Apprentissage supervisé - Regression, DecisionTreeRegressor, RandomForestRegressor

SOMMAIRE

- les différentes notions
- informations sur les données
- nettoyage des données
- sélection apprentissage et évaluation du modèle

LES DIFFÉRENTES NOTIONS :

- Les régressions
- La classification
- Decision TreeRegressor
- Randon Forestregressor

Les régressions

- La régression linéaire cherche à établir, sous forme d'une droite, u**ne relation entre une variable expliquée et une variable explicative.** En d'autres termes, les données d'une série d'observations sont représentées sous forme d'un nuage de points et l'on cherche à trouver une droite passant au plus près de ces points.
-
- La régression multiple utilise **plusieurs variables explicatives** (contrairement à la linéaire).Après la normalisation, nous passons à la prédiction. Ces deux méthodes prenant en compte les différentes variables explicatives mises à l'échelle dans le but de prédire la variable expliquée.
-
- La régression polynomiale , approche statistique employée pour modéliser une forme non linéaire entre X (variable explicative) et Y(réponse).
-

La classification

Type d'apprentissage supervisé qui consiste à produire des modèles prédictifs de classification.

Un modèle prend en entrée une observation caractérisée par un ensemble de variables explicatives et produit en sortie la classe (valeur discrète) à laquelle appartient l'observation.

Parmi ces algorithmes :

- régression logistique : algorithme de classification binaire. Il utilise une fonction sigmoïde pour prédire la probabilité d'appartenance à une classe(positive ou négative)
- arbres de décisions
- K plus proches voisins
- machines à vecteurs de support
- forêts aléatoires (random forest)

Les 4 derniers servent aussi bien pour la classification que la régression.

Decision TreeRegressor

Principe de fonctionnement

Un arbre de décision permet d'expliquer une variable cible à partir d'autres variables dites explicatives.

Du point de vue mathématique :une matrice X avec m observations et n variables, associée à un vecteur Y à expliquer

: il faut trouver une relation entre X et Y

Construction des règles

L’arbre de décision est un algorithme itératif qui, à chaque itération, va séparer les individus en k groupes (généralement k=2 et on parle d’arbre binaire) pour expliquer la variable cible.

La première division (on parle aussi de *split*) est obtenue en choisissant la variable explicative qui permet la meilleure séparation des individus. Cette division donne des sous-populations correspondant au premier nœud de l'arbre.

Le processus de split est ensuite répété plusieurs fois pour chaque sous-population (noeuds précédemment calculés) jusqu'à ce que le processus de séparation s’arrête.

RandomForestRegressor

Les forêts aléatoires sont un moyen de résoudre le problème de surapprentissage dû aux arbres de décisions.

Une forêt aléatoire est essentiellement une collection d'arbres de décision légèrement différents les uns des autres.(basés sur un échantillon aléatoire d’observations devant prédire la cible et être différent des autres.

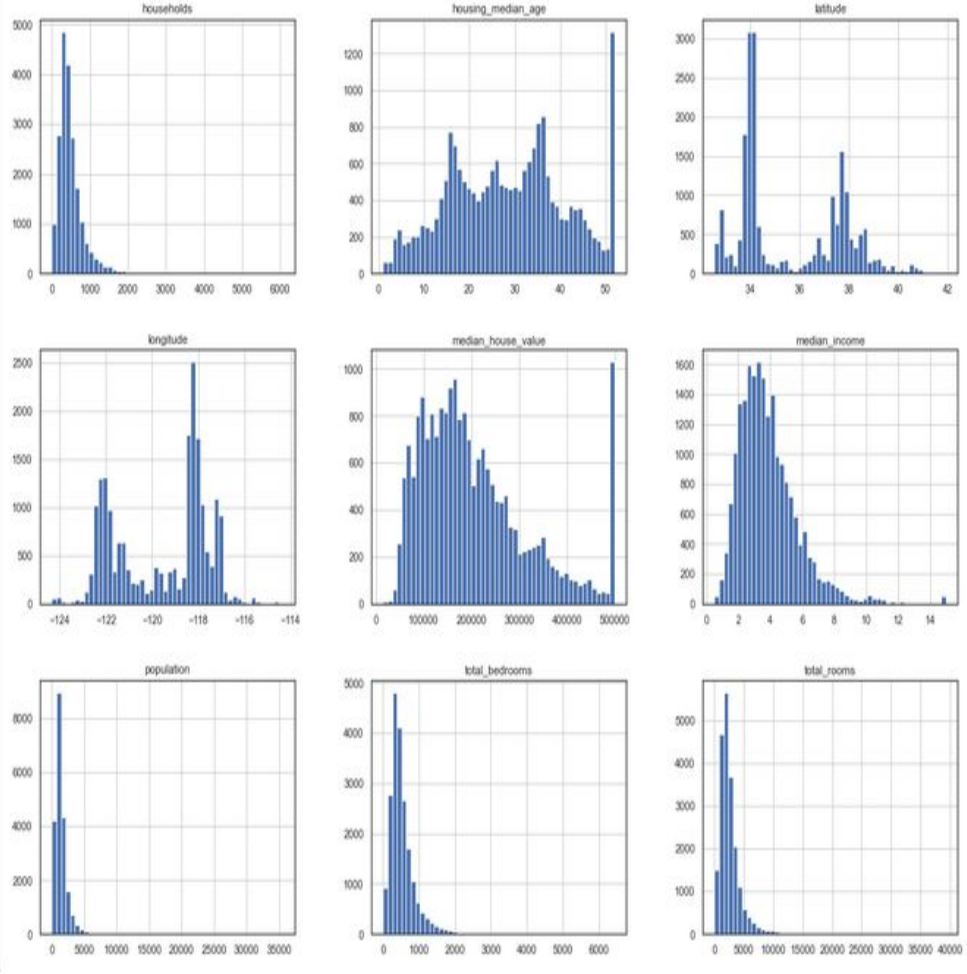
INFORMATIONS SUR LES DONNEES

Après le chargement des données, un rapide coup d'oeil permet d'avoir une première idée des données que nous manipulons :

- Notre base comprends 20 640 données, Ce qui permet une étude significative.
- L'attribut "total_bedrooms" possède 207 données nilles.
- La formule Value _count s, nous permet de mesurer la fréquence de chaque donnée dans la colonne "ocean proximity". On constate ainsi que la majorité des biens se situe à 1h de l'océan.
- On constate que le prix moyen des logements est de 207k$ et les biens ont en moyenne 29 ans.. les prix peuvent aller jusqu'à 500K$.
- Notre jeu de données se compose de plusieurs caractéristiques (features) et une target (les prix des maisons dans l'état de la californie.)

but : à partir de plusieurs caractéristiques estimer les prix des maisons en Californie,

à travers cet exemple nous travaillerons sur les 2 possibilités classification et regression.

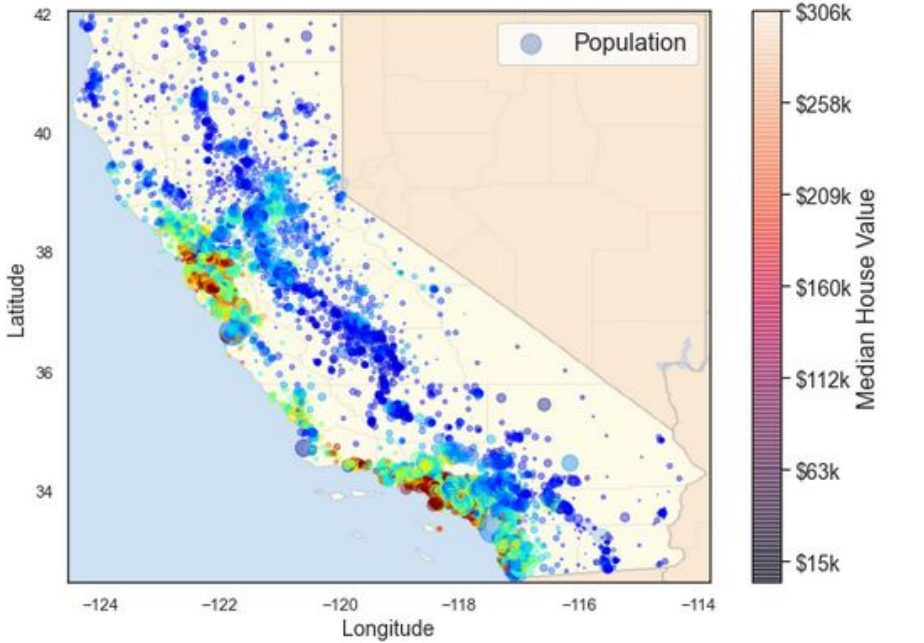


La méthode “hist” ignore les variables non numériques.

Les Variables en “forme de cloche” répondent plus au moins à la loi normale. (les revenus).

On remarque aussi les données abérahantes

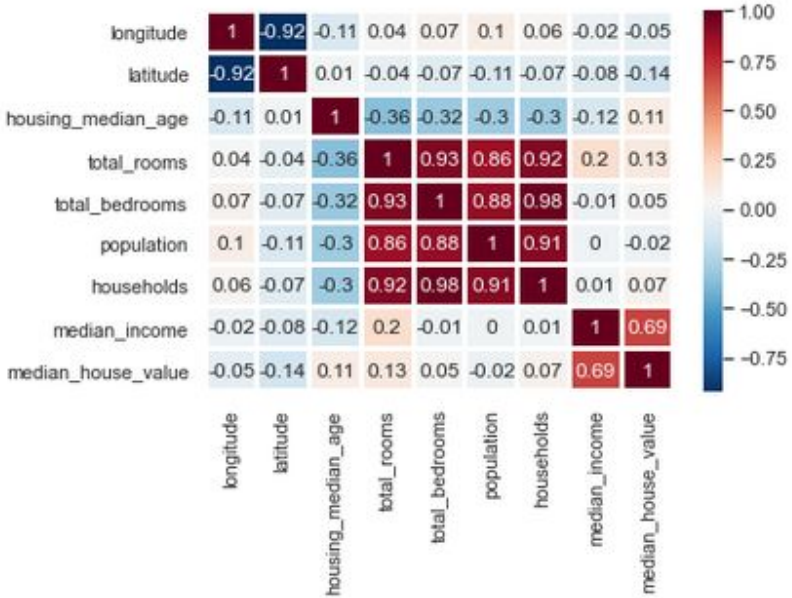
lien entre la position géographique et le prix des maisons (target).



Cette analyse, nous permet de confirmer visuellement que les biens les plus élevés sont situés au bord de la mer.

Et qu'il existe donc un lien entre la position géographique du bien et son prix.

la corrélation de l’attribut "median_ house_value" avec les autres attributs.



La corrélation nous permet de voir que le revenu et le prix des maisons sont fortement liés.

Ainsi que la latitude et la longitude mais par des relations inversées

On remarque que la valeur de la maison est liée positivement au revenu moyen et négativement à l'éloignement par rapport à la mer.

Nettoyage des données

Avant d’intégrer les données dans un algorithme d’apprentissage automatique, il est indispensable de nettoyer les données.

- créer deux variables
- éliminer les variables manquantes, via 3 possibilités : supprimer les valeurs manquantes (NaN), supprimer l’attribut "total_bedrooms", remplacer les valeurs manquantes par une autre valeur (0, la moyenne, la médiane. . .) Nous optons pour cette méthode.
- modifier les données numériques via l’encodage (cf"ocean_proximity”)

Sélection, apprentissage et évaluation du modèle

3. Calculez la mesure RMSE du modèle de la régression linéaire.

```
[65]: 1 mse2 = mean_squared_error(y_train, y_pred_train)
      2 print(mse2)

4810958229.787788
```

4 -Refaites les deux étapes précédentes avec le modèle DecisionTreeRegressor. Calculez la

mesure RMSE du modèle DecisionTreeRegressor qui existe dans le sous-module tree du module sklearn. Pour plus d'informations sur les arbres de décision:

```
[67]: 1 from sklearn.tree import DecisionTreeRegressor

[68]: 1 model_decision = DecisionTreeRegressor(random_state =42)
      2 model_decision.fit(X_train, y_train)

8]: DecisionTreeRegressor(random_state=42)

[87]: 1 y_pred = model_decision.predict(X_train)
      2 from sklearn.metrics import r2_score
      3 print((model_decision.score(X_train,y_train)))
      4 print((np.sqrt(mean_squared_error(y_train,y_pred))))

1.0
0.0
```

MSE = 0, Surapprentissage avec les arbres de regressions données non exploitables

.MSE = 0, Surapprentissage avec les arbres de regressions données non exploitables

juqu'à présent la régression semble plus adaptée

• L'écart type de tous les folds

rée [29]:
```
validcross = cross_val_score(model_decision, X_train, y_train, cv=10)
validcross
```

ut[29]:
```
array([0.63933283, 0.63989045, 0.63534847, 0.57349012, 0.61797954,
       0.66594384, 0.6373525 , 0.63962735, 0.59763641, 0.6389786 ])
```

La validation croisée nous donne 10 K-folds compris entre 0,6 et 0,7

rée [30]:
```
validcross.mean()
```

ut[30]:
```
0.6285580099000441
```

rée [31]:
```
validcross.std()
```

ut[31]:
```
0.024692742495461828
```

```
l'ecart type étant proche de 0, nous pouvons en déduire que nous avons une faible dispersionde valeurs
```

Avantage/inconvénients des arbres de décision

- **Avantages**
  - Facilité à manipuler des données « symboliques »
  - OK avec variables d'amplitudes très différentes
  - Multi-classe par nature
  - Interprétabilité de l'arbre !
  - Identification des inputs « importants »
  - Classification très efficace (en particulier sur inputs de grande dimension)
- **Inconvénients**
  - Sensibilité au bruit et points aberrants
  - Stratégie d'élagage délicate

Avantages et inconvénient des Forêts Aléatoires

- **Avantages**
  - **Reconnaissance TRES RAPIDE**
  - **Multi-classes par nature**
  - **Efficace sur inputs de grande dimension**
  - **Robustesse aux outliers**

- **Inconvénients**
  - **Apprentissage souvent long**
  - **Valeurs extrêmes souvent mal estimées dans cas de régression**

Dans la pratique les arbres de décisions sont plus exploitables.

Cette étude nous met en avant l’importance du choix des données, de leur fiabilité, et de l’importance de choisir le bon modèle afin d’exploiter au mieu ces données et de pouvoir en faire une analyse pertinente.

La régression peut être le meilleur choix sur des données d’apprentissage alors que la classification peut être idéale pour les données d'entraînement

Les princiipes sont compris, faute de temps je n’ai pas pu aller au bout de ce que je souhaitais.