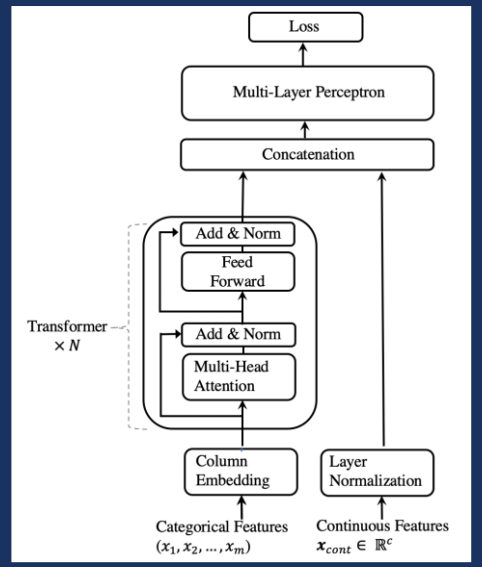
PROJET 7 PREUVE DE CONCEPT : UTILISATION DE TRANSFORMERS POUR LA MODÉLISATION DE DONNÉES TABULÉES

SOUTENANCE OPENCLASSROOMS, LE 31/08/2022

ERWAN CHESNEAU



PLAN :

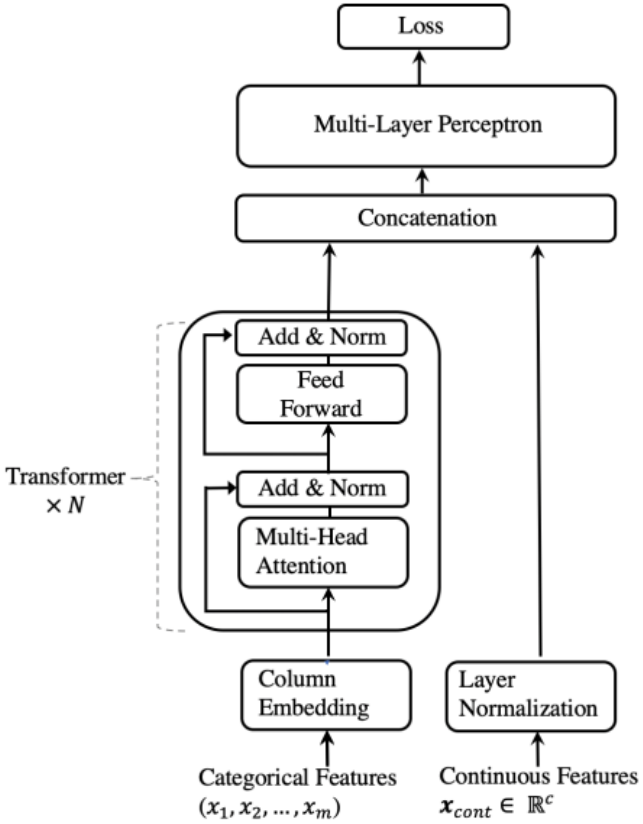
I. Motivations
II. Les TabTransformers
III. Collecte des données
IV. Nettoyage et exploration de la base de données
V. Features engineering
VI. Modélisation
VII. Conclusions
VIII. Améliorations à envisager

I. MOTIVATION:

- Les données tabulées :
  - Historiquement les données les plus utilisées en ML
  - Données avec lesquelles j'apprécie travailler
- ML : développement régulier de nouvelle méthode ou nouveau modèle :
  - Pour les données tabulées, souvent méthode Gboost les plus efficaces
  - Transformer très en vogue et efficace sur les donnéees textuelles (BERT)
- Utilisation de Transformer sur des données tabulées :
  - Proposé par Huang et al (AWS) en décembre 2020
  - Amélioration de la précision par rapport à un MLP classique
  - Précision proche d'un XGBoost
- Tester cette méthode sur des données réelles

II. LES TABTRANSFORMERS : PRINCIPE

- Proposé par Huang et al en décembre 2020 [1]
- Principe du transformateur :
  - Utiliser les méthodes du NLP pour améliorer les performances
  - Plongement de données catégorielle
  - Auto attention pour conserver un plongement contextualisé
- Architecture du modèle :
  - Couche de plongement de données
  - Un ou plusieurs transformateur(s)
    - Une couche d'auto attention
    - Une couche d'anticipation
  - Concaténation des données catégorielles et continues
  - Un perceptron multicouches



[1] : X. Huang, A. Khetan, M. Cvitkovic, et Z. Karnin, « Tabtransformer: Tabular data modeling using contextual embeddings », *ArXiv Prepr. ArXiv201206678*, 2020

II. LES TABTRANSFORMERS : UTILISATION

- Les auteurs ont testé le modèle sur 15 bases de données
  - Principalement issus de compétitions
  - Nombre de variables catégorielles entre 2 et 128
  - Amélioration des performances par rapport à un MLP
  - Performances similaires à celles accessibles par des arbres de décision
- D'autres auteurs ont également utilisés ce transformateur
  - Prédiction de la mortalité due à la COVID-19
  - Prédiction de maladie neurodégénérative
  - Etude d'attaques informatiques
- Des auteurs proposent de combiner TabTransformer et XGBoost

Dataset	Baseline MLP	TabTransformer	Gain (%)
albert	74.0	75.7	**1.7**
1995_income	90.5	90.6	**0.1**
dota2games	63.1	63.3	**0.2**
hcdr_main	74.3	75.1	**0.8**
adult	72.5	73.7	**1.2**
bank_marketing	92.9	93.4	**0.5**
blastchar	83.9	83.5	-0.4
insurance_co	69.7	74.4	**4.7**
jasmine	85.1	85.3	**0.2**
online_shoppers	91.9	92.7	**0.8**
philippine	82.1	83.4	**1.3**
qsar_bio	91.0	91.8	**0.8**
seismicbumps	73.5	75.1	**1.6**
shrutime	84.6	85.6	**1.0**
spambase	98.4	98.5	**0.1**

Model Name	Mean AUC (%)
TabTransformer	**82.8** $\pm$ 0.4
MLP	81.8 $\pm$ 0.4
GBDT	**82.9** $\pm$ 0.4
Sparse MLP	81.4 $\pm$ 0.4
Logistic Regression	80.4 $\pm$ 0.4
TabNet	77.1 $\pm$ 0.5
VIB	80.5 $\pm$ 0.4

[1] : X. Huang, A. Khetan, M. Cvitkovic, et Z. Karnin, « Tabtransformer: Tabular data modeling using contextual embeddings », *ArXiv Prepr. ArXiv201206678*, 2020

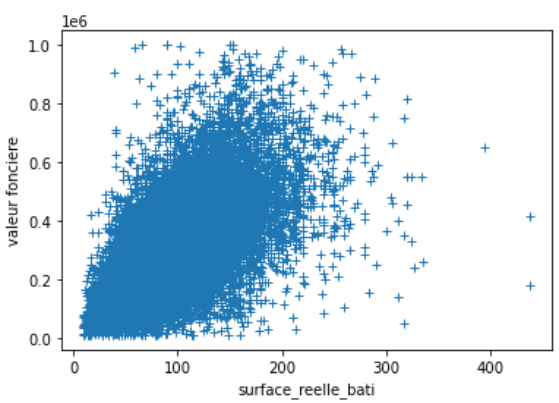
III. COLLECTE DES DONNÉES

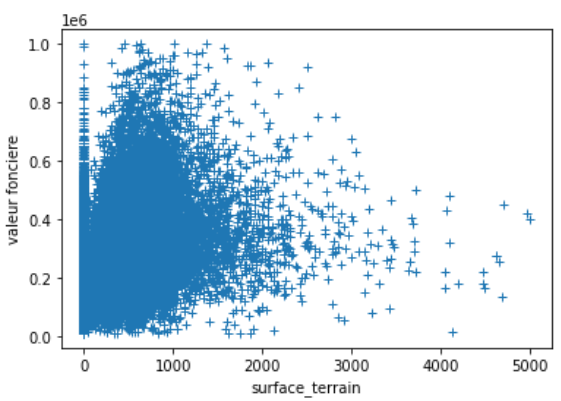
- Objectif : Prédire le prix de l'immobilier en France
- Utilisation uniquement de données publiques :
  - Récupération des données sur data.gouv.fr
- Les bases de données :
  - L'historique des mutations immobilières (descriptions des biens)
  - Données sur les communes (caractérisation de l'environnement)
  - Transport en commun (Accessibilité)
  - Educations (écoles à proximité)
  - Données sur les loyers (estimation locative)

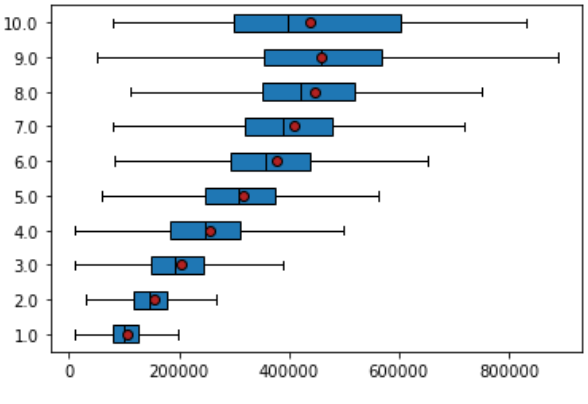
data
gouv.fr

IV. NETTOYAGE ET EXPLORATION : MUTATIONS IMMOBILIÈRES

- 40 variables : toutes ne sont pas utiles
  - Conservées : localisation, surface du bien et terrain, type de local, type de mutation, nombre de pièces et **valeur**
- Suppression des observations contenant des locaux commerciaux ou uniquement des dépendances
- Suppression des opérations avec un nombre de biens importants (Opérations immobilières particulières)
- Certaines mutation ont une valeur très élevée (>10M) ou très faible (1€)
  - On conserve uniquement les mutations dont la valeur est entre 10k et 1M€
- Suppression des biens de moins de 10m² et dont le terrain et > 5000m²
- Corrélation entre valeur et :
  - Surface du bien
  - Surface du terrain
  - Nombre de pièce



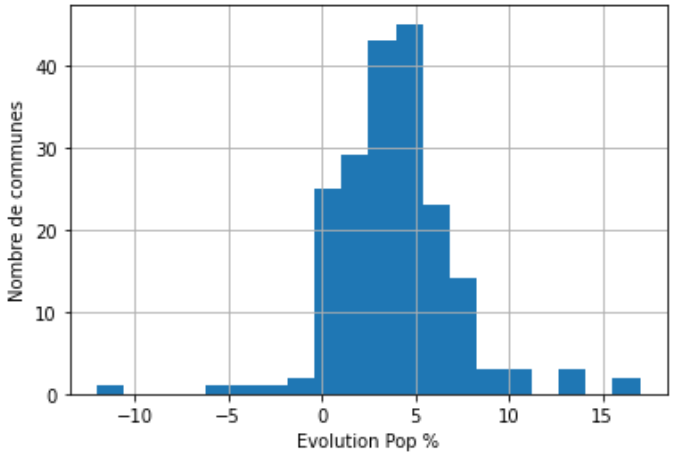




IV. NETTOYAGE ET EXPLORATION : COMMUNES

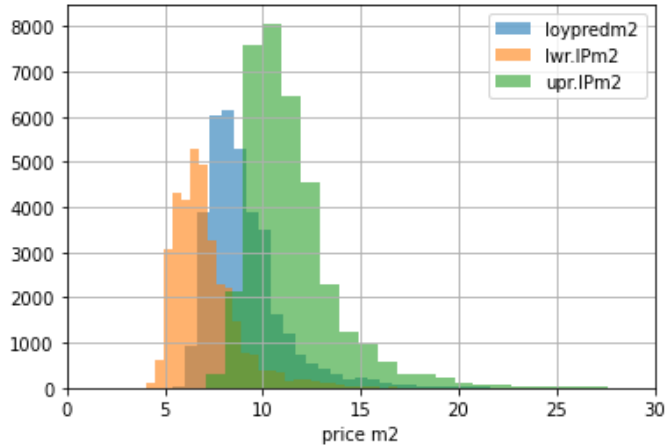
Données sur les communes :

- 101 variables initialement mais beaucoup peu renseignées
  - Conservées : code commune, nom, département, accès aux soins, informations démographiques et taux de propriété
- Vérification que les variables ne sont pas aberrantes et sont discriminantes :
  - Le nombre de pharmacie semble erroné (max 2)
  - Le taux d'étudiant non discriminant



Loyers :

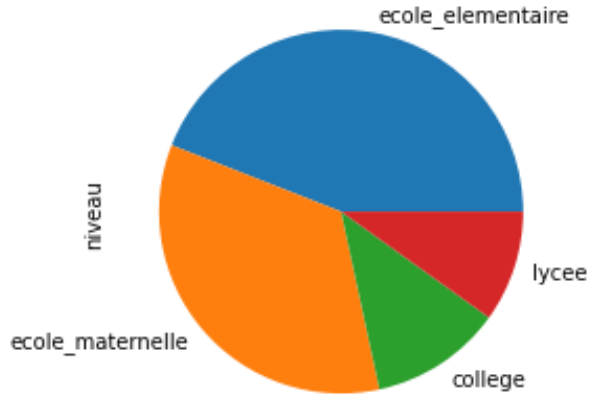
- On conserve le nom et le code de la commune et les estimations hautes, basses, et moyennes des loyers / m²
- Données disponibles pour les maisons et les appartements



IV. NETTOYAGE ET EXPLORATION

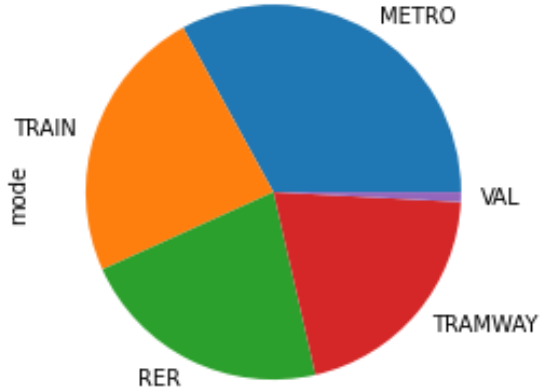
Educations :

- Liste de tous les établissements avec le niveau d'études, la localisation
- Simplification du niveau d'étude en niveau maternelle, élémentaire, collège, lycée
- Les universités sont traitées de la même manière
- Nombre d'étudiants inscrits par an moyens calculé



Transport en communs :

- Liste de toutes les stations de transports avec la localisation, le mode de transport



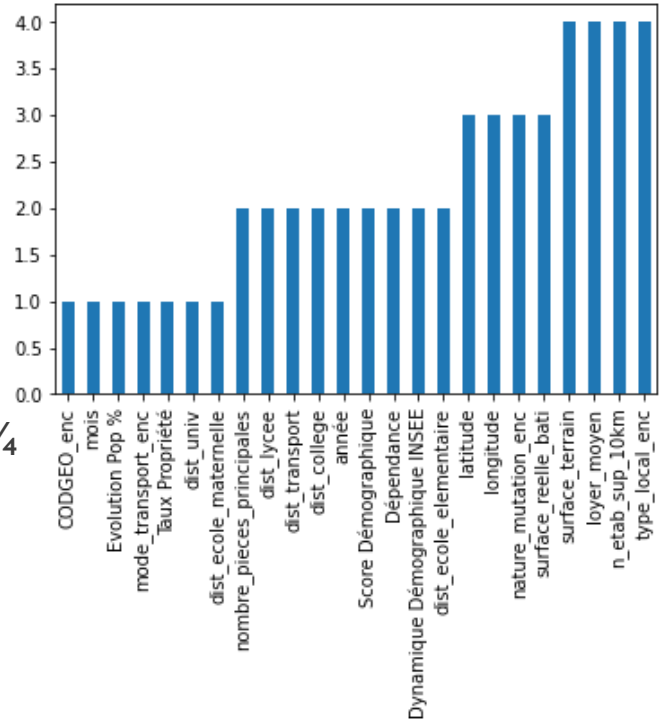
- Pour toutes les bases de données la localisation est vérifiée :
  - Latitude : entre 0 et 90
  - Longitude : entre -180 et 180

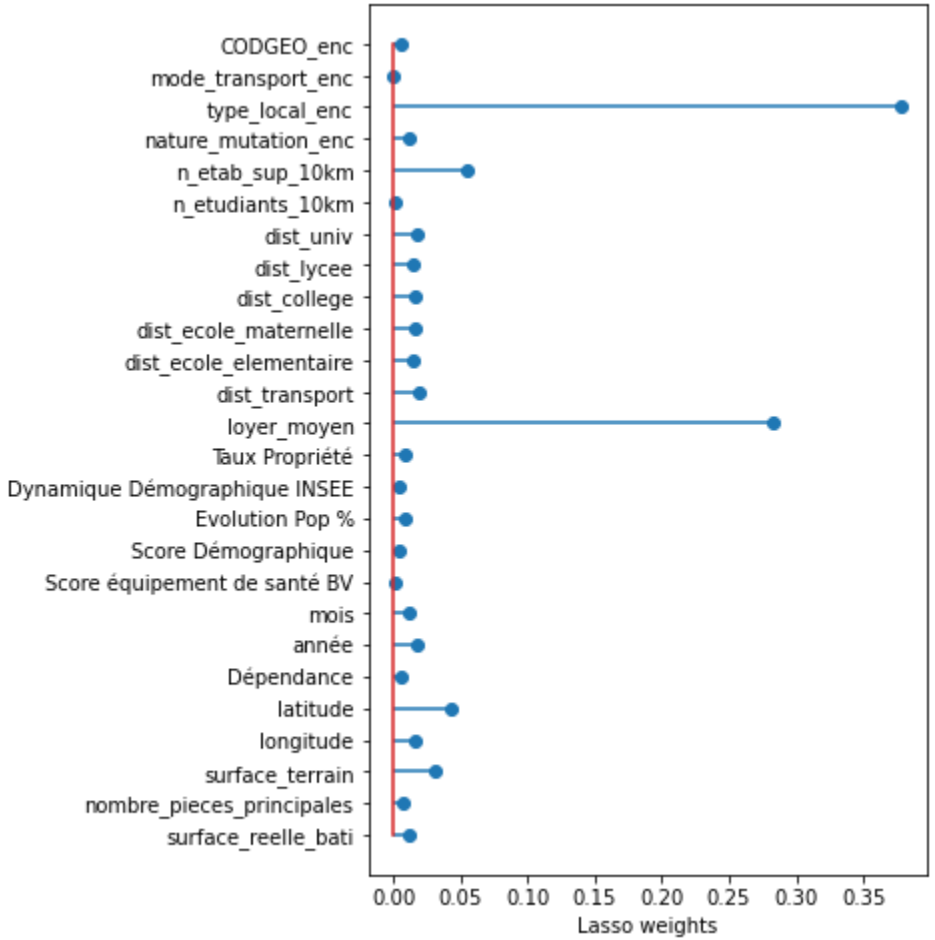
V. FEATURES ENGINEERING

- Combinaison de toutes les bases de données :
  - Informations sur les communes : grâce aux codes communes
  - Calcul du loyer estimé pour chaque bien
- Calcul de distances :
  - Utilisation de la librairie Sklearn
  - Distances entre le bien et les écoles
  - Nombre d'étudiants dans un rayon de 10km
  - Distance aux transports
  - Mode de transport à proximité

V. FEATURES ENGINEERING : FEATURES IMPORTANCES

- Sélection des variables les plus importantes parmi les 23 disponibles
- Combiner différentes méthodes :
  - F_score
  - Regularisation Lasso (L1)
  - Arbre de décision et XGBoost
- Seuil fixé arbitrairement
- Sélection des features présentes ¾
- Ajout du nombre de pièces
- Ajout du nombre de dépendance

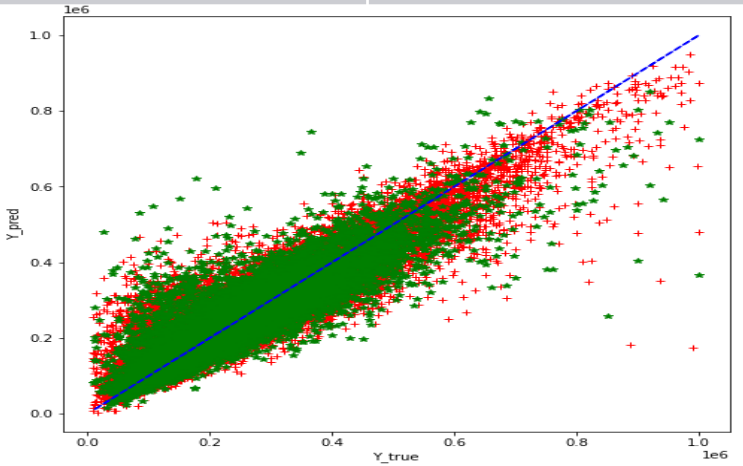




VI. MODELISATION : MODELE « CLASSIQUE »

- Des modèles classiques sont utilisés comme baseline
  - ElasticNet
  - KNN
  - Kernel SVR
  - XGBoost
  - CateBoost
- Meilleur modèle : XGBoost
  - Optimisation des hyperparamètres

Modèle	MAE
Dummy mean	92431
ElasticNet	47833
KNN	40993
Kernel SVR	91022
XGBoost	32809
XGBoost opt	31864
CateBoost	30711



VI. MODELISATION : RÉSEAUX DE NEURONES

- Les TabTransformers sont injectés dans un MLP :
  - Réseaux de neurones sert aussi de baseline
- Recherche de la meilleure architecture
  - Dans une architecture rectangulaire (l,n)
  - Optimum pour l=10, n=128 et l=15, n=256
- Réseaux moins performant que les méthodes de Boosting

Modèle	MAE
Dummy mean	92431
ElasticNet	47833
KNN	40993
Kernel SVR	91022
XGBoost	32809
XGBoost opt	31864
CateBoost	30711
NN_l-10_n-128	36755
NN_l-15_n-256	36896

VI. MODELISATION : TRANSFORMATEURS

- Librairies pytorch widedeep
- Différents modèles :
  - TabMLP : Plongement de données uniquement
  - TabResNet : bloc ResNet
  - TabTransformers : Huang et al
- TabMLP donne des faibles performances
- TabTransformers plus performants qu'un NN…
- Mais moins performant que CateBoost et XGBoost
- Données trop peu structurées ?

Modèle	MAE
Dummy mean	92431
ElasticNet	47833
KNN	40993
Kernel SVR	91022
XGBoost	32809
XGBoost opt	31864
CateBoost	30711
NN_l-10_n-128	36755
NN_l-15_n-256	36896
TabMLP	39247
TabResNet	36305
TabTransformers	35480

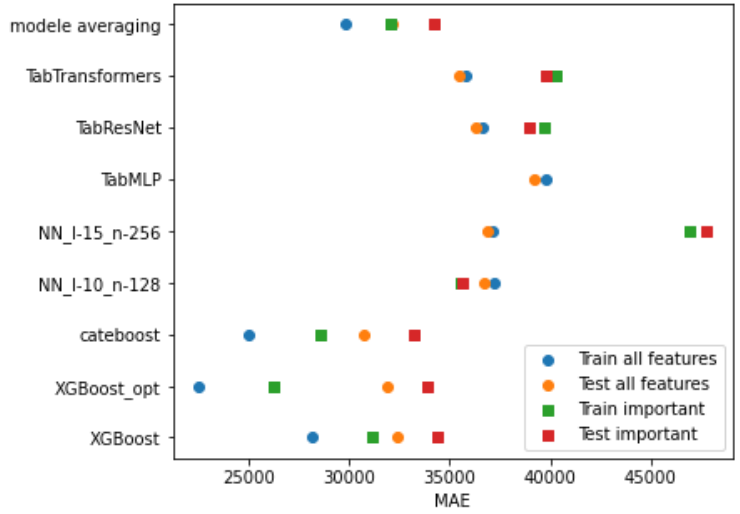
VI. MODELISATION : MODEL AVERAGING

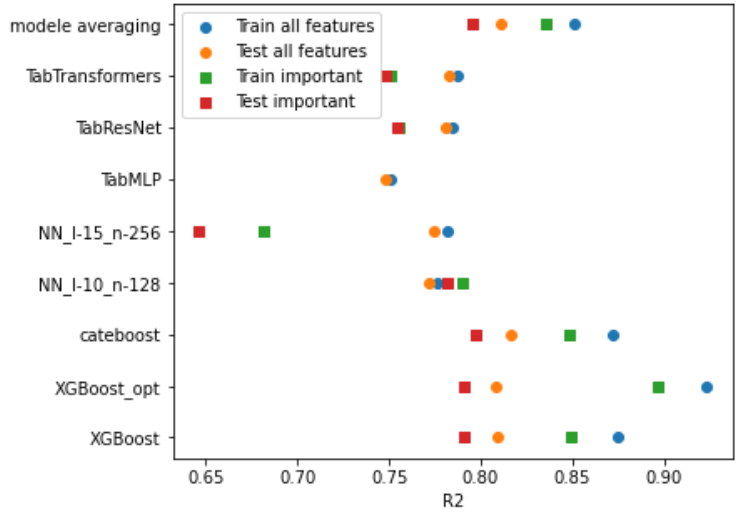
- Performances peu satisfaisantes
- Peut on améliorer la performance en combinant les modèles?
  - Sélection des modèles les plus performants
  - Hypothèse qu'une partie de l'erreur est aléatoire
  - Hypothèse que tous les modèles ne font pas les mêmes erreurs
- La performance est comparable à XGBoost
- Pas d'amélioration drastique :
  - L'erreur n'est pas aléatoire
  - L'erreur provient probablement d'imprécision de la base de données

Modèle	MAE
Dummy mean	92431
ElasticNet	47833
KNN	40993
Kernel SVR	91022
XGBoost	32809
XGBoost opt	31864
CateBoost	30711
NN_l-10_n-128	36755
NN_l-15_n-256	36896
TabMLP	39247
TabResNet	36305
TabTransformers	35480
Model averaging	32126

VI. MODELISATION : FEATURES IMPORTANCE

- 10 variables les plus importantes
- Apprentissage de tous les modèles
- Perte de performance systématique
- Les variables déclarées non importantes apportent de l'information
- Simplifie le modèle…
- … mais à utiliser avec précaution





VII. CONCLUSIONS

- Les transformers ont été adaptés pour pouvoir être appliqué aux données tabulées
- Test sur une base de données construite à partir de données publiques et donc imparfaites / incomplètes
  - Base de données nettoyée afin de conserver uniquement des données caractérisables et non aberrantes
  - Nouvelles variables créées pour caractériser la distance à des lieux d'intérêts
- Les TabTransformers permettent d'améliorer les performances accessibles à partir d'un NN
- La performance accessible par une méthode de boosting non égalée
  - Les données ne sont pas totalement adaptées à la méthode ➔ pas assez structurées
- La précision est assez faible et non annulable en moyenne
  - L'erreur causée par une base de données de trop faible qualitée

VIII. AMÉLIORATIONS

- Compléter la base de données avec des informations complémentaires (notamment sur la qualité des biens)
- Augmenter le nombre de variables (catégorielle) pour améliorer les performances
- Tester d'autres méthodes de sélection de variables
- Tester la combinaison de Xgboost avec les TabTransformers pour profiter du meilleur des deux mondes…

MERCI POUR VOTRE ATTENTION !