TP7: régression logistique
Statistiques pour Sciences Humaines II

Baptiste Perez et Anna Kiriliouk

Année 2021–2022

Commandes utiles pour le TP6

Commande	Description de la commande
`glm()`	ajuste un modèle de régression non-linéaire ; le modèle logistique est obtenu avec `family = "binomial"`
`summary()`	donne le résumé d'un objet créé par `lm` ou `glm`
`coef()`	donne les coefficients estimés à partir d'un objet crée par `lm` ou `glm`
`exp()`	fonction exponentielle
`seq()`	génère un vecteur avec valeurs de `from` jusqu'au `to`, de longueur `length`
`subset()`	sélectionne une sous-collection de variables et/ou observations d'un tableau
`cor()`	donne la/les corrélation(s) d'une collection des variables quantitatives
`predict()`	prédit une valeur à partir d'un objet crée par `lm` ou `glm`

Données "spotify" : recommandations musicales

Au cours des dix dernières années, les systèmes de recommandation sont devenus de plus en plus populaires. Amazon recommande des produits en fonction de votre historique de navigation et d'achat, Netflix recommande des films et des séries en fonction de votre historique de visionnage, et Spotify recommande des chansons que vous pourriez aimer en fonction de votre historique d'écoute. Ces systèmes de recommandation essaient tous d'atteindre le même objectif : utiliser les caractéristiques des produits / films / musique qu'un utilisateur apprécie pour déterminer les produits / films / musique que l'utilisateur pourrait apprécier mais qu'il n'a pas encore découvert.

Dans ce TP, on construira un modèle de régression logistique qui prédit la probabilité qu'un utilisateur aime une chanson en utilisant les caractéristiques pertinentes de la chanson. Le jeux de données `spotify.RDS` contient les caractéristiques des chansons de 2017 et indique si la personne a aimé ou non la chanson. Cet ensemble de données contient les préférences de chanson d'une seule personne ; par conséquent, la portée de notre analyse est limitée. On s'intéressera aux variables suivantes :

— `like` : indique si l'utilisateur aime la chanson (`yes`) ou pas (`no`).

— `mode` : indique si la chanson a été composée en mode `majeur` ou en mode `mineur`.

— `duration_ms` : la durée de la chanson en millisecondes.

— `danceability` : décrit à quel point une chanson est "dansante" en se basant sur le rythme. Une valeur de 0 est la moins dansante et 1 la plus dansante.

— `tempo` : le tempo d'une chanson, exprimé en Beats Par Minute (BPM).

— `instrumentalness` : plus la valeur est proche de 1, plus la chanson est susceptible d'être instrumentale, c'est-à-dire, sans vocaux.

— `loudness` : le volume moyen de la chanson, standardisé, entre -60 et 0 en décibel (dB).

Dans les questions 1–6, on ne considère que les variables `like`, `mode` et `danceability`.

1. Examinez les associations entre `mode` et `like` et entre `danceability` et `like` en faisant des graphiques appropriés. Interprétez ces graphiques.
2. Ajustez un modèle de régression logistique, appelé `m1`, pour prédire `like` en fonction de `mode` et `danceability`. Écrivez deux équations pour la cote d'aimer une chanson en fonction de `danceability`, une pour les chansons en `mineur` et l'autre pour les chansons en `majeur`.
3. Calculez la cote d'aimer une chanson avec `danceability = 0.35` pour une chanson en mineur et pour une chanson en majeur. Utilisez cet exemple pour interpréter $e^{\widehat{\beta}_{\text{mode}}}$.
4. Comment est-ce que la cote d'aimer une chanson change quand `danceability` augmente de 0.35 à 0.45 ? Et de 0.45 à 0.55 ?
5. Pour quelle valeur de `danceability` est-ce que l'utilisateur a une probabilité de 0.5 d'aimer (ou de pas aimer) une chanson ? Donnez la réponse pour une chanson en mineur et pour une chanson en majeur.
6. Faites une graphique qui montre deux courbes, une pour les chansons en mineur et une pour les chansons en majeur, de la probabilité estimée d'aimer une chanson en fonction de la `danceability`. Est-ce que l'on aurait pu faire une régression linéaire au lieu d'une régression logistique ?

Pour les questions suivantes, on considère toutes les variables explicatives.

7. Ajoutez une nouvelle variable `duration` à ce jeu de données qui représente la durée des chansons en secondes.
8. Vérifiez si un modèle de régression logistique contenant toutes les variables explicatives (`mode, duration, danceability, tempo, instrumentalness, loudness`) risque de souffrir de multicolinéarité.
9. Ajustez un modèle de régression logistique pour prédire `like` en fonction des variables listées dans la question 8. S'il y a des variables non-significatives (au niveau $\alpha = 0.05$), enlevez-les du modèle. Appelez le nouveau modèle `m2`.
10. Peut-on faire un graphique de la probabilité estimée d'aimer une chanson comme dans la question 7 ?
11. Considérons les données suivantes concernant deux nouvelles chansons :
    (a) "Sign of the times" : `mode = majeur`, `duration = 340.707`, `danceability = 0.516`, `tempo = 119.972, instrumentalness = 0`.
    (b) "Hotline Bling" : `mode = majeur`, `duration = 267.024`, `danceability = 0.896`, `tempo = 134.962, instrumentalness = 0.000258`.

Sur base des modèles `m1` et `m2`, est-ce que vous pensez que l'utilisateur aimera bien les chansons ? En réalité, l'utilisateur aimait bien "Hotline bling" mais pas "Sign of the times". Est-ce que les modèles sont adéquats ?