mlops

reds 2023

christophe bouder mlia

i introduction

ii outils fondamentaux

iii defis deploiement de modeles de ml

iv emballage ml modeles

v gerer le cycle de vie du ml

vi docker pour ml

vii creer des applications web de ml utilisation de l'api

viii ci/cd pour ml

ix deploiement de modeles de ml sur google cloud platform

x surveillance et debogage

xi post-production modeles ml

ii outils fondamentaux

python anaconda vscode
données structures
instructions de contrôle et boucles
fonctions
principes de base de git gitlab, github

iii defis deploiement de modeles de ml

les différentes approches permettant de déployer des modèles de ml en production.

- mlops comble le fossé entre l'étape des expériences d'apprentissage automatique et le déploiement du modèle dans l'environnement de production.
- la stratégie de déploiement du modèle dépend des besoins de l'entreprise, des utilisateurs et des applications.
- mlops combine les processus d'apprentissage automatique et les meilleures pratiques de devops pour fournir des résultats cohérents avec des pipelines et une gestion automatisés.

iv emballage ml modeles

•environnement virtuel
•dossier d'exigences
•sérialisation et désérialisation des modèles ml
•tester le code python à l'aide de pytest
•emballage python et gestion des dépendances
•développer, construire et déployer des packages ml
•configurer les variables d'environnement et les chemins d'accès

•le paquetage python contient plusieurs modules, et chaque module contient des fonctions, des classes et des variables.
•le fichier manifest.in contient la liste des fichiers à inclure et à exclure.
•les fichiers requirements.txt contiennent la liste des paquets à installer à l'aide de pip.
•python fait référence à la sérialisation et à la désérialisation par les termes pickling et unpickling, respectivement.
•le fichier init . py indique que le répertoire doit être traité comme un paquet.

v gerer le cycle de vie du ml

•introduction à mlflow
•suivi du mlflow
•projets mlflow
•modèles mlflow
•registre des modèles mlflow

●mlflow vous aide à passer de la phase d'expérimentation à la phase de déploiement
d'un projet de ml.
●à l'exception du registre des modèles, tous les composants peuvent être utilisés sans être intégrés à une base de données telle que mysql ;
toutefois, il est souhaitable de les intégrer à une base de données telle que mysql.
●par défaut, le projet mlflow utilise conda pour l'installation des dépendances ;
cependant, vous pouvez procéder sans conda en utilisant l'option -no-conda.
●chaque composant du mlflow est accessible séparément, mais il est possible de les relier pour créer le flux.

vi docker pour ml

une plateforme de conteneurisation

•docker est une plateforme de conteneurisation permettant d'empaqueter des applications et leurs dépendances sous la forme d'un conteneur.

•docker garantit la reproductibilité, la portabilité, la facilité de déploiement, les mises à jour granulaires, la légèreté et la simplicité.

•docker gère les données dans le conteneur docker à l'aide de docker volumes.

•une instance d'un conteneur est créée lorsque vous exécutez l'image docker.

vii creer des applications web de ml utilisation de l'api

rest apis
fastapi
streamlit
flask
gunicorn

•l'api restful fournit une plateforme commune pour communiquer avec
un système d'information. qui ont été conçues dans différents langages de programmation.
•fastapi nécessite python 3.6 et plus.
•streamlit est une bibliothèque open-source en python qui permet aux
utilisateurs de construire et de partager des interfaces utilisateur
attrayantes pour les modèles d'apprentissage automatique.
•fastapi dispose d'une documentation standard et interactive intégrée.
•les fichiers de configuration de nginx n'ont pas d'extension et chaque ligne
devrait être fermé à l'aide d'un ; (point-virgule).

viii ci/cd pour ml

processus d'intégration continue (ci), de
livraison continue (cd), de déploiement continu (cd) et de formation continue (ct)
dans le pipeline ci/cd

continuous integration/continuous delivery/deployment

liste d'outils populaires pour le pipeline ci/cd :
•jenkins
•actions github
•bambou
•circleci
•gitlab ci/cd
•travis ci

•cd désigne indifféremment la livraison continue ou le déploiement
continu,
en fonction du niveau d'automatisation que vous envisagez de mettre en
œuvre.
•jenkins est un outil d'automatisation ci/cd modulaire et open-source
écrit en java, qui s'accompagne d'un grand nombre de plugins.
•jenkins comprend le format xml des rapports de test junit.

ix deploiement de modeles de ml sur google cloud platform

Google Cloud Platform (GCP)
Google Kubernetes Engine (GKE)

•Le cluster Kubernetes peut être partagé entre plusieurs projets.
•L'identifiant Gmail est nécessaire pour créer un compte sur GCP.
•Cloud Build est une plateforme sans serveur qui vous permet d'automatiser la construction, le test et le déploiement de conteneurs rapidement.
•Les constructeurs de nuages sont des images de conteneurs dans lesquels sont installés des langages et des outils courants.
•Les mêmes étiquettes doivent être utilisées dans les fichiers service.yaml et deployment.yaml pour les services de
une communication réussie entre le service et le déploiement.

x surveillance et debogage

Une fois qu'un modèle de ML est déployé en production, il est essentiel de le surveiller afin de s'assurer que les performances du modèle restent à la hauteur et qu'il continue à fournir des résultats fiables de manière transparente. En fait, il existe de nombreuses raisons de défaillance dans les applications ou services de ML, telles que la défaillance du pipeline, la dégradation du modèle au fil du temps, la modification des données d'entrée du modèle, la défaillance du système ou du serveur, et la modification du schéma.

xi post-production modeles ml

•La sécurité des modèles est un élément essentiel des MLOps.
•Les tests A/B vous permettent d'effectuer une évaluation en ligne des modèles de ML.
•Les attaquants peuvent utiliser la méthode de la boîte noire ou la méthode de la boîte blanche lorsqu'ils tentent une attaque contradictoire.
•La bibliothèque Python Adversarial Robustness Toolbox (ART) et l'outil en ligne de commande Counterfit peuvent être utilisés pour la sécurité des modèles ML.
•Un bandit multi-bras (MAB) est une version avancée du test A/B. Il est suffisamment intelligent pour décider quel modèle devrait obtenir plus de trafic en évaluant plusieurs modèles. Il est suffisamment intelligent pour décider quel modèle devrait obtenir plus de trafic en évaluant plusieurs

I INTRODUCTION

La mise en production du modèle d'apprentissage automatique est une tâche complexe qui nécessite une compréhension approfondie des dernières technologies et du pipeline CI/CD. Les MLOps deviennent de plus en plus populaires dans le domaine de la science des données.

Ce cours est conçu pour fournir un guide complet sur la construction et le déploiement d'applications ML avec MLOps. Il couvre un large éventail de sujets, y compris les bases de la programmation Python, Git, le cycle de vie de l'apprentissage automatique, Docker, et des concepts avancés tels que l'empaquetage du code Python pour les modèles ML, la surveillance, la sécurité des modèles, Kubernetes, les tests à l'aide de pytest et l'utilisation du pipeline CI/CD pour la construction et le déploiement d'applications ML robustes et évolutives sur des plateformes cloud, y compris Azure, GCP et AWS.

Tout au long de ce cours, vous découvrirez les MLOps, les différents outils et les techniques de déploiement des modèles de ML. Vous apprendrez également à les utiliser pour produire des modèles et des applications de ML efficaces, évolutifs et faciles à maintenir. En outre, vous découvrirez les meilleures pratiques et les modèles de conception pour les MLOps.

Ce cours s'adresse aux data scientists, aux développeurs de logiciels, aux ingénieurs de données, aux analystes de données et aux managers qui découvrent les MLOps et souhaitent apprendre à mettre en production des modèles de ML. Il est également utile aux data scientists et aux ingénieurs ML expérimentés qui souhaitent approfondir leurs connaissances de ces technologies et améliorer leurs compétences en matière de déploiement de modèles ML en production.

Ce cours vous permettra d'acquérir les connaissances et les compétences nécessaires pour devenir compétent dans le domaine des MLOps. J'espère que vous trouverez ce cours instructif et utile.

II OUTILS FONDAMENTAUX

stocker et de gérer des données à l'aide de structures de
données courantes en Python. Vous devriez être en mesure d'utiliser les paquets de traitement de données de Python pour la manipulation des données, et vous devriez également savoir comment créer des classes, des méthodes et des objets.

A Python anaconda vscode

Python est un langage de programmation de haut niveau interprété, orienté objet et doté d'une sémantique dynamique. Sa syntaxe simple et facile à apprendre met l'accent sur la lisibilité et réduit donc le coût de la maintenance des programmes. Python prend en charge les modules et les paquets, ce qui encourage la programmation modulaire et la réutilisation du code. Il e s t développé sous une licence open-source approuvée par l'OSI, ce qui le rend librement utilisable et distribuable, même pour un usage commercial. Le Python Package Index (PyPI) héberge des milliers de modules tiers pour Python.

Anaconda va gérer des paquets

vscode comme jupyter notebook visualisateur

B Données structures

Les structures de données ne sont rien d'autre que des moyens particuliers de stocker et de gérer des données en mémoire afin de pouvoir y accéder facilement et les modifier ultérieurement. Python est livré avec un ensemble complet de structures de données, qui jouent un rôle important dans la programmation parce qu'elles sont réutilisables, facilement accessibles et gérables.

Tableau

Les tableaux sont des collections d'éléments homogènes. On peut utiliser le même type de données dans un même tableau.

Dictionnaire

Les dictionnaires sont définis comme des paires clé/valeur séparées par des virgules et placées entre accolades.

Liste

Il s'agit d'une collection d'éléments hétérogènes entre crochets. On peut utiliser le des types de données identiques ou différents dans une liste.

Set (jeu de mots)

Un ensemble est une collection d'éléments uniques (non dupliqués) entourés d'accolades. Un ensemble peut contenir des éléments hétérogènes.

Chaîne

Une chaîne de caractères est utilisée pour stocker des données textuelles. Elle peut être représentée par des guillemets simples (') ou doubles ("").

Tuple

Les tuples en Python (« n-uplets » en français) sont des collections de données différentes ou identiques qui sont désignées par un index et qui ne peuvent pas être modifiées

Les tuples sont des collections d'éléments entourés de crochets ronds (facultatifs) et sont immuables, c'est-à-dire qu'ils ne peuvent pas être modifiés.

Instructions de contrôle et boucles

Python dispose de plusieurs types d'instructions de contrôle et de boucles. Examinons-les.

If else if

Syntaxe :

si condition1 :
Code à exécuter
elif condition2 :
Code à exécuter
autre :
Code à exécuter

pour la boucle

Il est utilisé lorsque vous souhaitez itérer sur une séquence ou un itérable tel qu'une chaîne de caractères, une liste, un dictionnaire ou un tuple. Il exécute un bloc de code un certain nombre de fois.

Syntaxe :

for i in range (1,121)

boucle while

Elle est utilisée lorsque vous souhaitez exécuter un bloc de code indéfiniment jusqu'à ce que la condition spécifiée soit remplie. En outre, vous pouvez utiliser une instruction else pour exécuter un bloc de code une seule fois lorsque la première condition est fausse.

déclaration de passage

En Python, l'instruction pass fait office d'espace réservé. Si vous prévoyez d'ajouter du code plus tard dans des boucles, des définitions de fonctions, des définitions de classes ou des instructions if, vous pouvez écrire une instruction pass pour éviter toute erreur, car elle ne fait rien. Elle permet aux développeurs d'écrire la logique ou la condition par la suite et de poursuivre l'exécution du code restant.

Fonctions

Les développeurs utilisent des fonctions pour effectuer plusieurs fois la même tâche. Une fonction est un bloc de code qui peut prendre des arguments, effectuer certaines opérations et renvoyer des valeurs. Python permet aux développeurs d'utiliser des fonctions prédéfinies ou intégrées, ou d'écrire des fonctions définies par l'utilisateur pour effectuer une tâche spécifique.

Programmation orientée objet (OOP)

Python est un langage orienté objet, donc tout en Python est un objet.

Comprenons ses concepts clés et leur importance :

Classe : Une classe agit comme un modèle pour les objets. Elle est définie à l'aide du mot-clé class comme le mot-clé def est utilisé lors de la création d'une nouvelle fonction. Une classe Python contient des objets et des méthodes, et on peut y accéder en utilisant le point (.).

Objet : Un objet est une instance d'une classe. Il représente la structure de la classe et contient des variables de classe, des variables d'instance et des méthodes.

Méthode : En termes simples, il s'agit d'une fonction définie lors de la création d'une classe.

init : Pour l'initialisation automatique des membres de données par l'attribution de valeurs, vous devez utiliser la méthode init lors de la création d'une instance d'une classe. Elle est appelée à chaque fois que vous créez un objet de la classe. Son utilisation est équivalente à celle du constructeur en C++ et en Java.

Self : il nous permet d'accéder aux méthodes et aux attributs de la classe. Vous êtes libre de lui donner n'importe quel nom ; cependant, par convention et pour des raisons de lisibilité, il est préférable de le déclarer comme self.

Numérique Python (NumPy)

En bref, il s'agit d'un logiciel de traitement de tableaux. NumPy est l'abréviation de Numerical Python. Par défaut, le type de données d'un tableau NumPy est défini par ses éléments. Dans NumPy, la dimension des tableaux est désignée par le rang ; par exemple, un tableau à 2 dimensions signifie un tableau de rang 2. NumPy est populaire en raison de sa rapidité. Lorsque vous travaillez sur un grand ensemble de données, vous devez
voir la différence si vous utilisez NumPy.

Vous pouvez installer le paquetage NumPy à l'aide de pip :

```
pip install numpy
```

Pandas

Pandas est une bibliothèque open-source, sous licence BSD, qui fournit des structures de données et des outils d'analyse de données performants et faciles à utiliser pour le langage de programmation Python. Un DataFrame est une structure de données étiquetée en 2 dimensions avec des colonnes de types potentiellement

différents.
Vous pouvez installer le paquet Pandas à l'aide de pip :
pip install pandas
utilité
loc - sélection par étiquette
Un loc obtient des lignes (et/ou des colonnes) avec des étiquettes spécifiques. Pour obtenir toutes les lignes dans lesquelles une spécificité est établie
iloc - sélection par poste
iloc récupère les lignes (et/ou les colonnes) à l'emplacement de l'index. Pour obtenir la ligne par exemple à l'index 2.
Regrouper les données, puis utiliser une fonction d'agrégation
Enregistrer et charger des fichiers csv, Excel

Principes de base de Git gitlab, GitHub

Il est difficile de maintenir plusieurs versions de fichiers lorsque l'on travaille en équipe. Git résout ce problème en permettant aux développeurs de collaborer et de partager des fichiers avec d'autres membres de l'équipe. Ce chapitre couvre les concepts de base de Git et GitHub les plus nécessaires qui constituent la base de ce livre. Il explique la configuration de Git, la création d'un repo Git, le transfert de votre base de code sur GitHub et le processus de clonage. En outre, il couvre une introduction aux actions GitHub, les commandes Git courantes avec leur syntaxe et des exemples. Maintenir plusieurs versions de fichiers lorsque l'on travaille en équipe peut s'avérer difficile, mais Git est la solution.
Git est un système de contrôle de version distribué (DVCS) open-source. Une version
vous permet d'enregistrer les modifications apportées aux fichiers sur une période donnée.
Git est utilisé pour maintenir les versions historiques et actuelles du code source. Dans un projet, les développeurs disposent d'une copie de toutes les versions du code stockées sur le serveur central.
Git permet aux développeurs d'effectuer les opérations suivantes :
•Suivi des modifications, des personnes qui les ont effectuées et de la date à laquelle elles l'ont été
•Annulation/restauration des modifications
•Permettre à plusieurs développeurs de se coordonner et de travailler sur les mêmes fichiers
•Maintenir une copie des fichiers au niveau local et distant

Git + Hub = GitHub
Git et GitHub sont des entités distinctes. Git est un outil en ligne de commande, tandis que GitHub est une plateforme de collaboration. Vous pouvez stocker des fichiers et des dossiers sur GitHub et apporter des modifications à des projets existants. En créant une branche distincte, vous pouvez isoler ces modifications des fichiers de votre projet existant.
GitHub Actions facilite l'automatisation de tous les flux de travail de vos logiciels grâce à l'intégration continue et au déploiement continu. Vous pouvez construire, tester et déployer le code directement depuis GitHub.

Git + lab = Gitlab

Git et Gitlab sont des entités distinctes. Git est un outil en ligne de commande, tandis que Gitlab est une plateforme de collaboration. Vous pouvez stocker des fichiers et des dossiers sur Gitlab et apporter des modifications à des projets existants. En créant une branche distincte, vous pouvez isoler ces modifications des fichiers de votre projet existant.
Gitlab Actions facilite l'automatisation de tous les flux de travail de vos logiciels grâce à l'intégration continue et au déploiement continu. Vous pouvez construire, tester et déployer le code directement depuis Gitlab.
Commandes courantes de Git
git config --global user.name "[prénom nom]"
Initialiser un répertoire existant en tant que dépôt Git :
git init
Récupérer l'intégralité d'un référentiel à partir d'un site hébergé via une URL :
git clone [url]
git push -u origin master

III Défis liés au déploiement de modèles de ML

Une étude a révélé que 87 % des projets de science des données et d'apprentissage automatique ne sont jamais mis en production. Dans ce chapitre, vous étudierez les problèmes courants que vous pouvez rencontrer lors du déploiement de modèles d'apprentissage automatique.
Le déploiement de modèles de ML dépend entièrement de vos objectifs finaux, tels que la fréquence des prédictions, la latence, le nombre d'utilisateurs, les prédictions uniques ou par lots, et l'accessibilité.

Présentation des différentes approches permettant de déployer des modèles de ML en production. But d'utiliser MLOps pour surmonter les difficultés liées au déploiement manuel des modèles de ML. Etude des différentes phases du cycle de vie du ML.
Le cycle de vie de l'apprentissage automatique est un processus périodique. Il commence par le problème de l'entreprise et se termine par le contrôle et l'optimisation. Cependant, ce n'est pas si simple. Par exemple, en cas de modification des besoins de l'entreprise, il se peut que vous deviez exécuter à nouveau toutes les étapes du cycle de vie de l'apprentissage automatique. Dans certains scénarios, il peut être nécessaire de revenir aux étapes précédentes du cycle de vie du ML pour satisfaire aux critères de cette étape spécifique. Par exemple, si la précision du modèle est inférieure au seuil fixé, vous devez revenir aux étapes précédentes pour améliorer la précision. Cela peut se faire en ajoutant de nouvelles fonctionnalités, en optimisant les paramètres du modèle, etc.

La première étape, la plus importante, consiste à définir une idée et son impact sur le projet. Démarrer un projet sans tenir compte de l'impact sur l'entreprise, de la complexité, des résultats escomptés et du temps nécessaire peut entraîner des retards, des travaux répétitifs et une mauvaise utilisation des ressources.
L'impact sur l'entreprise peut être n'importe quoi, comme une augmentation des revenus, une diminution des dépenses ou une réduction des erreurs humaines. Il faut comprendre les problèmes de l'entreprise et essayer d'évaluer les possibilités d'avoir une solution de ML qui résoudra plusieurs problèmes de l'entreprise. Dans

certains scénarios, il est plus important d'obtenir des résultats rapides.

La collecte de données est le processus qui consiste à recueillir des données à partir d'une ou de plusieurs sources. Bien que cela semble facile, ce n'est pas le cas. Avant de collecter les données, vous devez répondre aux questions suivantes :

•Quelles sont les données à collecter ?
•Quelles sont les différentes sources de données ?
•Quel est le type de données ?
•Quelle est la taille des données ?

Dans le monde réel, vous pouvez être amené à collecter des données à partir de différentes sources, comme les bases de données relationnelles, les bases de données NoSQL, le web, etc. Pour éviter tout problème ou retard, vous devez mettre en place un pipeline pour la collecte des données.

Pour obtenir de meilleurs résultats, vous pouvez combiner les données disponibles avec des données externes, telles que les sites de médias sociaux, les données météorologiques et les données publiques.

Voici quelques moyens de collecter des données :

•Enquêtes électroniques
•Outil de scraping web autorisé
•Cliquez sur les données
•Plateformes de médias sociaux
•Suivi du site web
•Données d'abonnement/ d'enregistrement

Données vocales - service clientèle

Organisez une réunion avec les clients/parties prenantes et les experts du domaine pour connaître les données. Ils vous aideront à comprendre les données afin que vous puissiez décider quelles données doivent être collectées pour l'élaboration du modèle.

Défis :

•Les données sont dispersées à différents endroits
•Pas d'uniformité dans les données
•Combinaison de données provenant de différentes sources
•3V : volume, vitesse, variété
•Stockage des données

Préparation des données

Les données collectées sont généralement brutes. Il est nécessaire de les traiter afin qu'elles puissent être utilisées pour une analyse plus approfondie et le développement d'un modèle. Ce processus de nettoyage, de restructuration et de normalisation est connu sous le nom de préparation des données.

Environ 70 à 80 % du temps est consacré à cette étape du projet de ML. Il s'agit d'une tâche fastidieuse, mais inévitable. La technique de réduction des données est utilisée lorsque vous avez une grande quantité de données à traiter.

La préparation des données vise à transformer les données brutes en un format permettant l'EDA, c'est-à-dire l'analyse des données,

L'analyse exploratoire des données peut être effectuée efficacement pour obtenir des informations.

Défis :

•Valeurs manquantes
•Valeurs aberrantes

•Format de données différent
•Normalisation des données
•Bruit dans les données

Fonctionnalité ingénierie
À ce stade, vous préparez les données d'entrée qui peuvent être introduites dans le modèle, ce qui facilite le modèle d'apprentissage automatique en dérivant des caractéristiques significatives, des transformations de données, etc.

Supposons que vous combiniez les caractéristiques pour en créer une nouvelle qui pourrait aider les modèles ML à mieux comprendre les données et à identifier des modèles cachés. Par exemple, vous mélangez du sucre, de l'eau et du jus de citron pour faire de la limonade plutôt que de les consommer séparément.
Voici quelques techniques d'ingénierie des caractéristiques :
•Codage des étiquettes
•Combiner des caractéristiques pour en créer une nouvelle
•Encodage à chaud
•Imputation
•Mise à l'échelle
•Suppression des caractéristiques indésirables
•Transformation logarithmique
Défis :
•Manque de connaissances dans le domaine
•Création de nouvelles caractéristiques à partir de l'ensemble des caractéristiques existantes
•Sélection de caractéristiques utiles à partir d'un ensemble de caractéristiques

Construire et former le modèle
La construction d'un modèle de ML nécessite que les étapes précédentes soient franchies avec succès. Tout d'abord, vous devez disposer des ensembles de formation, de test et de validation (pour les algorithmes supervisés). Après la création d'un modèle de base, celui-ci peut être comparé à de nouveaux modèles. Ce processus implique le développement de plusieurs modèles afin de déterminer lequel est le plus efficace et donne les meilleurs résultats. Vous devez tenir compte des besoins en ressources informatiques pour cette étape.
Une fois que le modèle final est construit sur les données d'apprentissage, l'étape suivante consiste à vérifier son efficacité.
performance par rapport aux données non vues, c'est-à-dire les données d'essai.
Défis :
•Complexité du modèle
•Puissance de calcul
•Identifier un modèle approprié
•Temps d'apprentissage du modèle36

Tester et évaluer
C'est ici que vous élaborerez les cas de test et que vous vérifierez les performances du modèle par rapport aux nouvelles données. Les activités de pré-production ou de pré-déploiement sont réalisées ici. Les résultats des performances sont analysés et, si nécessaire, vous devez revenir aux étapes précédentes pour résoudre le problème.
Une fois cette étape franchie, vous pouvez faire passer le modèle ML en phase de

production.
Défis :
•Données d'essai insuffisantes
•Répéter le processus jusqu'à ce que le résultat réponde aux exigences.
•Identifier la plateforme permettant d'évaluer la performance du modèle sur des données réelles
•Décider du test à utiliser
•Enregistrement et analyse des résultats des tests

Modèle déploiement
Il s'agit d'exposer les modèles de ML formés aux utilisateurs du monde réel. Votre modèle fonctionne bien sur les ensembles de test et de validation, mais si un autre système ou d'autres utilisateurs ne peuvent pas l'utiliser, il ne remplit pas son objectif.
Par exemple, vous avez construit un modèle pour prédire les clients susceptibles de se désabonner, mais le travail n'est pas terminé tant que vous n'avez pas déployé un modèle qui commencera à fournir les prédictions sur des données réelles.
Le déploiement du modèle est crucial car vous devez prendre en compte les facteurs suivants :
•Nombre de fois où les prédictions doivent être livrées
•Temps de latence des prédictions
•Architecture du système de ML
•Coût de déploiement et de maintenance du modèle ML
•Complexité de l'infrastructure
•Ce point sera abordé en détail dans les chapitres suivants.
Défis :
•Questions de portabilité
•Problèmes d'évolutivité
•Défis liés aux données
•Menaces pour la sécurité

Surveillance et optimisation de
Enfin, la surveillance et l'optimisation constituent l'étape où l'observation et le suivi sont nécessaires. Vous devrez vérifier les performances du modèle au fur et à mesure qu'il se dégrade avec le temps.
Si les mesures du modèle, telles que la précision, sont inférieures au seuil prédéfini, il convient d'en assurer le suivi et d'entraîner à nouveau le modèle, automatiquement ou manuellement. De même, les données d'entrée doivent être surveillées, car il peut arriver que le schéma des données d'entrée ne corresponde pas ou qu'il contienne des valeurs manquantes.
En outre, les paramètres de l'infrastructure, tels que la mémoire vive, l'espace libre et les problèmes de système, doivent être suivis.
Il est bon de conserver les enregistrements des métriques, des résultats intermédiaires, des avertissements et des erreurs.
Défis :
•Dérive des données
•Décider de la valeur seuil pour différentes mesures
•Anomalies
•Finaliser les mesures d'évaluation du modèle qui doivent être suivies

Types de déploiement du modèle

Il existe plusieurs façons de déployer des modèles de ML en production, mais il n'y a pas de méthode générique. Cette section présente les méthodes les plus courantes de déploiement des modèles de ML.

Lot prédictions

Il s'agit de la méthode la plus simple. Ici, le modèle de ML est entraîné sur des données statiques pour faire des prédictions, qui sont sauvegardées dans la base de données, par exemple MS-SQL, et peuvent être intégrées dans des applications existantes ou consultées par l'équipe d'intelligence économique.

En général, les artefacts des modèles ML sont utilisés pour faire des prédictions car ils permettent de gagner du temps. L'artefact du modèle doit être mis à jour en fonction des nouvelles données afin d'améliorer les prédictions.

Cette méthode est bien adaptée aux petites organisations et aux débutants. Vous pouvez programmer la tâche cron pour qu'elle fasse des prédictions après certains intervalles de temps.

Pour :

•Abordable
•Moins complexe
•Facile à mettre en œuvre

Cons :

•Temps de latence modéré
•Ne convient pas aux organisations centrées sur le ML

Service web/REST API

Le service web/API REST est la méthode la plus répandue pour déployer les modèles. Contrairement aux prédictions par lots, il ne traite pas un ensemble d'enregistrements, mais un seul enregistrement à la fois. En temps quasi réel, il prend les paramètres des utilisateurs ou des applications existantes et fait des prédictions.

Il peut prendre des entrées et renvoyer les sorties au format JSON. JSON est un format populaire et compatible qui permet aux développeurs de logiciels ou de sites web de l'intégrer facilement dans des applications existantes.

Lorsqu'un événement est déclenché, l'API REST transmet les paramètres d'entrée au modèle ML et renvoie les prédictions.

Pour :

•Facile à intégrer
•Flexible
•Économique (plan de paiement à l'utilisation)
•Prédictions en temps quasi réel

Cons :

•Problèmes d'évolutivité
•Susceptible de faire l'objet de menaces de sécurité

Défis liés au déploiement de modèles dans l'environnement de production

Le déploiement de modèles de machines en production est crucial. Les défis techniques ne sont pas les seuls à se poser lors du déploiement des modèles ML. Voici les défis les plus courants que vous pouvez rencontrer.

Défis liés aux données

Habituellement, les scientifiques des données développent des modèles sur une quantité limitée de données dans la phase d'expérimentation, mais ils peuvent être confrontés à des défis lors du déploiement des modèles dans l'environnement de production, car les données à grande échelle peuvent avoir un impact sur les

performances du modèle.
Les nouvelles données ne cessent de modifier leur comportement, mais l'étape de préparation/pré-traitement des données peut ne pas être en mesure de gérer ces nouveaux problèmes. Par exemple, des enregistrements de chaînes de caractères sont présents dans une colonne numérique. Parfois, il est nécessaire de relancer les expériences en corrigeant les problèmes pour obtenir des résultats fiables.
Portabilité
Les problèmes de portabilité se posent lorsque vous déplacez la base de code de votre machine locale vers le serveur et vice versa. Supposons, par exemple, que vous ayez développé un modèle et que vous souhaitiez déplacer le code vers le serveur pour le réexécuter. Cependant, vous risquez de rencontrer des problèmes, comme l'incompatibilité de la bibliothèque avec la version actuelle du système d'exploitation, ou le fait que la ligne de code fonctionne avec la version x de Python et non avec la version actuellement installée.
Dans ce cas, il est nécessaire de créer un environnement virtuel et de réinstaller les paquets.
Évolutivité
Supposons que vous ayez déployé des modèles à l'aide d'un service web et que 100 fois plus d'utilisateurs tentent d'accéder à la même API. L'API REST peut cesser de répondre ou la latence peut augmenter.
La gestion des modèles de ML à petite échelle est relativement facile par rapport aux grandes organisations, où plusieurs modèles sont servis à l'échelle. Le déploiement de modèles à grande échelle est encore une nouveauté pour de nombreuses organisations.

Robustesse
En bref, la robustesse des modèles ML est mesurée par la fiabilité des résultats obtenus malgré les variations des données d'entrée. Dans le monde réel, il est pratiquement impossible d'obtenir une précision de 100 %. On peut donc dire que l'erreur de prédiction sur des données non vues doit être proche de l'erreur d'apprentissage.
Le comportement des utilisateurs peut être imprévisible. Par exemple, supposons qu'un modèle exige une valeur de type chaîne de caractères, mais que l'utilisateur fournisse une valeur alphanumérique. Dans ce cas, les résultats du modèle risquent de ne pas être fiables.
Sécurité
Les systèmes de ML sont sujets à des failles de sécurité. Ils peuvent être contraints de fournir de fausses prédictions en fournissant délibérément des données toxiques ou en ajoutant du bruit aux données. Les modèles sont plus performants lorsqu'ils sont entraînés sur de nouvelles données. Toutefois, une personne ou un groupe de personnes peut délibérément transmettre des données toxiques au modèle, dans l'intention de modifier les prédictions du modèle.
La sécurité des données est menacée car le modèle s'entraîne en permanence sur de nouvelles données. Lors de l'entraînement du modèle, les attaquants peuvent voler des informations sensibles.
MLOps
MLOps combine les processus d'apprentissage automatique et les meilleures pratiques de DevOps pour fournir des résultats cohérents avec des pipelines et une gestion automatisés. MLOps est un espace émergent dans l'industrie depuis quelques années.
MLOps comble le fossé entre les expériences d'apprentissage automatique et le déploiement des modèles dans l'environnement de production. Aujourd'hui, de

nombreux scientifiques des données sont contraints d'exécuter les processus MLOps manuellement.
Le processus MLOps fait intervenir plusieurs professionnels, et les scientifiques des données jouent un rôle essentiel. Les experts
recueillent les exigences du client. Ensuite, les ingénieurs de données collectent les données à partir de sources multiples et exécutent les travaux ETL. Une fois cette étape franchie, les data scientists construisent les modèles et les équipes DevOps construit le pipeline CI/CD et le surveille. Enfin, le retour d'information est envoyé au scientifique des données ou à l'équipe concernée pour validation.
Les MLOps rationalisent et automatisent ce processus afin d'accélérer les livraisons et de mettre en place des systèmes de gestion efficaces.
produits/services.
MLOps est une combinaison de trois disciplines :
Machine learning
devops
data engineering

MLOps est différent de DevOps parce que le code est généralement statique dans ce dernier cas, mais ce n'est pas le cas dans les MLOps.
Dans le cadre du MLOps, le modèle s'entraîne en permanence sur de nouvelles données, de sorte qu'une nouvelle version du modèle est générée de manière récurrente. Si elle répond aux exigences, elle peut être transférée dans l'environnement de production. C'est pourquoi le MLOps nécessite une formation continue (CT) ainsi qu'une intégration continue (CI) et une livraison/déploiement continus (CD).
Dans le cadre de DevOps, les développeurs écrivent le code en fonction des exigences et le diffusent ensuite dans l'environnement de production, mais dans le cas de l'apprentissage automatique, les développeurs doivent d'abord collecter les données et les nettoyer. Ils écrivent le code pour l'élaboration du modèle, puis construisent le modèle d'apprentissage automatique. Enfin, ils le diffusent dans l'environnement de production.

Avantages de MLOps
Les processus MLOps permettent non seulement d'accélérer le parcours de l'apprentissage automatique, des expériences à la production, mais aussi de réduire le nombre d'erreurs. MLOps automatise le déploiement des modèles de Machine Learning et de Deep Learning dans un environnement de production. En outre, il réduit la dépendance à l'égard d'autres équipes en rationalisant les processus.

Gestion efficace de la vie de la ML cycle
MLOps prend en charge le cycle de vie de l'apprentissage automatique en suivant les principes fondamentaux des projets d'apprentissage automatique. Il est basé sur une méthodologie agile. Un pipeline CI/CD automatisé permet d'accélérer le processus de recyclage des modèles sur de nouvelles données, les tests avant le déploiement, la surveillance et les boucles de rétroaction. L'étape actuelle dépend des résultats des étapes précédentes, ce qui réduit les risques de problèmes dans le processus de déploiement.
Si le volume de trafic ou le nombre de demandes augmente pour un modèle déployé, les ressources requises augmentent ; de même, les ressources requises diminuent lorsque le volume de trafic ou le nombre de demandes diminue.
Il existe de nombreuses plateformes (comme GitHub Actions) disponibles sur le marché qui aident à
vous configurez le flux de travail de MLOps.

Reproductibilité
Développeur : Cela fonctionne sur ma machine.
Le directeur : Ensuite, nous allons expédier votre machine au client.
•La reproductibilité joue un rôle crucial dans le cycle de vie du ML. Lors du transfert des fichiers de code vers une autre machine ou un autre serveur, la reproductibilité réduit le temps de débogage. MLOps fonctionne selon les principes DRY (Don't Repeat Yourself) et vous permet d'obtenir des résultats cohérents.
•Avec les mêmes données d'entrée, le flux de travail répliqué doit produire un résultat identique. Pour ce faire, les développeurs utilisent des outils d'orchestration de conteneurs tels que Docker afin de créer et de mettre en place le même environnement avec les dépendances sur une autre machine ou un autre serveur pour un résultat cohérent.

Automatisation

La plupart du temps, les développeurs déploient les modèles plusieurs fois avant le déploiement final pour s'assurer que tout est en place et que le résultat est conforme aux attentes. Sans automatisation, ce processus serait long et fastidieux.

L'automatisation augmente la productivité, car il est moins probable que vous testiez, déployiez, mettiez à l'échelle et surveilliez les modèles de ML manuellement.

À chaque étape, certaines règles et conditions sont mises en œuvre, et le modèle ne passe à l'étape suivante que lorsque ces conditions sont remplies. Cela réduit la participation active d'autres équipes chaque fois que vous envisagez de le déployer en production.

Il n'y a pas ou peu de retard dans le déploiement lors des tests, car la dépendance entre les équipes est réduite. Après avoir apporté les modifications nécessaires au code, un déploiement rapide

est effectué de manière automatisée (s'il passe les tests et remplit les conditions). Cela permet d'augmenter la productivité globale de l'équipe.

Suivi et rétroaction boucle

Le suivi des performances du modèle, des métriques, des résultats des tests et de la production devient facile si vous configurez correctement le flux de travail de MLOps. Les performances du modèle peuvent se dégrader avec le temps, d'où la nécessité d'entraîner à nouveau le modèle sur de nouvelles données.

Grâce au suivi et aux boucles de rétroaction, il envoie des alertes sur les performances du modèle, les mesures, etc.

Par exemple, si la boucle de rétroaction envoie l'information selon laquelle la précision du modèle est tombée en dessous de 68 %, le modèle doit être réentraîné. La boucle de rétroaction vérifie à nouveau les performances du modèle. Si elle est supérieure au seuil fixé, elle passe à l'étape suivante ; dans le cas contraire, le modèle doit être recalibré à l'aide des données les plus récentes.

Conclusion

Dans ce chapitre, vous avez étudié les principaux défis à relever lors du déploiement de modèles de ML en production. Au début de ce chapitre, vous avez appris comment fonctionne le cycle de vie du ML, et vous avez compris ses différentes étapes et leurs défis. Ensuite, vous avez été exposé à différents types de techniques de service de modèle et vous avez examiné leurs avantages et leurs inconvénients. Ensuite, vous avez analysé les défis auxquels vous pouvez être confronté lors du déploiement de modèles de ML en production. Enfin, vous avez appris les avantages des MLOps.

Dans le chapitre suivant, vous apprendrez à développer, construire et installer des packages python personnalisés pour les modèles ML à l'aide d'un cas d'utilisation.

Points à retenir

•MLOps comble le fossé entre l'étape des expériences d'apprentissage automatique et le déploiement du modèle dans l'environnement de production.

•La stratégie de déploiement du modèle dépend des besoins de l'entreprise, des utilisateurs et des applications.

•MLOps combine les processus d'apprentissage automatique et les meilleures pratiques de DevOps pour fournir des résultats cohérents avec des pipelines et une gestion automatisés.

IV Emballage ML Modèles

•Environnement virtuel
•Dossier d'exigences
•Sérialisation et désérialisation des modèles ML
•Tester le code Python à l'aide de pytest
•Emballage Python et gestion des dépendances
•Développer, construire et déployer des packages ML
•Configurer les variables d'environnement et les chemins d'accès
*

Environnements virtuels
Lorsque vous travaillez sur plusieurs projets, le projet A nécessite la version 2.6 du paquetage Y, tandis que le projet B nécessite la version 2.8 du paquetage Y. Dans ce cas, vous ne pouvez pas conserver les deux versions d'un même paquetage au niveau mondial. Dans ce cas, il n'est pas possible de conserver globalement les deux versions d'un même paquet.
L'environnement virtuel est la solution. Il vous permet d'isoler les dépendances pour chaque projet. Vous pouvez créer l'environnement virtuel n'importe où et y installer les paquets nécessaires.

Le fichier requirements contient la liste des paquets qui peuvent être installés à l'aide de pip.
Pour créer le fichier d'exigences :
pip freeze > requirements.txt
Pour installer la liste des paquets du fichier d'exigences, vous pouvez utiliser ce qui suit
commande :
pip install -r requirements.txt
Où -r fait référence à --requirement.
Elle demande à pip d'installer tous les paquets du fichier d'exigences donné.

Sérialisation et désérialisation des modèles ML
La sérialisation est un processus par lequel une hiérarchie d'objets Python est convertie en un flux d'octets, tandis que la désérialisation est l'opération inverse, c'est-à-dire qu'un flux d'octets (provenant d'un fichier binaire ou d'un objet de type octet) est reconverti en une hiérarchie d'objets. En Python, la sérialisation et la désérialisation se réfèrent respectivement au décapage et au dépicage.
Ici, joblib.dump() et joblib.load() seront utilisés en remplacement de pickle.dump() et pickle.load respectivement, pour travailler efficacement sur des objets Python arbitraires contenant de grandes données, de grands tableaux NumPy en particulier, comme illustré ici.
Importer la bibliothèque joblib :
1. import joblib
Créez ensuite un objet à persister :
.dump(
,
)
Un objet peut être rechargé :
.load(
)

Tester le code Python avec pytest

Tester votre code vous assure qu'il donne les résultats escomptés et que ses fonctions fonctionnent sans bug. L'outil pytest vous permet de construire et d'exécuter des tests en toute simplicité. Il est doté d'une syntaxe simple et de plusieurs fonctionnalités.

Emballage de Python et gestion des
dépendances
Supposons que vous ayez créé le modèle ML final dans un bloc-notes Python. Ce cahier Python
contient toutes les étapes depuis le chargement des données jusqu'à la prédiction des données de test.
Cependant, cet ordinateur portable ne doit pas être utilisé dans un environnement
•Difficile à déboguer
•Nécessité d'apporter des modifications à plusieurs endroits
•Beaucoup de dépendances
•Pas de modularité dans le code
•Conflit de variables et de fonctions
•Extraits de code dupliqués
Python est un langage de programmation modulaire. La programmation modulaire est une approche de conception dans laquelle le code est divisé en fichiers distincts, de sorte que chaque fichier contient tout ce qui est nécessaire à l'exécution d'un élément logique défini et peut renvoyer la sortie attendue lorsqu'il est importé par d'autres fichiers qui leur serviront d'entrée. Ces fichiers séparés sont appelés modules.
Module
Un module est un fichier Python qui peut contenir des classes, des fonctions et des variables. Par exemple, un module est un fichier Python qui peut contenir des classes, des fonctions et des variables,
load.py est un module, et son nom est load.
Paquet
Un paquet contient un ou plusieurs modules (pertinents), de sorte qu'ils sont liés les uns aux autres. Un paquet peut contenir un sous-paquet qui contient les modules. Il utilise la hiérarchie de fichiers intégrée des répertoires pour faciliter l'accès. Un répertoire avec des sous-répertoires peut être appelé paquetage s'il contient le fichier init . py.
Les paquets seront installés dans l'environnement de production dans le cadre du déploiement. Par conséquent, avant de procéder à l'installation des paquets, vous devez avoir des réponses aux questions suivantes concernant le déploiement :
•Qui sont les utilisateurs de votre application ? Votre application sera-t-elle installée par d'autres développeurs chargés du développement de logiciels,
par le personnel d'exploitation d'un centre de données ou par un groupe moins averti en matière de logiciels ?
•Votre application est-elle destinée à fonctionner sur des serveurs, des ordinateurs de bureau, des clients mobiles (téléphones) ?
tablettes, etc.), ou intégrés dans des dispositifs dédiés ?
Configurer les variables d'environnement et les
chemins d'accès à
Il peut être nécessaire d'ajouter le chemin d'accès aux variables d'environnement.
Il vous permet d'importer des modules et des fonctions :
Ouvrez le fichier .bashrc à l'aide du terminal
sudo vi ~/.bashrc
Points à retenir

•Le paquetage Python contient plusieurs modules, et chaque module contient des fonctions, des classes et des variables.
•Le fichier MANIFEST.in contient la liste des fichiers à inclure et à exclure.
•Les fichiers requirements.txt contiennent la liste des paquets à installer à l'aide de pip.
•Python fait référence à la sérialisation et à la désérialisation par les termes pickling et unpickling, respectivement.
•Le fichier Init . py indique que le répertoire doit être traité comme un paquet.

V Gérer le cycle de vie du ML

Introduction
Le cycle de vie de l'apprentissage automatique comporte de nombreux défis. Par exemple, les scientifiques des données doivent essayer différents modèles contenant de multiples paramètres et hyperparamètres. Ils doivent garder une trace du modèle qui fonctionne bien et de ses paramètres. Enfin, ils doivent sauvegarder le modèle sérialisé pour pouvoir le réutiliser. Ce chapitre explique le rôle de MLflow dans un cycle de vie ML. MLflow est une plateforme permettant de rationaliser le développement de l'apprentissage automatique, y compris le suivi des expériences, l'empaquetage du code dans des exécutions reproductibles, ainsi que le partage et le déploiement des modèles. Elle peut gérer un cycle de vie complet de l'apprentissage automatique.

•Introduction à MLflow
•Suivi du Mlflow
•Projets MLflow
•Modèles MLflow
•Registre des modèles Mlflow

entraîner, de réutiliser et de déployer des modèles ML en utilisant Mlflow

MLflow est une plateforme open-source pour la gestion de bout en bout de l'apprentissage automatique.

MLflow permet aux scientifiques des données d'effectuer autant d'expériences qu'ils le souhaitent avant de déployer le modèle en production ;
Il suit également les hyperparamètres utilisés lors de la construction du modèle. Il vous permet de sauvegarder le modèle entraîné ainsi que ses meilleurs hyperparamètres. Enfin, il vous permet de déployer un modèle de ML dans un serveur de production ou un nuage. Vous pouvez même garder une trace des modèles utilisés dans la phase d'essai et dans la phase de production afin que les autres membres de l'équipe puissent être au courant de ces informations.
MLflow est agnostique, c'est-à-dire que vous pouvez utiliser n'importe quelle bibliothèque ML populaire. De plus, vous pouvez utiliser n'importe quel langage de programmation car les fonctions de MLflow sont accessibles via l'API REST et l'interface de ligne de commande (CLI).
L'intégration de MLflow à votre code existant est assez facile car elle ne nécessite que des changements minimes. Si vous travaillez sur un système local, il créera automatiquement un répertoire mlrun, dans lequel il stockera la sortie, les artefacts et les métadonnées. Il crée un répertoire distinct pour chaque exécution.

Cependant, vous pouvez spécifier le chemin pour créer le répertoire mlrun.
MLflow vous permet de stocker les informations de chaque exécution dans des bases de données, telles que MySQL ou PostgreSQL.
MLflow est plus utile dans les scénarios suivants :
•Comparaison de différents modèles :
MLflow offre une interface utilisateur qui permet aux utilisateurs de comparer différents modèles. Vous pouvez comparer Random Forest et Logistic regression côte à côte, avec la métrique de leur modèle et les paramètres utilisés. MLflow supporte une large gamme de modèles.
• Déploiement cyclique du de modèle :
En production, il est nécessaire de pousser une nouvelle version du modèle après chaque changement de données, lorsque de nouvelles exigences apparaissent, ou après avoir construit un modèle meilleur que le modèle actuel. Dans ces scénarios, MLflow aide à garder une trace des modèles qui sont en staging (pré-production) et des modèles qui sont en production avec des versions et de brèves descriptions.
•Dépendances multiples : Si vous travaillez sur différents projets ou différents frameworks, chacun d'entre eux aura un ensemble différent de dépendances. MLflow vous aide à maintenir les dépendances en même temps que votre modèle.
•Travailler avec une grande équipe de science des données : MLflow stocke les métriques du modèle, les paramètres, l'heure de création, les versions, les utilisateurs, etc. Ces informations sont accessibles aux autres membres de l'équipe travaillant sur les mêmes projets. Ils peuvent suivre toutes les métadonnées en utilisant l'interface MLflow ou une table SQL (si vous les stockez dans la base de données).

Utilisation installation conda
create env conda
pip install mlflow
le projet MLflow utilise conda pour l'installation des dépendances

MLflow composants
MLflow se compose de quatre éléments :
•Suivi du MLflow
•Projets MLflow
•Modèles MLflow
•Registre MLflow
Ces composants sont conçus de manière à pouvoir fonctionner ensemble de manière transparente ; toutefois, vous êtes libre d'utiliser les composants individuels selon vos besoins.

MLflow tracking
Le suivi MLflow est une API et une interface utilisateur permettant d'enregistrer les paramètres, les versions de code, les métriques et les artefacts lors de l'exécution de votre code d'apprentissage automatique et de visualiser les résultats.
MLflow saisit les informations suivantes sous la forme d'exécutions, où chaque

exécution
signifie exécuter un bloc de code :
•Heure de début et de fin : Il enregistre les heures de début et de fin d'une expérience.
•Source : Il peut s'agir du nom du fichier de lancement de l'exécution ou du projet ML.
nom.
•Les paramètres : Ils contiennent les données sous forme de paires clé-valeur. Ce ne sont rien d'autre que les paramètres d'entrée du modèle que vous souhaitez capturer, tels que le nombre d'arbres utilisés dans un algorithme de forêt aléatoire. Par exemple, n_estimators est une clé, et sa valeur est 100. Vous devez appeler la fonction log_param() de MLflow pour stocker les paramètres.
•Métriques : Une métrique est utilisée pour mesurer la performance du modèle, par exemple sa précision. Elle contient les données dans une paire clé-valeur ; cependant, la valeur ne doit être que numérique. Vous devez appeler la fonction log_metric() de MLflow pour stocker la métrique.
•Artéfacts : Lorsque vous souhaitez stocker un fichier ou un objet (tel qu'un fichier pickle du modèle entraîné), la fonction des artefacts vient à la rescousse. Vous pouvez stocker un modèle entraîné sérialisé, un tracé ou un fichier CSV à l'aide de cette fonction, qui peut être appelée à l'aide de log_artifacts().
Tout d'abord, vous devez stocker le fichier ou l'objet dans le répertoire local et, à partir de là, vous pouvez enregistrer le fichier ou l'objet en indiquant le chemin d'accès à ce répertoire.

Projets MLflow
Une fois que vous avez terminé la phase d'expérimentation, votre prochaine étape sera d'empaqueter tout le code en tant que projet avec ses dépendances. Imaginons que vous souhaitiez déplacer la base de code et les dépendances sur le serveur ou sur une autre machine ; MLflow fera le travail pour vous.
MLflow vous permet d'empaqueter la base de code et ses dépendances pour la rendre reproductible et réutilisable. Les projets MLflow fournissent des API et des CLI qui vous aideront à intégrer votre modèle dans MLOps.
Vous pouvez exécuter le projet MLflow directement à partir du dépôt git distant (à condition qu'il contienne tous les fichiers nécessaires) ; vous pouvez également l'exécuter à partir de la CLI locale.
Les champs du fichier MLproject sont les suivants :
•Nom : Il s'agit du nom du projet, qui peut être n'importe quel texte.
•Environnement : Il s'agit de l'environnement qui sera utilisé au moment de l'exécution de la commande du point d'entrée. Il contient les dépendances/paquets requis par le point d'entrée ou le projet Mlflow.
•Point d'entrée : La section point d'entrée contient la commande à exécuter dans l'environnement du projet MLflow. Cette commande peut recevoir des arguments ; il s'agit d'un champ obligatoire qui ne peut pas être laissé vide.
•Paramètres : Cette section contient un ou plusieurs arguments qui seront utilisés par les commandes du point d'entrée, mais elle est facultative.

MLflow modèles
Le module MLflow models vous permet d'empaqueter le modèle de différentes manières, telles que python function, Scikit-learn (sklearn), et Spark MLlib (spark). Cette flexibilité vous aide à connecter les outils associés en aval sans effort.
Lorsque vous enregistrez le modèle en utilisant mlflow.sklearn.log_model(model, name), un répertoire de modèle est créé, et il stocke les fichiers et les métadonnées associés aux modèles. Vous verrez la structure de répertoire suivante :

```
Régression logistique/
├── conda.yaml
├── MLmodel
├── model.pkl
└── requirements.txt
```

Registre MLflow
Le registre MLflow est une plateforme de stockage et de gestion de modèles ML par le biais d'une interface utilisateur et d'un système de gestion des données.
un ensemble d'API.
Il permet de suivre l'évolution du modèle, les différentes versions et les transitions des modèles d'un état à l'autre, par exemple de la mise en scène à la production. Chaque membre autorisé de l'équipe peut suivre toutes les informations précédentes.

Conclusion
Ce chapitre explique l'importance du MLflow dans le cycle de vie du ML et dans l'environnement de production. On a exploré le rôle, la fonctionnalité et l'utilisation de MLflow, avec des exemples de quatre composants de MLflow. MLflow aide les développeurs en science des données à différentes étapes du cycle de vie du ML. Le suivi de MLflow permet de choisir le meilleur modèle en gardant une trace des métriques du modèle, des hyperparamètres utilisés et d'autres informations utiles. Le composant de projet MLflow vous aide à gérer les dépendances et peut être exécuté à partir du dépôt GitHub. Le registre MLflow sert d'emplacement central pour les modèles enregistrés et la modification des états, tels que staging et production.
Dans le chapitre suivant, vous apprendrez à utiliser un docker pour la portabilité lors du transfert de projets ML d'une machine à l'autre ou vers le serveur.
Points à retenir
- MLflow vous aide à passer de la phase d'expérimentation à la phase de déploiement d'un projet de ML.
- À l'exception du registre des modèles, tous les composants peuvent être utilisés sans être intégrés à une base de données telle que MYSQL ; toutefois, il est souhaitable de les intégrer à une base de données telle que MYSQL.
- Par défaut, le projet MLflow utilise conda pour l'installation des dépendances ;

Cependant, vous pouvez procéder sans conda en utilisant l'option -no-conda.

●Chaque composant du MLflow est accessible séparément, mais il est possible de les relier pour créer le flux.

VI Docker pour ML

Introduction

Les conteneurs sont une abstraction au niveau de la couche applicative qui regroupe le code et les dépendances. Plusieurs conteneurs peuvent être exécutés sur la même machine et partager le noyau du système d'exploitation. Chaque conteneur en cours d'exécution est considéré comme un processus isolé dans l'espace utilisateur.
Docker automatise les tâches de configuration répétitives et fastidieuses, ce qui permet au développeur d'économiser du temps et des efforts. Docker est une solution complète qui comprend l'interface utilisateur, les API, les CLI et, enfin, la sécurité. Docker fonctionne sous Linux, Windows et Mac OS.

Introduction à Docker

Docker est une plateforme de conteneurisation qui permet d'empaqueter les applications et leurs dépendances sous la forme d'un conteneur. Il garantit que toutes les bibliothèques et dépendances nécessaires sont intégrées dans un environnement isolé afin que l'application fonctionne sans problème dans les environnements de développement, de test et de production. Docker est populaire parmi les développeurs car il est léger, rapide, portable, sécurisé et plus efficace que les machines virtuelles. Une application conteneurisée commence à fonctionner dès que vous lancez le conteneur Docker.
Docker dispose de son propre registre Docker, appelé Docker hub. Le hub Docker permet aux développeurs de stocker et de distribuer des images de conteneurs sur l'internet. Une étiquette d'image permet aux développeurs de différencier les images. Le registre Docker comporte des dépôts publics et privés. Un développeur peut stocker une image de conteneur sur le hub Docker à l'aide de la commande push et la récupérer à l'aide de la commande pull.

Il arrive souvent que le code fonctionne parfaitement sur votre machine, mais qu'il génère une erreur lorsque vous l'exécutez sur une autre machine. Cela arrive aussi bien aux développeurs qu'aux data scientists. La raison peut être n'importe quoi : un système d'exploitation différent, une version différente d'un système d'exploitation, des versions différentes de python ou des problèmes de dépendance. Ainsi, lorsqu'ils sont confrontés à ce problème, ils peuvent finir par passer beaucoup de temps à le résoudre. Avec Docker, vous ne devriez pas rencontrer ces problèmes, car les paquets Docker nécessitent des fichiers, une configuration et des commandes pour un flux continu.

La méthode traditionnelle de déploiement dans des environnements de production prend beaucoup de temps, car il faut déplacer les fichiers nécessaires, installer les dépendances, configurer et enregistrer le résultat manuellement.
Souvent, les développeurs doivent d'abord configurer l'environnement. Cela prend encore plus de temps lorsque vous devez répéter ce processus pour les différentes étapes du projet, telles que le développement, la pré-production et la production. Les scripts de déploiement manuel sont difficiles à gérer.
Avec Docker, vous pouvez placer tous les fichiers nécessaires dans le répertoire et écrire la configuration, la version du système d'exploitation et les commandes à

exécuter séquentiellement dans un fichier Docker. Vous pouvez également connecter les deux conteneurs Docker au même réseau. En outre, vous pouvez utiliser le même fichier Docker pour le développement, la pré-production et la production.
Docker garantit la reproductibilité, la portabilité, la facilité de déploiement, les mises à jour granulaires, la légèreté et la simplicité.

Docker compose
Docker compose permet aux développeurs de configurer et d'exécuter plusieurs conteneurs. Il lit la configuration à partir du fichier docker-compose.yml. Une seule commande docker-compose up permet de démarrer les services et d'exécuter les applications multi-conteneurs. D'autre part, vous pouvez détruire tout cela à l'aide de la commande docker-compose down. Vous pouvez également supprimer les volumes en ajoutant le drapeau – volumes.

Docker objets
Lorsque vous travaillez avec Docker, vous pouvez créer et utiliser des objets Docker tels que des images, des conteneurs, des volumes et des réseaux. Dans ce chapitre, vous allez découvrir certains de ces objets.
Fichier Docker
Dockerfile peut être considéré comme un ensemble de commandes ou d'instructions permettant aux développeurs de construire des images Docker. Ces commandes ou instructions sont exécutées de manière séquentielle. Il s'agit d'un document de texte brut sans extension.
Image Docker
Pour créer un conteneur Docker, il faut créer une image Docker. Elle stocke tout le code et les dépendances nécessaires à l'exécution de l'application et sert de modèle pour l'exécution d'une instance de conteneur. Une image Docker peut être téléchargée sur le hub Docker, d'où elle peut être tirée vers le serveur ou le système pour exécuter le conteneur.

Docker conteneurs
Une instance d'un conteneur est créée lorsque vous exécutez l'image Docker. Vous pouvez utiliser la même image Docker pour exécuter autant de conteneurs que vous le souhaitez. Il s'agit d'un composant important de l'écosystème Docker. Docker exécute les conteneurs dans un environnement isolé.
Une couche supplémentaire, appelée couche de conteneur, est automatiquement créée au-dessus des couches d'images existantes lorsque le développeur exécute un conteneur. Un conteneur Docker possède sa propre couche de lecture et d'écriture, ce qui permet aux développeurs d'apporter des modifications spécifiques à ce conteneur. Supposons que vous fassiez fonctionner trois conteneurs à l'aide de la même image Docker et que vous installiez une autre version du paquetage python dans un conteneur en cours d'exécution. Cela n'affectera pas la version existante du paquet python dans les autres conteneurs.
Docker gère les données à l'intérieur du conteneur Docker à l'aide de Docker Volumes.
Tous les fichiers que vous avez créés dans une image ou un conteneur font partie intégrante du système de fichiers Union. Cependant, le volume de données fait partie du système de fichiers de l'hôte Docker et il est simplement monté à l'intérieur du conteneur.
Il est initialisé lors de la création du conteneur. Par défaut, il n'est pas supprimé lorsque le conteneur est arrêté. Les volumes de données peuvent également être

partagés entre les conteneurs et peuvent être montés en mode lecture seule.
La commande pour vérifier tous les conteneurs en cours d'exécution est docker ps.
La commande pour vérifier tous les conteneurs en cours d'exécution et arrêtés est docker ps -a.
Mode détaché
Pour exécuter Docker en arrière-plan, exécutez le conteneur en mode détaché. Vous pouvez utiliser l'option -d pour exécuter le conteneur en mode détaché.
Conteneur Docker réseau
En règle générale, un hôte Docker comprend plusieurs conteneurs Docker. Les conteneurs Docker doivent également interagir et collaborer avec des conteneurs locaux et distants pour créer des applications distribuées. Le bridge network est l'interface réseau par défaut que Docker Engine attribue à un conteneur.

Créer un fichier Docker
Dockerfile suit une structure simple et facile à comprendre, à savoir # comment, suivi d'une instruction et d'un argument. Il est d'usage d'écrire les instructions en majuscules pour les différencier des arguments.
DE
•L'instruction FROM définit l'image de base pour les instructions suivantes.
•Un fichier Docker valide doit comporter une instruction FROM.
•FROM peut apparaître plusieurs fois dans le fichier Docker.
CMD
•CMD définit une commande par défaut à exécuter lors de la création d'un conteneur.
•CMD n'est pas exécuté lors de la construction d'une image.
•Peut être remplacé au moment de l'exécution.
RUN
Il exécute une commande dans un nouveau calque au-dessus de l'image actuelle et enregistre les résultats.
COPIE
L'instruction COPY copie de nouveaux fichiers ou répertoires à partir de <src> et les ajoute à la base de données <src>.
système de fichiers du conteneur au chemin <dest>.
POINT D'ENTRÉE
Cela vous aide à configurer le conteneur comme un exécutable ; c'est similaire à CMD. Il peut y avoir au maximum une instruction pour ENTRYPOINT ; si plus d'une instruction est spécifiée, seule la dernière sera prise en compte.
WORKDIR <chemin>
Elle définit le répertoire de travail pour les instructions RUN, CMD et ENTRYPOINT qui suivent.
EXPOSER
Cela permet d'exposer les ports réseau du conteneur, sur lesquels il écoutera au moment de l'exécution.
ENV
Cela définira les variables d'environnement <clé> à <valeur> dans le conteneur. Lorsqu'un conteneur est exécuté à partir de l'image résultante, il transmet et conserve toutes les informations à l'application qui s'exécute à l'intérieur du conteneur.

Construire une image Docker
Grâce à la commande docker build, les utilisateurs peuvent automatiser une construction de docker qui exécute plusieurs instructions de ligne de commande à la suite.
La commande docker build construit une image à partir d'un fichier Docker et d'un contexte :
docker build -t ImageName:TagName dir
Où ?
●-t : Balise d'image
●ImageName : Le nom que vous souhaitez donner à votre image
●TagName : La balise que vous voulez donner à votre image
●dir : Le répertoire où se trouve le fichier Docker.
Pour le répertoire actuel, il suffit d'utiliser . (point) :
sudo docker build -t myimage:v1 .
Vérifiez l'image nouvellement créée à l'aide de la commande docker images. L'étape suivante consiste à construire le conteneur à partir de l'image nouvellement créée.
Note : Pour vérifier toutes les commandes exécutées sur cette image, exécutez la commande suivante docker history [Image_id]

Exécuter un conteneur Docker

Un conteneur Docker est une instance d'exécution d'une image docker
l'option docker run
peut exécuter l'image Docker comme suit :
docker run --name test -it myimage:v1

•-it : Elle est utilisée pour indiquer que vous souhaitez exécuter le conteneur en mode interactif.
•--name : permet de donner un nom au conteneur.
•monimage : Il s'agit du nom de l'image qui doit être exécutée.
•v1 : Il s'agit de la balise de l'image.
L'outil docker
inspect [container_id] permet d'obtenir les
informations complètes du conteneur au format JSON.
La commande docker top [container_id] affiche les processus de premier niveau dans un conteneur.

Points à retenir
•
Docker est une plateforme de conteneurisation permettant d'empaqueter des applications et leurs dépendances sous la forme d'un conteneur.

•Docker garantit la reproductibilité, la portabilité, la facilité de déploiement, les mises à jour granulaires, la légèreté et la simplicité.
•Docker gère les données dans le conteneur Docker à l'aide de Docker Volumes.
•Une instance d'un conteneur est créée lorsque vous exécutez l'image Docker.

VII Créer des applications Web de ML Utilisation de l'API

Lorsque vous développez une application web en Python, il est très probable que vous utilisiez un framework. Un framework est une bibliothèque qui facilite la vie du développeur lors de la construction d'applications web évolutives, standard et prêtes à la production.
Ce chapitre commence par les API REST et explique comment utiliser les API pour déployer les modèles ML. Il abordera ensuite différents frameworks web et illustrera enfin les étapes de construction d'une interface utilisateur pour l'API de modèle ML. A la fin de cette partie, vous devriez savoir comment déployer une application web de ML.

REST APIs
REST est un acronyme pour Representational State Transfer. REST est un style architectural créé principalement pour guider le développement et la conception de l'architecture du World Wide Web (WWW). En d'autres termes, un service web ou une API web suivant l'architecture REST est une API REST.
REST est un modèle permettant de créer des API qui peuvent être utilisées pour accéder à des ressources telles que des images, des vidéos, du texte, JSON et XML hébergées sur le serveur. L'API RESTful fournit une plateforme commune pour la communication entre des applications conçues dans différents langages de programmation.
Ce qui est intéressant avec l'API, c'est que le client n'a pas besoin de connaître les opérations internes effectuées du côté du serveur et vice versa. L'API REST traite toutes les données demandées/traitées par l'utilisateur comme une ressource ; il peut s'agir d'un texte, d'une image, d'une vidéo, etc.
L'API REST est sans état, ce qui signifie que le client doit fournir tous les paramètres dans la demande chaque fois que l'API est appelée. Le serveur ne stocke pas les paramètres précédents transmis par le client avec la requête.

FastAPI
FastAPI est un framework web pour le développement d'API RESTful en Python. FastAPI est un framework léger (comparé à Django), facile à installer, facile à coder et pourtant très performant. Il permet le développement d'API REST avec un minimum de code. FastAPI est livré avec une documentation standard et interactive intégrée. Une fois que vous avez développé et exécuté l'API, vous pouvez accéder à la documentation de votre application à l'adresse suivante {API endpoint}/docs ou {API endpoint}/redoc.
Note : FastAPI nécessite Python 3.6 et plus.
Installons FastAPI et uvicorn, une interface de passerelle de serveur asynchrone. (ASGI), pour la production.
pip install fastapi uvicorn

STREAMLIT

Streamlit est une bibliothèque open-source en Python qui permet aux utilisateurs de construire et de partager des interfaces utilisateur attrayantes pour les modèles d'apprentissage automatique. Elle est accompagnée d'une documentation complète permettant d'apprendre et d'explorer. Avec streamlit, vous pouvez ajouter de beaux widgets interactifs pour obtenir les entrées de l'utilisateur avec quelques lignes de code, comme une boîte de sélection déroulante et un curseur pour changer les valeurs.

Selon ses créateurs, streamlit est le moyen le plus rapide de créer des applications de ML et de les déployer sur le cloud. Il s'agit d'un cadre idéal pour déployer des applications de ML à l'aide de Python.

FLASK

Flask est un framework web qui vous permet de construire des applications web en utilisant Python. C'est un framework léger comparé à Django. Il suit l'architecture REST. Vous pouvez développer des applications web simples avec Flask, car il nécessite moins de code de base. Flask est basé sur l'interface WSGI (Web Server Gateway Interface) et le moteur Jinja2.Créer des applications Web de

Pour démarrer une application Flask, vous devez utiliser la fonction run(). Si vous définissez debug=True à l'intérieur de la fonction run(), il devient alors facile de repérer l'erreur. Lorsque vous activez le mode débogage, le serveur redémarre chaque fois que vous apportez des modifications au fichier d'application et que vous le sauvegardez. Si une erreur se produit, la raison en est affichée dans le navigateur lui-même. Cependant, vous ne devez pas utiliser cette fonctionnalité lorsque vous déployez des modèles dans l'environnement de production.

Dans un Flask, vous pouvez appeler des fichiers statiques tels que des fichiers CSS ou JavaScript pour rendre la page web.

de l'application. La fonction route() guide Flask vers l'URL appelée par la fonction.

pip install Flask

Gunicorn

Gunicorn est un serveur d'application permettant d'exécuter une application Python. Gunicorn est compatible WSGI, il peut donc communiquer avec plusieurs applications WSGI. Dans le cas présent, Gunicorn traduit la requête reçue de Ngnix pour l'application Flask et vice versa.

NGINX

NGINX est un serveur web open-source et reverse proxy hautement performant et évolutif. Il peut assurer l'équilibrage de la charge et la mise en cache des instances d'application. Il accepte les connexions entrantes et décide où elles doivent aller ensuite. Dans le cas présent, il est installé au sommet d'une Gunicorn.

Points à retenir

•L'API RESTful fournit une plateforme commune pour communiquer avec un système d'information.

qui ont été conçues dans différents langages de programmation.

•FastAPI nécessite Python 3.6 et plus.
•Streamlit est une bibliothèque open-source en Python qui permet aux utilisateurs de construire et de partager des interfaces utilisateur attrayantes pour les modèles d'apprentissage automatique.
•FastAPI dispose d'une documentation standard et interactive intégrée.
•Les fichiers de configuration de NGINX n'ont pas d'extension et chaque ligne devrait être fermé à l'aide d'un ; (point-virgule).

VIII CI/CD pour ML

Introduction

De nos jours, l'automatisation est omniprésente. En automatisant les processus manuels, vous économisez du temps, des efforts et des coûts. Le meilleur aspect de l'automatisation est qu'elle réduit les erreurs humaines. Vous pouvez automatiser la plupart des parties tout en incorporant de nouvelles modifications ou mises à jour dans l'application.
Le pipeline CI/CD permet de déployer les mises à jour de l'application de manière automatisée. Ce chapitre aborde les éléments du pipeline CI/CD et les moyens de l'exploiter pour les applications ML. Dans ce chapitre, vous découvrirez les différentes étapes du pipeline CI/CD, leur importance dans les MLOps et la manière d'en construire un.

Pipeline CI/CD pour ML

CI/CD est l'acronyme de Continuous Integration/Continuous Delivery/Deployment (intégration continue/livraison continue/déploiement).
L'objectif du pipeline CI/CD est d'automatiser la chaîne d'étapes interconnectées pour déployer une application ou publier une nouvelle version du logiciel.
Lorsqu'une nouvelle fonctionnalité est ajoutée à l'application, toute amélioration doit être intégrée à l'application. Cependant, cela implique différentes équipes qui exécutent de multiples tâches et les valident avant de passer à l'étape de la production. La plupart du temps, il s'agit d'un processus manuel et chronophage, qui peut entraîner un retard dans la publication de la nouvelle version.
Le pipeline CI/CD permet d'automatiser les tests, d'exécuter un code sans erreur pour l'application, d'accélérer les déploiements, de faire gagner du temps et de l'argent aux développeurs, d'assurer une grande fiabilité, etc.
Le pipeline CI/CD vous permet de faire passer rapidement les changements du développement au déploiement, qui se fait généralement en quatre étapes :

•Validation des modifications du code : Après avoir apporté des modifications au fichier ou au code, le développeur transfère les mises à jour dans le référentiel des sources. Cette activité est souvent réalisée en équipe. Le pipeline CI/CD permet à n'importe quel membre de l'équipe de vérifier l'intégrité du code. Il est donc possible de pousser automatiquement les modifications vers le référentiel une fois qu'elles ont passé les tests.

•Construction : Dans cette phase, il récupère les modifications du référentiel pour la construction. Il surveille le référentiel source pour détecter d'éventuelles modifications. Dès qu'il détecte des modifications, il lance le processus de construction et valide les résultats de la construction une fois celle-ci terminée.
•Test : La phase de test exécute les tests automatisés, tels que les tests unitaires, pytest et les tests d'API, au-dessus de la construction. Il s'agit d'une étape essentielle du pipeline CI/CD. Cette phase de garantit l'intégrité globale du code et empêche tout code défectueux de passer à la phase suivante.
• Déploiement : Cette phase permet de déployer les modifications dans l'environnement de production.

Liste d'outils populaires pour le pipeline CI/CD :
•Jenkins
•Actions GitHub
•Bambou
•CircleCI
•GitLab CI/CD
•Travis CI

Intégration continue (CI)
Dans l'intégration continue, l'équipe de développeurs construit, exécute et teste d'abord le code dans des environnements locaux. Si tout se passe bien, ils poussent les mises à jour vers le référentiel. Ensuite, la chaîne d'étapes commence à se dérouler, ce qui implique les étapes de construction, d'exécution et de test. Les membres du projet sont informés à chaque étape et reçoivent des mises à jour opportunes, telles que les résultats de la construction et des tests. Enfin, les artefacts sont stockés et le rapport sur l'état actuel est envoyé par courriel ou notifié via Slack.

Lorsqu'une équipe de développeurs travaille sur la même application, ils poussent les modifications du code vers le référentiel.
Dans un environnement de production, le code peut se casser ou générer une erreur. D'autre part, il peut y avoir un conflit entre les mises à jour, lorsque plusieurs développeurs essaient de pousser les mises à jour de l'application vers le dépôt de code central. Ce problème est pris en charge par l'étape de l'IC, où les développeurs peuvent pousser les changements qui passeront par les étapes définies, telles que la construction, l'exécution et le test, afin de s'assurer que le code fonctionne correctement sans aucun problème. Toutefois, si l'une des étapes de l'IC échoue, vous en serez informé et la suite du processus s'arrêtera. De cette manière, vous pouvez éviter d'intégrer un code défectueux dans la production.
Les développeurs peuvent fréquemment pousser et vérifier le fonctionnement du code, le flux et l'intégrité du code ou des applications avant de les pousser à l'étape suivante pour le déploiement.

Livraison/déploiement continu (CD)
CD fait référence à la livraison continue (Continuous Delivery) ou au déploiement continu (Continuous Deployment) (les termes sont utilisés de manière interchangeable) en fonction du niveau d'automatisation que vous envisagez de mettre en œuvre. L'étape CD dépend de l'étape CI. Une fois l'étape CI terminée, elle déclenche l'étape CD dans le pipeline. L'objectif de l'étape de livraison continue (CD) est de livrer une base de code ou un artefact sans erreur

à l'environnement de pré-production. À ce stade, vous pouvez ajouter une série de cas de test (si nécessaire) pour garantir une construction stable et une application fonctionnelle. Il envoie le rapport sur l'état des tests à l'équipe ou au développeur, puis l'application est poussée manuellement vers l'environnement de production. Si l'une des étapes échoue, il se peut que vous deviez recommencer l'ensemble du processus avec les mises à jour nécessaires.
D'autre part, le déploiement continu va plus loin et déploie rapidement l'application de l'environnement de pré-production à l'environnement de production. Il supprime l'étape du déploiement manuel d'une application dans l'environnement de production. Toutefois, il est facultatif, car il appartient au développeur et à l'équipe opérationnelle de choisir le niveau d'automatisation qu'ils souhaitent mettre en œuvre en fonction de l'activité de l'entreprise et de la nature de l'application.

Formation continue (CT)
Une nouvelle étape a été introduite dans le pipeline CI/CD traditionnel : l'entraînement continu (EC). À ce stade, les modèles doivent être formés en continu à mesure que de nouvelles données arrivent ou qu'un événement se produit, par exemple lorsque la précision du modèle tombe en dessous du seuil acceptable. Cela peut ajouter une légère complexité au pipeline CI/CD, mais c'est essentiel pour le déploiement de l'apprentissage automatique.
Avec la CI/CD, la formation continue (CT) est tout aussi importante dans les MLOps. Le recyclage des modèles dépend de scénarios et de divers autres facteurs, tels que la fréquence de modification des données et le schéma des données d'entrée. Il dépend également d'événements tels que la chute de la précision en dessous d'un seuil acceptable, de périodes spécifiques telles que la fin de chaque semaine, ou de déclencheurs manuels.

Introduction à Jenkins
Jenkins est un outil d'automatisation CI/CD modulaire et open-source, écrit en Java, qui s'accompagne de plusieurs plugins. Jenkins permet de construire, de tester et de déployer en douceur et en continu des applications dont le code a été récemment mis à jour ou développé. Il bénéficie du soutien d'une large communauté et est très apprécié des développeurs.
Si la construction est réussie, Jenkins exécute automatiquement une série d'étapes à partir du référentiel de code et, si tout se passe bien, il déploie l'application sur le serveur.
Voici les principales caractéristiques de Jenkins :
•Jenkins est un outil open-source, gratuit et modulaire.
•Il est créé par des développeurs et pour des développeurs.
•Jenkins est facile à i n s t a l l e r , à configurer et peut être installé sur Linux et MacOS,
et Windows.
•Un grand nombre de plugins Jenkins sont disponibles pour les plateformes cloud les plus courantes.
•Le maître de Jenkins peut répartir la charge entre plusieurs esclaves et permettre un traitement plus rapide.

Construire un pipeline CI/CD en utilisant GitHub, Docker et Jenkins

Ici, Jenkins est utilisé pour automatiser le flux de travail de ML. Tout d'abord, vous devez créer une base de code sur la machine locale et exécuter l'application ML localement pour vous assurer qu'elle fonctionne correctement sur la machine locale. Ensuite, vous pousserez les modifications vers le dépôt GitHub. Ensuite, intégrez le dépôt GitHub et Jenkins par webhook. Jenkins doit être installé sur les serveurs de pré-production ou de production. Jenkins va extraire la dernière base de code du dépôt GitHub lié et déployer l'application ML sur le serveur.

Créer un pipeline CI/CD en utilisant Jenkins

On peut alors construire un pipeline CI/CD simple en utilisant GitHub et Jenkins. Lorsque les développeurs apportent des mises à jour au dépôt GitHub, le webhook GitHub détecte les changements et envoie une notification à Jenkins. Jenkins extrait le code le plus récent du dépôt GitHub, construit l'image Docker et exécute le conteneur à l'aide de cette image. Ensuite, il entraîne le modèle et exporte l'objet pickle du modèle entraîné. À l'étape suivante, les résultats des tests sont exportés et affichés dans Jenkins. Après avoir passé tous les tests, l'application web ML est exécutée. Enfin, le feedback est envoyé par e-mail au développeur ou à l'équipe concernée.

Conclusion

Dans ce chapitre, on a exploré les processus d'intégration continue (CI), de livraison continue (CD), de déploiement continu (CD) et de formation continue (CT) dans le pipeline CI/CD. Vous avez également appris à créer un pipeline CI/CD simple à l'aide de l'outil open-source populaire Jenkins. Vous avez ensuite intégré GitHub et Jenkins à l'aide de GitHub webhook, et vous avez construit une image Docker et exécuté le conteneur en tant que tâche de Jenkins. Vous avez également exécuté et exporté les résultats de pytest à l'aide du plugin JUnit. Dans la dernière étape, une application web ML a été déployée sur le port 8005. Enfin, elle a déclenché l'envoi d'un email avec l'état de la construction, le nom du job et les logs de construction.

Dans le chapitre suivant, vous apprendrez à construire des pipelines CI/CD qui déploient des applications ML sur la plateforme Heroku à l'aide d'Actions GitHub.

Points à retenir

- CD désigne indifféremment la livraison continue ou le déploiement continu, en fonction du niveau d'automatisation que vous envisagez de mettre en œuvre.
- Jenkins est un outil d'automatisation CI/CD modulaire et open-source écrit en Java, qui s'accompagne d'un grand nombre de plugins.
- Jenkins comprend le format XML des rapports de test Junit.

IX Déploiement de modèles de ML sur Google Cloud Platform

Introduction

L'informatique en nuage est la fourniture à la demande de ressources informatiques, telles que les serveurs, les bases de données, l'analyse, le stockage et la mise en réseau. Ces ressources et services sont disponibles en dehors des locaux ; toutefois, il est possible d'y accéder via le nuage (l'internet) en fonction des besoins. Cela signifie qu'il n'est pas nécessaire de mettre en place une grande infrastructure sur place. Les entreprises et les particuliers y trouvent leur compte, car ils obtiennent instantanément les ressources et les services requis dans le cadre d'un plan de paiement à l'utilisation. Les ressources et les services peuvent être ajoutés ou supprimés rapidement, ce qui permet de dépenser l'argent de manière efficace.

Google Cloud Platform (GCP)

Google Cloud Platform (GCP) comprend les services de cloud computing proposés par Google, qui utilisent la même infrastructure que celle utilisée par YouTube, Gmail et d'autres plateformes ou services Google. La plateforme offre une gamme de services pour le calcul, l'apprentissage automatique et l'IA, la mise en réseau, l'IoT et le BigData. Voici quelques services proposés par le GCP :

- Le moteur de calcul de Google fournit des instances VM pour exécuter le code et déployer les applications.
- Des services d'IA et d'apprentissage automatique comme Vertex AI, qui est une plateforme unifiée de gestion du cycle de vie de l'intelligence artificielle de bout en bout. Les scientifiques des données peuvent télécharger les données, construire, former et tester les modèles d'apprentissage automatique en toute simplicité.
- Les modules d'IA, tels que Vision AI, permettent de tirer des enseignements des images à l'aide d'AutoML.
- Les services de conteneurs, tels que Container Registry et Google Kubernetes Engine (GKE), gèrent les images Docker et permettent aux développeurs de créer des applications évolutives.
- BigQuery et data proc pour traiter et analyser de grandes quantités de données.
- Les bases de données, telles que Cloud SQL et Cloud Bigtable, stockent les données dans le nuage.
- Outils pour développeurs, tels que Cloud Build, Cloud Source Repositories et Google Cloud Deploy pour automatiser le processus CI/CD.
- Les outils de gestion, tels que Deployment Manager et Cost Management, vous aident à suivre le déploiement et le coût des outils ou des services utilisés dans les projets.
- Les services de mise en réseau, tels que le nuage privé virtuel (VPC), vous permettent de créer un environnement de nuage privé virtuel au sein d'un nuage public. Plusieurs projets créés dans différentes régions peuvent communiquer entre eux sans passer ouvertement par l'internet public.
- Services de sécurité, tels que la gestion des clés dans le nuage, les pare-feu et la sécurité centre.
- Les services de stockage, tels que le stockage en nuage, vous permettent de stocker des artefacts.
- L'informatique sans serveur, telle que Cloud Function, est une plateforme de calcul sans serveur pilotée par les événements. Cette fonction en tant

que service (FaaS) vous permet d'exécuter le code sans serveur ni conteneur.
•Les services d'exploitation, tels que Cloud Logging et Cloud Monitoring, vous permettent de suivre les performances, les délais et les modèles ou applications déployés.
•Elle fournit également d'autres services, tels que la migration, l'IoT, la gestion des événements, l'identité et l'accès, l'hybride et le multicloud, la sauvegarde et la récupération.
Vous devez avoir compris l'essentiel de GCP et de ses services ; maintenant, vous allez apprendre à déployer le modèle d'apprentissage automatique sur GCP. Après avoir terminé ce chapitre, vous serez en meilleure position pour travailler sur GCP pour les déploiements de modèles de ML.
GCP propose un compte d'essai gratuit de 90 jours avec 300 crédits. Cela permettra aux nouveaux clients d'acquérir une expérience pratique et d'explorer les services offerts par GCP.

Registre des conteneurs
Container Registry est un service de gestion et de stockage d'images de conteneurs privés offert par GCP. Il permet aux utilisateurs de pousser et de tirer des images de conteneurs en toute sécurité. Dans ce cas, l'image de conteneur Docker sera poussée vers le Container Registry. Le moteur Kubernetes tirera l'image pour déployer l'application ML sur le cloud.

Kubernetes
L'orchestration de conteneurs est une automatisation du processus opérationnel nécessaire pour exécuter des charges de travail et des services conteneurisés. Elle automatise des tâches telles que le déploiement, la montée en charge, la gestion du cycle de vie, l'équilibrage de la charge, la configuration, la sécurité, l'allocation des ressources, la surveillance de l'état de santé et la mise en réseau des conteneurs.
Kubernetes (également connu sous le nom de K8s ou Kube) est une plateforme open-source pour l'orchestration de conteneurs. Elle permet à l'application d'évoluer à la volée, sans interrompre l'application en cours d'exécution dans la production. Elle crée rapidement plusieurs répliques d'applications conteneurisées pour faire face à l'augmentation du trafic et réduit automatiquement le nombre de répliques lorsque le trafic diminue.
Le déploiement sur Kubernetes crée des pods avec des conteneurs à l'intérieur. Les pods sont la plus petite unité de l'environnement Kubernetes. Un Pod peut contenir un ou plusieurs conteneurs. Il s'exécute toujours sur un nœud ; toutefois, un nœud peut contenir plusieurs pods. Un nœud n'est qu'un travailleur (VM ou machine physique) dans un environnement Kubernetes. Tous les nœuds sont gérés par un plan de contrôle.
Chaque nœud exécute le Kubelet et le runtime du conteneur, tel que Docker. Le Kubelet est le moyen de communication entre le plan de contrôle et le nœud. Il gère également les pods et les conteneurs en cours d'exécution à l'intérieur du nœud. Le moteur d'exécution de conteneurs, tel que Docker, extrait l'image du registre et exécute l'application conteneurisée.

Google Kubernetes Engine (GKE)
Google Kubernetes Engine (GKE) offre l'infrastructure nécessaire pour gérer, déployer et mettre à l'échelle des applications conteneurisées. L'architecture sous-jacente de GKE consiste en un ensemble d'instances de moteur de calcul assemblées pour former un cluster.
Le manifeste est un fichier (JSON ou YAML) contenant une description de tous les composants que vous souhaitez déployer. Ces fichiers manifestes guident Kubernetes dans la mise en réseau des conteneurs. Kubernetes planifie le déploiement des conteneurs dans les clusters et identifie le meilleur hôte pour le conteneur. Après avoir choisi un hôte, il gère le cycle de vie du conteneur sur la base de spécifications préétablies.

Points à retenir
•Le cluster Kubernetes peut être partagé entre plusieurs projets.
•L'identifiant Gmail est nécessaire pour créer un compte sur GCP.
•Cloud Build est une plateforme sans serveur qui vous permet d'automatiser la construction, le test et le déploiement de conteneurs rapidement.
•Les constructeurs de nuages sont des images de conteneurs dans lesquels sont installés des langages et des outils courants.
•Les mêmes étiquettes doivent être utilisées dans les fichiers service.yaml et deployment.yaml pour les services de
une communication réussie entre le service et le déploiement.

X Surveillance et débogage

Introduction
Jusqu'à présent, vous avez appris diverses techniques pour déployer des modèles de ML en production. Cependant, le déploiement d'un modèle en production n'est pas une fin en soi ; la surveillance est l'étape suivante. Ce chapitre traite des concepts et des techniques de surveillance des modèles. La surveillance du modèle ne se limite pas au point final où le modèle est déployé ; vous devez également l'intégrer dans les étapes intermédiaires, le cas échéant. Le modèle ML statique est entraîné sur la base de données historiques. Cependant, ce modèle peut ne pas fournir des performances constantes à tout moment. Les principales raisons peuvent être des changements dans les données d'entrée, les exigences de l'entreprise et la dégradation du modèle au fil du temps.

Importance de la surveillance du site
Une fois qu'un modèle de ML est déployé en production, il est essentiel de le surveiller afin de s'assurer que les performances du modèle restent à la hauteur et qu'il continue à fournir des résultats fiables de manière transparente. En fait, il existe de nombreuses raisons de défaillance dans les applications ou services de ML, telles que la défaillance du pipeline, la dégradation du modèle au fil du temps, la modification des données d'entrée du modèle, la défaillance du système ou du serveur, et la modification du schéma.
Plusieurs équipes sont impliquées lors du déploiement des modèles en production, ce qui inclut généralement une équipe de Data Scientists, d'ingénieurs de données et de DevOps. Si les erreurs de prédiction augmentent, qui sera tenu pour responsable ? Qui est le propriétaire du modèle en production ?

En raison de COVID-19, les performances du modèle ML des banques ont été fortement affectées. Les prédictions du modèle étaient très éloignées des chiffres réels. C'est le cas d'un changement dans les données d'entrée. Cela devrait vous donner un aperçu des défis posés par le déploiement des modèles en production et de la nécessité de surveiller les modèles de ML.
Le suivi est essentiel lorsqu'il s'agit de comparer les performances des nouveaux et des anciens modèles et leurs prévisions dans le temps. Il peut y avoir des problèmes de latence, c'est-à-dire des retards dans les résultats. Pour suivre et étudier les problèmes de latence, un suivi efficace est nécessaire. Vous devez surveiller les données d'entrée pour vous assurer que les données d'entrée de production sont traitées comme les données de formation. Pour suivre et traiter les valeurs extrêmes, les valeurs hors plage ou les cas particuliers avant de les transmettre aux modèles, une surveillance est nécessaire. Le dernier point, mais non le moindre, est la sécurité du modèle, c'est-à-dire que la surveillance est nécessaire pour repérer toute attaque externe sur le modèle ou le système.
Les objectifs du suivi du ML sont les suivants :

• Détecter les problèmes ou les défaillances à un stade précoce afin que les mesures nécessaires puissent être prises.
• Garder une trace de l'utilisation des ressources et des prédictions du modèle pour évaluer les performances du modèle et du système dans l'environnement de production.
•Détecter les changements dans la distribution, le schéma et les anomalies dans les données d'entrée qui peuvent entraîner une erreur dans les prédictions.
•Garantir la disponibilité des prévisions du modèle et leur explicabilité
•Suivre et stocker les mesures dans une base de données ou un espace de stockage spécifique.
La surveillance des modèles aide les scientifiques des données à réduire les défaillances des modèles, à éviter les temps d'arrêt et à garantir des résultats fiables aux utilisateurs.

Principes fondamentaux de la surveillance ML

La surveillance des ML consiste à suivre les performances, les erreurs, les métriques, etc. des modèles déployés et à envoyer des alertes (le cas échéant) pour s'assurer que les modèles continuent à fonctionner au-dessus d'un seuil acceptable. Elle ne se limite pas au suivi des données d'entrée ou de la dégradation du modèle. Cependant, elle doit prendre en compte tous les éléments susceptibles d'affecter directement ou indirectement les performances du modèle. La surveillance du ML permet de décider si un modèle existant doit être mis à jour.
Les étapes essentielles à prendre en compte sont les suivantes :
•
Le suivi est la clé : Le suivi est essentiel une fois qu'un modèle est déployé en production. Le suivi vous permet de repérer et de résoudre les problèmes ou les erreurs plus rapidement. Le suivi peut être divisé en deux types :
oContrôle fonctionnel : Dans le cas de la ML, le contrôle fonctionnel consiste à suivre les mesures, les erreurs et les performances liées aux modèles de ML, telles que la précision et les valeurs aberrantes.
oSurveillance opérationnelle : La surveillance opérationnelle

consiste à suivre les paramètres spécifiques au système, tels que l'utilisation du processeur et de la mémoire vive, le temps de fonctionnement et le débit.
•Intégration évolutive : La surveillance doit pouvoir s'intégrer facilement à l'infrastructure et au flux de travail existants. Le système de surveillance doit pouvoir être intégré de manière transparente et évolutive. Il doit pouvoir suivre plusieurs modèles s'il y en a plus d'un. La solution de surveillance doit être agnostique, de sorte qu'elle puisse être utilisée pour différents types de déploiement, avec différentes piles technologiques.

•Suivi des mesures : Le suivi de la précision n'est pas suffisant. Les systèmes de surveillance doivent être en mesure de suivre toutes les mesures susceptibles d'affecter les performances du modèle et du système au fil du temps. Un système de contrôle centralisé doit utiliser plusieurs mesures de performance pour donner l'état général de la solution. Utilisez différentes mesures pour différents types de caractéristiques. Par exemple, les valeurs min, max, moyenne, écart type et valeurs aberrantes peuvent être utilisées pour les données numériques. Les enregistrements des mesures et les journaux doivent être stockés dans une base de données afin de pouvoir être analysés ultérieurement.

•Système d'alerte pour les événements importants : Il n'est pas possible de suivre manuellement plus de 100 paramètres 24 heures sur 24 et 7 jours sur 7 ; c'est pourquoi le système d'alerte doit faire partie de la solution de surveillance. Tout ne doit pas faire l'objet d'une alerte. Vous pouvez suivre plus de 20 paramètres, mais seuls quelques-uns d'entre eux doivent faire l'objet d'une attention particulière. Par exemple, si la dérive des données d'entrée dépasse le seuil, le système d'alerte doit envoyer une notification à l'équipe responsable ou aux scientifiques des données. L'objectif du système d'alerte est d'informer les équipes ou les personnes concernées afin que les problèmes ou les erreurs puissent être évités ou résolus au plus tôt pour garantir la cohérence des performances du modèle de ML.
•Analyse de la cause première et débogage : Une fois que vous avez reçu l'alerte, il se peut que vous deviez agir en conséquence. Vous pouvez commencer l'analyse des causes profondes pour déterminer le problème et le résoudre par le débogage. Par exemple, si la précision du modèle passe en dessous du seuil, cela peut être dû à un changement dans la distribution des données d'entrée ou à des anomalies.
Pour concevoir un système de surveillance efficace, il faut tenir compte des canalisations existantes et de l'infrastructure. Le suivi repose principalement sur trois piliers :
•Traitement et stockage : Il traite et stocke les mesures critiques dans une base de données ou une source de données avec un horodatage. Il est possible d'interroger les mesures en cas de besoin.
•Graphiques et tableau de bord : Il interroge ou récupère les mesures de surveillance à partir d'une base de données ou d'une source de données connectée. Vous pouvez choisir la visualisation en fonction du type de mesures. Ici, vous devez décider des mesures qui doivent être affichées. Il doit résumer l'ensemble de la surveillance à l'aide de graphiques intuitifs.
•Alerte : enfin, il doit envoyer une alerte à l'équipe concernée ou aux scientifiques des données afin qu'ils prennent des mesures immédiates, le cas

échéant. Cela permet de prévenir les défaillances futures ou d'en atténuer l'impact.

Avant de mettre en place un système de surveillance, voici la liste des questions auxquelles il faut répondre :

- Que prévoyez-vous de contrôler ?
- Quels sont les outils et le langage utilisés ?
- Quelle plate-forme ou bibliothèque sera utilisée pour le suivi ?
- Comment prévoyez-vous d'intégrer le dispositif de surveillance dans le système existant ?

l'environnement et les outils ?

- Quel est le seuil à utiliser pour les alertes ?
- Quelles sont les mesures à prendre après la détection d'un problème ou d'une défaillance ?

Enfin, conformément au cycle de vie du modèle, la surveillance doit compléter la boucle de rétroaction, c'est-à-dire qu'après avoir détecté un problème ou une défaillance, elle doit envoyer une notification à l'équipe concernée ou aux scientifiques des données afin qu'ils puissent prendre les mesures nécessaires, telles que le réentraînement du modèle ML.

Paramètres pour le suivi de votre système ML

Vous devez décider du type de mesure à surveiller. Les métriques opérationnelles permettront aux développeurs de suivre les défaillances ou les avertissements liés au système ou au serveur, tels qu'une utilisation élevée des ressources susceptible d'augmenter la latence, ce qui affectera l'expérience de l'utilisateur final. Le suivi des mesures opérationnelles est essentiel car le serveur doit être en bon état pour exécuter le modèle ML et d'autres tâches. Un autre facteur est le coût. Les équipes opérationnelles doivent fixer des limites aux coûts et doivent savoir quelle ressource ou quel service est à l'origine des coûts les plus élevés.

Voici quelques indicateurs opérationnels importants :

- Système ou serveur
  - Utilisation des ressources
  - Disponibilité
  - Temps de latence
  - Débit
- Coût
  - Coût de l'infrastructure
  - Coût du stockage
  - Service supplémentaire (le cas échéant)

Une fois que le serveur est opérationnel sans problème, vous pouvez vous concentrer sur les métriques fonctionnelles ou de modèle ML. Dans ce chapitre, vous étudierez la détection des métriques de surveillance des modèles et les moyens d'y remédier.

Voici quelques mesures importantes du modèle :

- Données d'entrée
  - Version du modèle
  - Dérive des données
  - Valeurs aberrantes

- Modèle ML
    - Version du modèle
    - Hyperparamètres du modèle
    - Métadonnées
    - Prédictions ou résultats
    - Mesures d'évaluation du modèle de classification
        - Précision
        - Matrice de confusion
        - Score ROC-AUC
        - Scores de précision et de rappel
        - Score F1
    - Mesures d'évaluation du modèle de régression
        - Erreur quadratique moyenne (RMSE)
        - R-carré et R-carré ajusté
        - Erreur absolue moyenne (MAE)
        - Erreur moyenne absolue en pourcentage (MAPE)
- Dérive prévisionnelle

Vous pouvez décider des indicateurs à suivre de près et de ceux qui nécessitent des alertes.

Dérive dans ML
La dérive du modèle signifie un changement dans le comportement des modèles de ML ou que le modèle ne fonctionne pas comme prévu ou selon l'accord de niveau de service (SLA). Les performances du modèle peuvent se dégrader après son déploiement en production, car le modèle peut recevoir des données qui n'ont pas été introduites lors de la formation du modèle.
Types de dérives dans ML
Il existe principalement trois types de dérive en ML :
•Dérive des données
•Dérive prévisionnelle
•Changement de concept

Dérive des données
On parle de dérive des données lorsque la distribution ou les caractéristiques des éléments d'entrée changent par rapport aux données d'apprentissage. Elle est également connue sous le nom de dérive des caractéristiques, dérive de la population ou déplacement des covariables. Si la distribution des données d'entrée change, les prédictions peuvent ê t r e affectées car le modèle n'y est pas préparé.
Si une nouvelle catégorie est ajoutée à la caractéristique après le déploiement, elle peut entraîner une erreur lors de la prédiction, car elle n'existait pas au moment de l'apprentissage du modèle.
Autre exemple : supposons que les données d'apprentissage contiennent une caractéristique appelée "notation de crédit", dont le poids est élevé, ce qui signifie qu'une modification de cette caractéristique peut entraîner un changement majeur dans les résultats du modèle. L'entreprise a décidé d'opter pour la notation de crédit de Moody's plutôt que pour celle de S&P. Cela entraînera une modification des données d'entrée. Par exemple, la note AA- de S&P est équivalente à la note Aa3 de Moody's.
L'équation suivante montre que la distribution des données de formation ne

correspond pas à la distribution des données de référence (production).
Mathématiquement, la dérive des données peut être définie comme suit :
P(H) s Pref(H)
Où P(X) représente la distribution de probabilité des
données d'entrée. Dérive de prédiction
Lorsque les données d'entrée changent avec la dérive des données, cela peut entraîner une modification de la variable cible ou de la variable de prédiction. Il s'agit d'un changement dans les prédictions au fil du temps. On parle également de changement de probabilité préalable, de dérive des étiquettes ou de changement de classe inconditionnel. Ce phénomène peut également être dû à la suppression ou à l'ajout de nouvelles classes. Le réentraînement du modèle peut contribuer à atténuer la dégradation du modèle due à la dérive des prédictions.
Mathématiquement, la dérive de prédiction peut être définie comme suit :
P(Y) s Pref(Y)
Où P(Y) représente la distribution de probabilité préalable des étiquettes cibles.

Changement de concept
Un changement de concept se produit lorsque la relation entre les variables indépendantes et les variables dépendantes ou cibles change. Il est également connu sous le nom de changement de classe postérieure, de changement conditionnel ou de dérive réelle du concept. Il s'agit de changements dans la relation entre les variables indépendantes et les variables dépendantes ou cibles. entre les variables d'entrée et les variables cibles. Si vous détectez un changement de concept significatif, il est très probable que les prédictions de votre modèle ne soient pas fiables. Un changement de concept fait référence à la relation entre les variables indépendantes et dépendantes.

Techniques de détection de la dérive dans ML
La mesure de la distance statistique à l'aide de la métrique de distance entre deux distributions est utile pour détecter la dérive dans la ML.
Si l'ensemble de données comporte de nombreuses variables indépendantes, vous pouvez utiliser des techniques de réduction de la dimensionnalité, telles que l'ACP.
Le suivi de nombreuses caractéristiques peut augmenter la charge du système de surveillance et il devient parfois difficile d'atténuer la dérive en ciblant des caractéristiques spécifiques.
Des mesures statistiques de base, telles que la valeur moyenne, l'écart-type, la corrélation et la comparaison des valeurs minimales et maximales, peuvent être utilisées pour calculer la dérive entre la formation et les variables indépendantes actuelles.
Les mesures de distance telles que l'indice de stabilité de la population (PSI), l'indice de stabilité des caractéristiques (CSI), la divergence de Kullback-Leibler (KL-
Divergence), la divergence de Jensen-Shannon (JS-Divergence) et les statistiques de Kolmogorov-Smirnov (KS) peuvent être utilisées pour les caractéristiques continues.
Les contrôles de cardinalité, le test du Khi-deux et l'entropie peuvent être utilisés pour les variables catégorielles.
Les cartes de contrôle et les intersections d'histogrammes peuvent être utilisées pour détecter une dérive dans les données.
Enfin, il existe plusieurs plateformes de suivi des modèles, telles que WhyLabs, et des bibliothèques, telles que deepchecks et alibi-detect. Elles s'intègrent facilement et offrent un cadre prêt à l'emploi pour la détection des dérives. Le plus intéressant est que la plupart d'entre elles fournissent un cadre de détection de la dérive, un stockage pour les journaux et les données historiques, des tableaux de

bord de surveillance intuitifs et des mécanismes d'alerte pour les événements critiques.

S'attaquer à la dérive sur le site ML

Une fois la dérive détectée dans le modèle ML, elle peut être traitée de la manière suivante :

Problèmes de qualité des données

S'il y a un problème avec les données d'entrée, il peut être facilement corrigé. Par exemple, des images à haute résolution ont été fournies pour l'entraînement des modèles de reconnaissance des visages, mais des images à basse résolution ont été transmises au modèle déployé.

Réentraînement du modèle

Après avoir détecté le changement de données ou de concept, le réentraînement du modèle avec des données récentes peut améliorer ses performances. Parfois, les données de production ne sont pas suffisantes pour entraîner le modèle. Dans ce cas, vous pouvez combiner des données historiques avec des données de production récentes et donner plus de poids aux données récentes.

Voici quatre stratégies pour recycler le modèle :

•Recyclage périodique : Le programmer à un moment fixe, par exemple, chaque année.

Lundi à 22 heures

•Axé sur les données ou les événements : Lorsque de nouvelles données sont disponibles

•Modèle ou métrique : Lorsque la précision est inférieure à un seuil ou à un accord de niveau de service.

•Apprentissage en ligne : Le modèle apprend continuellement en temps réel ou presque sur la base des données les plus récentes.

Reconstruction ou mise au point du modele

Si le réentraînement du modèle ne fonctionne pas, vous devrez peut-être envisager de le reconstruire ou de le régler sur des données récentes. Vous pouvez automatiser cette opération à l'aide d'un pipeline.

Surveillance opérationnelle avec Prometheus et Grafana

Prometheus est un système open-source utilisé pour la surveillance des événements et les alertes. Il récupère les données en temps réel des tâches instrumentées et les stocke avec un horodatage dans la base de données. Le terme "instrument" fait référence à l'utilisation d'une bibliothèque client qui permet à Prometheus de suivre et d'extraire ses métriques et de les stocker localement. Prometheus propose des bibliothèques clientes qui peuvent être utilisées pour instrumenter votre application. Dans ce scénario, vous utiliserez le client Prometheus Python dans l'application FastAPI pour exposer ses métriques qui doivent être suivies.

Un serveur Prometheus recueille et stocke les métriques des tâches instrumentées sous forme de séries temporelles. Ces données peuvent être récupérées à l'aide du langage de requête PromQL et peuvent être utilisées pour visualiser les métriques. Il est livré avec un gestionnaire d'alertes pour gérer les alertes.

Selon GrafanaLabs, Grafana est une plateforme open-source de visualisation et de surveillance interactive qui permet aux utilisateurs de visualiser les métriques,

les journaux et les traces collectées à partir d'applications déployées. Grafana est facile à intégrer avec les bases de données les plus courantes, telles que Prometheus, Influx DB, ElasticSearch, MySQL et PostgreSQL.
Grafana étant un outil open source, vous pouvez écrire un plugin d'intégration à partir de zéro pour vous connecter à plusieurs sources de données. Le tableau de bord Grafana récupère les données des sources de données connectées et vous permet de choisir le type de visualisation parmi de nombreuses options de visualisation, telles que les cartes thermiques, les diagrammes à barres et les graphiques linéaires. Vous pouvez facilement exécuter la requête, visualiser les mesures et configurer des alertes pour les événements critiques.
Dans ce scénario, vous utiliserez Grafana et Prometheus. Prometheus et Grafana sont tous deux des outils open-source. En fait, Prometheus et Grafana sont des combinaisons populaires dans l'industrie pour les systèmes de surveillance. Le tableau de bord Grafana sera utilisé pour la visualisation et la gestion des alertes. Il récupérera les données de la base de données Prometheus pour interroger les métriques et affichera la visualisation intuitive sur le tableau de bord.
Prometheus et Grafana peuvent être installés et configurés séparément sur une machine locale ou un serveur distant. Cependant, vous utiliserez des images docker de Prometheus et Grafana en exécutant le fichier docker-compose.yaml.
Pour maintenir la cohérence, les fichiers similaires des chapitres précédents avec des modifications mineures seront utilisés.

Points à retenir
•La surveillance du ML consiste à suivre les performances, les erreurs, les mesures, etc. du modèle déployé et à envoyer des alertes (le cas échéant) pour s'assurer que le modèle continue à fonctionner au-dessus du seuil acceptable.
•Une fois qu'un modèle de ML est déployé en production, il est essentiel de le surveiller afin de s'assurer que ses performances sont à la hauteur et qu'il continue à fournir des résultats fiables de manière transparente.
•Un changement de concept se produit lorsque la relation entre les variables indépendantes et les variables dépendantes ou cibles change.
•La surveillance doit compléter la boucle de rétroaction, c'est-à-dire qu'après avoir détecté un problème ou une défaillance, elle doit envoyer une notification à l'équipe concernée ou aux scientifiques des données afin qu'ils puissent prendre les mesures nécessaires.

XI Post-production Modèles ML

Introduction

Jusqu'à présent, on a étudié les différentes étapes du cycle de vie du ML et les moyens d'empaqueter, de déployer et de contrôler les modèles de ML. On a également appris à mettre en œuvre des tests pour garantir l'intégrité et le fonctionnement des modules. On peut créer des applications locales à partir de modèles de ML qui peuvent fonctionner sur des appareils Windows et Android ; déployer des modèles sur des plateformes de cloud populaires comme Microsoft Azure, GCP et AWS ; des solutions de surveillance pour la surveillance opérationnelle et la surveillance des modèles de ML.
Cependant, l'étape suivante consiste à en tirer une valeur commerciale. Après le déploiement, lorsqu'un nouveau modèle est prêt, vous devrez décider si le nouveau modèle est plus performant que le modèle existant afin de fournir des prédictions fiables. La sécurité des modèles est également essentielle après leur déploiement en production.

Combler le fossé entre le modèle ML et la création de valeur commerciale

Les utilisateurs professionnels s'appuient sur les prédictions du modèle ML pour élaborer la stratégie de l'entreprise et prendre des décisions. L'obtention d'une bonne précision n'est pas synonyme d'un bon impact commercial. Parfois, des modèles moins précis augmentent le chiffre d'affaires d'une entreprise. Vous devez combler le fossé entre l'élaboration de solutions de ML et la création d'une valeur commerciale à partir de ces solutions. Les scientifiques des données doivent être en mesure de convertir les prédictions des modèles en actions faciles à comprendre. Par exemple, avec un modèle de classification, vous pouvez créer cinq catégories basées sur le score de probabilité des classes, puis les classer de manière à ce que la probabilité la plus élevée ait le rang le plus élevé.
Après avoir déployé le modèle ML dans l'environnement de production à l'aide d'un système de surveillance, l'étape suivante consiste à s'assurer que la solution MLOps profitera aux utilisateurs professionnels. Vous pouvez obtenir un retour d'information de la part des parties prenantes, des utilisateurs professionnels ou des clients pendant qu'ils consomment les résultats du modèle. Par exemple, vous pouvez demander aux utilisateurs si une caractéristique est devenue plus importante que les autres après le déploiement des modèles, afin de lui donner plus de poids. De cette manière, vous pouvez ajuster le modèle pour fournir des prédictions plus utiles aux utilisateurs professionnels.
Les scientifiques des données doivent être en mesure d'expliquer le fonctionnement du modèle (à un niveau élevé) aux utilisateurs professionnels, ce qui les aidera à faire confiance aux résultats du modèle, car beaucoup d'entre eux ne connaissent pas les termes de ML. Les utilisateurs professionnels ou les clients préfèrent consommer les résultats sous une forme lisible, comme des tableaux de bord interactifs ou des chatbots. Vous pouvez également utiliser des outils de Business Intelligence (BI), tels que Tableau ou Power BI, pour créer un tableau de bord.

Modèle sécurité

La sécurité des modèles est un élément essentiel des MLOps. Lors du traitement des données, il peut être important de protéger les informations sensibles qu'elles peuvent contenir. Les attaques peuvent prendre dans la phase de formation au modèle ou dans la phase de production. Dans ce chapitre, vous vous familiariserez avec les différents types d'attaques afin de pouvoir mettre en œuvre des solutions adéquates pour les prévenir.

Les termes suivants sont liés à la sécurité des modèles :

•L'empoisonnement : Transmission de données malveillantes au processus de formation dans le but de modifier les résultats du modèle.

•Extraction : Reconstruction d'un nouveau modèle à partir du modèle cible qui fonctionnera de la même manière que le modèle ciblé.

•Evasion : Tenter de changer l'étiquette d'une classe particulière en faisant de petites variations dans les données d'entrée.

•Inférence : Déterminer si un ensemble de données spécifique faisait partie de la formation.
données

•Inversion : Obtenir des informations sur les données de formation à partir du modèle formé par ingénierie inverse.

Attaque adverse

Il s'agit d'une méthode permettant de générer des données hostiles. Les attaquants envoient intentionnellement des données malveillantes au modèle afin que celui-ci fasse des prédictions incorrectes. Lorsque le modèle apprend les nouveaux modèles de données, cette attaque provoque une erreur dans les prédictions du modèle. Ces données d'entrée malveillantes peuvent sembler normales aux yeux des humains ; cependant, de petites modifications des données d'entrée peuvent avoir un impact important sur les prédictions du modèle.

Les attaques adverses peuvent être classées en deux catégories principales en fonction de l'objectif de l'attaque.
attaquants :

•Attaques ciblées : Les attaquants tentent de modifier l'étiquette en fonction d'une cible particulière. Pour ce faire, ils peuvent modifier la source de données d'entrée en fonction d'une cible spécifique. Cela demande plus de temps et d'efforts.

•Attaques non ciblées : Les attaquants n'ont pas de cible spécifique que le modèle devrait prédire. Cependant, ils modifient l'étiquette sans cible spécifique. Cela demande moins de temps et d'efforts.

Les attaquants peuvent utiliser les méthodes suivantes pour mener des attaques contradictoires :

•Méthode de la boîte noire : Dans cette méthode, les attaquants peuvent envoyer les données d'entrée au modèle et obtenir la sortie en fonction de celles-ci.

• Méthode de la boîte blanche : Dans cette méthode, l'attaquant connaît presque tout du modèle de ML, comme les données d'apprentissage et les poids attribués aux caractéristiques.

Empoisonnement des données attaque

Dans une attaque par empoisonnement de données, les attaquants ont accès aux données d'entrée ou aux données sources. Ils modifient les données d'entrée de manière à ce que le modèle fasse des prédictions incorrectes qui ne sont pas fiables pour prendre des décisions commerciales ou entreprendre une action. Les attaquants peuvent ajouter du bruit aux données d'origine afin de modifier les prédictions du modèle. Cette attaque cible les données d'entraînement et les modifie intelligemment.

Attaque par déni de service distribué (DDoS)

Il s'agit de transmettre des données complexes à des modèles qui mettront plus de temps à faire des prédictions. Ce type d'attaque limite l'utilisation des modèles pour les utilisateurs. Pour ce faire, les attaquants peuvent injecter des logiciels malveillants pour contrôler le système ou le serveur.

Confidentialité des données attack

La confidentialité des données fait référence à la confidentialité des informations personnelles identifiables (PII) et des informations personnelles sur la santé (PHI). Les attaquants tentent d'apprendre ce type d'informations sensibles par le biais de ce type d'attaque. Il peut s'agir d'informations sur le modèle ou les données d'apprentissage. Les modèles ML tels que les machines à vecteurs de support (SVM) peuvent divulguer ces informations, car les vecteurs de support sont des points de données provenant des données d'entraînement.

Les attaques contre la confidentialité des données peuvent être classées dans les catégories suivantes :

•Attaque par inférence d'appartenance : L'objectif de cette attaque est de déterminer si l'entrée X fait partie des données d'apprentissage. La plupart du temps, des modèles fantômes sont utilisés dans ce type d'attaque contre la protection de la vie privée. La formation de modèles fantômes utilise un ensemble de données fantômes afin d'imiter le modèle cible. La sortie des modèles fantômes est ensuite transmise au méta-modèle. Enfin, la sortie du méta-modèle est utilisée pour extrapoler les propriétés des données d'apprentissage ou du modèle. Les modèles surajustés sont sujets à des attaques contre la confidentialité des données.

•Attaque par inférence d'entrée : Elle est également connue sous le nom d'inversion de modèle ou d'attaque par extraction de données. Il s'agit du type d'attaque le plus courant. L'objectif de cette attaque est d'extraire des informations de l'ensemble des données d'apprentissage en procédant à une rétro-ingénierie du modèle. Elle peut également viser l'apprentissage des propriétés statistiques des données d'entrée, telles que la distribution des probabilités. Les attaquants peuvent tenter d'apprendre les caractéristiques qui ne sont pas codées explicitement lors de l'apprentissage du modèle.

•Attaque par extraction de modèle : Elle est également connue sous le nom d'attaque par inférence de paramètres. L'objectif de cette attaque est d'apprendre les hyperparamètres du modèle, puis de reconstruire le modèle qui se comporte comme le modèle ciblé. Il est intéressant de noter que les modèles surajustés sont difficiles à extraire en raison des erreurs de prédiction élevées basées sur les données de test.

Atténuer le risque d'attaques par le modèle

Le pipeline de ML peut être divisé en deux phases : la phase de formation et la phase de test. Vous pouvez vous attaquer à diverses attaques de modèles de ML en fonction de ces phases.

Phase de formation

Un scientifique des données effectue des activités telles que la collecte et le nettoyage des données, l'ingénierie des caractéristiques, le choix d'algorithmes appropriés, l'ajustement des hyperparamètres et la construction de modèles. Les attaquants ciblent cette phase par le biais d'attaques telles que l'empoisonnement des données. Si un modèle est entraîné sur des données empoisonnées, vous ne pouvez pas vous fier à ses prédictions.

Vous pouvez mettre en œuvre les techniques suivantes pour réduire le risque d'attaques pendant la phase de formation :

•Cryptage des données
•Protéger l'intégrité des données de formation
•Statistiques robustes
•Assainissement des données

Phase de test

Dans cette phase, les attaquants ciblent les modèles de ML. Les attaques par extraction de modèle sont courantes parmi les attaquants. Ils tentent de déterminer si l'entrée X fait partie des données d'apprentissage ou de voler les paramètres du modèle définis pendant la phase d'apprentissage.

Les techniques suivantes peuvent être mises en œuvre pour atténuer le risque pendant la phase de test :

•Formation contradictoire
•Autoencodeurs
•Distillation
•Techniques d'ensemble
•Limiter le nombre de demandes par utilisateur

Vous pouvez utiliser la bibliothèque Python Adversarial Robustness Toolbox (ART) et l'outil en ligne de commande Counterfit pour la sécurité des modèles ML. Cette bibliothèque permet de se défendre contre les types d'attaques ML les plus courants. Elle prend également en charge les frameworks, bibliothèques et types de données ML les plus courants.

Tests A/B

Les tests A/B sont largement utilisés dans le domaine du marketing, de la conception de sites web et des campagnes d'e-mailing afin de connaître et de comprendre les préférences des utilisateurs. L'objectif des tests A/B est d'augmenter le taux de conversion, le taux de réussite, le chiffre d'affaires, etc. Les tests A/B consistent à diviser les audiences ou les clients de la population en groupes égaux. Ces ensembles seront dirigés vers la version de contrôle et la version expérimentale. La version de contrôle est la version existante, tandis que la version expérimentale est la nouvelle version ou la version challenger. Tout d'abord, définissez l'énoncé du problème pour le test A/B, l'hypothèse nulle et l'hypothèse alternative pour l'énoncé du problème. Ensuite, concevez l'expérience pour suivre et analyser la métrique, puis exécutez et validez l'expérience. Ensuite, comparez les statistiques des résultats. Enfin, prenez une décision en fonction des résultats. Si le test A/B n'est pas effectué correctement, ses résultats ne seront pas fiables.

Les scientifiques des données effectuent une évaluation en ligne des modèles de

ML en divisant les données en ensembles de formation et de validation. Cependant, le test A/B vous permet d'effectuer l'évaluation en ligne des modèles de ML en mesurant les métriques commerciales ou les taux de réussite. Le test A/B peut être mis en œuvre lorsque vous effectuez des opérations sur les données d'entraînement, telles que la mise à l'échelle et la normalisation ou l'application de différents algorithmes et hyperparamètres lors de la construction du modèle. Un ingénieur MLOps ou un data scientist peut déployer plusieurs modèles simultanément pour tester les modèles en production. Google Cloud (GCP) et Amazon SageMaker permettent le déploiement de plusieurs modèles en production derrière le même point de terminaison pour décider quel modèle est le plus performant du point de vue de l'entreprise.
Un Bandit Multi-Armé (MAB) est une version avancée du test A/B. Il est plus complexe que le test A/B, car il utilise des algorithmes ML tout en allouant dynamiquement plus de trafic à la version la plus performante et moins de trafic à la version la moins performante sur la base des données.

MLOps est l'avenir

Le domaine de l'apprentissage automatique est en plein essor. Les industries font de l'apprentissage automatique un élément essentiel du processus de développement des entreprises, et le système d'apprentissage automatique permet de relever de nombreux nouveaux défis.
Les MLOps peuvent gérer la complexité et l'évolutivité des modèles de ML. C'est pourquoi la demande de MLOps augmente dans l'ensemble du secteur. Il y a une pénurie d'ingénieurs MLOps dans le secteur, car les MLOps sont à l'intersection de l'apprentissage automatique et du développement de logiciels. Les entreprises doivent mettre en place des équipes distinctes pour les ingénieurs MLOps, ou elles peuvent renforcer les compétences de leurs data scientists pour qu'ils exécutent les tâches MLOps. Les MLOps réduisent les coûts et les efforts manuels du cycle de vie global de l'apprentissage automatique. Cela permet aux data scientists et aux ingénieurs ML de se concentrer sur d'autres tâches productives.
Le MLOps devient plus populaire que le DevOps. Votre modèle peut être performant dans l'environnement local, mais s'il n'atteint pas les utilisateurs finaux ou les clients, son impact commercial est faible. Plus de 85 % des modèles de ML ne sont pas déployés dans l'environnement de production. Les ingénieurs MLOps peuvent combler le fossé entre la recherche et la production.

Points à retenir

•La sécurité des modèles est un élément essentiel des MLOps.
•Les tests A/B vous permettent d'effectuer une évaluation en ligne des modèles de ML.
•Les attaquants peuvent utiliser la méthode de la boîte noire ou la méthode de la boîte blanche lorsqu'ils tentent une attaque contradictoire.
•La bibliothèque Python Adversarial Robustness Toolbox (ART) et l'outil en ligne de commande Counterfit peuvent être utilisés pour la sécurité des modèles ML.
•Un bandit multi-bras (MAB) est une version avancée du test A/B. Il est suffisamment intelligent pour décider quel modèle devrait obtenir plus de trafic en évaluant plusieurs modèles. Il est suffisamment intelligent pour décider quel modèle devrait obtenir plus de trafic en évaluant plusieurs modèles.