MEUTER Romain

Rapport ML : “Absenteeism at work”

Rendu 25/05/2020

UE 801 EC1

Apprentissage machine

Introduction

J’ai choisi le dataset “Absenteeism at work”, dans le but de comprendre la complexité et l’impact que des données nombreuses et plutôt variées peuvent avoir sur l’algorithme d’apprentissage. Notamment, j’ai découvert l’importance d’un bon “nettoyage” des données sur l’algorithme.

Les points qui ont retenu mon attention :

- L’équiprobabilité des données classant qu’une catégorie,
- La distinction variable nominale/quantitative et leurs différents traitements,
- Les valeurs aberrantes/manquantes (mois, raison et temps d’absences),

Ces points m’ont permis de faire évoluer la prédiction de l’algorithme à 73 %. Pour une question de place dans ce rapport je ne parlerai que la solution optimale, c’est-à-dire le meilleur dataset transformé, ainsi que le meilleur paramétrage de l'algorithme d’apprentissage trouvé. Le dataset optimal choisi était celui type 2 (explicité ci-dessous).

Les données

En tous 740 lignes sont comptabilisées pour 21 colonnes. Les données sont de trois types différents :

- **Booléen :** “Manque de discipline”, “Buveur”, “Fumeur”
- **Catégorie :** “Raison de l’absence”, “Mois de l’absence”, “Jour de la semaine”, “Saison”, *“ID”* (Ce dernier est compté comme catégorie car il fait référence à un individu, et n’a de sens d'être calculé seulement compté)
- **Numérique (“float” et “int” compris) :** “Frais de transport”, “Distance jusqu’au travail”, “Temps de service (en année)”, “Age”, “Charge de travail par jour”, “Objectif de travail accompli”, “Animaux”, “Poids”, “Taille”, “IMC”, “Temps d’absence en heures”

Transformation données X :

Après analyse, l’objectif de prédiction était de déterminer par les différents facteurs donnés quel sera le temps l'intervalle de temps d’absence des employés selon leurs caractéristiques connus. Le dataset a été formé de la manière suivante :

- Suppression des lignes de mois d'absence à 0,
- Suppression des lignes où le temps d'absence est à 0.

Ensuite, les données numériques sont normalisées, et les données catégorielles subissent un encodage à chaud (Tranformation en plusieurs tableau booléen d’une colonne catégorie). Le but étant de se ramener au maximum à une échelle de tous les axes compris entre 0 et 1 inclus.

Transformation donnée cibles :

Dans l’idée d’une équi-probabilité d’apparition et d’une classification sensée, j’ai partager les données de temps d’absences en 3 classes de cette manière :

- De 0 à 2 heures compris (soit 35 % des données),
- De 3 à 7 heures compris (soit 26 % des données),
- Plus de 8 heures compris (soit 39 % des données).

Perceptron multicouches :

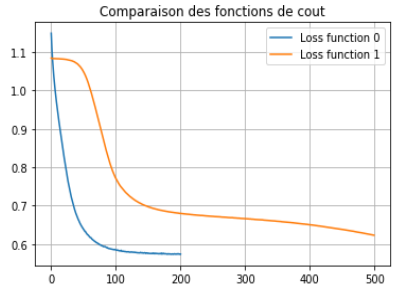
Régulation des hyperparamètres des RN :

Durant la mise en oeuvre du RN, j'ai cherché à trouver les paramètres maximisant la prédiction. Pour cela, j'ai établi une fonction qui cherche à trouver la meilleure moyenne de prédiction pour toutes les possibilités d'un paramètre, ensuite réitère le processus. Cela permet dans une première phase de trouver une combinaison générale puis de modifier sur cette base.

Résultat & remarque :

Mon dataset retenu est le numéro 1, avec une prédiction de 73 %, ci-contre. Selon la configuration suivante :

- Solveur : Adam,
- Fonction d'activation : Logistique,
- Valeur alpha : 10**(-5)
- Ratio initiale d'apprentissage : 0.001
- Forme du ratio d'apprentissage : adaptive
- Nombre d'itération : 500
- Etat aléatoire : 9
- Couches cachées du MLP : 20, 20, 20



La matrice de confusion :

```
[[47  3  7]
 [17 13  6]
 [ 3  2 42]]
```

	precision	recall	f1-score	support
0	0.70	0.82	0.76	57
1	0.72	0.36	0.48	36
2	0.76	0.89	0.82	47
accuracy			0.73	140
macro avg	0.73	0.69	0.69	140
weighted avg	0.73	0.73	0.71	140

J'ai testé différentes combinaisons et ai observé une mauvaise même si la prédiction étant améliorée, cela se voit par la matrice de confusion, où la classe deux est plus souvent classés comme 1 que 2. De plus, nous pouvons observé comme ci-joint la tendance critique de la courbe d'apprentissage se modifier, selon la fonction d'activation (*Loss function 0* : identité et *Loss function 1* : logistique).

Conclusion & Axe d'amélioration :

Ce modèle n'est pas optimal, à la fois par la formation des données mais également par le modèle choisi. Cependant, l'apprentissage se base sur un petit dataset, et les poids étant initialisés de façon aléatoire ont marqué une difficulté sur le pourcentage de prédiction de réussite.

Les données transformées n'ont montré que peu de différence, tournant plus au hasard dû à l'initialisation du modèle qu'à la différence dû à la formation. Cependant, la normalisation et l'encodage à chaud ont réellement montré une progression de la prédiction.
Vis-à-vis du modèle, après la formation initiale des données, le nombre de caractéristique à relativement augmenter. Ce qui est problématique pour des modèles comme les RN, car ils

sont assez lourds. Un modèle tel que XGBoost ou les forêts aléatoires seraient tout aussi intéressants à exploiter.

Enfin, j'ai découvert tardivement, les fonctionnalités de test des combinaisons de Sklearn nommée : "gridsearchcv" et "randomizedsearchcv" qui sont deux méthodes distinct pour la paramétrisation des modèles d'apprentissage et se montre relativement utile et a utilisé pour la suite de mon apprentissage.