

ADVANCED ML TD2

Advanced machine learning TD2

Etudiant :
TRANG THOMAS

Encadrant :
M. OBLIN DENIS

Remerciements

Merci à monsieur Oblin Denis pour avoir assuré ce TD2, ses conseils pour l'implémentation et l'optimisation des modèles de machine learning.

Table des matières

| | | |
|----------|--|----------|
| 1 | Présentation du dataset | 3 |
| 2 | Data Exploration | 3 |
| 3 | Data cleaning & Processing | 4 |
| 4 | Modélisation & tuning | 5 |
| 4.1 | Reporting en prenant toutes les features | 5 |
| 4.2 | Reporting en prenant uniquement les 10 features les plus corrélées à la target | 6 |
| 5 | Notebook | 6 |

1 Présentation du dataset

Le dataframe étudié contient 1460 lignes et 81 colonnes. La valeur à prédire dans ce dataset est le prix de vente. Parmi ces 81 colonnes, il y a 38 colonnes contenant des valeurs numériques. Ce dataset est un dataset classique concernant le prix de l'immobilier en fonction des caractéristiques des produits.

2 Data Exploration

Dans un premier temps, nous allons explorer le dataset afin de voir comment est rempli le dataset. Pour les colonnes contenant des valeurs nulles, voici le nombre de valeurs nulles par colonnes.

Nombre de valeurs nulles colonnes :

```
- LotFrontage : 259
- Alley : 1369
- MasVnrType : 8
- MasVnrArea : 8
- BsmtQual : 37
- BsmtCond : 37
- BsmtExposure : 38
- BsmtFinType1 : 37
- BsmtFinType2 : 38
- Electrical : 1
- FireplaceQu : 690
- GarageType : 81
- GarageYrBlt : 81
- GarageFinish : 81
- GarageQual : 81
- GarageCond : 81
- PoolQC : 1453
- Fence : 1179
- MiscFeature : 1406
```

FIGURE 1 – Liste des colonnes contenant des valeurs non nulles

Etant donné que nous travaillons sur un sujet de prédiction numérique, la corrélation des données à la target est intéressante. Voici la liste des 10 colonnes les plus corrélées au prix de vente.

| Correlation to the target | |
|---------------------------|----------|
| SalePrice | 1.000000 |
| OverallQual | 0.790982 |
| GrLivArea | 0.708624 |
| GarageCars | 0.640409 |
| GarageArea | 0.623431 |
| TotalBsmtSF | 0.613581 |
| 1stFlrSF | 0.605852 |
| FullBath | 0.560664 |
| TotRmsAbvGrd | 0.533723 |
| YearBuilt | 0.522897 |
| YearRemodAdd | 0.507101 |

FIGURE 2 – Top 10 des colonnes corrélées à la target

3 Data cleaning & Processing

Le but de ce TD n'est pas concentré sur la préparation des données mais plutôt sur la modélisation et le tuning des modèles. Cependant il est nécessaire de réaliser tout de même un processing de la donnée. Dans cette section, nous allons expliquer le traitement des données dans l'optique d'entraîner nos modèles.

Dans un premier temps, nous allons retirer les colonnes non numériques étant donné qu'elles ne peuvent nous servir pour notre sujet de régression. Même si nous avons déterminé les colonnes les plus corrélées à la target, nous allons utiliser toutes les colonnes étant donné que certaines des colonnes peuvent permettre de régler certains détails. De plus, nous avons supprimé toutes les colonnes ne contenant des valeurs Na ou NaN qui sont des valeurs non attribuées, nous n'avons pas pris l'initiative de les remplacer des d'autres valeurs par exemple la moyenne, car cela aurait pu fausser le modèle étant donné que le modèle dépend également d'autres features.

Nous avons également décidé de normaliser les données d'entraînement et de tests pour les modèles tels que le random forest regressor ou l'extra Trees.

4 Modélisation & tuning

Nous devons ici prédire le prix de vente de l'immobilier. Nous sommes sur de l'apprentissage supervisé et plus précisément sur un sujet de régression. Nous allons tester 4 modèles de régression :

- La régression linéaire multiple
- XG Boost
- Random Forest regression
- Extra Trees

Le score de métrique que nous utilisons est le **R²**. Après avoir implémenté ces modèles, nous allons ensuite les tuner afin de trouver les hyper-paramètres optimaux. Pour tuner nos modèles, nous allons faire varier les valeurs des hyper paramètres et retenir la combinaison d'hyper-paramètres pour lesquels nous avons le meilleur score.

Pour le random Forest, les paramètres que nous faisons varier sont :

```
param = {'n_estimators' : [int(x) for x in np.linspace(start=80, stop=100, num=10)],
        'max_depth' : [80, 90, 100],
        'min_samples_split' : [4, 8],
        'min_samples_leaf' : [2, 4],
        'bootstrap' : [True, False]
}
```

FIGURE 3 – Reporting des metrics R² des modèles sur le test set après tuning

Pour le XG Boost regressor, les paramètres que nous faisons varier sont :

```
param = {'n_estimators' : [int(x) for x in np.linspace(start=80, stop=100, num=10)],
        'max_depth' : [80, 90, 100],
}
```

FIGURE 4 – Reporting des metrics R² des modèles sur le test set après tuning

Pour le Extras Trees regressor, les paramètres que nous faisons varier sont :

```
param = {'n_estimators' : [int(x) for x in np.linspace(start=150, stop=170, num=10)],
        'max_depth' : [90, 100],
        'min_samples_split' : [6, 8],
        'min_samples_leaf' : [3, 4],
        'bootstrap' : [True, False]
}
```

FIGURE 5 – Reporting des metrics R² des modèles sur le test set après tuning

4.1 Reporting en prenant toutes les features

Voici le reporting en prenant toutes les features dans le dataset, afin de comparer l'efficacité des modèles avec et sans feature selection. Le reporting suivant présente les résultats des différents modèles après tuning. Nous retrouvons le XG Boost regressor en première position avec un score de 0.87, ce qui est très satisfaisant. On constate que le modèle s'est bien généralisé sur les données.

| | Score R2 |
|--------------------------|----------|
| model | |
| XG Boost regression | 0.872517 |
| Random Forest Regression | 0.846871 |
| Extra Trees Regressor | 0.822541 |
| Linear regression | 0.800714 |

FIGURE 6 – Reporting des metrics R2 des modèles sur le test set après tuning sans feature selection

4.2 Reporting en prenant uniquement les 10 features les plus corrélées à la target

Les modélisations suivantes ont été faites en sélectionnant uniquement les 10 features numériques les plus corrélées à la target. Voici le reporting présentant les résultats des différents modèles après tuning. Nous retrouvons le XG Boost regressor en première position avec un score de 0.88, ce qui est très satisfaisant. **Nous retrouvons de meilleurs résultats en faisant de la feature importance selection que sans en faire.** On en déduit que les autres données confusioennent les modèles étant donné que nous voyons que les performances sont meilleures en ayant moins de colonnes.

| | Score R2 |
|--------------------------|----------|
| model | |
| XG Boost regression | 0.882891 |
| Extra Trees Regressor | 0.853611 |
| Random Forest Regression | 0.849791 |
| Linear regression | 0.781361 |

FIGURE 7 – Reporting des metrics R2 des modèles sur le test set après tuning avec feature selection

5 Notebook

Vous pourrez retrouver le notebook sur mon GitHub : https://github.com/thomastrg/Real_Estate_predictions ou bien sur le dépôt DVO.