Prédiction du succès ou de l'échec d'une candidature à un poste de chercheur d'or

Jeremy Kulcsar

Le détail technique des réponses aux questions se trouve dans le fichier jupyter notebook *QuantMetry_réponse.ipynb* joint avec ce fichier.

1. Statistiques descriptives

1. Il nous est donné un jeu de donné avec 12 colonnes et 20 000 lignes. La première colonne est l'index, nous allons donc l'ignorer.

Les 11 autres colonnes sont les variables, qui se divisent en trois groupes :
Variables catégoriques : embauche, cheveux, sexe, diplome, specialite, dispo
Variables numériques : age, exp, salaire, note
La date est un cas à part que nous traiterons différemment.

La variable que nous souhaitons prédire ici est l'embauche, variable catégorique binaire. Il s'agit donc d'un problème de classification binaire.

L'analyse détaillée dans le notebook nous montre que :

- Les variables catégoriques gardent grossièrement les mêmes proportions entre le set complet et la restriction du set aux candidats embauchés -> pas de discrimination très explicite à ce niveau, sauf peut-être pour le diplôme et la spécialité, où les masters/licence et la géologie augmentent un peu en proportion.

- Un groupement des dates en semestres montrer que les embauches sont peu variables i.e. il n'y a pas de période plus prône à l'embauche qu'une autre. On peut donc retirer cette donnée du set.

- Les données continues contiennent des incohérences (Par exemple, âge minimal pour un embauché d'un an ou notes obtenues supérieures à 100). Les variables contenant des incohérences suivent cependant une gaussienne symétrique, on peut donc se dire qu'il s'agit ici de la manière dont la donnée a été enregistrée par le logiciel et non des erreurs d'entrées par les candidats. On va donc garder les incohérences pour les données qui ont une distribution symétrique gaussienne.

- De manière générale, toutes les variables continues suivent une distribution gaussienne symétrique.

Pour les variables catégoriques, les plus importantes semblent être le diplôme et la spécialisation.

Pour les variables numériques, la note est intuitivement importante. L'âge, l'expérience et le salaire demandé semblent intuitivement corrélées : plus on est âgé, plus on a d'expérience,

et plus on demande un salaire haut. Les trois one une distribution similaire : on pourrait se contenter d'en choisir une sur les trois si les analyses de dépendances multivariées nous le prouvent.

2. Pour les dépendances statistiques, les détails sont donnés dans le notebook.

a) On remarque une p-valeur égale à 0.0. Cela implique qu'elle est si faible que le logiciel a choisi d'arrondir le résultat. La p-valeur étant inférieure à 0.001, les variables sexe et spécialité sont hautement corrélées.

Le graphe dans le notebook nous montre que si l'on est un homme, on a 64.03% de chances d'être un géologue contre 30.51% pour une femme. A contrario, on a moins de 20% de chances d'être un détective ou un foreur contre au moins 24% pour une femme.

b) La distribution de salaire entre les différentes couleurs de cheveux semble uniforme : la couleur de cheveux ferait donc un mauvais prédicteur de salaire.

La p-valeur est de 0.23, ce qui est largement supérieur à 0.05 : Cela confirme que les deux variables ne sont pas statistiquement dépendantes.

c) On va regarder la covariance et le coefficient de corrélation de Pearson.

On remarque que les diagonales sont négatives : il semblerait que les deux variables soient donc négativement corrélées. Si l'expérience augmente, la note diminue et vice-versa. Cependant, les valeurs sont faibles.

Ce résultat semblant surprenant, allons un peu plus loin avec le coefficient de Pearson, qui a pour valeur -0.012. Cette valeur signifie que bien que négative, la corrélation entre les deux variables est très faible.

2. Machine Learning

1. Etant donné que nous avons ici affaire à un problème de classification binaire, nous allons en premier temps adapter la donnée à un problème de classification avec un one hot encoding, et ensuite effectuer un feature scaling pour éviter les soucis de distorsion associés aux ranges (l'age qui ne dépasse pas 100 contre le salaire demandé qui est 1000 fois plus élevé).

Ensuite, nous allons entraîner différents modèles de classification (KNN, Logistic Regression, Decision Tree, Random Forest et Grandient Boosting Classifier) puis choisir le meilleur. Nous sommes dans un cas où la base de données est relativement petite (20k lignes, et 15k de training) avec seulement 11% d'embauches, il faudra donc un modèle qui n'a pas besoin de beaucoup de données pour bien généraliser. On peut s'attendre à ce que les modèles arborescents et boostés donnent les meilleurs scores.

L'idée est ensuite de tuner les hyperparamètres avec un gridsearch afin de ré-entraîner les modèles sur une cross-validation pour avoir le modèle le plus entraîné avec les hyperparamètres les mieux ajustés.

Le modèle qui performe le mieux est le Grandient Boosting Classifier.

Les détails se trouvent dans le notebook.

2. Comme le montre les calculs dans le notebook, les features les plus importantes pour notre modèle sont la note qui compte pour plus de 25% dans la prise de décision, le salaire demandé qui compte pour plus de 15%, et la disponibilité qui compte pour plus de 5% si celle-ci est négative.

3. Il s'agit ici de savoir qui est susceptible d'être embauché. S'il est important de bien classer les vrais positifs et que l'on est sensible aux faux positifs car l'on tient à embaucher des candidats qualifiés, il faudra voir quel modèle a le meilleure score Precision. Il s'agit du nombre de vrais positifs sur le nombre total de positifs trouvés par le modèle.

4. Première piste : Fournir plus de données pour permettre à l'algorithme de mieux apprendre et améliorer sa capacité à généraliser

Seconde piste : Utiliser des méthodes plus avancées comme l'ensemble learning entre les différents algorithmes les plus performants trouvés plus haut