Détection des mails de phishing à l'aide de QSVM

Sommaire

Contexte du projet

Le phishing est une technique de fraude en ligne utilisée pour obtenir des informations sensibles telles que des noms d'utilisateur, des mots de passe et des informations de carte de crédit en se faisant passer pour une entité de confiance. Avec l'augmentation de ces attaques, il devient crucial de développer des techniques automatisées pour détecter les mails de phishing. Dans ce contexte, nous allons explorer l'utilisation des Machines à Vecteurs de Support (SVM) pour identifier ces mails malveillants, en les appliquant sur des ordinateurs quantiques.

SVM sur Ordinateur quantique

Les ordinateurs quantiques exploitent les principes de la mécanique quantique pour effectuer des calculs bien plus rapidement que les ordinateurs classiques pour certaines tâches. Un SVM quantique (QSVM) utilise les états quantiques pour représenter et manipuler les données, permettant ainsi de traiter des ensembles de données beaucoup plus vastes et complexes.

Avantages des QSVM

Vitesse de Calcul Accrue : Les ordinateurs quantiques peuvent effectuer des calculs en parallèle grâce à la superposition et à l'intrication, réduisant ainsi le temps nécessaire pour entraîner les modèles SVM sur des ensembles de données volumineux.

- **Capacité à gérer des données complexes** : Les QSVM peuvent traiter des données dans des espaces de grande dimension de manière plus efficace que les SVM classiques, offrant des performances supérieures pour des tâches complexes.
- **Amélioration des précisions** : Les algorithmes quantiques peuvent explorer des solutions optimales plus rapidement, potentiellement améliorant la précision des modèles de détection de phishing.

Modalités pédagogiques

Ce travail pratique est réalisé en groupe et se déroule sur deux jours. Les étudiants travailleront en autonomie sur les différentes étapes du projet, avec un accompagnement pour répondre aux questions techniques et méthodologiques.

Jour 1 - matin

1. Importation du jeu de données :
Objectif : Charger les données nécessaires pour l'analyse.

Pistes :

- Rechercher comment utiliser des bibliothèques Python comme pandas pour importer des fichiers CSV.
- Explorer les différentes méthodes de pandas pour charger des données, comme **read_csv()**.
- Vérifier la structure des données importées en utilisant des fonctions comme **head()** ou **info()** pour obtenir un aperçu initial.

Questions guidantes :

- Comment importer un fichier CSV en Python ?
- Quelles informations peut-on obtenir avec les méthodes **head()** et **info()** de pandas ?

2. Visualisation du jeu de données :
Objectif : Comprendre la distribution des données et identifier les tendances ou anomalies potentielles.

Pistes :
- Utiliser des bibliothèques de visualisation telles que **Matplotlib** ou **Seaborn**.
- Réfléchir à la meilleure façon de visualiser les proportions des différentes classes de données (e.g., mails de phishing vs. mails légitimes).
- Explorer comment créer des graphiques comme des histogrammes, des barres ou des nuages de mots pour visualiser les données textuelles.

Questions guidantes :
➡ Quelles bibliothèques de visualisation sont disponibles en Python ?
➡ Comment peut-on visualiser la distribution des classes dans un jeu de données ?
➡ Comment peut-on représenter les termes fréquents dans des données textuelles ?

3. Analyse statistique :
Objectif : Obtenir des statistiques de base sur les données.

Pistes :
- Explorer les méthodes de pandas pour obtenir des statistiques descriptives (e.g., **describe()**).
- Identifier quelles caractéristiques des mails (e.g., longueur du texte, nombre de mots) pourraient être importantes pour l'analyse.

Questions guidantes :
➡ Comment obtenir des statistiques descriptives pour un jeu de données en utilisant pandas ?
➡ Quelles caractéristiques des mails pourraient être intéressantes à analyser ?

4. Analyse de la corrélation :
Objectif : Identifier les relations entre les différentes caractéristiques.

Pistes :
- Rechercher des méthodes pour calculer et visualiser les corrélations entre les caractéristiques.
- Utiliser des cartes de corrélation pour visualiser ces relations.

Questions guidantes :
➡ Qu'est-ce qu'une corrélation et pourquoi est-elle importante dans l'analyse des données ?
➡ Comment peut-on visualiser une carte de corrélation en utilisant **Seaborn** ou **Matplotlib** ?

5. Identification des caractéristiques importantes :
Objectif : Déterminer quelles caractéristiques ont le plus d'impact sur la classification.

Pistes :
- Explorer des techniques d'ingénierie des caractéristiques pour sélectionner les plus pertinentes.
- Examiner des méthodes comme l'analyse de la variance (**ANOVA**) ou les modèles de sélection de caractéristiques intégrés dans les bibliothèques de machine learning.

Questions guidantes :
➡ Quelles techniques peuvent être utilisées pour sélectionner les caractéristiques les plus importantes dans un jeu de données ?
➡ Comment peut-on utiliser des méthodes comme l'ANOVA pour l'analyse des caractéristiques ?

Jour 1 - après-midi

1. Nettoyage des données

Objectif : Assurer que les données sont propres et prêtes pour l'analyse en traitant les valeurs manquantes et les doublons.

Pistes :

- Explorer différentes méthodes pour gérer les valeurs manquantes, comme la suppression des lignes avec des valeurs manquantes ou l'imputation de ces valeurs avec des techniques comme la moyenne, la médiane ou des valeurs spécifiques.
- Vérifier quelles colonnes contiennent des valeurs manquantes et d'évaluer l'impact de ces valeurs manquantes sur l'analyse.

Questions guidantes :

➡ Quelles méthodes peuvent être utilisées pour traiter les valeurs manquantes dans un jeu de données ?
➡ Comment déterminer quelles colonnes ou lignes contiennent des valeurs manquantes ?
➡ Quels sont les avantages et les inconvénients de supprimer des lignes par rapport à l'imputation des valeurs manquantes ?

2. Traitement des doublons :

Pistes :

- Rechercher comment identifier et supprimer les doublons dans un jeu de données.
- Réfléchir aux critères de duplication : doivent-ils considérer l'ensemble des colonnes ou seulement certaines colonnes pour identifier les doublons ?

Questions guidantes :

➡ Comment identifier les doublons dans un jeu de données ?
➡ Quels critères utiliser pour déterminer si une ligne est un doublon ?
➡ Quels impacts les doublons peuvent-ils avoir sur l'analyse des données et la performance des modèles de machine learning ?

3. Transformation des textes en vecteurs

Objectif : Convertir les données textuelles en une forme numérique que les algorithmes de machine learning peuvent utiliser, en utilisant des techniques comme TF-IDF.

Pistes :

- Rechercher des techniques courantes de vectorisation de texte, notamment le Bag of Words, TF-IDF et les embeddings de mots.
- Suggérez-leur de comprendre les principes de base de TF-IDF (Term Frequency-Inverse Document Frequency) et pourquoi cette technique est utile pour la représentation des textes.

Questions guidantes :

➡ Quelles sont les différentes méthodes pour transformer des textes en vecteurs ?
➡ En quoi consiste la technique TF-IDF et quels sont ses avantages par rapport à d'autres méthodes de vectorisation de texte ?

4. Implémentation de TF-IDF :

Pistes :

- Explorer les bibliothèques Python disponibles pour la vectorisation de texte, comme **scikit-learn**.
- Réfléchir à la manière de configurer le vectoriseur TF-IDF, par exemple en ajustant les paramètres pour tenir compte des mots les plus fréquents ou les moins fréquents.

Questions guidantes :

- ➡ Comment utiliser la bibliothèque `scikit-learn` pour appliquer la vectorisation TF-IDF ?
- ➡ Quels paramètres peuvent être ajustés dans le vectoriseur TF-IDF pour améliorer la qualité de la transformation des textes ?

5. Application de la vectorisation :

Pistes :

- Transformer la colonne de texte en une matrice de vecteurs TF-IDF et de vérifier la forme et le contenu de la matrice résultante.
- Analyser les vecteurs résultants pour comprendre quelles caractéristiques textuelles sont capturées.

Questions guidantes :

- ➡ Comment appliquer la transformation TF-IDF à une colonne de texte dans un jeu de données ?
- ➡ Comment interpréter la matrice de vecteurs résultante ?
- ➡ Quels défis pourraient survenir lors de la transformation de grands volumes de texte et comment les surmonter ?

Jour 2 - matin

1. Implémentation d'un QSVM

Objectif : Utiliser des outils et des bibliothèques pour implémenter et simuler un modèle de Machine à Vecteurs de Support Quantique (QSVM).

Pistes :

- Familiariser avec les bibliothèques de simulation d'ordinateurs quantiques, notamment **Qiskit**, développé par IBM.
- Consulter la documentation officielle de **Qiskit** pour comprendre les concepts de base et les fonctionnalités offertes par la bibliothèque.
- Commencer par des tutoriels ou des exemples de code pour se faire une idée des étapes de base de l'utilisation de **Qiskit**.

Questions guidantes :

- ➡ Qu'est-ce que Qiskit et quelles sont ses principales fonctionnalités ?
- ➡ Où peut-on trouver la documentation et des tutoriels pour débuter avec Qiskit ?
- ➡ Quels sont les concepts de base d'un ordinateur quantique nécessaires pour comprendre Qiskit ?

2. Installation et configuration de Qiskit :

Pistes :

- Suivre les instructions d'installation de **Qiskit** sur leur machine locale ou dans un environnement de développement en ligne comme **Google Colab**.
- Tester l'installation en exécutant un exemple simple, comme créer un circuit quantique basique et le simuler.

Questions guidantes :

- ➡ Comment installer Qiskit sur une machine locale ou dans un environnement cloud ?
- ➡ Comment vérifier que l'installation de Qiskit a été réussie ?
- ➡ Comment créer et simuler un circuit quantique de base avec Qiskit ?

3. Compréhension des composants de Qiskit :
Pistes :

- Familiariser avec les composants de Qiskit, tels que **QuantumCircuit, Aer** (simulateur quantique), et **QuantumInstance**.
- Comprendre comment ces composants interagissent pour permettre la création, la simulation et l'analyse de circuits quantiques.

Questions guidantes :

- Quels sont les principaux composants de Qiskit et leurs rôles respectifs ?
- Comment créer et manipuler un circuit quantique avec `QuantumCircuit` ?
- Comment utiliser `Aer` pour simuler un circuit quantique ?

4. Entraînement du modèle QSVM avec des hyperparamètres choisis
Objectif : Entraîner un modèle de **Machine à Vecteurs de Support Quantique** (QSVM) en ajustant les hyperparamètres pour optimiser les performances.

Pistes :

- Rechercher comment configurer un modèle QSVM avec Qiskit.
- Comprendre les hyperparamètres clés qui influencent le comportement du modèle QSVM, comme le choix du noyau quantique.

Questions guidantes :

➡ Comment configurer un modèle QSVM avec Qiskit ?
➡ Quels sont les hyperparamètres clés d'un QSVM et comment influencent-ils le modèle ?

5. Préparation des données :
Pistes :

- Préparer les données de manière à ce qu'elles soient compatibles avec les entrées du modèle QSVM.
- Utiliser des techniques de prétraitement des données pour normaliser ou transformer les données d'entrée en une forme appropriée pour le modèle quantique.

Questions guidantes :

➡ Comment préparer les données pour qu'elles soient compatibles avec un modèle QSVM ?
➡ Quelles techniques de prétraitement des données sont nécessaires pour les modèles quantiques ?

6. Entraînement et validation du modèle :
Pistes :

- Diviser les données en ensembles d'entraînement et de validation.
- Définir une procédure pour entraîner le modèle QSVM en ajustant les hyperparamètres et en utilisant des techniques de validation croisée pour évaluer les performances.

Questions guidantes :

➡ Comment diviser les données en ensembles d'entraînement et de validation pour un modèle QSVM ?
➡ Quelles sont les meilleures pratiques pour entraîner un modèle QSVM et évaluer ses performances ?

7. Analyse des résultats :

Pistes :

- Analyser les résultats de l'entraînement du modèle en utilisant des métriques d'évaluation appropriées, comme la précision, le rappel, la F-mesure, et l'AUC.
- Interpréter les résultats pour identifier les points forts et les limites du modèle QSVM.

Questions guidantes :

➡ Quelles métriques d'évaluation utiliser pour analyser les performances d'un modèle QSVM ?
➡ Comment interpréter les résultats obtenus pour améliorer le modèle ?

Jour 2 - après-midi

Évaluation du modèle :

Utilisation de métriques comme la précision, le rappel, la F-mesure, et l'AUC pour évaluer les performances.
Visualisation des résultats et interprétation des performances du modèle.

Présentation des résultats

- Préparation d'une présentation sur PowerPoint.
- Discussion sur les avantages et les inconvénients des SVM classiques et quantiques.
- Exploration des autres applications possibles des SVM quantiques.

Modalités d'évaluation

Participation active :

- Engagement dans les discussions et les travaux pratiques.
- Collaboration efficace au sein du groupe.

Qualité de l'analyse et du modèle :

- Pertinence et précision de l'analyse des données.
- Qualité de l'implémentation du modèle QSVM.

Présentation finale :

- Clarté et exhaustivité des diapositives.
- Capacité à expliquer et justifier les choix techniques.
- Discussion critique sur les avantages et les inconvénients des SVM classiques et quantiques.

Livrables

Présentation PowerPoint

Avantages et inconvénients des SVM :

- **Avantages** : efficacité dans les espaces de grande dimension, robustesse contre l'overfitting, etc.
- **Inconvénients** : complexité computationnelle, choix des hyperparamètres, etc.

Mise en pratique du QSVM :

- Description des étapes suivies pour implémenter le modèle.
- Résultats obtenus et leur interprétation.

Autres applications du QSVM :

- Classification d'images, reconnaissance vocale, bio-informatique, etc.

Code source :

- Scripts de préparation des données, implémentation du modèle QSVM et évaluation des performances.

Critères de performance

Qualité de la préparation des données :

- Correction et complétude du nettoyage et de la transformation des données.

Implémentation du modèle :

- Exactitude et optimisation du code QSVM.

Évaluation et interprétation :

- Pertinence des métriques utilisées et qualité de l'interprétation des résultats.

Présentation :

- Clarté, structure et profondeur de l'analyse des avantages et inconvénients des SVM classiques et quantiques.
- Richesse des exemples d'autres applications du QSVM.

Travail en groupe :

- Collaboration efficace et répartition équitable des tâches.