Dr. Christian Tsoungui Obama

Fiche mémo ML #1

Votre guide pratique en 7 étapes de l'idée au modèle de régression déployable

Guide fondamental pour aspirants Data Scientists

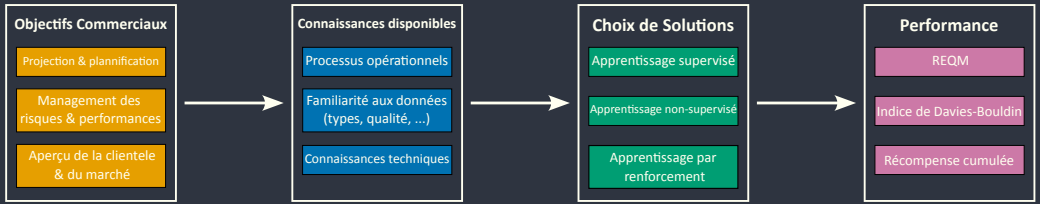
Basé sur
"Machine Learning avec Scikit-learn, Keras & TensorFlow"
par

Aurélien Géron

Étape 1 : Bien définir le problème

- Imaginez que vous collectez des données météorologiques avec un capteur. Vous vous demandez : « Si je connais l'humidité, puis-je prédire la température ? »
- Pensez maintenant à un magasin qui suit son budget publicitaire et ses ventes. Quelqu'un demande : « Si nous modifions notre budget publicitaire, pouvons-nous prédire nos ventes ? »

L'utilisation de modèles d'apprentissage automatique (ML) pour répondre à de telles questions nécessite une définition adéquate du problème et de son contexte.



Feuille de route d'un projet de ML : illustration des points clés à considérer lors de la définition du problème

Les étapes pour construire des modèles de régression sont décrites ci-après.

Étape 2 : Acquérir les données

Sources de données possibles :

- données propriétaires (ex. : bases de données de l'entreprise, données opérationnelles)
- sources en ligne (ex. : web scraping, API)
- dépôts de données publics et ouverts (ex. : Kaggle, UCI)

```python
import kagglehub
from kagglehub import KaggleDatasetAdapter

# Chargement d'un jeu de donnees depuis Kaggle

adsSales = kagglehub.dataset_load(
                                KaggleDatasetAdapter.PANDAS,
                                "thorgodofthunder/tvradionewspaperadvertising/versions/1",
                                "Advertising.csv",
                                  )
```

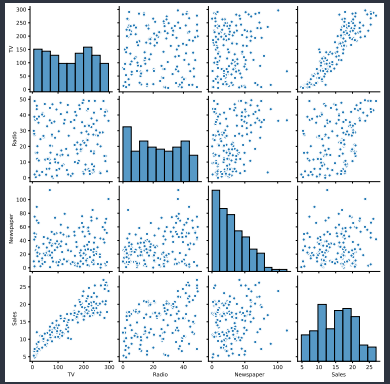
Exemple de code pour importer le jeu de données "Advertising" depuis Kaggle[1]

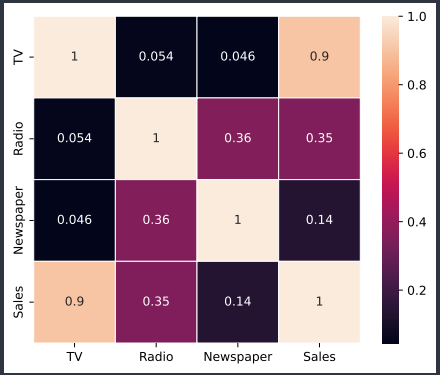
[1] www.kaggle.com/datasets/thorgodofthunder/tvradionewspaperadvertising

Étape 3 : Explorer les données

Analyse exploratoire des données (AED)

1. diviser les données en ensembles d'entraînement et de test
2. examiner la corrélation linéaire (attributs numériques)
3. transformer les données à l'aide de pipelines (ex: imputation, one-hot-encoding)





Matrice de nuages de points : le nuage de points ressemble à une ligne droite pour les attributs linéairement corrélés

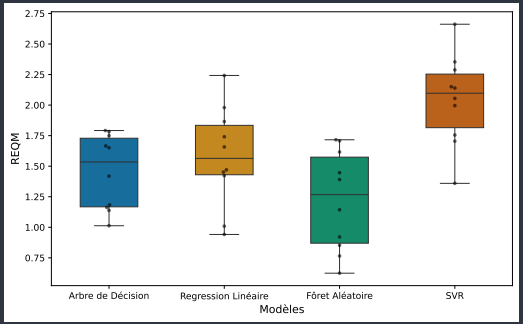
Matrice de corrélation : les valeurs quantifient la force et la direction de la corrélation linéaire

Étape 4 : Entraîner et évaluer les modèles candidats

Quelques modèles candidats pour la régression :

Modèles	Avantages	Inconvénients
Régression linéaire	simple, interprétable, rapide à entraîner	suppose la linéarité, sensible aux données aberrantes et à la multicolinéarité
Régresseur par arbre de décision	modélise la non-linéarité, facile à visualiser et à interpréter	sujet au surajustement, sensible aux changements de données et à l'asymétrie
Régression par machine à vecteurs de support (SVR)	modélise la non-linéarité, efficace pour les données de haute dimension et de taille moyenne	coûteux en calcul pour les larges jeux de données, difficile à interpréter et sensible au choix du noyau

```
1  # Entrainement d'un modele de regression lineaire
2
3  from sklearn.linear_model import LinearRegression
4  from sklearn.metrics import mean_squared_error
5  import numpy as np
6
7  lin_reg = LinearRegression()
8  lin_reg.fit(adsSales_prepared, adsSales_labels)
9
10 adsSales_predictions = lin_reg.predict(
       adsSales_prepared)
11 lin_mse = mean_squared_error(adsSales_labels,
       adsSales_predictions)
12
13 lin_rmse = np.sqrt(lin_mse)
```



Performance des modèles : scores de racine d'erreur quadratique moyenne (REQM – le plus petit est le meilleur)

Étape 5 : Affiner le meilleur candidat (Forêts aléatoires)

Les hyperparamètres du modèle sont affinés pour améliorer ses performances :

1. explorer un large espace avec la recherche aléatoire
2. affiner la recherche avec la recherche par grille
3. utiliser la validation croisée dans les deux cas

```
# Reglage des hyperparametres par recherche par grille
from sklearn.model_selection import GridSearchCV

param_grid = [
    {'n_estimators': [3, 10, 30, 50], 'max_features': [6, 8, 10]},
    {'bootstrap': [False], 'n_estimators': [3, 10], 'max_features': [2, 3, 4]},
]

forest_reg = RandomForestRegressor()
grid_search = GridSearchCV(forest_reg, param_grid, cv=5,
                           scoring='neg_mean_squared_error',
                           return_train_score=True)

# Entrainer le modele pour toutes les combinaisons de parametres
grid_search.fit(adsSales_prepared, adsSales_labels)

# Selection du meilleur modele de la recherch par grille
final_model = grid_search.best_estimator_
```

Étape 6 : Évaluer la généralisation du modèle grâce au jeu de données de test

Préparer le jeu de données de test avec le même pipeline que pour les données d'entraînement

```
# Definition de le jeu de donnees de test
X_test = strat_test_set.drop("Sales", axis=1)
y_test = strat_test_set["Sales"].copy()

X_test_prepared = full_pipeline.transform(X_test)
```

Évaluer la performance du modèle

```
# Execution des predictions sur l'ensemble de test
final_predictions = final_model.predict(X_test_prepared)

final_mse = mean_squared_error(y_test, final_predictions)
final_rmse = np.sqrt(final_mse)
```

Étape 7 : Sauvegarder le modèle pour un déploiement éventuel

```python
# Sauvegarde du modele en tant que fichier joblib

def save_model(model, name="model", model_path="./"):
    """
    Sauvegarde le modele avec un nom donne a un emplacement specifique.

    Parametres
    ----------
    model :
        Le modele entraine.
    name:
        Le nom de fichier sous lequel sauvegarder le modele
    model_path:
        Le chemin dans lequel le modele est sauvegarde
        Exemple:
            "./models/"

    """
    os.makedirs(model_path, exist_ok=True)
    model_name = name + ".pkl"
    pkl_path = os.path.join(model_path, model_name)
    joblib.dump(model, pkl_path)

MODEL_PATH = os.path.join("models", "adsSales")
save_model(final_model, name="Foret_aleatoire_V1.1", model_path=MODEL_PATH)
```

Avez-vous trouvé cela utile ?

Obtenez le notebook complet : » Dépôt GitHub «

Commentez ci-dessous : Comment choisissez-vous vos modèles candidats pour la régression ?

Suivez-moi : pour la prochaine fiche mémo de cette série.

Aimez ce post : pour l'aider à atteindre plus de monde.

Restons en contact !

/in/christiantsoungui @chrisTs