Projet de traitement de données massives

Pour ce projet, vous allez implémenter deux solutions différentes pour un problème de traitement de données de Kaggle. Il s'agit d'un projet en équipe de 2, qui compte pour 25% de votre semestre. Vous êtes libre du choix de projet Kaggle, ainsi que du choix d'environnement et de langage de développement.

Il est fortement recommandé de choisir un projet de traitement de données numériques ou texte (pour ceux d'entre vous ayant préalablement pris un cours de traitement du langage) et dont l'objectif est de prédire un comportement ou une classe d'objets. Ces projets s'aligneront le mieux avec la matière du cours. À l'inverse, des projets de traitement d'images ou de reconnaissance d'objets (particulièrement par réseaux de neurones) sont à l'extérieur des sujets du cours et augmenteront grandement votre charge de travail dans le cours. Pour vous aider, voici une liste de suggestions de projets intéressants :

- Movie Recommendation (https://www.kaggle.com/c/movie-recommendation)
- San Francisco Crime Classification (https://www.kaggle.com/c/sf-crime)
- Bike Sharing Demand (https://www.kaggle.com/c/bike-sharing-demand)
- Swimming Pool Visitor Forecasting (https://www.kaggle.com/c/swimming-pool-visitor-forecasting)
- Train occupancy prediction (https://www.kaggle.com/c/train-occupancy-prediction)
- Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic)
- House Prices (https://www.kaggle.com/c/house-prices-advanced-regression-techniques)

L'équipe produisant le meilleur projet sera considérée pour le prix Pierre Ardouin (voir plus bas). Afin de se qualifier pour ce prix, les étudiants doivent produire un travail innovateur, écrire un rapport complet de haute qualité, et démontrer une performance parmi les meilleurs rangs de la compétition en question. Ce standard supérieur n'est appliqué que pour le prix Pierre Ardouin, et n'influence pas votre note finale au projet.

Première Partie : Planification (1%)

Noël est passé, le jour de l'an a été fêté, c'est le temps de se remettre au travail. Mais ne vous en faites pas, on va commencer tranquillement.

Vous devez vous trouver un partenaire et vous créer un compte sur https://inclass.kaggle.com/. Consultez la liste de compétitions actives, et choisissez-en une qui vous intéresse et qui est ouverte au public. Ça peut être une compétition en cours ou une déjà terminée, et ça peut être une compétition ordinaire plutôt qu'une de la section *inclass*. L'important est que vous puissiez télécharger les jeux de données, et que vous puissiez soumettre une solution pour obtenir une évaluation.

Je vous recommande fortement de créer immédiatement un petit script pour générer une soumission aléatoire pour votre compétition. Ça va vous permettre de vous assurer que vous comprenez bien ce qui est demandé et attendu.

Décrivez brièvement le projet. Quels sont les objectifs? Qu'est-ce qui constituerait une bonne solution ou un bon résultat? Comment prévoyez-vous vous y prendre, quelles idées ou intuitions avez-vous pour parvenir aux objectifs? Ce rapport devrait être d'environ 2 pages et est dû le 30 janvier 2019.

Deuxième Partie : Les données (7%)

Maintenant que vous avez choisi un projet Kaggle et téléchargé les fichiers de données de la compétition, vous allez regarder les données avec lesquelles vous devrez travailler un peu plus proche.

Discutez des données, les types d'attributs, et les propriétés statistiques des données. Quelles difficultés se présentent? (ex. : bruit, fléau de dimensionnalité, informations manquantes, déséquilibre des classes, etc.). Implantez des algorithmes de prétraitement des données afin de corriger ces difficultés. Décrivez ces algorithmes et discutez de leurs résultats. L'objectif ici n'est pas de faire une grande liste de statistiques sur les données, mais d'en tirer des leçons pour guider la réalisation du reste du projet. (2 points)

Discutez également de ce qui manque dans vos données. Quelles informations additionnelles voyez-vous comme nécessaire ou utile à avoir pour réaliser votre projet mais manquent au jeu de données de Kaggle? Où et comment pourriez-vous obtenir ces informations supplémentaires afin d'enrichir votre jeu de données? Comment est-ce que ces différents jeux de données vont être combinés dans votre système? (1 point)

Discutez également de la procédure de tests que vous envisagez. La majorité des projets Kaggle ne viennent pas avec des données tests, et l'option de soumettre un fichier de résultats pour évaluation ne retourne qu'un score numérique sans indications de ce qui a ou n'a pas fonctionné. On ne peut pas faire un projet en tâtant dans le noir! Donc, comment prévoyez-vous tester vos solutions afin d'obtenir une rétroaction qui pourra guider votre développement? (1 point)

Allez chercher des idées dans les travaux antérieurs. Votre première destination devrait être le forum de discussion associé à votre compétition Kaggle, où d'autres équipes peuvent avoir échangé des idées. Vous pouvez également contacter certaines des équipes plus hautes dans le classement pour leur poser des questions. Finalement, vous pouvez vous tourner vers la littérature scientifique, où des projets similaires ont déjà été réalisés. Ceci vous donnera des idées sur comment traiter vos données Kaggle, et comment planifier pour les prochaines étapes. N'oubliez pas de bien inclure vos références! (2 points)

Finalement, faites un lien avec votre premier rapport. À la lumière de ce que vous avez appris maintenant, les idées et intuitions que vous aviez précédemment sont-elles encore valide? Si oui,

laquelle prévoyez-vous implémenter en premier, et pourquoi semble-t-elle la plus prometteuse? Si non, comment allez-vous les mettre à jour? (1 point)

Le rapport pour cette partie devrait être d'environ 6 pages et est dû le 20 février 2019.

Troisième Partie : Premier traitement des données (7%)

Pour ce rapport, vous devez présenter le premier algorithme de traitement de données que vous avez implémenté, son fonctionnement et les résultats que vous avez obtenu.

Décrivez l'algorithme que vous avez choisi d'implanter. Décrivez, d'un point de vue technique, comment il fonctionne et les composantes clefs. Justifiez vos choix pour les décisions de design et d'implémentation que vous avez pris. (2 points)

Décrivez également les tests que vous avez faits jusqu'à présent. Pour chaque test, présentez les résultats obtenus. Présentez des statistiques pertinentes (taux de succès, précision, rappel, temps moyen de calcul, complexité algorithmique, etc.) et des études de cas comme exemples spécifiques. Décrivez autant les cas qui fonctionnent bien que ceux pour lesquels le test échoue, et discutez des raisons pour cette différence. (2 points)

En lien avec le rapport précédent, décrivez quels attributs des données sont utilisés par votre algorithme et comment. Quels attributs ont une valeur prédictive plus importante? Est-ce qu'ils correspondent aux attributs plus importants dans le rapport précédent? (2 points)

Basé sur les résultats expérimentaux que vous avez, quels nouveaux problèmes avez-vous identifiés, et quelles actions pourriez-vous prendre pour les résoudre? (1 points)

Le rapport pour cette partie devrait être d'environ 8 pages et est dû le 20 mars 2019.

Quatrième partie : Résultat final (10%)

Pour ce rapport, vous devez présenter le deuxième algorithme de traitement de données que vous avez implémenté, son fonctionnement et les résultats que vous avez obtenu. Cet algorithme doit être différent du premier, et non seulement un raffinement ou une variation du premier.

Ce rapport est très similaire au troisième rapport. Les premiers 7 points sur 10 sont identiques au rapport précédent.

Comparez les deux solutions que vous avez réalisées. Dans quelles conditions est-ce que une est préférable à l'autre, et pourquoi? Discutez autant les différences provenant de la nature des algorithmes, que les différences résultant des leçons prises dans le rapport précédent. (2 points)

Offrez une rétrospective sur le projet. En quoi avez-vous eu raison dans votre plan initial, et en quoi avez-vous eu tort? Si le projet était à refaire, que feriez-vous différemment? (1 point)

Soumettez votre résultat final à Kaggle et indiquez votre score et position. Incluez également la version finale de votre code (lien vers votre dépôt GIT ou fichier zip) avec votre soumission.

Le rapport pour cette partie devrait être d'environ 10 pages et est dû le 17 avril 2019.

Évaluation des rapports

Les rapports seront remis en-ligne à travers le site web du cours. Une seule soumission par équipe. Chaque rapport doit inclure une page titre indiquant les membres de l'équipe, leur niveau (1er, 2e, ou 3e cycle), la date de soumission, et le nom de projet Kaggle choisi (avec un lien vers la page web du projet). Veuillez inclure l'identification de votre projet Kaggle au début de chacun de vos rapports. Les rapports doivent être écrits en Word ou LaTeX (pas de rapports écrits à la main) et soumis en format PDF.

La majorité des points du rapport seront donnés sur l'analyse et la discussion de votre système et de vos résultats. Il est donc important (pour vous) d'écrire une analyse approfondie et scientifique. La question centrale n'est donc pas « qu'est-ce qui se produit », mais « pourquoi est-ce que ça se produit » et « qu'est-ce qu'on peut y faire ». Vous ne devez pas simplement écrire un algorithme et générer des résultats. Vous devez être en mesure de justifier vos décisions qui ont mené à votre algorithme, et expliquer pourquoi il a généré ces résultats.

Un exemple peut clarifier les choses. Supposons que vous avez créé un système de recommandation de films, que vous l'avez testé, et que vous avez trouvé qu'il recommande le film Sharknado de manière démesurée. Vous pouvez rapporter ce résultat de plusieurs manières :

- « Notre algorithme recommande le film Sharknado trop souvent. » Ceci n'est pas une analyse, mais simplement une observation des faits. Les points donnés seront minimaux.
- « Notre algorithme recommande le film Sharknado trop souvent parce que ce film obtient un score élevé dans notre algorithme trop souvent. » Ceci est l'inverse d'une analyse utile. Je ne donne pas de points, et je me réserve le droit de rire de vous.
- « Notre algorithme recommande le film Sharknado trop souvent parce que la description du film contient une immense liste de tous les mots clefs possibles, donc il se trouve à obtenir un score élevé dans un grand nombre de requêtes par mots clefs. » Vous avez identifié et analysé le problème et découvert sa source, bien joué! Vous avez des points.
- « Notre algorithme recommande le film Sharknado trop souvent parce que la description du film contient une immense liste de tous les mots clefs possibles, donc il se trouve à obtenir un score élevé dans un grand nombre de requêtes par mots clefs. Nous allons résoudre ce problème en assignant un poids aux mots clefs en fonction de la longueur de la description. » Non seulement vous avez découvert la source du problème, mais vous l'avez comprise assez bien pour proposer une solution, c'est fantastique. Vous aurez une bonne note.
- « Notre algorithme recommande le film Sharknado trop souvent parce que la description du film contient une immense liste de tous les mots clefs possibles, donc il se trouve à obtenir un score élevé dans un grand nombre de requêtes par mots clefs. Une solution possible serait d'éliminer la description des films de nos données, mais nous jugeons que ceci nous ferait perdre trop d'information utile étant donné que la vaste majorité des films ont des descriptions précises et informatives. Une autre solution possible serait de demander aux utilisateurs de noter la valeur

des descriptions. Cependant, nous anticipons plusieurs problèmes avec cette solution, par exemple que faire d'une nouvelle description qui n'a pas encore été notée par un utilisateur, et comment éviter le spam de notes positives? Finalement, on pourrait assigner un poids aux mots clefs en fonction de la longueur de la description, ce qui pénaliserait les films avec des descriptions trop longues sans affecter ceux qui ont une description brève et informative. C'est la solution que nous avons choisie d'appliquer. » Vous avez identifié le problème, vous l'avez analysé pour trouver sa source, puis vous avez exploré plusieurs pistes de solutions et justifié votre choix d'une en particulier. C'est parfait. 100%.

Pour les étudiants de premier cycle, ces critères seront appliqués en gardant en tête qu'il s'agit ici d'une de vos premières expériences à écrire ce type de rapport. Pour les étudiants de deuxième et troisième cycle, je tiens pour acquis que vous avez déjà une bonne expérience en rédaction et présentation scientifique, et l'application de ces critères sera plus stricte.

Notez que votre performance et rang dans la compétition Kaggle, quoique intéressant à souligner à titre indicatif, n'influencera pas votre note d'aucune façon. De même, des solutions qui donnent de bons résultats pour la compétition Kaggle mais ne nécessitent aucun traitement de données (exemple : trouver une base de données avec les bonnes réponses et écrire quelques requêtes SQL) ne vaudront aucuns points.

Notez finalement que jusqu'à 10% des points d'un rapport peuvent être enlevés en pénalité pour une mauvaise qualité. Ceci inclut particulièrement les fautes d'orthographe et de grammaire, les figures mal préparées (ou dessinées à la main), les rapports écrits à la main, les irrégularités de polices et tailles de caractère, et les textes incohérents.

Répétition dans les rapports

Chaque rapport doit être un document entier et cohérent. Donc, évitez les phrases du type « on a appliqué la technique du rapport précédent » ou les explications incomplètes qui ne peuvent être comprises qu'en lisant vos autres rapports. Réexpliquez les points importants.

Étant donné que les quatre rapports se font suite et traitent d'un même projet, il y aura naturellement des répétitions, particulièrement sur les parties du projet qui n'ont pas changé et les explications fondamentales. C'est acceptable de reprendre le texte de vos rapports précédents dans ce cas. Indiquez clairement dans ce cas quelle explication est reprise, et quelle est la nouvelle addition dans ce rapport. Prenez bien entendu compte des commentaires que j'ai indiqué dans le rapport précédent; je peux vous faire reperdre des points si les corrections demandées n'ont pas été faites.

Alternativement, il est acceptable pour vous de reprendre le rapport précédent et de le mettre à jour avec les nouvelles informations. Une fois de plus, indiquez clairement les nouveaux ajouts de ce rapport, et tenez bien compte des commentaires que j'ai fait sur le rapport précédent.

Il n'est pas acceptable de produire un nouveau rapport avec l'ancien rapport entier en prélude ou en annexe.

Équipes

Le projet doit être réalisé en équipes de 2 étudiants. La note sera donnée pour l'équipe, et non par individu. Étant donné la variation d'expérience à réaliser ce genre de projets entre les étudiants de 1er cycle et les étudiants de 2e/3e cycle, je ne pourrai pas permettre les équipes mixtes formées d'un étudiant du 1er cycle et un étudiant du 2e/3e cycle. La seule exception à cette règle est pour les étudiants de 1er cycle dans le profil distinction. Les équipes mixtes de 2e cycle et 3e cycle sont acceptables.

Dans le cas qu'un des deux coéquipiers d'une équipe abandonne le cours durant la session, le coéquipier restant aura deux options. Il peut premièrement choisir de continuer leur projet seul. Alternativement, il peut se joindre à un autre étudiant solitaire pour former une nouvelle équipe de deux, ou à une équipe de deux pour former une nouvelle équipe de trois. La nouvelle équipe choisira alors lequel des deux projets continuer et lequel laisser tomber.

Plagiat

Le plagiat est une offense académique sérieuse. Tout étudiant qui tente de soumettre un travail qui n'est pas le sien sera pénalisé. Ceci inclut de copier le travail ou rapport d'un autre étudiant du cours, la soumission d'un autre compétiteur sur Kaggle, ou un système trouvé ailleurs. Un étudiant coupable de plagiat recevra automatiquement la note de zéro pour le projet entier (c'est-à-dire toutes les quatre parties) et s'exposera à d'autres sanctions telles que décidées par l'Université.

Conseils

- Essayez plusieurs de vos idées et parlez-en dans vos rapports. Décrivez quelle est l'idée, pourquoi vous pensez que c'est intéressant à essayer (qu'est-ce que vous voulez découvrir ou que pensez-vous va arriver), et quel est le résultat obtenu (est-ce celui que vous attendiez, et sinon pourquoi). À force de réfléchir et d'expérimenter, vous trouverez une bonne solution. Et ce n'est pas mauvais que plusieurs de vos idées ne fonctionnent pas; c'est la nature même de la recherche! De plus, ça justifie expérimentalement que la version finale de votre système est la meilleure, et non simplement la première que vous avez essayé. Pour l'évaluation, je donne des points pour les explorations intéressantes (à condition qu'elles soient bien présentées, justifiées, et analysées, bien entendu). Je ne donnerai pas de points pour des idées farfelues ou mal présentées. Mais par contre je n'enlèverai jamais de points pour avoir essayé quelque chose. Et en contrepartie, si vous ne décrivez pas vos idées et expériences dans votre rapport, je ne peux pas vous donner de points du tout.
- Considérez les extrêmes logiques de vos idées. Par exemple, si augmenter le poids d'un attribut améliore les résultats, pourquoi ne pas l'augmenter encore plus, ou ne conserver que cette variable? Ce sera rarement le bon choix, mais d'explorer le comportement de votre système dans

les cas extrêmes peut souvent aider à mieux comprendre le problème et à développer une nouvelle intuition pour sa solution.

- Justifiez votre analyse avec des démonstrations mathématiques lorsque possible.
- Ne soumettez pas un copier-coller de votre code au complet dans votre rapport. Expliquez comment votre algorithme fonctionne en utilisant des descriptions du processus et des étapes, la logique du système, des formules mathématiques, et du pseudo-code.
- Je suis disponible durant mes heures de bureau pour vous aider en discutant de votre projet, des difficultés que vous rencontrez, et en suggérant des idées et des pistes. Je n'ai pas de solutions pour chaque projet. Et je ne vais pas déboguer votre code pour vous.

Prix Pierre Ardouin

Depuis l'automne 2013, le Département d'informatique et de génie logiciel a mis en place un concours récompensant l'équipe qui aura produit le meilleur TP/projet dans le cadre d'un cours. Ces travaux de session ont l'envergure d'un mini-projet qui est admissible par rapport aux normes fixées par le Département. À la suite des évaluations des travaux, l'enseignant du cours détermine l'équipe gagnante; chaque membre de l'équipe gagnante reçoit alors une bourse de 50$ ainsi qu'une attestation remises par le Département.

De plus, le Département d'informatique et de génie logiciel a mis en place une bourse Élite, appelée bourse « Pierre Ardouin », qui vise à récompenser le meilleur projet de session, tous cours confondus. Deux principaux critères guident le choix des évaluateurs dans l'identification du lauréat : l'excellence du travail (par rapport à ce qui est demandé dans l'énoncé) et l'aspect créativité/innovation. Il est actuellement prévu une bourse de 200$ pour récompenser chaque membre de l'équipe « élite » gagnante (pour un maximum de 1000$ pour toute l'équipe). Aussi, le Département veille à publier l'information sur un site Web dédié : http://www.ift.ulaval.ca/vie-etudiante/prix-pierre-ardouin.

À la deuxième moitié du mois de mai de chaque année universitaire, le Département organise une cérémonie pour honorer les finalistes et le lauréat du prix «Pierre Ardouin» des sessions d'automne et d'hiver, et leur remettre une attestation.