similarités

Le modèle sentencetransformers/all-deMiniLM-L6-v2 de Hugging Face Le modèle "sentence-transformers/all-MiniLM-L6-v2" est un modèle de traitement de langage naturel (NLP) conçu par la bibliothèque Hugging Face. Il utilise une architecture de réseau de neurones basée sur la transformer pour produire des embeddings de phrases de haute qualité, c'est-à-dire des représentations vectorielles d'une phrase qui capturent ses caractéristiques sémantiques et syntaxiques. Le modèle "sentence-transformers/all-MiniLM-L6-v2" est entraîné sur de grandes quantités de données textuelles dans plusieurs langues. Il utilise une version miniaturisée de la transformer, appelée MiniLM, qui permet d'obtenir une performance élevée tout en réduisant considérablement la taille du modèle. En outre, le modèle est régularisé avec une méthode appelée distillation, qui permet de transférer les connaissances d'un modèle plus grand vers un modèle plus petit.

Phase d'encodage Dans le modèle "sentence-transformers/all-MiniLM-L6-v2", la phase d'encodage est l'étape où la phrase d'entrée est transformée en un embedding de haute qualité(un vecteur numérique dense qui représente la phrase d'entrée de manière précise et efficace). Cette étape est cruciale car elle permet de représenter la phrase sous forme de vecteur numérique dense qui capture ses caractéristiques sémantiques et syntaxiques. La phase d'encodage dans ce modèle est réalisée en utilisant une version miniaturisée de la transformer (une architecture de réseau de neurones), appelée MiniLM. Cette architecture utilise des couches de self-attention pour permettre à chaque mot de la phrase de "s'auto-encoder", c'est-à-dire de prendre en compte les autres mots de la phrase pour construire sa propre représentation vectorielle. Cette méthode permet de capturer des relations complexes entre les différents mots de la phrase, ce qui améliore la qualité de l'embedding produit. Le vecteur résultant de la phase d'encodage a une taille de 384. Cela signifie que la phrase d'entrée est transformée en un vecteur numérique de 384 dimensions qui capture ses caractéristiques sémantiques et syntaxiques.

Approche de Similarité Le modèle "sentence-transformers/all-MiniLM-L6-v2" utilise une approche de similarité pour calculer la similitude entre les embeddings de phrases (vecteurs denses) . Cette approche est basée sur la mesure de la distance cosinus, qui permet de calculer la similarité cosinus entre deux vecteurs dans un espace vectoriel. Plus précisément, pour calculer la similarité entre deux phrases, le modèle utilise la fonction "cosine-similarity" de la bibliothèque scikit-learn pour calculer la distance cosinus entre les embeddings des deux phrases (vecteurs denses) . Cette distance cosinus est une mesure de la similarité entre les deux phrases, qui varie de 0 à 1, où 0 indique une différence totale entre les deux phrases et 1 indique une similitude totale. L'approche de similarité basée sur la mesure de la distance cosinus est une méthode efficace pour calculer la similarité entre les embeddings de phrases, et elle permet au modèle "sentence-transformers/all-MiniLM-L6-v2" d'obtenir des résultats de haute qualité dans différentes tâches de NLP.

3.2.5 Interface Graphique

Nous avons développé une interface web pour encapsuler notre modèle de suggestion de musique basé sur les images. Cette interface web permet à l'utilisateur de télécharger une image et de recevoir en retour une liste de musiques qui s'adaptent à l'ambiance dégagée par l'image donnée.L'interface web que nous avons développée offre plusieurs avantages par rapport à l'utilisation du modèle seul dans un environnement de développement intégré (IDE). Tout d'abord, l'interface web rend notre modèle accessible à un public plus large, car il n'est plus nécessaire de posséder des connaissances en programmation ou en apprentissage automatique pour l'utiliser. Les utilisateurs peuvent simplement télécharger une image et recevoir des suggestions de musique en retour, ce qui rend l'expérience plus conviviale et plus accessible. De plus, l'interface web permet également une meilleure visualisation des résultats. Au lieu de simplement afficher les suggestions de musique dans une console de terminal, l'interface web présente les résultats sous forme de tableau, avec des liens pour écouter chaque suggestion de musique. Cela permet aux utilisateurs de mieux explorer les résultats et de découvrir de nouvelles musiques en fonction de leur image d'origine. Enfin, encapsuler le modèle derrière une interface web offre une plus grande flexibilité et permet des intégrations plus faciles avec d'autres outils et applications. Par exemple, notre interface web pourrait être intégrée dans une application de musique existante ou utilisée comme composant d'une plateforme plus large de recommandation de produits ou de services. En somme, l'interface web que nous avons développée permet à un plus grand nombre de personnes d'utiliser notre modèle de suggestion de musique, rend les résultats plus faciles à visualiser et à explorer, et offre une plus grande flexibilité pour l'intégration avec d'autres outils et applications.

3.2.6 Pipeline du Projet

Notre projet suit un pipeline précis. Tout commence lorsque l'utilisateur télécharge une image via notre interface web. Cette image est envoyée à l'API Azure Cognitive Services pour obtenir une description détaillée. Cette description est ensuite transmise à notre modèle, qui la traite en effectuant une opération d'encodage pour transformer la

description en un vecteur de dimension 384. Ensuite, nous calculons la similarité entre ce vecteur et chaque vecteur qui représente une émotion parmi notre ensemble de 276 émotions. Nous parcourons notre dataset de 9031 lignes pour chaque ligne, nous calculons un vecteur de taille 276, dont chaque valeur représente la similarité entre le vecteur de chaque émotion dans la combinaison et le vecteur de description. Nous calculons la moyenne de chaque ligne, qui est la somme des valeurs du vecteur résultant divisé par le nombre d'émotions dans la combinaison. Nous classons ensuite les combinaisons selon cette moyenne de façon décroissante et renvoyons les trois meilleures combinaisons. Enfin, nous affichons dans l'interface web l'ensemble des musiques associées à ces trois meilleures combinaisons. Ce processus permet à l'utilisateur d'obtenir une suggestion de musique pertinente en fonction de l'image téléchargée, avec un niveau élevé de précision et de cohérence.

4 Difficultés rencontrées

Au niveau de la recherche du dataset, l'une des difficultés était de trouver un dataset de musiques avec les tags émotionnels associés. Cela a nécessité une recherche approfondie pour trouver le bon dataset qui contient suffisamment de données et de qualité pour entraîner notre modèle.

Quant à l'API de description d'images, elle peut parfois générer des descriptions imprécises ou peu informatives selon la qualité de l'image en entrée. Cela peut impacter directement la qualité de la suggestion de musique générée par notre modèle. Cela a nécessité une attention particulière pour choisir la bonne API et une validation manuelle pour assurer la qualité des descriptions générées. trop de temps pour dataset et pour ingenierie de donnees afin d'organiser le dataset

5 Perspectives

— Amélioration du modèle : Le modèle pourrait être amélioré en utilisant des techniques plus avancées de traitement de texte et d'apprentissage automatique. Cela pourrait améliorer la précision de la recommandation de musique et rendre le modèle plus efficace.
— Extension de la base de données : Étendre la base de données de musique avec plus de musiques, de genres et d'artistes pourrait améliorer la qualité de la recommandation
— Intégration de sources de données supplémentaires : En plus de la description générée par l'API, d'autres sources de données telles que les tags de musiques, les paroles, les notes de critiques musicales pourraient être intégrées pour améliorer la précision de la recommandation.
— Intégration de l'interactivité avec l'utilisateur : L'interface web pourrait être améliorée pour permettre à l'utilisateur d'interagir avec les recommandations, par exemple en permettant à l'utilisateur d'indiquer quels morceaux ont été appréciés et lesquels n'ont pas été appréciés, pour affiner la recommandation de musique à l'avenir. Une

perspective intéressante pour ce projet serait de l'intégrer dans le réseau social Instagram. En effet, notre modèle de suggestion de musique peut être utilisé pour proposer des musiques qui s'adaptent avec les stories partagées sur Instagram. De cette manière, les utilisateurs pourront bénéficier d'une expérience plus immersive et personnalisée. Il serait également possible de développer une application mobile dédiée à notre projet. Cette application permettrait aux utilisateurs d'uploader des images et de recevoir des suggestions de musiques en temps réel, en utilisant la même approche de suggestion que celle mise en place dans l'interface web développée pour notre projet.

6 Conclusion

En conclusion, notre projet de suggestion de musique basé sur une image et une description générée par l'API d'Azure Cognitive Services a abouti à un pipeline efficace. Nous avons utilisé un modèle de transformer de Hugging Face pour encoder la description et calculer la similarité avec des émotions prédéfinies. Ensuite, nous avons parcouru notre dataset pour calculer la similarité entre chaque combinaison d'émotions et la description et avons classé les résultats en fonction de leur moyenne. L'interface web développée permet à l'utilisateur de télécharger une image et de recevoir trois combinaisons d'émotions et les musiques associées. Nous avons rencontré des difficultés dans la collecte des datasets et la qualité de la description générée par l'API, mais nous avons pu trouver des solutions pour y remédier. Nous envisageons des perspectives intéressantes pour ce projet, telles que l'intégration dans les réseaux sociaux pour proposer des musiques qui s'adaptent avec les stories ou le développement d'une application mobile pour une utilisation plus conviviale. Ce projet montre l'efficacité des modèles de transformer pour la suggestion de contenu personnalisé et ouvre des perspectives pour d'autres applications dans le domaine de l'IA.