

Challenge Otto de Kaggle

Robin Duraz - Jiaxin Gao

TC1 - Apprentissage

6 novembre 2018

1 Présentation du challenge

Ce challenge a été proposé par le groupe Otto, une des plus grandes compagnies d'e-commerce, vendant des produits dans de nombreux pays autour du monde. Analyser et classifier des produits peut se révéler être une tâche complexe, et en effet, une de leurs tentatives classifiait des produits identiques venant de différents endroits comme différents.

De ce fait, la capacité à analyser correctement des produits dépend de la précision avec laquelle on regroupe des produits similaires.

Le jeu de données est composé de plus de 200 000 produits, tous ayant 93 features. Le but de ce challenge est de classifier correctement ces produits en 9 classes. De ces données, un peu plus de 60 000 servent de données d'entraînement, tandis que tout le reste sert de données de test pour l'évaluation par la plateforme de Kaggle. Pour ces données de test, nous devons calculer les probabilités d'appartenance à chaque classe, ce qui est testé par le challenge.

L'erreur sur les données est calculée de la façon suivante :

$$-\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log(p_{ij})$$

2 Methodology

Pour commencer, on a effectué une analyse des données. Nous avons observé la répartition des classes, des résumés statistiques sur les valeurs contenues dans les features, comme les minimum, maximum, moyenne, afin de nous faire une première idée du type de données.

Ensuite, nous avons simplement essayé d'utiliser séparément différents modèles, principalement venant de sci-kit learn afin de voir les performances sur les données brutes, puis normalisées/« scalées ». De cela, nous avons choisi de « scaler » tout le temps car cela améliorerait les performances pour pratiquement tous les modèles (et à défaut ne les baissait pas). L'effet de la normalisation dépendant des modèles, et étant globalement inconstant, nous avons choisi de ne pas le faire.

Après cela, nous avons essayé de « tuner », e.g. améliorer les performances de nos modèles individuellement en changeant les valeurs de leurs paramètres. Pour y arriver, nous avons utilisé la méthode de « grid search » pour tester différentes valeurs pour chaque paramètre, et quelque fois pour des groupes de deux paramètres.

Avant de tenter des combinaisons, nous avons éliminé SVM qui était trop lent et trop peu performant. Ensuite, nous avons essayé de combiner les prédictions de plusieurs modèles ensemble, deux par deux, en effectuant une moyenne pondérée sur les probabilités prédites. Nous avons également effectué du « grid search » de trouver de bonnes valeurs de poids sur les combinaisons.

Pour finir, nous avons aussi essayé de combiner plus de deux modèles pour voir si cela pouvait encore améliorer nos résultats, ce que nous avons réussi.

3 Results