ENSAE ParisTech

Python pour le Data Scientist

Prédiction de la fidélisation de la clientèle d'une entreprise de e-commerce basée à Berlin

Réalisé par : Khaled Mansour

2017/2018

Table des matières

1 Introduction

Les clients passent souvent une seule commande pour une boutique en ligne spécifique. Un parmi les objectifs du système de gestion de la relation client (CRM) est de maximiser la valeur vie client (CLV) en incitant les clients à retourner au magasin (que ce soit en ligne ou pas).

Une méthode courante pour le faire est d'envoyer des coupons et des bon de réduction quelque temps (jours ou semaines) après une commande. Cependant, un coupon représente un cout de manque à gagner pour l'entreprise lorsqu'il est utilisé. Dans le cas où un client aurait fait un achat de suivi même sans l'incitation d'un coupon, la valeur de ce dernier serait effectivement gaspillée. Pour cette raison, l'entreprise veut optimiser en utilisant seulement des coupons prometteurs et qui sont bien ciblés.

Dans le cadre de ce projet, un détaillant en ligne basé à Berlin a fourni des données réelles à l'université Humboldt à Berlin. Les données sont la propriété intellectuelle de l'université et une permission à réutiliser les données a été bien prise en compte avec l'accord de Professeur Stefan Lessmann ; chef du département « système d'information » à Humboldt (Annexe).

L'objectif consiste à identifier les clients dont on peut s'attendre à ce qu'ils achètent à nouveau dans les 90 prochains jours en fonction des caractéristiques du client, conditions de commande et les produits commandés. Les clients identifiés ayant un potentiel retour ou achat de suivi ne recevront pas de coupon. La boutique en ligne estime que l'envoi d'un coupon à un client, qui n'envisage pas refaire une autre commande, a une probabilité de 20% pour convaincre le client à refaire une autre commande d'une valeur moyenne de 25 euro (avant la réduction du coupon).

L'objectif de cet étude est de maximiser les revenus du détaillant en fournissant une liste des clients prometteurs à cibler par les coupons. Cette liste est une estimation binaire (0/1) si le client revient naturellement dans les 90 jours suite à l'achat initial. La performance du modèle de prédiction devrait prendre en compte le gain net des revenus. Dans le cas de cette boutique en ligne, les couts et les gains sont asymétriques. Envoyer un coupon à un client qui aurait retourné de toute façon entraine une perte effective de 10 euro. Ne pas envoyer de coupon à un client qui ne prévoit pas retourner entraine un bénéfice attendu de 3 euro. La matrice des coûts est représentée dans le tableau ci-dessous où client non fidèle est celui qui n'envisage pas faire un achat de suivi et le client fidèle est celui qui compte faire un autre achat naturellement.

		Valeur réelle	
		Non fidèle	Fidèle
Prédiction	Client non fidèle (0) / coupon	3	-10
	Client fidèle (1) / pas de coupon	0	0

Tableau1. Matrice de coût pour le modèle.

Parmi les questions qu'on cherche à répondre dans cette étude sont :

- Quel modèle de machine learning arrive le mieux à faire la prédiction de cette liste des clients à cibler en termes de performance et de précision ?
- Comment peut le modèle de machine learning maximiser le profit du détaillant avec les résultats de la prédiction ?

2 Aperçu et Analyse du jeu des données

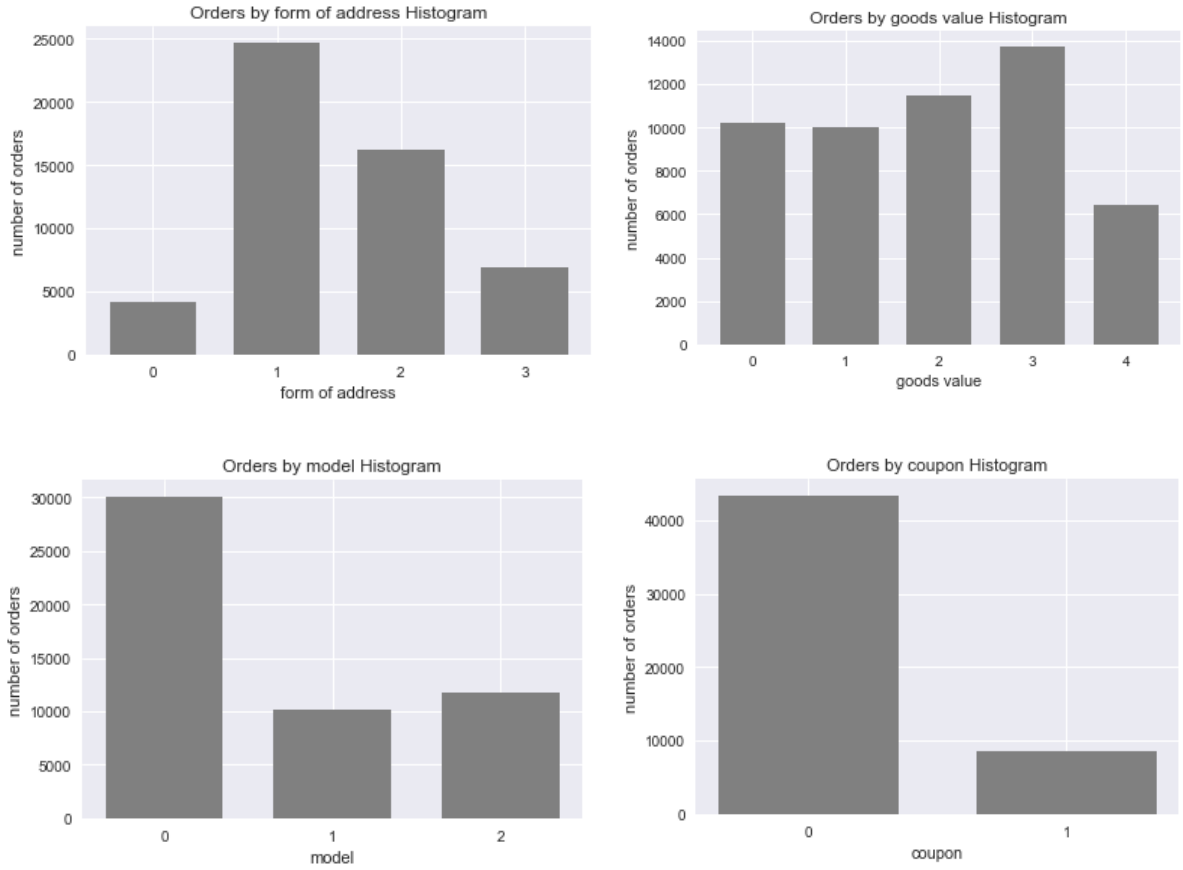
Afin d'implémenter un modèle de machine Learning pour la prédiction de la fidélisation de cette entreprise, des données réelles du détaillant en ligne ont été utilisées. Il y a deux ensembles de données. Le premier ensemble est nommé « train » et présente l'enregistrement de 51884 ordres ou commandes avec 38 variables (l'explication de chaque variable se trouve dans l'annexe). Le deuxième ensemble des données est appelé « test ». C'est l'ensemble de données que j'ai utilisé pour appliquer les algorithmes de machine Learning afin de prédire le retour des clients. Les deux ensembles de données ont 37 variables explicatives qui décrivent les caractéristiques de chaque client mais aussi le comportement de l'ordre (période de temps, quantité).

Notre variable cible est "return customer" qui est seulement fourni dans « train ». La variable cible "Return customer" est une variable binaire avec "1" qui signifie que le client a passé une autre commande dans les 90 jours et dans un autre cas est "0". Selon l'ensemble de données « train », plus de 81% des clients étaient des clients non-retour (9773 sur 51884).

Selon l'analyse descriptive du jeu des données, le montant moyen des articles remis et annulés est de 0,12, ce qui signifie que sur 100 commandes, une moyenne de 12 commandes sont soit remises, soit annulées. Ceci est un taux élevé. Le nombre moyen d'articles achetés est d'environ 2 articles par commande. La durée de livraison réelle moyenne est de 6 jours, mais l'écart entre la durée réelle et la durée estimée est remarquable ; 18 jours en moyenne. Ceci implique l'existence d'un système d'estimation de durée de livraison qui n'est pas précis.

Encore, 20% de toutes les commandes sont en période de Noel, ce qui peut expliquer la commande unique sans rachat dans les 90 prochains jours. Le délai pour faire une commande depuis la création du compte client est de 1,4 jour en moyenne (cela reflète que la plupart des commandes ont été effectuées le même jour que celui de la création de compte). Quand il s'agit du nombre de commande par type de produit, on remarque que le nombre de livres est le plus élevé (moyenne égale à 0,9 ce que veut dire 90% de toutes les commandes) suivi en deuxième position par le nombre de livres de poche « paperbook » et en 3eme position par le nombre de livres d'école « school book ». Enfin, le coupon est utilisé en moyenne 0,16, ce qui signifie 16 fois sur 100 commandes ce qui reflètent la performance faible du système des coupons mais le plus important c'est que 41% de toutes les commandes ont été référés reflétant l'importance du référent pour les prochaines commandes.

Les quatre graphes ci-dessous présentent des histogrammes pour quatre variables ;
« coupon », « goods value » , « form of address » et « Model » (code python dans le notebook)



Graph1. Histogrammes de quatre variables

Selon les graphes, on constate que « form of address » la plus fréquente est 1 (Mr) puis 2 (Mrs) puis 3 (other) puis 0 (company). Ce que veut dire que la plupart des commandes sont faites par des individus et non pas par des sociétés. En fait, presque 50% de retour sont de forme 1 (homme) et 30% de forme 2 (femme). La « good value » 3 est la plus élevée avec 2738 sur 9773 clients récurrents (environ 27%). La valeur 0 et la valeur 2 viennent en deuxième position (20%) puis 1 et en dernière position la valeur 4. Les clients avec la forme d'adresse 1 avec une commande de « good value » 3 ou 0 ou 2 ont une forte probabilité de retour. Quand il s'agit du modèle, le modèle 0 compte pour plus de 50% des clients qui reviennent suivi par le modèle 2 puis finalement le modèle 1. Selon l'histogramme de la variable coupon et la table ci-dessous, clairement le problème ici est le nombre élevé de clients ou de commandes utilisant coupon (valeur 1), mais ne reviendra pas dans les 90 prochains jours, ce qui conduit à gaspiller des coupons. Sur les clients revenant seulement 1388 (environ 14%) ont utilisé des coupons.

coupon	0	1
non returning	35031	7080
returning	8385	1388

Tableau2. « returning » et « non returning » par coupon

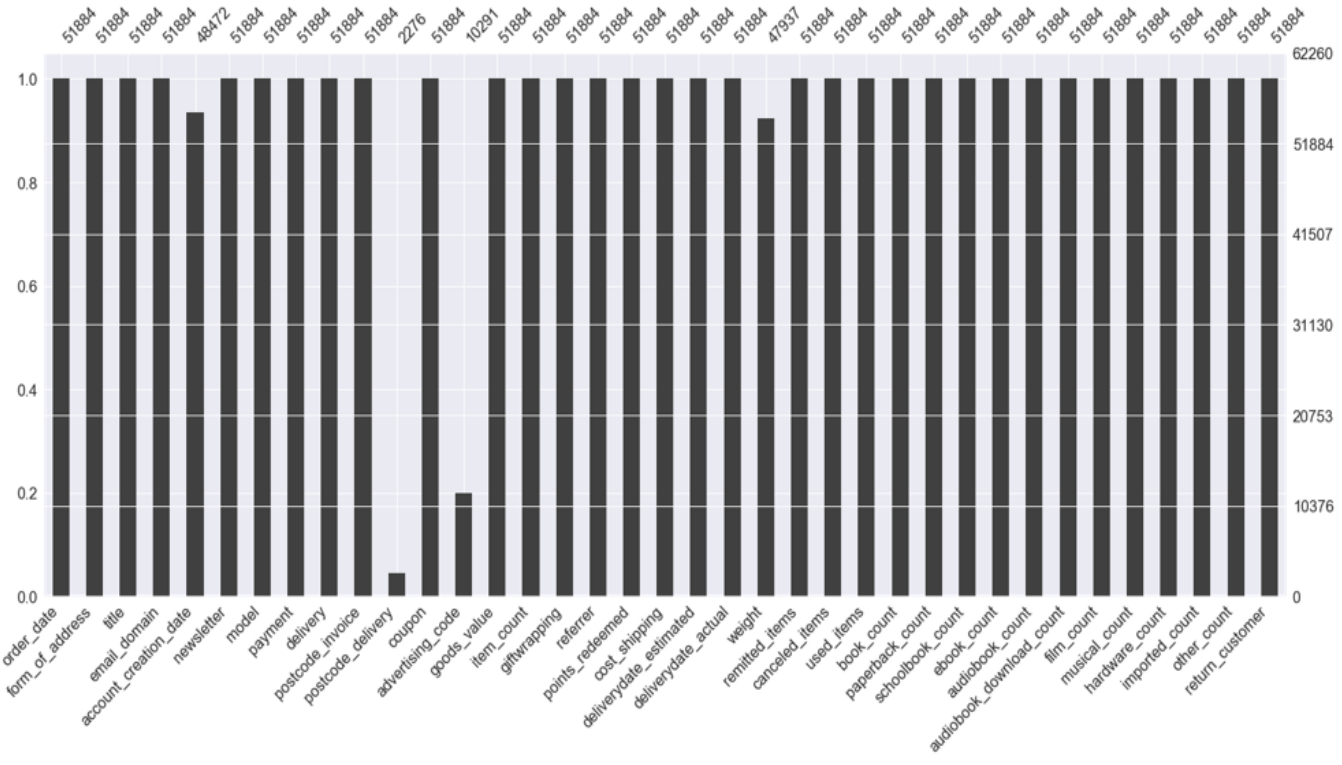
3 Démarche

3.1 Préparation des données

Valeurs manquantes

Dans cette étape « train » et « test » ont été rassemblé ensemble sous un seul ensemble de donnée « comdata » afin de faire le prétraitement des données. Après avoir faire la visualisation pour vérifier s'il y a des observations manquantes (Graph2 ci-dessous), on a conclu que ces trois variables ; « weight », « postcode_delivery » et « advertising_code » doivent être traitées.

- Postcode_delivery : Cette variable manque plus de 90% des observations. L'Imputation de ces valeurs sera une hypothèse forte. Cependant, une hypothèse possible derrière cette grande quantité de valeurs manquantes est que les valeurs de « postcode_delivery » sont les mêmes que celles de « postcode_invoice ». Par conséquence, cette variable sera supprimée de l'analyse et des prédictions suivantes et seule la mention "postcode_invoice" sera prise en compte.
- Weight : cette variable sera abandonnée aussi en supposant que cela n'a aucun effet sur le désabonnement ou la rétention des clients vu que l'information de poids est fournie sur le site et que c'est la volonté du client de choisir le produit en connaissant son poids donc cela ne devrait pas l'empêcher de revenir. Aussi beaucoup de produits sont téléchargeables et n'ont aucun poids ce qui explique les valeurs manquantes.
- Advertising_code: une variable fictive (dummy) sera créée à la place en prenant la valeur 1 si un code publicitaire a été utilisé et 0 dans le cas contraire .



Graph2. Visualisation des valeurs manquantes

Variables supplémentaires et type de variables

Dans cette étape on a commencé par supprimé la variable « points_redeemed » puisque toutes les observations prennent la valeur 0 et par conséquent cette variable n'a aucun effet et doit être supprimée. Ensuite j'ai créé les nouvelles variables suivantes :

- "timelapse_to_order" : l'objectif est de calculer l'intervalle de temps entre la création d'un compte et la première commande. Pour faire ceci, order_date et account_creation_date doivent être convertis en datetime.
- « estimated_delivery_duration » : Cette variable est la différence entre la date de livraison estimée par l'entreprise et la date de la commande. L'objectif est de calculé la durée de cette livraison estimée
- « actual_delivery_duration » : Même chose que la variables précédente mais celle-là pour calculer la durée pour la livraison actuelle. C'est la différence entre date de livraison actuelle et date de la première commande
- « delivery_gap » : c'est la différence entre la date de livraison estimée et la date de livraison actuelle pour capturer la différence entre ce qui est promis et ce qui est réel et voir son influence sur le comportement de l'acheteur.
- « total_remitted_canceled » : ici , on a groupé ensemble les articles annulés avec les articles remis puisqu'il s'agit de la même chose ; la non satisfaction du client.
- « purchased_item_coun » : cette variable et la différence entre « item_count » ou bien le nombre total des articles par ordre et « total-remitted_canceled » ou bien le nombre total des articles remis et annulés . Cette variable reflète le nombre réel des articles finalement retenus ou achetés par les clients.
- « multiple_items »: une variable fictive « dummy » qui prend la valeur 1 si plusieurs articles ont été achetés et 0 si seulement 1 ou 0 article a été acheté (0 possible quand la commande est totalement annulée ou remise)
- « Christmas » : une variable fictive « dummy » qui prend la valeur 1 si la commande a été passée en période de Noel (entre 2013-11-17 et 2013-12-24 dans notre cas) et 0 autrement.

Maintenant, nous allons mettre des variables à leurs types correspondants où les dates comme datetime et convertir le type d'objet en type catégoriel (y compris le client de retour) et le reste en numérique comment suivant :

- En type « datetime » : « order_date » , « account_creation_date » , « deliverydate_estimated", "deliverydate_actual".
- En type catégorielle: « form_of_address » , « email_domain », « payment » , « advertising_code » , « title » , « newsletter » , « model" , "delivery" , "postcode_invoice" , "coupon", "goods_value", "giftwrapping", "referrer", "cost_shipping" , "multiple_items" , "Christmas" , "return_customer"
- En type entiers (numérique) : le reste des variables

Ensuite on passe à l'encodage des valeurs catégorielles à l'aide d'un « one hot encoding » avec LabelEncoder de la librairie « sklearn.preprocessing ». Ceci est une étape importante. Même si de nombreux algorithmes d'apprentissage automatique peuvent prendre en charge des valeurs catégorielles sans autre manipulation, il existe de nombreux autres algorithmes qui ne le sont

pas. Donc, il vaut toujours mieux transformer ces valeurs catégorielles en valeurs numériques pour les analyses suivantes avec n'importe quel algorithme.

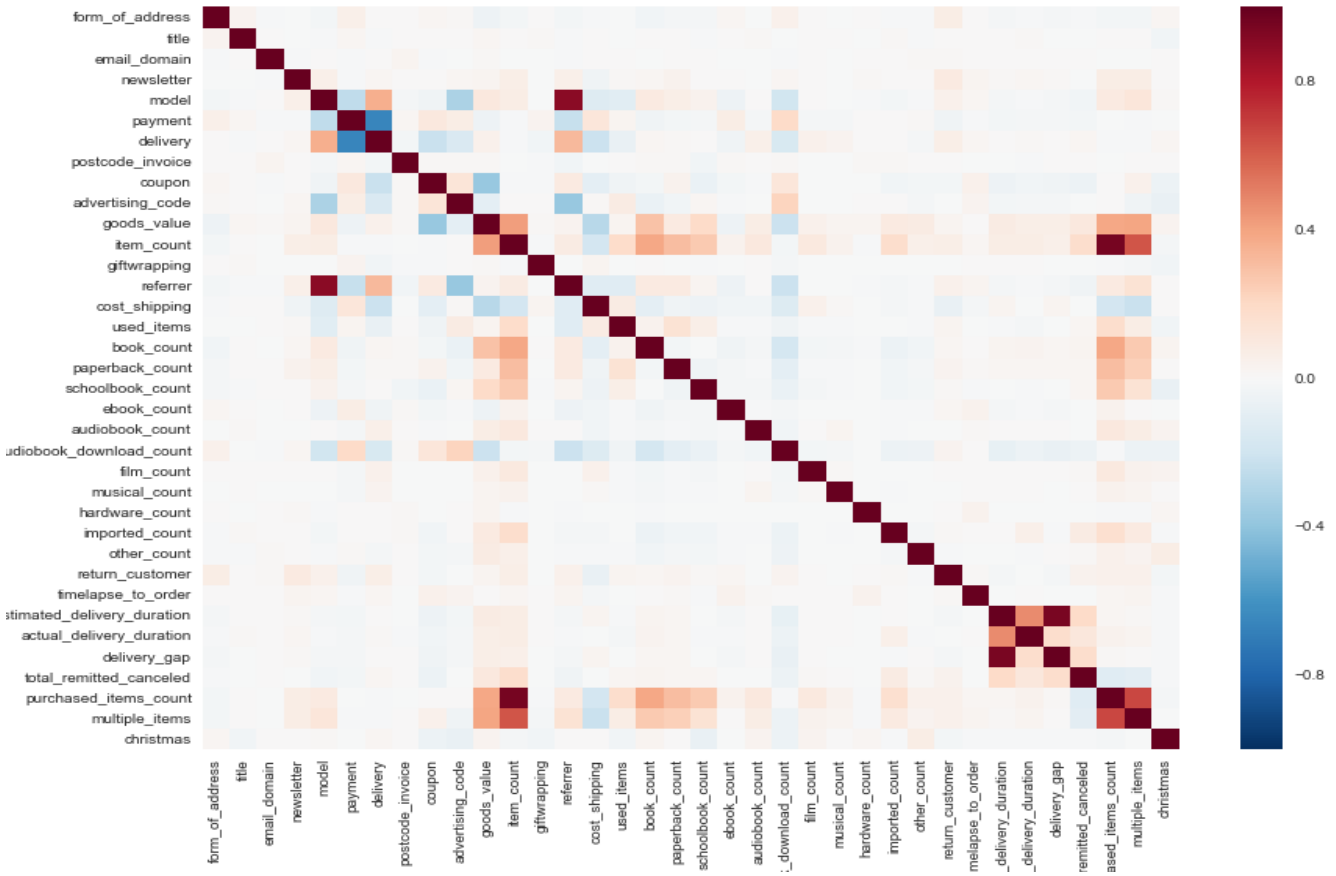
Après avoir fini cette étape de prétraitement des données, on divise les données de l'ensemble « comdata » en deux ensembles : « train » et « test » pour l'analyse exploratoire des données et les modèles d'apprentissage dans les prochaines étapes.

3.2 Sélection de variables et Cross Validation

Sélection de variables

La sélection des variables a été faite sur 3 étapes :

- Etape 1 : Sélection manuelle en supprimant la variable « item_count » puisque la nouvelle variable « purchased_item_count » reflète mieux le nombre des articles achetés à chaque commande.
- Etape 2 : Sélection par l'élimination des variables avec forte corrélation. Le seuil limite était fixé à 0.7. Le graphe 3 ci-dessous présente la corrélation intra-features.



Graph3. Corrélation intra-features

- ⇨ Selon le graph, on constate une corrélation positive entre items_count et purchased_items_count , ce qui était attendue. Cependant, nous avons une corrélation positive entre « model » et « referrer ». Enfin, cette corrélation positive existe également entre l'écart de livraison « delivery_gap » et la durée de livraison estimée. Une corrélation négative remarquable existe entre les variables de paiement et de livraison.

- Etape 3 : Sélection automatique des features basé sur un algorithme. L'algorithme utilise la fonction intégrée de « feature_importances_ » sur sklearn tout en dépendant de l'input. Dans notre cas, c'est le « automatic_feature_selection_params ». Ce dernier est déterminé lors du choix de modèle prédictif.

Cross Validation

La conception était initialement basée uniquement sur la validation croisée k-fold. Cette méthode divise les données en partitions paires et effectue k cycles différents, chaque partition étant utilisée une seule fois pour la validation. Comme il est nécessaire de régler les différents paramètres du modèle et d'obtenir ensuite une estimation non biaisée des performances du modèle, la manière souhaitée de faire cela nécessite une validation croisée imbriquée k-fold, c'est-à-dire une boucle externe qui traite les ensembles « test » et une boucle interne où la sélection et l'ajustement du modèle sont effectués. Cela représentait trop de ressources pour toutes les prestations proposées et le coût supplémentaire de la boucle extérieure n'était pas justifié par les avantages qu'il offrait. La procédure utilisée pour la formation de toutes les méthodes reposait sur les éléments suivants :

- Diviser les données en deux grands ensembles avec respectivement 85% et 15% des données.
- En utilisant le grand ensemble, effectuer la validation croisée et la sélection du modèle en utilisant la validation croisée k-fold. Dans ce projet, k = 4.
- Une fois que tous les modèles ont été formés et que le meilleur a été identifié, utilisez ce modèle sur l'ensemble de test et évaluez les performances. Ceci est l'estimateur final pour notre modèle. La décision n'est pas basée sur l'ensemble de test, l'ensemble de test sert uniquement comme estimation.
- Entraînez le modèle final avec toutes les données en utilisant les paramètres que nous avons trouvés.

3.3 Sélection de model et mesure de performance et précision

Mesures

Comme l'objectif de la société est de maximiser les bénéfices, il est clair que la mesure de performance primaire doit être basée sur le bénéfice attendu. Étant donné que nous souhaitons comparer les bénéfices entre différentes tailles d'échantillon et différentes divisions, il est plus pratique de calculer le bénéfice attendu par client. Je marque cette quantité comme ROI (retour sur investissement), tout au long du code et du papier. Bien que cette mesure prenne normalement en compte l'investissement initial, je n'en tienne pas compte et le dénominateur total de « l'investissement » n'a pas de coût. La mesure obtenue est alors calculée comme suit :

$$ROI(y^) = 1/N\ (3 *TN - 10* FN)$$

où TN et FN sont les nombres de vrais négatifs et de faux négatifs respectivement, avec un négatif dans notre cas étant une personne qui ne reviendrait pas faire un autre achat. En outre, il est utile d'examiner l'impact du seuil utilisé (threshold) dans les modèles qui affectent les probabilités aux classes et son impact sur les bénéfices. Nous utilisons cela au lieu de regarder les courbes ROC. Ne souhaitant pas les jeter complètement, nous utilisons également la mesure de la zone sous la courbe (AUC) et la mesure de précision aussi bien que la « confusion matrix ».

Modèles

La sélection du modèle, l'évaluation de l'hyperparamètre et la formation ont été effectuées en utilisant le package scikit-learn pour Python, avec les paquets NumPy, SciPy et matplotlib sur lesquels il est construit. Pandas étaient utilisés pour gérer les trames de données dans Python. Le imbalanced-learn package a également été utilisé, qui est une branche de sklearn.

J'ai essentiellement utilisé le package sklearn afin d'extraire les features. J'ai ensuite implémenté trois modèles prédictifs :

- Random Forest
- Gradient Boosting
- Logistic Regression

Mon premier choix de classificateurs était basé sur des arbres de décision, à savoir les forêts aléatoires et le renforcement du gradient (Gradient Boosting). Ils sont connus vu qu'ils peuvent bien fonctionner avec des types de données mixtes, résistent à une mauvaise mise à l'échelle des variables et, grâce à leurs diverses techniques de régularisation intégrées, sont probablement plus tolérants aux erreurs possibles dans la phase de traitement des données. Ils ont aussi l'avantage d'avoir la méthode sous-jacente la plus compréhensible pour générer des prédictions et sont les plus intuitifs parmi les méthodes étudiées.

La régression logistique a été incluse en raison de sa popularité dans diverses sources que j'ai rencontrées. De plus, de bonnes performances pourraient indiquer l'applicabilité potentielle d'autres méthodes linéaires au problème.

Enfin, en raison de la nature embarrassante et parallèle de l'entraînement de l'algorithme « random forest », c'est-à-dire des tâches d'entraînement de divers arbres complètement indépendants les uns des autres, j'ai pu utiliser les capacités de parallélisation intégrées de Sklearn. Le code résultant pourrait efficacement utiliser tous les noyaux (cores) pendant la formation accélérant sensiblement le processus.

4 Résultats

On remarque qu'en générale les trois algorithmes sont proches en termes de performance et de précision avec une performance meilleur pour les algorithmes basés sur les arbres de décision ; notamment « random forest » et « GBM » et un avantage léger pour l'algorithme des forets aléatoires.

Les résultats obtenus sont optimisés dès le début avec la fonction randomsearchCV pour l'optimisation des hyperparamètres. La comparaison entre les 3 modèles est faite essentiellement en comparant leur performance avec les 3 mesures : ROI , ROC AUC et ACCURACY en jouant sur le changement à chaque fois sur la sélection des variables soit d'une façon automatique soit manuelle soit en éliminant les variables avec forte corrélation. Après en deuxième lieu, le temps de calcul ou vitesse d'apprentissage de chaque modèle prédictif avec les différents scénarios de sélection de variables est comparé avec le temps de calcul des autres modèles.

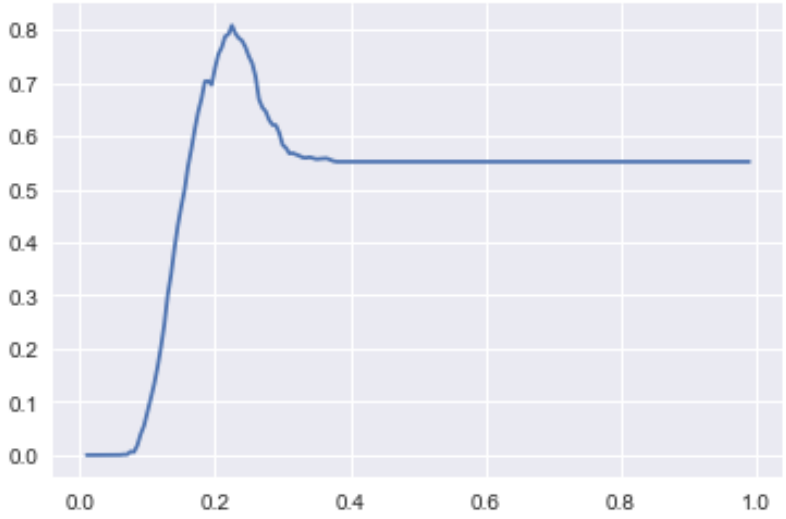
Le tableau suivant regroupe l'ensemble des résultats de performance des trois modèles sur le test set, en fonction des features utilisées en entrée dans le train set.

	ROI (moyenne)	**Score ROC**	**Accuracy**	**Vitesse d'apprentissage**
1/Random forest (les forêts aléatoires)				
Toutes les features	0.8074	0.6494	71.2%	143.8 mn
Sélection automatique des features	0.7492	0.6343	76.5%	12.3 mn
Sélection par élimination des variables corrélées	0.7936	0.6484	69.6%	112.9
Sélection automatique + élimination des variables corrélées	0.7417	0.6328	75.2%	11.8 mn
2/Gradient Boosting				
Toutes les features	-	-	-	-
Sélection automatique des features	0.7490	0.6314	69.9%	38.1 mn
Sélection par élimination des variables corrélées	-	-	-	-
Sélection automatique + élimination des variables corrélées	0.7598	0.6292	71.2%	35.1 mn
3/Régression logistique (logistic regression)				
Toutes les features	0.7291	0.6276	68.2%	9.5 mn
Sélection automatique des features	0.7094	0.6099	74.5%	5.2 mn
Sélection par élimination des variables corrélées	0.7382	0.6272	69.1%	7.5 mn
Sélection automatique + élimination des variables corrélées	0.7096	0.6108	74.5%	5.1 mn

Tableau3. Résultats comparatifs des trois modèles prédictifs

Selon le tableau3, le modèle qui a fourni les meilleur résultats sur tous les niveaux (accuracy , ROI , ROC score) est le modèle de « random forest » avec toutes les features et pas de sélection variables automatique ni par élimination des variables corrélées. Le graph4 ci-dessous nous montre une visualisation assez intuitive, la courbe qui reflète la relation entre la moyenne retour sur

investissement sur chaque client (ROI) et le seuil optimal (Threshold) qui affecte les probabilités dans le modèle prédictif.



Graph4. Calcul du seuil optimal en fonction de ROI

Pour conclure, l'exercice de prévision des clients fidèles ou bien les clients qui vont refaire une autre commande dans les prochaines 90 jours, semble un peu complexe mais surtout prend beaucoup de temps pour avoir une prédiction de haute précision et performance. Le meilleur résultat était fourni par le modèle des forêts aléatoires avec toutes les features en termes de performance, précision et Retour sur investissement (ROI). Néanmoins, le seul problème est le temps de calcul ou la vitesse d'apprentissage qui est un peu élevée pour ce modèle ce qui nous mène à d'autres améliorations pour réduire ce temps de calcul tout en gardant cette performance. Une alternative serait de choisir l'implémentation de l'algorithme des forêts aléatoires aussi mais avec la sélection automatique des features ce qui donne une prédiction un peu moins bonne mais avec un gain énorme de temps de calcul (12.3 mn contre 143.8 mn).

Malheureusement, le modèle « Gradient Boosting » avec toutes les features et celui avec sélection des variables avec élimination par corrélation seulement n'ont pas pu fournir un résultat et on a dû interrompre le « kernel » après plus que 12 heures de calcul sans résultat vu la contrainte du temps disponible avant la date limite de ce projet.

5 Conclusion

Transformer les données clients en informations utiles comme trouver le meilleur moyen de guider et contrôler le processus de coupon pour obtenir le plus grand bénéfice est d'une grande importance dans le monde commercial, ce qui est par ailleurs très difficile dans une étude empirique.

Dans ce projet, on a réussi à construire un modèle prédictif des clients fidèles d'un boutique en ligne. Le modèle est exploitable avec ses performance et prêt à être implémenté dans un cas industriel réel tel que notre cas. Néanmoins, le modèle a toujours besoin d'autres améliorations en

termes de performance et vitesse d'apprentissage. Il se peut que la performance se dégrade dans le temps ce qui exige la nécessité d'utilisé un autre algorithme plus complexe ou un framework de deep learning plus avancé tel que Keras ou Tensorflow. Mais le modèle final peut passer avec toute fluidité à l'échelle et être entraîné sur des jeux 10, 100, 1000 fois plus grand mais le temps de calcul restera toujours une préoccupation et un problème à résoudre et à optimiser d'une façon constante. Une autre possibilité d'augmenter sa performance aussi c'est d'obtenir plus de variables ou information à l'échelle du client lui-même et non pas seulement le produit ou la commande. Des variables qui reflètent l'âge, l'occupation (étudiant , cadre , … etc) ou niveau d'étude et la région ou pays ou ville seront peut-être pas couteux à obtenir et peuvent facilement augmenter la précision et la performance du modèle. Une autre variable importante serait de détecter si la commande a été effectuée à partir d'un téléphone mobile ou bien un laptop. Le modèle présent même si peut toujours être amélioré dans plusieurs façons, peut être implémenté, maintenu et rafraichi d'une façon automatique avec un coût négligeable. Une fois que l'entreprise a décidé d'acheter ce modèle de machine learning sous forme d'un modèle seule tel qu'il est ou bien sous forme d'une application téléchargeable et peut être intégré dans une solution BI existante, elle doit choisir la fréquence avec laquelle ce modèle est réglé pour être rafraichi régulièrement soit une ou deux fois par jour soit une fois par semaine ou n'importe quelle fréquence sans l'intervention humaine pour faire ceci.

Une fois l'exécution de ce modèle se fait après avoir réglé la fréquence, un rapport avec la liste des clients ou comptes clients à contacter serait envoyer sous forme PDF ou Excel au manager marketing ou « customer success » pour décider sur la partition et l'envoi des coupons. Si la mise à jour a échoué, il serait facile de détecter ceci en recevant une notification sur la boite email de l'administrateur de l'application ou bien de l'outil BI ou ce modèle est intégré. Aussi, l'estimation d'un nouveau modèle une fois il y a du changement dans les variables « inputs » ou autre, pourrait être faite à partir de ce même modèle en demandant une simple mise à jour. Cependant, cette version du modèle est la version beta et d'autres versions plus robuste, rapide et précise vont être mis en vente une fois la mise à jour est complète.

Références

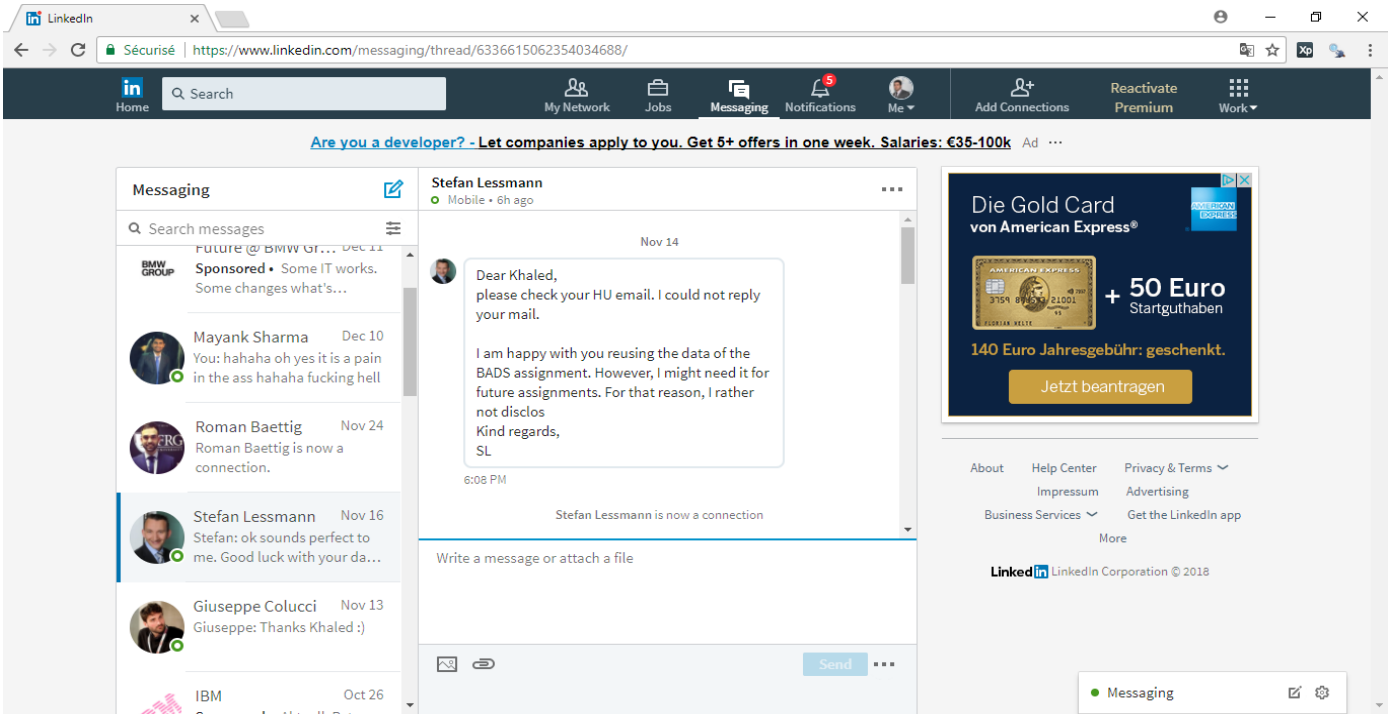
1. Larivière, Bart, and Dirk Van den Poel. "Predicting customer retention and profitability by using random forests and regression forests techniques." Expert Systems with Applications 29.2 (2005): 472-484.
2. Prasad, U. Devi, and S. Madhavi. "Prediction of churn behavior of bank customers using data mining tools." Business Intelligence Journal 5.1 (2012): 96-101.
3. Xie, Y., Li, X., Ngai, E. W. T., & Ying, W. (2009). Customer churn prediction using improved balanced random forests. Expert Systems with Applications, 36(3), 5445-5449.

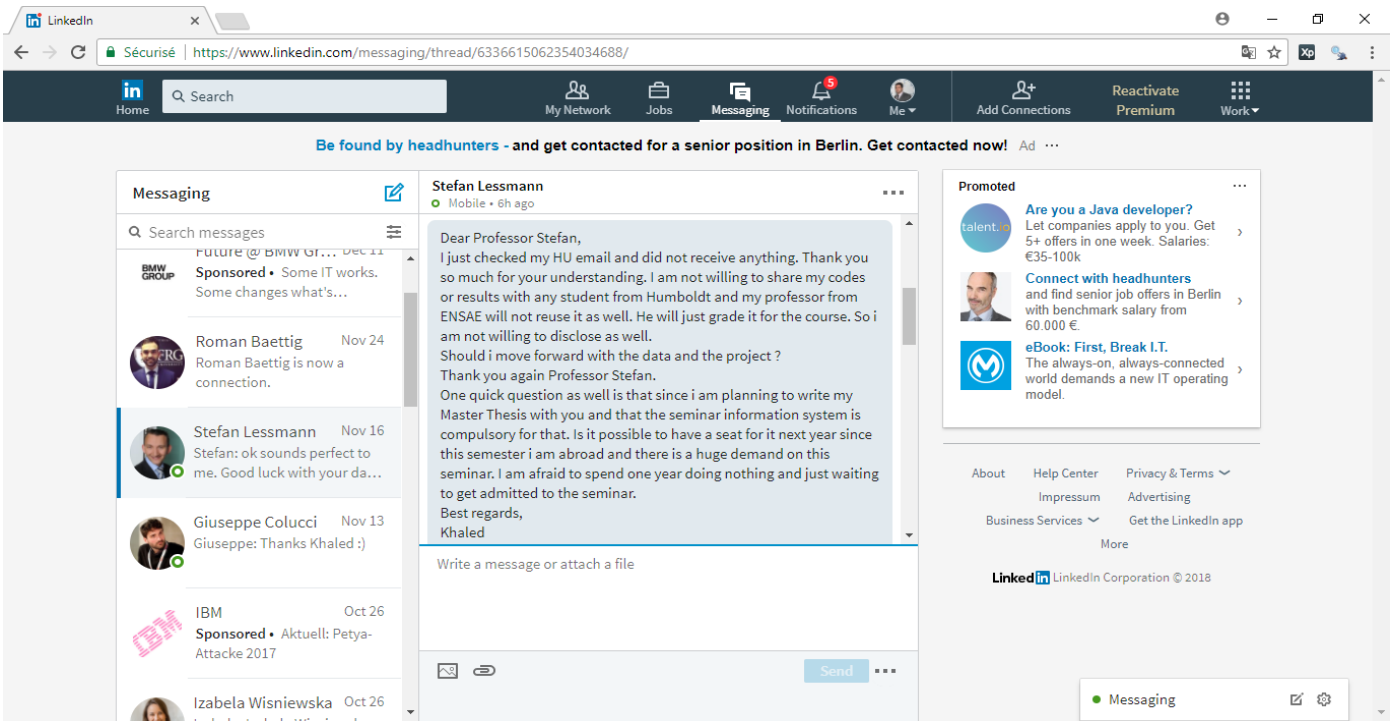
ANNEXE

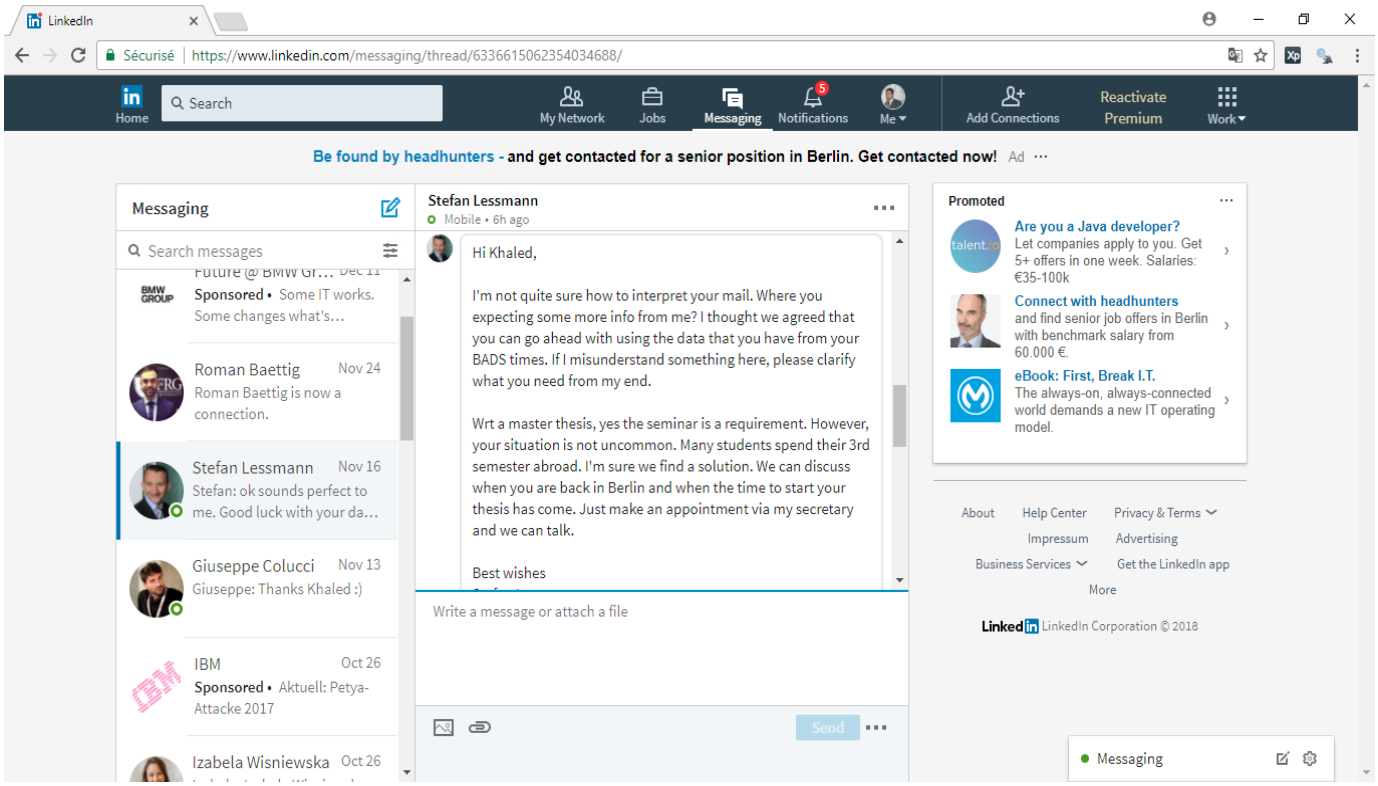
Dictionnaire des variables dans le jeu des données

Variable	Type de variable	Explication
ID	Nominal	ID unique du client
Order_date	Cardinal	Date de la première commande
Form_of_address	Nominal	Comment addresser le client (“Mr” , “ Mrs “ ..)
Title	Nominal	Si le client a un titre comme “Prof” ou “Dr”. oui =1 , non = 0
Email_domain	Nominal	Le non de domaine de l’email du client
Account_creation_date	Cardinal	Date de la creation du compte client
Newsletter	Nominal	Est ce que le client s’est enregistré pour une newsletter ? oui =1 , non = 0
Model	Nominal	Type de conception de site Web montré au client
Payment	Nominal	Méthode de paiement
Delivery	Nominal	Type de livraison , 0 = Livraison de la porte, 1 = Collection au bureau de poste
Postcode_invoice	Nominal	Code postal de l'endroit où la facture a été envoyée
Postcode_delivery	Nominal	Code postal de l'endroit où le produit commandé a été envoyé
Coupon	Nominal	Le client a utilisé un coupon pour cette commande Oui = 1, Non = 0
Advertising_code	Nominal	Code de données pour la publicité
Goods_value	Ordinal	La valeur du produit (5 = le plus élevé)
Item_count	Cardinal	Le nombre d'articles par ordre
Giftwraping	Nominal	Le produit vendu a-t-il été emballé comme cadeau? Oui = 1, Non = 0
Referrer	Nominal	L'entrée du client à la boutique par lien partenaire? Oui = 1, Non = 0
Points_redeemed	Nominal	Le client a-t-il utilisé des points de remboursement pour payer la commande? Oui = 1, Non = 0
Cost_shipping	Nominal	Le client a-t-il dû payer les frais de la livraison? Oui = 1, Non = 0
Deliverydate_estimated	Cardinal	Date de livraison estimée
Deliverydate_actual	Cardinal	Date de livraison actuelle
Weight	Cardinal	Poids du panier / commande en gramme
Remitted_items	Cardinal	Nombre des articles remis
Canceled_items	Cardinal	Nombre des articles annulés
Used_items	Cardinal	Nombre des articles 2éme main (occasion)
Book_count	Cardinal	Nombre des livres
Paperback_count	Cardinal	Nombre des livres poche
Schoolbook_count	Cardinal	Nombre des livres d’école
Ebook_count	Cardinal	Nombre des E-livre
Audiobook_count	Cardinal	Nombre des livres audio
Audiobook_download_count	Cardinal	Nombre de téléchargement livre audio
Film_count	Cardinal	Nombre des films
Musical_count	Cardinal	Nombre de musique
Hardware_count	Cardinal	Nombre des hardware
Imported_count	Cardinal	Nombre des articles importés
Other_count	Cardinal	Autre nombre des article
Return_customer	Nominal	Le client a passé une autre commande dans les 90 jours: Oui = 1, Non = 0

Accord du Prof. Stefan Lessmann sur la réutilisation du jeu des données






LinkedIn
Sécurisé | https://www.linkedin.com/messaging/thread/6336615062354034688/
Home
Search
My Network
Jobs
Messaging
Notifications
Me
Add Connections
Reactivate Premium
Work
Be found by headhunters - and get contacted for a senior position in Berlin. Get contacted now! Ad
Messaging
Search messages
Sponsored • Some IT works. Some changes what's...
Roman Baettig Nov 24
Roman Baettig is now a connection.
Stefan Lessmann Nov 16
Stefan: ok sounds perfect to me. Good luck with your da...
Giuseppe Colucci Nov 13
Giuseppe: Thanks Khaled :)
IBM Oct 26
Sponsored • Aktuell: Petya-Attacke 2017
Izabela Wisniewska Oct 26
Stefan Lessmann
Mobile • 6h ago
Hi Khaled,
I'm not quite sure how to interpret your mail. Where you expecting some more info from me? I thought we agreed that you can go ahead with using the data that you have from your BADS times. If I misunderstand something here, please clarify what you need from my end.
Wrt a master thesis, yes the seminar is a requirement. However, your situation is not uncommon. Many students spend their 3rd semester abroad. I'm sure we find a solution. We can discuss when you are back in Berlin and when the time to start your thesis has come. Just make an appointment via my secretary and we can talk.
Best wishes
Write a message or attach a file
Send
Promoted
Are you a Java developer?
Let companies apply to you. Get 5+ offers in one week. Salaries: €35-100k
Connect with headhunters
and find senior job offers in Berlin with benchmark salary from 60.000 €.
eBook: First, Break I.T.
The always-on, always-connected world demands a new IT operating model.
About
Help Center
Privacy & Terms
Impressum
Advertising
Business Services
Get the LinkedIn app
More
LinkedIn Corporation © 2018
Messaging