

# PROJET EN PYTHON

# Classification binaire

BARRY Issa | le 28/03/2020

# Contexte

Ce projet consiste à entrainer un modèle de classification binaire sur des données relative à une campagne de prêts bancaires réalisée pour une institution bancaire.

# Objectif

* Gestion des données manquantes
* Trouver les meilleurs modèles
* Soumettre les résultats pour voir le score

La méthode des *k* plus proches voisins et le model de régression logistique sont les algorithmes qui ont vraiment marché pour moi.

# Méthode  k-NN :

La méthode des **K plus proches voisins** (**KNN**) a pour but de classifier des points cibles en fonction de leurs distances par rapport à des points constituant un échantillon d’apprentissage (c’est-à-dire dont la classe est connue a priori).

Quand on l’utilise en classification le résultat est une classe d'appartenance. Un objet d'entrée est classifié selon le résultat majoritaire des statistiques de classes d'appartenance de ses k plus proches voisins, (*k* est un nombre entier positif généralement petit). Si *k* = 1, alors l'objet est affecté à la classe d'appartenance de son proche voisin.

# Comment c’est dérouler le travail ?

* D’abord j’ai chargé les données (bank\_train\_data et bank\_train\_csv)
* Ensuite j’ai utilisé « get\_dummies » pour le codage
* Puis pour chacun des deux fichiers j’ai pris un échantillon de 60% pour l’entrainement
* Et les fonctions utilisées sont :

• euclidian\_distance (pour calculer la distance euclidienne)

• neighbors (Pour trouver le k plus proche voisin)

• prediction (Qui renvoie le résultat à savoir si l’individu va souscrire ou non)

* En fin j’ai  effectué le test avec le fichier (bank\_test\_data), puis le résultat enregistré dans un fichier nommé (test\_result) au format csv

# Model de régression logistique :

## La régression logistique est largement répandue dans le domaine bancaire et des assurances. C’est un modèle de régression binomiale.

# Ces points forts :

• Il nous permet à expliquer la survenue d’un évènement

• Il nous permet à cherche la probabilité de succès

# Comment c’est dérouler le travail ?

Après le chargement des données, l’encodage et la création du vecteur de poids. J’ai utilisé des fonctions pour entrainer le model comme :

• Une fonction pour calculer le taux d’erreur

• Une fonction de prédiction

• Le cross entropy

• Le gradient

• La sortie du model

Et enfin le résultat est enregistré dans un fichier (bank\_test\_result).

# Conclusion

Après avoir le résultat de chacun des algorithmes (KNN et régression logistique). J’ai constaté qu’à l’exécution l’algorithme de régression logistique est beaucoup plus rapide, tandis que la méthode des K plus proche voisin prenait énormément de temps avant d’obtenir le résultat.

Mais sur codalab la régression logistique affichai un meilleur score (0.8104) para-port à la méthode des k plus proche voisin.