DATA SCIENCE DIVISION FELLOWSHIP PROGRAM: TAKEHOME EXAM

STATISTICS CANADA

AFFECTATION : APPRENTISSAGE AUTOMATIQUE

Vous avez **jusqu'à 18 heures , le 3 décembre 2022 pour terminer et soumettre votre examen.**

Retournez l'examen achevé à : statcan.fellowship-fellowship.statcan@statcan.gc.ca.

La date et l'heure de votre courriel serviront à prouver que vous avez terminé l'examen dans le délai alloué. Aucun retard ne sera accepté. Donc, si l'examen n'est pas soumis par la date requise; votre candidature ne sera plus considérée dans le cadre de ce processus.

Toutes les tâches doivent être complétées pour que votre soumission soit revue et évaluée dans le cadre de ce processus de sélection.

Le contenu de cet examen est confidentiel. Ne divulguez pas le contenu de l'examen à d'autres personnes.

OBJECTIF

Le but de l'examen à domicile est de mettre en valeur vos compétences. Nous évaluerons votre capacité à appliquer des techniques courantes liées à la science des données et à l'apprentissage automatique dans Python, à exprimer les résultats de manière cohérente et compréhensible et à créer des visualisations appropriées pour présenter et interpréter les données.

L'examen permettra d'évaluer les aptitudes et les compétences suivantes :

- Programmation et apprentissage automatique;
- Gestion de données;
- Pensée analytique;
- Narration et visualisation de données;
- Communication.

Lisez attentivement toutes les consignes ci-dessous avant de commencer.

CONSIGNES

Deux scénarios différents sont proposés. **Complétez un (1)** des deux scénarios.

Chaque scénario nécessite l'accès à un ensemble de données accessible au public et l'exécution d'une séquence de tâches.

Certaines tâches sont structurées et précises, tandis que d'autres sont plus ouvertes. Des liens sont fournis pour les ensembles de données. Si les liens ne fonctionnent pas, il est de votre devoir de trouver l'ensemble de données par d'autres moyens.

DOCUMENTS À SOUMETTRE

Lorsque vous aurez complété le scénario, vous devrez fournir le code en entier pour les tâches du scénario. Cela inclut le code entier utilisé pour effectuer les tâches et tout autre code auxiliaire requis. N'hésitez pas à inclure le code pour toute exploration de données que vous pourriez choisir d'effectuer.

Le code doit être sous forme d'un bloc-notes Jupyter (ou l'équivalent) et doit être bien documenté. Nous ne devrions pas être obligés d'exécuter votre code pour produire des chiffres, ce qui signifie que les cellules de votre bloc-notes devraient déjà avoir été exécutées lorsqu'elles sont visualisées dans Gitlab ou Github. Vous devrez créer un dépôt Git publique (Gitlab ou Github) et y insérer votre bloc-notes Jupyter.

Votre soumission nécessite l'envoi par courriel du lien vers votre dépôt, ce qui nous permettra d'évaluer le code, les chiffres et la documentation dans le dépôt.

Nous ne souhaitons pas que vous produisiez un modèle ayant une précision de validation de 99 %, alors veuillez ne pas perdre de temps et d'efforts sur un **quelconque** réglage des hyperparamètres. Nous nous intéressons plutôt à votre processus de réflexion, à votre capacité à expliquer ce que vous avez fait, à la raison pour laquelle vous l'avez fait et, compte tenu des contraintes de l'examen, à la façon dont vous auriez pu faire les choses différemment. Soyez prêt à répondre aux questions concernant cet examen lors de l'entretien.

Voici quelques autres éléments que nous recherchons dans votre examen :

- Reproductibilité;
- Maintenabilité de la solution;
- Suivi du guide de style PEP8 / Google / Numpy;
- Choix de noms significatifs pour les fonctions et les variables;
- Pratiques de codage exemplaires.

NOTES IMPORTANTES

- Vous devez passer l'examen de manière autonome et sans assistance.
- Assurez-vous que le dépôt Git que vous configurez et soumettez n'est pas modifiable par une autre personne.
- Les soumissions évaluées seront récupérées dans le dépôt Git au plus tard à 18 h 00 le samedi 2 décembre 2022.

Bonne chance et amusez-vous bien!

SCÉNARIO 1 : ENSEMBLE DE DONNÉES SUR LES VOITURES

Ensemble de données

- L'ensemble de données sur les voitures se trouve à l'adresse suivante : https://ai.stanford.edu/~jkrause/cars/car_dataset.html
- L'ensemble de données sur les voitures contient 16 185 images de 196 classes de voitures. Les données sont réparties en 8 144 images d'entraînement et 8 041 images d'essai, et chaque classe a été divisée approximativement en deux.

Objectif

- Vos clients aimeraient avoir un modèle capable de classer les voitures trouvées à partir d'images prises par des caméras de circulation.
- Ce scénario porte sur l'évaluation de l'importance de la taille d'un ensemble de données étiqueté dans un environnement d'apprentissage supervisé.

Remarque : Pour les questions où l'on vous demande d'expliquer ou d'interpréter les résultats, veuillez le faire dans le cahier Jupyter (ou équivalent) en utilisant Markdown.

TÂCHE 1 — *Créer une fonction qui permet de convertir un ensemble de données étiqueté en sous-ensembles étiquetés et non étiquetés.*

☐	Créez une fonction qui comprend les entrées ou les arguments suivants : dataset_labels (liste de nombres entiers), proportion (nombre flottant entre [0,1]). Dans ce cas-ci, les indices de dataset_labels représentent les indices des instances dans votre ensemble de données, et dataset_labels[idx] représente l'indice de classe de l'instance de données idx.
☐	Cette fonction « supprime » les étiquettes de (100 * proportion)% de l'ensemble de données d'origine. Une fois que cette proportion d'instances de l'ensemble de données a été échantillonnée, ces étiquettes doivent être supprimées afin que ces instances soient traitées comme un ensemble de données sans étiquette.
☐	La fonction doit garantir que toutes les classes ont au moins une instance étiquetée dans l'ensemble de données.
☐	Le produit de la fonction dépend de vous. Cependant, à partir du résultat de cette fonction, vous devez : • savoir quelles instances de données (indices) dans votre ensemble de données sont étiquetées ou non étiquetées; • connaître l'étiquette si une instance de données est étiquetée; • quand même savoir quelle était l'étiquette d'origine si une instance de données n'est pas étiquetée.

TÂCHE 2 — *Nettoyage des données*

☐	Parcourez les images de l'ensemble de données et retirez du disque (supprimez) toutes les instances qui ne sont pas rouges, vertes, bleues (RVB). Plus précisément, supprimez toutes les images du disque qui n'ont pas trois canaux.

TÂCHE 3 — *Représentation des ensembles de données*

☐	Créez un dictionnaire vide qui fait correspondre les indices des instances de votre ensemble de données à un dictionnaire ayant cette structure (ensemble de données d'instances de taille N) : {1 : {'embedding' : <np.ndarray>, 'class_idx' : <int>, 'labelled' : <booléen ou int>}, 2 : {...}, …, N : {…}}
☐	À l'aide de PyTorch, chargez un ResNet-18 prépréentraîné et utilisez-le comme intégrateur de fonctionnalités. Pour chaque image de l'ensemble de données, chargez l'image à partir du disque et passez-la dans le ResNet-18 préentraîné pour recevoir le résultat (qui devrait être un vecteur de 1000 dimensions). Ensuite, mettez à jour le dictionnaire avec l'intégration nouvellement acquis et la commande init class_idx. ayant une étiquette « True » (ou 1). Méfiez-vous des types de données pour votre tenseur!
☐	Une fois cela fait, enregistrez ce dictionnaire sur le disque en utilisant torch.save(). Il s'agit de votre ensemble de données complet qui sera utilisé pour toutes les étapes suivantes.

TÂCHE 4 — *Créer un ensemble de données partiellement étiqueté*

☐	À l'aide de la fonction de la tâche 1 et de l'ensemble de données traité de la tâche 3, créez un ensemble de données en utilisant seulement 40 % des données étiquetées et 60 % des données non étiquetées.

TÂCHE 5 — *Créer une répartition entraînement et validation*

☐	Créez une fonction qui accepte deux arguments, dataset_inputs et dataset_labels, et un nombre flottant se situant entre [0,1], « training_proportion ». Cette fonction permet de prendre un ensemble de données (les entrées et les étiquettes) et de le répartir en données d'entraînement et en données de validation selon le nombre flottant, ou « training_proportion ». La fonction produit quatre objets : training_inputs, training_labels, test_inputs, test_labels.
☐	Remarque : N'hésitez pas à utiliser n'importe quel progiciel ou n'importe quelle bibliothèque (p. ex. Scikit learn)

TÂCHE 6 — *Créer des expériences pour convaincre les clients que davantage de données étiquetées améliorera le rendement du modèle*

☐	C'est à vous de convaincre vos clients d'étiqueter un peu plus de données, puisque vous pensez que le modèle aura un meilleur rendement en ayant plus de données étiquetées. Cependant, vos clients sont occupés et l'étiquetage de données supplémentaires prend du temps et coûte cher.

☐	Pour tous les modèles entraînés, utilisez un modèle linéaire simple qui se sert des intégrations comme entrées à l'aide de sklearn.linear_model.SGDClassifier. Utilisez le nombre d'itérations nécessaires pour former un modèle en 1 minute, MAXIMUM (ne vous inquiétez pas, nous ne jugeons pas les résultats en fonction de la précision).
☐	Créez une sorte de figure, en fonction des essais ou des expériences que vous avez effectués, que vous montrerez à vos clients pour les convaincre que l'étiquetage de plus de données conduira à un meilleur classificateur pour ce cas d'utilisation. Assurez-vous d'utiliser votre fonction créée à la tâche 5. Analysez le ou les tracés produits et expliquez les à vos clients. N'oubliez pas que dans ce scénario, vous n'avez que 40 % des données étiquetées; les 60 % restants ne sont pas étiquetés.

TÂCHE 7 — *Apprentissage actif pour sélectionner de nouvelles instances à étiqueter*

☐	Vos clients ont accepté d'étiqueter 25 % plus de l'ensemble de données (pour un total de 65 % de l'ensemble de données d'origine).
☐	En utilisant le modèle formé à partir de 40 % des données et des entrées non étiquetées, utilisez la fonction predict_proba de votre modèle sklearn.linear_model.SGDClassifier pour obtenir les scores de probabilité pour chaque classe selon vos instances non étiquetées. Utilisez scipy.stats.entropy pour calculer l'entropie des prédictions pour les instances non étiquetées. Sélectionnez les instances K ayant l'entropie la plus élevée pour l'étiquetage, où K est le nombre d'instances restantes qui porteront votre ensemble de données étiqueté total à 65 %. Celui-ci sera ajouté au sous-ensemble étiqueté et constituera votre ensemble de données « final ».
☐	Remarque : Étant donné que l'ensemble de données d'origine est entièrement étiqueté, utilisez simplement les véritables étiquettes de l'ensemble de données d'origine. Par conséquent, le suivi des indices des instances aura été important pour les consignes précédentes.

TÂCHE 8 — *Entraînement et évaluation du modèle final*

☐	Vous avez maintenant des données étiquetées supplémentaires.
☐	À l'aide de votre ensemble de données d'entraînement, entraînez un modèle final puis utilisez-le pour évaluer votre modèle selon votre ensemble exclu de données de validation.
☐	Expliquez vos résultats et votre mise en œuvre.
☐	D'après votre évaluation, la décision d'étiqueter plus d'instances était-elle une bonne décision?
☐	Faites état du rendement final.

TÂCHE 9 — *Soumettre l'examen à Statistique Canada*

☐	Référez-vous aux consignes qui se trouvent sous « Documents à soumettre » à la première page du présent document.

	Soumettez votre examen à statcan.fellowship-fellowship.statcan@statcan.gc.ca. Veuillez inclure : - le code entier utilisé pour accomplir la tâche et tout autre code auxiliaire requis; - le lien vers votre dépôt Git afin que nous puissions évaluer le code, les chiffres et la documentation.

SÉNARIO 2 : ENSEMBLE DE DONNÉES SUR LES CATASTROPHES NATURELLES

Ensemble de données

- L'ensemble de données sur les catastrophes naturelles est disponible à l'adresse suivante : https://www.kaggle.com/competitions/nlp-getting-started/
- Il s'agit d'un ensemble de données textuel (traitement du langage naturel) qui contient des tweet sur des catastrophes réelles, ou non (classification binaire).

Objectif

- Vos clients aimeraient classer les tweet selon qu'ils portent sur une véritable catastrophe ou non.
- Ils aimeraient que vous utilisiez le modèle le plus simple possible qui permet de faire le travail et d'atteindre une précision suffisamment élevée (un nombre qu'ils ne veulent pas définir). De plus, il est important pour eux de comprendre la façon dont vous êtes parvenu à la classification du modèle, si possible.
- Vous entraînerez et évaluerez trois algorithmes d'apprentissage de complexité croissante.
- Chaque algorithme d'apprentissage nécessitera un prétraitement différent des données textuelles que vous devrez mettre en œuvre vous-même.
- Votre objectif est de présenter à vos clients des options qui permettent de maintenir un équilibre entre le caractère explicable des modèles et leur rendement.

Remarque : Pour les questions où l'on vous demande d'expliquer ou d'interpréter les résultats, veuillez le faire dans le cahier Jupyter (ou équivalent) en utilisant Markdown.

TÂCHE 1 — *Modèle de sac de mots*

☐	Créez un modèle de sac de mots.
☐	Effectuez tout le prétraitement de l'ensemble de données pour le préparer à un modèle de sac de mots. Expliquez tous vos choix concernant le prétraitement.
☐	Entraînez le modèle et fournissez des tracés sur l'évaluation.

TÂCHE 2 — *Génération de fonctionnalités et modèle d'apprentissage automatique traditionnel*

☐	Vous allez maintenant utiliser TF-IDF pour générer des fonctionnalités pour votre ensemble de données. De plus, n'hésitez pas à créer tout autre ensemble de caractéristiques (unigramme, bigramme, trigramme, etc.).
☐	Au moyen des fonctionnalités que vous avez créées, sélectionnez n'importe quel modèle traditionnel (réseau non neuronal) pour former un modèle et fournir des tracés sur l'évaluation.
☐	Expliquez tous vos choix. Remarque : Nous ne jugerons pas nécessairement le rendement du modèle, mais vous devrez expliquer le choix du modèle et la raison pour laquelle il a été sélectionné, en fonction des fonctionnalités générées.

TÂCHE 3 — *Plongements de mots préentraînés et modèle de classificateur linéaire*

☐	Sélectionnez n'importe quel plongement de mots préentraîné (p. ex. GloVe).
☐	Vous exécuterez vos entrées textuelles au moyen du plongement de mots pour faire correspondre chaque instance d'entrée (séquence de mots) à un seul plongement qui représente la séquence entière de mots. C'est à vous de décider de la façon dont vous allez vous y prendre. La seule contrainte est que si votre plongement de mots est de dimension W, alors chacune de vos entrées textuelles sera désormais de dimension W, et donc les entrées de l'ensemble de données devraient pouvoir s'intégrer dans une matrice de forme (N,W), où N est la taille de l'ensemble de données et W est la dimension du plongement de mots.
☐	Expliquez tous vos choix.
☐	Entraînez un modèle linéaire simple en utilisant les représentations vectorielles comme entrées à l'aide de sklearn.linear_model.SGDClassifier. Utilisez le nombre d'itérations nécessaires pour former un modèle en 10 minutes, MAXIMUM (ne vous inquiétez pas, nous ne recherchons pas la précision dans ce cas-ci).
☐	Faites état des mesures pertinentes.

TÂCHE 4 — *Recommandations aux clients*

☐	Créez un ou plusieurs tracés finaux des mesures de rendement pertinentes de chaque expérience.
☐	Votre travail consiste à présenter les mesures et les tracés à chaque client, en fournissant une recommandation aux clients qui tient compte de tous leurs désirs et leurs besoins.
☐	Expliquez vos décisions.

TÂCHE 5 — *Soumettre l'examen à Statistique Canada*

☐	Référez-vous aux consignes qui se trouvent sous « Documents à soumettre » à la première page du présent document.

	Soumettez votre examen à statcan.fellowship-fellowship.statcan@statcan.gc.ca. Veuillez inclure : - le code entier utilisé pour accomplir la tâche et tout autre code auxiliaire requis; - le lien publique vers votre dépôt Git afin que nous puissions évaluer le code, les chiffres et la documentation.