PROJET À COMPLÉTER

Segmentez des clients d'un site e-commerce

Note d'accompagnement

Si votre étudiant a commencé ce projet avant le 01/12/2021, il a débuté son travail sur **ce projet archivé**. Il peut continuer l'existant ou recommencer avec ce nouveau projet ; voici également le **guide mentorat de l'ancien projet**.

Le nommage des livrables à déposer sur la plateforme a été changé et des indications sur les temps de soutenance ont été ajoutées le 11/05/2022.
Une soutenance mérite un refus dans les cas suivants :

- Critères d'évaluation non-validés pour une ou plusieurs compétences.
- Plagiat (veillez à poser des questions méthodologiques ou sur la raisonnement de la solution pour s'assurer que le travail a bien été réalisé par l'étudiant.
- Une présentation en dessous de 15 min ou au-dessus de 25 minutes. Néanmoins, une tolérance de +/- 20% est permise, selon les cas.

L'intention de ce projet est de réaliser une **segmentation de clients**, au travers de la préparation et de la mise en œuvre de transformations de features numériques et catégorielles (features engineering), de la mise en œuvre des modèles d'apprentissage non supervisés, de la sélection d'un modèle et l'optimisation de ses hyperparamètres.

Le nombre de segments de clients doit permettre de bien différencier les clients, afin que l'équipe Marketing puisse mettre en place des actions ciblées, mais doit rester raisonnable afin de maîtriser leur charge de travail proportionnelle à ce nombre.

L'étudiant pourra s'inspirer de démarches de segmentation connues en fonction des données à disposition (segmentation RFM, segmentation Personae, segmentation "à la performance", segmentation comportementale...), afin de créer les features adaptées.

Incitez l'étudiant à mettre en œuvre des fonctions de comparaison des résultats des modèles selon différents indicateurs (score, temps de calcul...), permettant de choisir le plus adapté à la problématique et aux moyens techniques à disposition.

Remarque importante : Le fichier est assez pauvre en termes de données utilisables pour le clustering ; notamment, seuls 3 % ont réalisé plus d'une commande, ce qui peut paraître trop limité au départ pour les étudiants. Mais il est assez riche pour réaliser un clustering pertinent d'un point de vue métier, et acquérir les compétences techniques et métier attendues pour ce projet.

Si votre étudiant est sur le parcours Ingénieur IA, et qu'il demande les critères d'évaluation des compétences, n'hésitez pas à les lui communiquer (ceci ne s'applique pas aux étudiants du parcours Data Scientist).

Milestones

Voici un aperçu des étapes suggérées (mais notez bien que les étudiants ne sont pas tenus de réaliser les étapes du projet dans cet ordre). Cette section met également en évidence les difficultés potentielles. En particulier :

- La vitesse d'avancement du projet pour chaque étape est donnée à titre indicatif, mais varie en fonction de chaque élève.
- Le livrable fait référence aux résultats attendus de l'étudiant. Les étudiants ne sont pas non plus tenus de remplir ces "sous-livrables" suggérés pour valider le projet ; ils ne seront évalués que sur les livrables finaux.

Milestone 1 : Analyse exploratoire

- **Livrable** :
  - Notebook, partie préparation du fichier des commandes et de son analyse exploratoire.
- **Niveau d'avancement** : 10 %
- **Problèmes et erreurs courants** :
  - L'identifiant unique d'un client est le « customer_unique_id », à utiliser pour regrouper les commandes par client (un « customer_id » différent est associé à chaque commande).
- **Recommandations** :
  - Analyser le contenu de chaque table mise à disposition (features, valeurs).
  - Préparer un fichier de commandes, par « merge » des différentes tables.
  - Réaliser un describe(), vérifier s'il y a des valeurs manquantes.
  - Analyser par exemple la distribution du nombre de commandes par client, la distribution des montants, la distribution des catégories des produit etc.
  - L'étudiant constatera que seuls 3 % des clients ont réalisé plus d'une commande.

Milestone 2 : Création d'un fichier par client

- **Livrable** :

    - Notebook, partie feature engineering de création de features par client, à partir du fichier des commandes.
- **Niveau d'avancement** : 20 %
- **Problèmes et erreurs courants** :
    - Erreur consistant à supprimer les 3 % de clients qui ont plus d'une commande : ce sont les meilleurs clients, donc à conserver d'un point de vue métier marketing afin de les cibler et les gérer.
    - Certaines données sont liées à la commande (exemple : « payment_value » montant de la commande, ou « review_score ») ; d'autres sont liées à la ligne de commande (exemple : « price »), donc attention lors de la réalisation d'agrégations, en fonction du fichier utilisé (par commande « order_id », ou ligne de commande « order_item_id »).
    - Erreur consistant à prendre trop de features au départ et de natures très différentes, qui conduiraient à des clusters qui n'auraient aucun sens métier.
- **Recommandations** :
    - Conserver toutes les commandes, notamment celles des 3 % de clients qui en ont fait plusieurs.
    - Se concentrer dans un premier temps sur quelques features qui ont du sens du point de vue marketing, pour cibler les clients plus ou moins intéressants en termes de vente : RFM (Récence : durée depuis la dernière commande, Fréquence : nombre de commandes, Montant: par exemple montant cumulé des commandes).
    - Attention sur RFM : il faut garder des valeurs continues des features R, F, M, et non pas calculer des quantiles qui créeraient une distorsion de données (comme ce qui était réalisé autrefois sans machine learning).
    - Réaliser l'étape suivante de clustering avec ces 3 features, puis faire d'autres simulations avec des features supplémentaires, par exemple « review_score », voire d'autres features qui pourraient apporter de la valeur pour séparer les bons clients des autres.
    - Attention : l'ajout de features catégorielles comme la catégorie peut provoquer un brouillage du clustering, qui n'aurait plus de sens d'un point de vue métier. Cela peut faire partie de tests de la part de l'étudiant, mais n'est pas obligatoire.
    - Idéalement, la création du fichier par client sera développée via un « df.groupby(...).**agg**(...) ».
- **Ressources :**
    - Méthode RFM : https://www.definitions-marketing.com/definition/segmentation-rfm/

Milestone 3 : Élaboration d'un modèle de clustering

- **Livrable** :
    - Notebook de simulation d'algorithmes de clustering
- **Niveau d'avancement** : 75 %
- **Problèmes et erreurs courants** :

    - Erreur consistant à se contenter d'une approche purement technique pour déterminer le meilleur nombre de clusters.
- **Recommandations** :
    - Tester dans un premier temps l'algorithme k-means avec les 3 features RFM, avec pour objectif de déterminer le nombre optimal « k » de clusters d'un point de vue métier marketing :
        - Faire une simulation k-means avec un nombre « k » de clusters entre 2 et 15 (éviter d'avoir trop de clusters, ce qui serait trop complexe à gérer d'un point de vue métier).
        - Réaliser des mesures techniques afin de cibler les valeurs candidates de « k » : au minimum coefficient de silhouette et distorsion (méthode « elbow » ou « ducoude ») :
            - Le coefficient de silhouette permet un premier ciblage des « k » candidats, sans se limiter au coefficient de silhouette le plus élevé. Il apparaît avec les 3 features RFM que les valeurs « k » candidates sont entre 3 et 7 (peu de différence de la valeur du coefficient de silhouette).
            - La méthode du coude (avec le graphique de distorsion) permet de se focaliser sur les valeurs « k » qui correspondent au coude du graphique ou juste avant le coude. Dans notre cas des 3 features RFM, les valeurs « k » sont 3 et 4.
        - Ensuite réaliser des mesures et analyses orientées métier, afin de vérifier que les valeurs « k » sélectionnées permettent un clustering pertinent d'un point de vue métier, et déterminer le « k » qui répond le mieux au besoin métier :
            - Pour chaque valeur « k » sélectionnée, vérification du nombre de clients par cluster. Si ce nombre est trop faible pour certains clusters (par exemple <500, pour 95 000 clients), la valeur « k » n'est pas pertinente d'un point de vue métier.
            - Si elle est pertinente, l'analyse se poursuit par l'analyse des clusters, afin de déterminer les profils de clients de chaque cluster et s'assurer de la pertinence de chaque cluster d'un point de vue métier, par exemple :
                - Pour chaque feature, un graphique de boxplot par cluster.
                - Et/ou inversement, pour chaque cluster un graphique de boxplot par feature (c'est complémentaire aux premiers graphiques).
                - Des graphiques d'analyse bivariée entre 2 features, ou 3D entre 3 features.
                - Pour chaque feature, un calcul de moyenne par cluster.
            - L'analyse des graphiques permet de formaliser un profil de clients par cluster, par exemple « cluster 0 = clients avec des montants élevés et une fréquence de commande élevée », etc.

        - Cette analyse permet de valider que chaque cluster décrit une typologie de clients qui a du sens d'un point de vue métier (dans notre cas, pour séparer les clients plus ou moins intéressants), et qui est différente des autres clusters.
        - Le découpage en 3 ou 4 clusters devrait permettre de garder de bons candidats d'un point de vue métier.
    - Tester dans un deuxième temps un k-means avec une autre feature, par exemple le review_score, qui peut contribuer à séparer des clients plus ou moins intéressants, notamment traiter en marketing les clients plus mécontents et avec du potentiel. La même démarche sera mise en œuvre, mesure technique, puis approche métier pour déterminer les valeurs « k » optimales. Dans ce cas, le découpage en 4 ou 5 clusters devrait donner de bons candidats d'un point de vue métier.
    - Tester ensuite éventuellement avec d'autres features, afin de voir l'impact métier sur les clusters. Des features de natures différentes, non orientées « clients intéressants ou pas » (par exemple « catégorie ») risquent de brouiller les clusters et leur intérêt d'un point de vue métier ; il faut donc rester vigilant à ne pas aller trop loin, voire constater que cela n'apporte rien.
    - Tester ensuite d'autres algorithmes (au moins 2) pour comparer les résultats avec le k-means. Par exemple, il peut être intéressant de tester un DBSCAN et un agglomerative clustering.
        - Pour le DBSCAN, l'étudiant pourra faire varier la valeur de l'hyperparamètre « eps », avec un hyperparamètre « min_sample » fixé à 100 (en dessous, le nombre de clients par cluster n'est pas suffisant, comme mentionné plus haut) :
            - Il constatera qu'il n'obtiendra pas un nombre de clusters intéressant.
            - Il peut également mettre un « min_sample » =10, et constater que pour chaque valeur de l'eps, les clusters proposés ne contiennent pas tous un nombre suffisant de clients, et/ou le nombre de clients sans cluster (cluster « -1 ») est trop élevé.
            - Donc l'analyse graphique métier n'est pas utile. Ceci démontre que le DBSCAN n'est pas approprié. Intuitivement, cela s'explique par le fait que le DBSCAN fonctionne par densité, et que dans notre cas la densité des 3 000 bons clients (qui ont commandé plusieurs fois) est faible.
        - Pour l'agglomerative clustering, les résultats devraient être assez similaires au k-means ; par contre, les temps de traitement risquent d'obliger l'étudiant à travailler sur un sample du fichier.
- **Ressources :**
    - Librairie Yellowbrick d'évaluation technique de clusters (coefficient de silhouette, distorsion) : https://www.kaggle.com/kautumn06/yellowbrick-clustering-evaluation-examples

Milestone 4 : Contrat de maintenance – Simulations

- **Livrable** :
  - Notebook de simulation d'évolution des clusters
- **Niveau d'avancement** : 100 %
- **Problèmes et erreurs courants** :
  - Erreur consistant à ne pas utiliser le « transformer » StandardScaler du fichier qui a servi à l'entraînement du modèle (« fit » du modèle).
  - Erreur consistant à comparer des listes de clusters prédits à 2 dates différentes (cela ne concerne pas les mêmes clients et pas le même nombre, ni les mêmes données de clients).
- **Recommandations** :
  - L'objectif est de déterminer au bout de combien de temps le modèle de clustering entraîné initialement proposé (donc « fit ») devient obsolète (quand les prédictions, « predict », ne sont plus pertinentes), nécessitant d'entraîner un nouveau modèle de clustering.
  - Pour prendre un exemple, supposons que l'entraînement du modèle initial M0 ait été réalisé à T0 pour un fichier clients F0, qui donne la liste des numéros de clusters C0, CO=M0.fit(F0).
  - À T1=T0 +n jours, un nouveau modèle M1 est entraîné sur le nouveau fichier clients F1 à T1, et donne une nouvelle liste de clusters C1, C1_new = M1.fit(F1).
  - Si on utilise le modèle initial M0, à T1 la prédiction des numéros de clusters du fichier F1 des clients à T1 donne C1_init = M0.predict(F1).
  - Il s'agit de comparer les numéros de clusters à T1 du fichier F1, selon que l'on utilise le modèle initial créé à T0(M0) via un « predict », ou le modèle créé à T1 via un « fit ».
  - Pour un k-means, les numéros de clusters ne correspondent pas forcément d'une simulation à l'autre. Pour mesurer la divergence des clusters, il est conseillé d'utiliser l'ARI, indépendant de la numérotation des clusters.
  - Il s'agit donc de simuler plusieurs périodes T1, T2 à Tt, et d'afficher l'évolution de l'ARI. Si l'ARI passe en dessous de 0.8 (correspond environ à 0,9 en accuracy), il est sûrement pertinent de reproposer un entraînement de modèle au client.
  - Le fichier clients Fi à date Ti sera créé à partir de toutes les commandes passées jusqu'à Ti. Une fonction permettra de générer ce fichier juste en passant la date du fichier (filtrage de toutes les commandes jusqu'à Ti et création des features).
  - Attention, les fichiers Fi sont les fichiers clients transformés par un StandardScaler, celui qui a servi à standardiser les données d'entraînement du modèle concerné. Donc pour le calcul de C1_init, il faut utiliser le StandardScaler du modèle M0 (fit sur le F0), et pour C1_new, celui du M1 (fit sur le F1).
  - Le délai entre 2 simulations (« n jours » entre Ti et Ti+1) doit être suffisamment court pour déterminer assez précisément le délai de maintenance du modèle (1 semaine, 15 jours).
  - La date de début T0 doit être la plus proche possible de la date de fin du fichier (août 2018), tout en intégrant le délai de simulation de maintenance, non connu au départ. Il faudra donc procéder par itération. Par exemple tester avec T0 au

31/12/2017, et si le délai de maintenance déterminé est de 3 mois (ARI <0.8), refaire une simulation à T0 = août 2018 – 3 ou 4 mois, soit avril ou mai 2018, pour finir la simulation courant août au plus près de la date du fichier complet.
    - Une autre solution est de faire une simulation en marche arrière, en faisant attention à faire le « predict » sur le modèle le plus ancien.
    - Il peut être intéressant, mais pas obligatoire, de calculer l'accuracy à la place de l'ARI, ce qui nécessite de « recaler » les numéros de clusters entre les 2 listes. Ceci peut se faire en réalisant une matrice de confusion et en utilisant la fonction argmax() pour déterminer la translation de numéro de cluster. Cette transformation ne fonctionne que si l'accuracy est élevée.
    - Il peut être intéressant, mais pas obligatoire, de regarder la divergence des clusters par numéro de cluster, le but étant de s'assurer de la stabilité en priorité des clusters de « bons clients ».
- **Ressources :**
    - ARI : https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html

Évaluation des compétences

Transformer les variables pertinentes d'un modèle d'apprentissage non supervisé

❒ Les variables pertinentes ont été transformées pour permettre leur exploitation.

❒ Une ou plusieurs variables pertinentes permettant d'améliorer la solution proposée ont été créées.

Mettre en place le modèle d'apprentissage non supervisé adapté au problème métier

❒ Le nombre de segments et la répartition sont adaptés à la problématique métier.

❒ La stratégie d'ajout de nouveaux clients a été explicitée.

❒ La nature des variables d'entrée a été prise en compte dans le choix de l'algorithme et de la distance.

Évaluer les performances d'un modèle d'apprentissage non supervisé

❒ La forme des clusters est évaluée.

❒ La stabilité des clusters est évaluée.

Adapter les hyperparamètres d'un algorithme non supervisé afin de l'améliorer

❒ Les étapes d'évaluation sont automatisées pour tester facilement plusieurs combinaisons de paramètres.

❒ Les valeurs de paramètres testés sont pertinemment choisies.

Respecter la convention PEP8

❒ La convention PEP8 est respectée.

❒ Le code est commenté (commentaires réguliers, docstrings dans les fonctions).