Adaboost ou XGBoost - Exercice

March 13, 2024

1 Adaboost ou XGBoost ?

Deux algorithmes de boosting connus sont Adaboost & XGBoost, voyons voir la performance de chacun de ces algorithmes sur le dataset d'AIRBNB Seattle. Notre but va être de prédire le prix d'un appartement en fonction des caractéristiques qu'on nous a donné.

1. Importez les librairies usuelles

[31]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import AdaBoostRegressor
from xgboost import XGBRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[31], line 6
      4 from sklearn.metrics import mean_squared_error
      5 from sklearn.ensemble import AdaBoostRegressor
----> 6 from xgboost import XGBRegressor
      7 from sklearn.preprocessing import StandardScaler
      8 from sklearn.pipeline import make_pipeline

ModuleNotFoundError: No module named 'xgboost'
```

2. Importez le dataset `listings.csv`

[32]:
```
listings= pd.read_csv('/Users/msm/Downloads/listings.csv')
```

[33]:
```
listings.head()
```

[33]:

	id	listing_url	scrape_id	last_scraped	\
0	241032	https://www.airbnb.com/rooms/241032	20160104002432	2016-01-04	
1	953595	https://www.airbnb.com/rooms/953595	20160104002432	2016-01-04	
2	3308979	https://www.airbnb.com/rooms/3308979	20160104002432	2016-01-04	

```
3  7421966  https://www.airbnb.com/rooms/7421966  20160104002432   2016-01-04
4   278830   https://www.airbnb.com/rooms/278830  20160104002432   2016-01-04

                                name  \
0        Stylish Queen Anne Apartment
1   Bright & Airy Queen Anne Apartment
2  New Modern House-Amazing water view
3                  Queen Anne Chateau
4       Charming craftsman 3 bdm house

                                             summary  \
0                                                NaN
1  Chemically sensitive? We've removed the irrita…
2  New modern house built in 2013.  Spectacular s…
3  A charming apartment that sits atop Queen Anne…
4  Cozy family craftman house in beautiful neighb…

                                               space  \
0  Make your self at home in this charming one-be…
1  Beautiful, hypoallergenic apartment in an extr…
2  Our house is modern, light and fresh with a wa…
3                                                NaN
4  Cozy family craftman house in beautiful neighb…

                                         description experiences_offered  \
0  Make your self at home in this charming one-be…                 none
1  Chemically sensitive? We've removed the irrita…                 none
2  New modern house built in 2013.  Spectacular s…                 none
3  A charming apartment that sits atop Queen Anne…                 none
4  Cozy family craftman house in beautiful neighb…                 none

                            neighborhood_overview  … review_scores_value  \
0                                             NaN  …                10.0
1  Queen Anne is a wonderful, truly functional vi…  …                10.0
2  Upper Queen Anne is a charming neighborhood fu…  …                10.0
3                                             NaN  …                 NaN
4  We are in the beautiful neighborhood of Queen …  …                 9.0

  requires_license license jurisdiction_names instant_bookable  \
0                f     NaN         WASHINGTON                f
1                f     NaN         WASHINGTON                f
2                f     NaN         WASHINGTON                f
3                f     NaN         WASHINGTON                f
4                f     NaN         WASHINGTON                f

  cancellation_policy  require_guest_profile_picture  \
0            moderate                              f
```

```
1              strict                              t
2              strict                              f
3            flexible                              f
4              strict                              f

  require_guest_phone_verification calculated_host_listings_count  \
0                                f                               2
1                                t                               6
2                                f                               2
3                                f                               1
4                                f                               1

  reviews_per_month
0              4.07
1              1.48
2              1.15
3               NaN
4              0.89

[5 rows x 92 columns]
```

3. On a beaucoup de données dans ce dataset. Affichez toutes les colonnes du dataset

[34]:
```
# Afficher les noms des colonnes
listings.info()
```

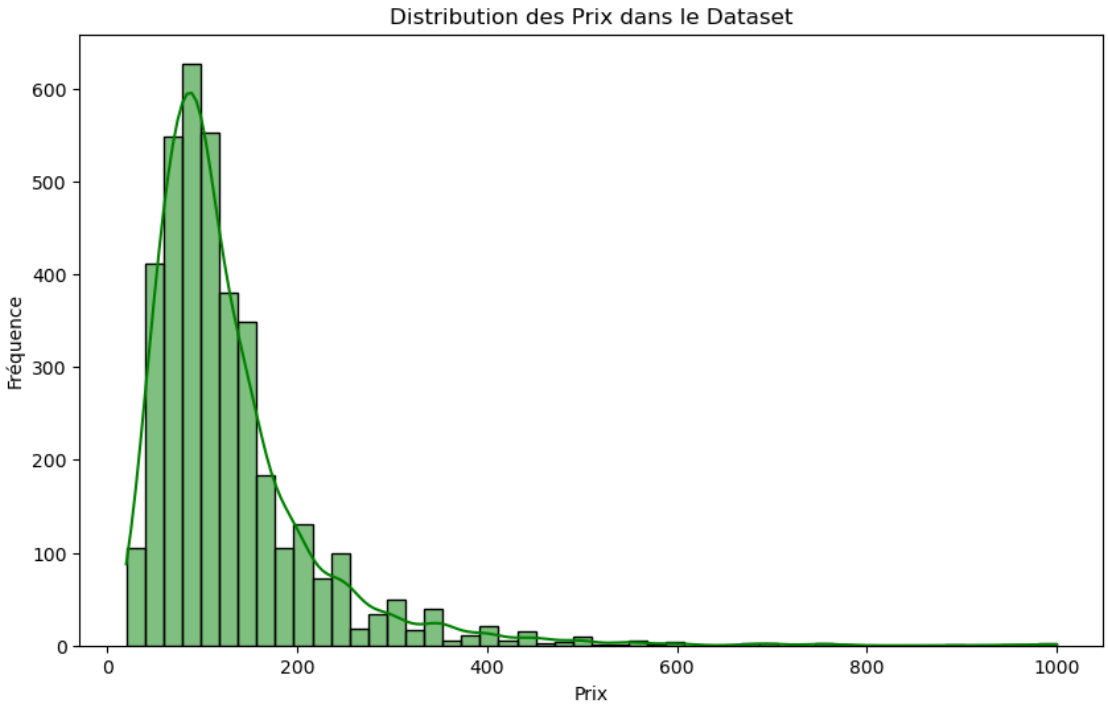
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3818 entries, 0 to 3817
Data columns (total 92 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   id                                3818 non-null   int64
 1   listing_url                       3818 non-null   object
 2   scrape_id                         3818 non-null   int64
 3   last_scraped                      3818 non-null   object
 4   name                              3818 non-null   object
 5   summary                           3641 non-null   object
 6   space                             3249 non-null   object
 7   description                       3818 non-null   object
 8   experiences_offered               3818 non-null   object
 9   neighborhood_overview             2786 non-null   object
 10  notes                             2212 non-null   object
 11  transit                           2884 non-null   object
 12  thumbnail_url                     3498 non-null   object
 13  medium_url                        3498 non-null   object
 14  picture_url                       3818 non-null   object
 15  xl_picture_url                    3498 non-null   object
 16  host_id                           3818 non-null   int64
```

```
17  host_url                        3818 non-null   object
18  host_name                       3816 non-null   object
19  host_since                      3816 non-null   object
20  host_location                   3810 non-null   object
21  host_about                      2959 non-null   object
22  host_response_time              3295 non-null   object
23  host_response_rate              3295 non-null   object
24  host_acceptance_rate            3045 non-null   object
25  host_is_superhost               3816 non-null   object
26  host_thumbnail_url              3816 non-null   object
27  host_picture_url                3816 non-null   object
28  host_neighbourhood              3518 non-null   object
29  host_listings_count             3816 non-null   float64
30  host_total_listings_count       3816 non-null   float64
31  host_verifications              3816 non-null   object
32  host_has_profile_pic            3816 non-null   object
33  host_identity_verified          3816 non-null   object
34  street                          3818 non-null   object
35  neighbourhood                   3402 non-null   object
36  neighbourhood_cleansed          3818 non-null   object
37  neighbourhood_group_cleansed    3818 non-null   object
38  city                            3818 non-null   object
39  state                           3818 non-null   object
40  zipcode                         3811 non-null   object
41  market                          3818 non-null   object
42  smart_location                  3818 non-null   object
43  country_code                    3818 non-null   object
44  country                         3818 non-null   object
45  latitude                        3818 non-null   float64
46  longitude                       3818 non-null   float64
47  is_location_exact               3818 non-null   object
48  property_type                   3817 non-null   object
49  room_type                       3818 non-null   object
50  accommodates                    3818 non-null   int64
51  bathrooms                       3802 non-null   float64
52  bedrooms                        3812 non-null   float64
53  beds                            3817 non-null   float64
54  bed_type                        3818 non-null   object
55  amenities                       3818 non-null   object
56  square_feet                     97 non-null     float64
57  price                           3818 non-null   object
58  weekly_price                    2009 non-null   object
59  monthly_price                   1517 non-null   object
60  security_deposit                1866 non-null   object
61  cleaning_fee                    2788 non-null   object
62  guests_included                 3818 non-null   int64
63  extra_people                    3818 non-null   object
64  minimum_nights                  3818 non-null   int64
```

```
 65  maximum_nights                    3818 non-null   int64
 66  calendar_updated                  3818 non-null   object
 67  has_availability                  3818 non-null   object
 68  availability_30                   3818 non-null   int64
 69  availability_60                   3818 non-null   int64
 70  availability_90                   3818 non-null   int64
 71  availability_365                  3818 non-null   int64
 72  calendar_last_scraped             3818 non-null   object
 73  number_of_reviews                 3818 non-null   int64
 74  first_review                      3191 non-null   object
 75  last_review                       3191 non-null   object
 76  review_scores_rating              3171 non-null   float64
 77  review_scores_accuracy            3160 non-null   float64
 78  review_scores_cleanliness         3165 non-null   float64
 79  review_scores_checkin             3160 non-null   float64
 80  review_scores_communication       3167 non-null   float64
 81  review_scores_location            3163 non-null   float64
 82  review_scores_value               3162 non-null   float64
 83  requires_license                  3818 non-null   object
 84  license                           0 non-null      float64
 85  jurisdiction_names                3818 non-null   object
 86  instant_bookable                  3818 non-null   object
 87  cancellation_policy               3818 non-null   object
 88  require_guest_profile_picture     3818 non-null   object
 89  require_guest_phone_verification  3818 non-null   object
 90  calculated_host_listings_count    3818 non-null   int64
 91  reviews_per_month                 3191 non-null   float64
dtypes: float64(17), int64(13), object(62)
memory usage: 2.7+ MB
```

4. Révisons un peu Seaborn, affichez la distribution des prix dans le dataset

[39]:
```
import seaborn as sns
import matplotlib.pyplot as plt
# Convertissez la colonne 'price' en un format numérique
listings['price'] = listings['price'].replace('[\$,]', '', regex=True).
 ↪astype(float)
# Configurez les paramètres du graphique
plt.figure(figsize=(10, 6))
sns.histplot(listings['price'], bins=50, kde=True, color='green')
# Ajoutez des étiquettes et un titre
plt.title('Distribution des Prix dans le Dataset')
plt.xlabel('Prix')
plt.ylabel('Fréquence')
# Affichez le graphique
plt.show()
```



5. Supprimez les outliers pour ne garder que les appartements qui ont un prix inférieur à 400$/nuit

[41]:
```
listings_filtered = listings[listings['price'] <= 400]
listings_filtered
```

[41]:

	id	listing_url	scrape_id	\
0	241032	https://www.airbnb.com/rooms/241032	20160104002432	
1	953595	https://www.airbnb.com/rooms/953595	20160104002432	
3	7421966	https://www.airbnb.com/rooms/7421966	20160104002432	
5	5956968	https://www.airbnb.com/rooms/5956968	20160104002432	
6	1909058	https://www.airbnb.com/rooms/1909058	20160104002432	
…	…	…	…	
3813	8101950	https://www.airbnb.com/rooms/8101950	20160104002432	
3814	8902327	https://www.airbnb.com/rooms/8902327	20160104002432	
3815	10267360	https://www.airbnb.com/rooms/10267360	20160104002432	
3816	9604740	https://www.airbnb.com/rooms/9604740	20160104002432	
3817	10208623	https://www.airbnb.com/rooms/10208623	20160104002432	

	last_scraped	name	\
0	2016-01-04	Stylish Queen Anne Apartment	
1	2016-01-04	Bright & Airy Queen Anne Apartment	
3	2016-01-04	Queen Anne Chateau	
5	2016-01-04	Private unit in a 1920s mansion	

```
6      2016-01-04     Queen Anne Private Bed and Bath
…           …                                  …
3813   2016-01-04  3BR Mountain View House in Seattle
3814   2016-01-04   Portage Bay View!-One Bedroom Apt
3815   2016-01-04   Private apartment view of Lake WA
3816   2016-01-04   Amazing View with Modern Comfort!
3817   2016-01-04           Large Lakefront Apartment

                                                summary  \
0                                                   NaN
1     Chemically sensitive? We've removed the irrita…
3     A charming apartment that sits atop Queen Anne…
5     We're renting out a small private unit of one …
6     Enjoy a quiet stay in our comfortable 1915 Cra…
…                                                   …
3813  Our 3BR/2BA house boasts incredible views of t…
3814  800 square foot 1 bedroom basement apartment w…
3815  Very comfortable lower unit. Quiet, charming m…
3816  Cozy studio condo in the heart on Madison Park…
3817  All hardwood floors, fireplace, 65" TV with Xb…

                                                 space  \
0     Make your self at home in this charming one-be…
1     Beautiful, hypoallergenic apartment in an extr…
3                                                   NaN
5     If you include a bit of your background in you…
6     Enjoy a quiet stay in our comfortable 1915 Cra…
…                                                   …
3813  Our 3BR/2BA house bright, stylish, and wheelch…
3814  This space has a great view of Portage Bay wit…
3815                                                NaN
3816  Fully furnished unit to accommodate most needs…
3817                                                NaN

                                           description experiences_offered  \
0     Make your self at home in this charming one-be…                 none
1     Chemically sensitive? We've removed the irrita…                 none
3     A charming apartment that sits atop Queen Anne…                 none
5     We're renting out a small private unit of one …                 none
6     Enjoy a quiet stay in our comfortable 1915 Cra…                 none
…                                                   …                  …
3813  Our 3BR/2BA house boasts incredible views of t…                 none
3814  800 square foot 1 bedroom basement apartment w…                 none
3815  Very comfortable lower unit. Quiet, charming m…                 none
3816  Cozy studio condo in the heart on Madison Park…                 none
3817  All hardwood floors, fireplace, 65" TV with Xb…                 none
```

```
                                neighborhood_overview  …  \
0                                                 NaN  …
1     Queen Anne is a wonderful, truly functional vi…  …
3                                                 NaN  …
5     This part of Queen Anne has wonderful views an…  …
6     Close restaurants, coffee shops and grocery st…  …
…                                                  …  …
3813  We're located near lots of family fun. Woodlan…  …
3814  The neighborhood is a quiet oasis that is clos…  …
3815                                              NaN  …
3816  Madison Park offers a peaceful slow pace upsca…  …
3817                                              NaN  …

     review_scores_value requires_license license jurisdiction_names  \
0                   10.0                f     NaN         WASHINGTON
1                   10.0                f     NaN         WASHINGTON
3                    NaN                f     NaN         WASHINGTON
5                   10.0                f     NaN         WASHINGTON
6                   10.0                f     NaN         WASHINGTON
…                      …              …     …                   …
3813                 8.0                f     NaN         WASHINGTON
3814                10.0                f     NaN         WASHINGTON
3815                 NaN                f     NaN         WASHINGTON
3816                 NaN                f     NaN         WASHINGTON
3817                 NaN                f     NaN         WASHINGTON

     instant_bookable cancellation_policy  require_guest_profile_picture  \
0                   f            moderate                              f
1                   f              strict                              t
3                   f            flexible                              f
5                   f              strict                              f
6                   f            moderate                              f
…                 …                  …                                 …
3813                f              strict                              f
3814                f            moderate                              f
3815                f            moderate                              f
3816                f            moderate                              f
3817                f            flexible                              f

     require_guest_phone_verification calculated_host_listings_count  \
0                                   f                               2
1                                   t                               6
3                                   f                               1
5                                   f                               1
6                                   f                               1
…                                   …                               …
3813                                f                               8
```

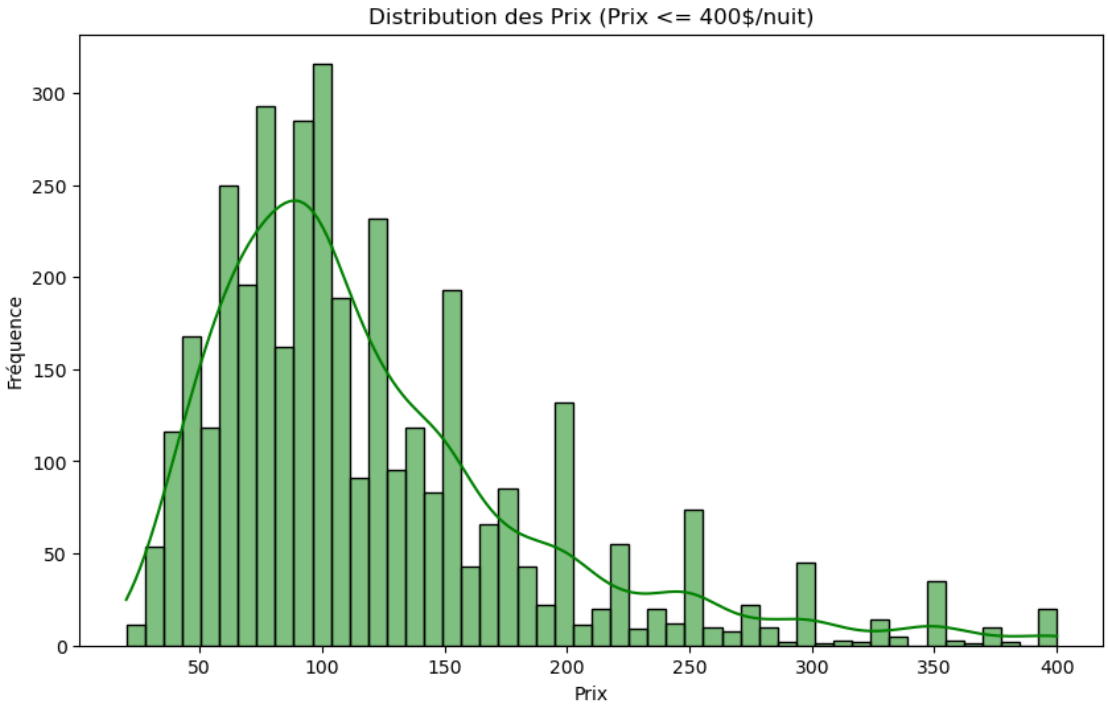
```
3814                                f                                1
3815                                f                                1
3816                                f                                1
3817                                f                                1

     reviews_per_month
0                 4.07
1                 1.48
3                  NaN
5                 2.45
6                 2.46
…                    …
3813              0.30
3814              2.00
3815               NaN
3816               NaN
3817               NaN

[3755 rows x 92 columns]
```

[ ]:

[42]:
```
import seaborn as sns
import matplotlib.pyplot as plt
# Configurez les paramètres du graphique
plt.figure(figsize=(10, 6))
sns.histplot(listings_filtered['price'], bins=50, kde=True, color='green')
# Ajoutez des étiquettes et un titre
plt.title('Distribution des Prix (Prix <= 400$/nuit)')
plt.xlabel('Prix')
plt.ylabel('Fréquence')
# Affichez le graphique
plt.show()
```

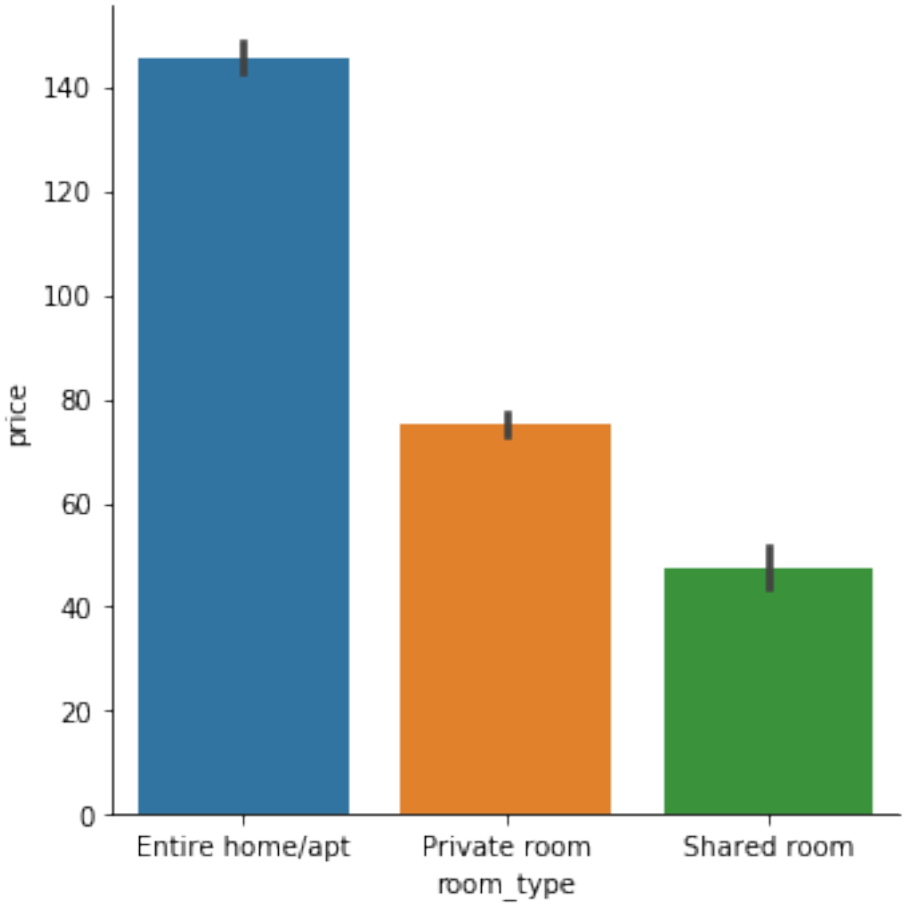


6. Exprimez le prix en fonction des variables suivantes :

- room type
- beds
- property type

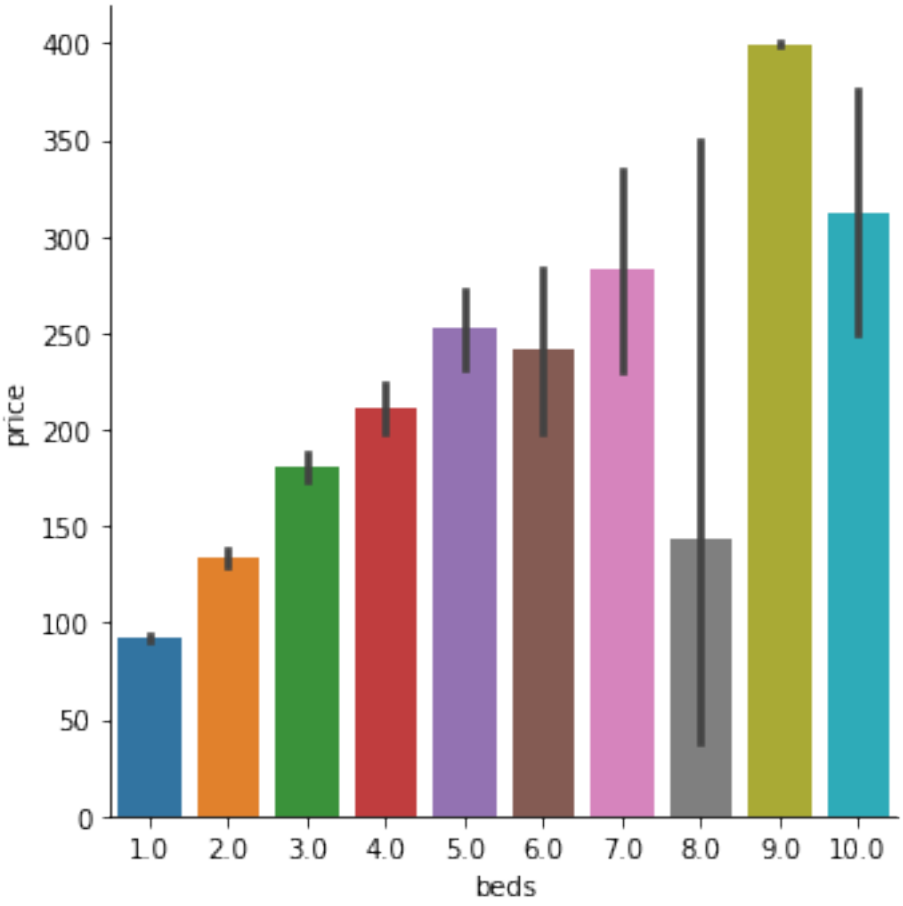
[ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
# Configurez les paramètres du graphique
plt.figure(figsize=(12, 6))
sns.boxplot(x='room_type', y='price', data=listings_filtered, palette='Set2')
# Ajoutez des étiquettes et un titre
plt.title('Prix en fonction du Type de Chambre')
plt.xlabel('Type de Chambre')
plt.ylabel('Prix ($)')
plt.yscale('log') # Utilisez une échelle logarithmique pour mieux visualiser␣
↪les variations
# Affichez le graphique
plt.show()
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7fdd316300f0>
```



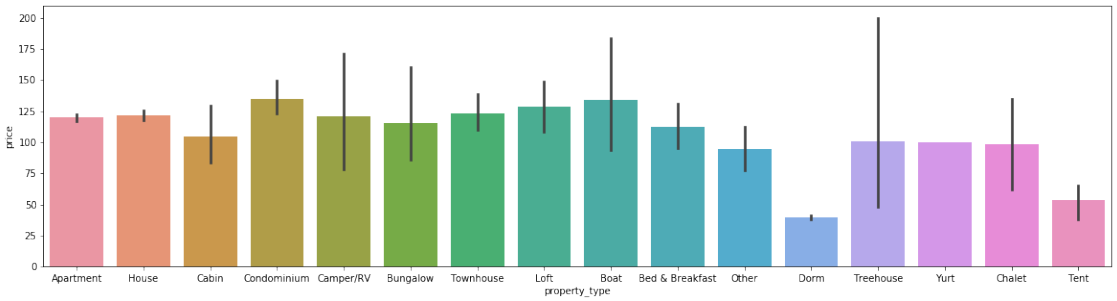
[ ]:
```
# Configurez les paramètres du graphique
plt.figure(figsize=(12, 6))
sns.scatterplot(x='beds', y='price', data=listings_filtered, hue='room_type',␣
↪palette='Set2')
# Ajoutez des étiquettes et un titre
plt.title('Prix en fonction du Nombre de Lits')
plt.xlabel('Nombre de Lits')
plt.ylabel('Prix ($)')
plt.yscale('log') # Utilisez une échelle logarithmique pour mieux visualiser␣
↪les variations
# Affichez le graphique
plt.show()
```

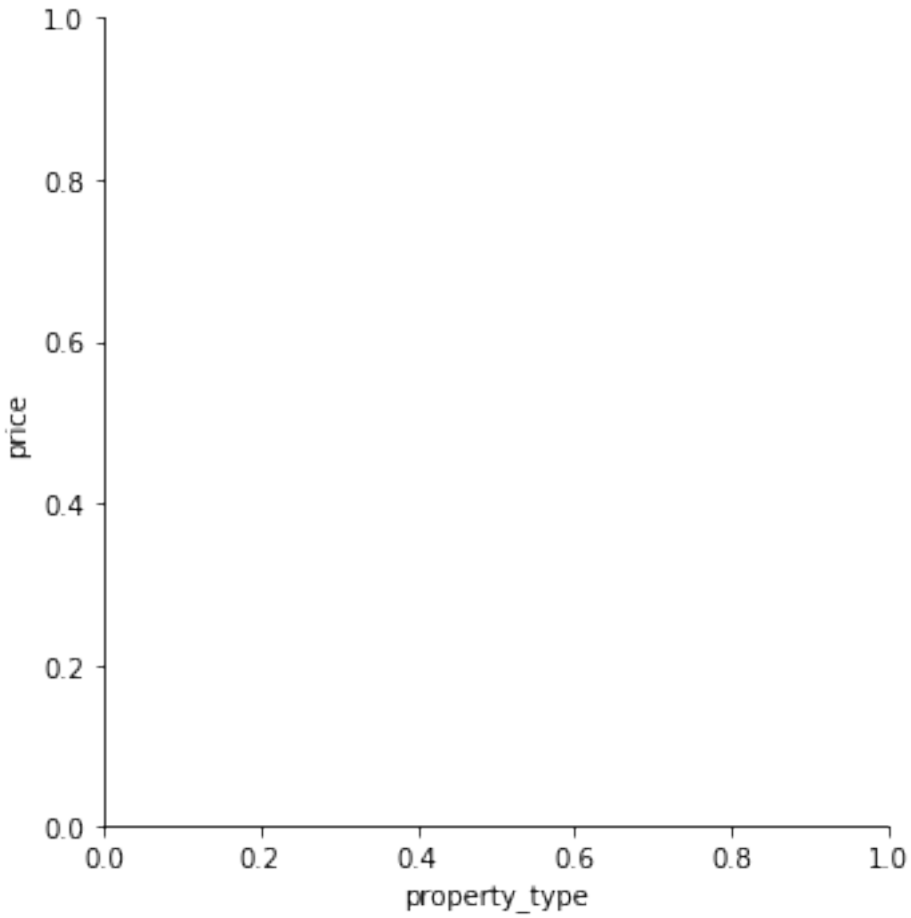
[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7fdd31654be0>
```



[ ]:
```
# Configurez les paramètres du graphique
plt.figure(figsize=(16, 6))
sns.boxplot(x='property_type', y='price', data=listings_filtered,␣
 ↪palette='Set2')
# Ajoutez des étiquettes et un titre
plt.title('Prix en fonction du Type de Propriété')
plt.xlabel('Type de Propriété')
plt.ylabel('Prix ($)')
plt.xticks(rotation=45, ha='right') # Faites pivoter les étiquettes sur l'axe x␣
 ↪pour une meilleure lisibilité
# Affichez le graphique
plt.show()
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7fdd3431c9e8>
```





7. Séparez la variable cible de votre dataset

```
[ ]: # Séparez la variable cible
y = listings_filtered['price']
# Supprimez la colonne 'price' du DataFrame pour obtenir les features
X = listings_filtered.drop('price', axis=1)
# Affichez les premières lignes des features pour vérification
```

```
print(X.head())
# Affichez les premières lignes de la variable cible pour vérification
print(y.head())
```

8. Il faut qu'on écrème quelques variables explicatives de notre jeu de données. Commencer par simplement enlever les variables qui expriment un `id` quelconque ou urls. On enlèvera également les variables qui contiennent des textes longs comme `notes`

De la même manière, on enlèvera toutes les variables qui décrivent le prix mensuel ou hebdomadaire comme `monthly price`

Votre dataset devrait contenir uniquement les variables catégoriques et numériques une fois votre nettoyage fait.

A la fin, votre dataset contiendra les variables suivantes :

```
Index(['host_response_time', 'host_response_rate', 'host_acceptance_rate',
       'host_is_superhost', 'host_neighbourhood', 'host_listings_count',
       'host_total_listings_count', 'host_has_profile_pic',
       'host_identity_verified', 'neighbourhood_group_cleansed', 'zipcode',
       'latitude', 'longitude', 'property_type', 'room_type', 'accommodates',
       'bathrooms', 'bedrooms', 'beds', 'bed_type', 'square_feet',
       'security_deposit', 'cleaning_fee', 'guests_included', 'extra_people',
       'minimum_nights', 'maximum_nights', 'calendar_updated',
       'has_availability', 'availability_30', 'availability_60',
       'availability_90', 'availability_365', 'number_of_reviews',
       'review_scores_rating', 'review_scores_accuracy',
       'review_scores_cleanliness', 'review_scores_checkin',
       'review_scores_communication', 'review_scores_location',
       'review_scores_value', 'requires_license', 'license',
       'instant_bookable', 'cancellation_policy',
       'require_guest_profile_picture', 'require_guest_phone_verification',
       'calculated_host_listings_count', 'reviews_per_month'],
      dtype='object')
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:
```
Index(['host_response_time', 'host_response_rate', 'host_acceptance_rate',
       'host_is_superhost', 'host_neighbourhood', 'host_listings_count',
       'host_total_listings_count', 'host_has_profile_pic',
       'host_identity_verified', 'neighbourhood_group_cleansed', 'zipcode',
       'latitude', 'longitude', 'property_type', 'room_type', 'accommodates',
       'bathrooms', 'bedrooms', 'beds', 'bed_type', 'square_feet',
       'security_deposit', 'cleaning_fee', 'guests_included', 'extra_people',
```

```
       'minimum_nights', 'maximum_nights', 'calendar_updated',
       'has_availability', 'availability_30', 'availability_60',
       'availability_90', 'availability_365', 'number_of_reviews',
       'review_scores_rating', 'review_scores_accuracy',
       'review_scores_cleanliness', 'review_scores_checkin',
       'review_scores_communication', 'review_scores_location',
       'review_scores_value', 'requires_license', 'license',
       'instant_bookable', 'cancellation_policy',
       'require_guest_profile_picture', 'require_guest_phone_verification',
       'calculated_host_listings_count', 'reviews_per_month'],
      dtype='object')
```

9. Gérez les valeurs NaN. Utilisez les stratégies que vous préférez

[ ]:
```
# Liste des colonnes à conserver dans le dataset final
cols_to_keep = [
'experiences_offered', 'host_response_time', 'host_response_rate',␣
 ↪'host_acceptance_rate', 'host_is_superhost', 'host_listings_count',␣
 ↪'host_total_listings_count', 'host_has_profile_pic',␣
 ↪'host_identity_verified', 'neighbourhood_group_cleansed', 'latitude',␣
 ↪'longitude', 'property_type', 'room_type', 'accommodates', 'bathrooms',␣
 ↪'bedrooms', 'beds', 'bed_type', 'security_deposit', 'cleaning_fee',␣
 ↪'guests_included', 'extra_people', 'minimum_nights', 'maximum_nights',␣
 ↪'has_availability', 'availability_30', 'availability_60', 'availability_90',␣
 ↪'availability_365', 'number_of_reviews', 'review_scores_rating',␣
 ↪'review_scores_accuracy', 'review_scores_cleanliness',␣
 ↪'review_scores_checkin', 'review_scores_communication',␣
 ↪'review_scores_location', 'review_scores_value', 'requires_license',␣
 ↪'instant_bookable', 'cancellation_policy', 'require_guest_profile_picture',␣
 ↪'require_guest_phone_verification', 'calculated_host_listings_count',␣
 ↪'reviews_per_month'
]
# Créez le nouveau DataFrame avec les colonnes à conserver
cleaned_listings = listings_filtered[cols_to_keep]
# Affichez les premières lignes du nouveau DataFrame pour vérification
cleaned_listings.head()
```

[ ]:
```
host_response_time                True
host_response_rate                True
host_acceptance_rate              True
host_is_superhost                 True
host_neighbourhood                True
host_listings_count               True
host_total_listings_count         True
host_has_profile_pic              True
host_identity_verified            True
neighbourhood_group_cleansed     False
zipcode                           True
```

```
latitude                            False
longitude                           False
property_type                        True
room_type                           False
accommodates                        False
bathrooms                            True
bedrooms                             True
beds                                 True
bed_type                            False
square_feet                          True
security_deposit                     True
cleaning_fee                         True
guests_included                     False
extra_people                        False
minimum_nights                      False
maximum_nights                      False
calendar_updated                    False
has_availability                    False
availability_30                     False
availability_60                     False
availability_90                     False
availability_365                    False
number_of_reviews                   False
review_scores_rating                 True
review_scores_accuracy               True
review_scores_cleanliness            True
review_scores_checkin                True
review_scores_communication          True
review_scores_location               True
review_scores_value                  True
requires_license                    False
license                              True
instant_bookable                    False
cancellation_policy                 False
require_guest_profile_picture       False
require_guest_phone_verification    False
calculated_host_listings_count      False
reviews_per_month                    True
dtype: bool
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3755 entries, 0 to 3817
Data columns (total 48 columns):
host_response_time                  3755 non-null object
host_response_rate                  3755 non-null float64
host_acceptance_rate                3755 non-null float64
host_is_superhost                   3755 non-null object
host_neighbourhood                  3755 non-null object
host_listings_count                 3755 non-null float64
host_total_listings_count           3755 non-null float64
host_has_profile_pic                3755 non-null object
host_identity_verified              3755 non-null object
neighbourhood_group_cleansed        3755 non-null object
zipcode                             3748 non-null object
latitude                            3755 non-null float64
longitude                           3755 non-null float64
property_type                       3755 non-null object
room_type                           3755 non-null object
accommodates                        3755 non-null int64
bathrooms                           3755 non-null float64
bedrooms                            3755 non-null float64
beds                                3755 non-null float64
bed_type                            3755 non-null object
square_feet                         3755 non-null float64
security_deposit                    3755 non-null object
cleaning_fee                        3755 non-null float64
guests_included                     3755 non-null int64
extra_people                        3755 non-null object
minimum_nights                      3755 non-null int64
maximum_nights                      3755 non-null int64
calendar_updated                    3755 non-null object
has_availability                    3755 non-null object
availability_30                     3755 non-null int64
availability_60                     3755 non-null int64
availability_90                     3755 non-null int64
availability_365                    3755 non-null int64
number_of_reviews                   3755 non-null int64
review_scores_rating                3755 non-null float64
review_scores_accuracy              3755 non-null float64
review_scores_cleanliness           3755 non-null float64
review_scores_checkin               3755 non-null float64
review_scores_communication         3755 non-null float64
```

```
review_scores_location              3755 non-null float64
review_scores_value                 3755 non-null float64
requires_license                    3755 non-null object
instant_bookable                    3755 non-null object
cancellation_policy                 3755 non-null object
require_guest_profile_picture       3755 non-null object
require_guest_phone_verification    3755 non-null object
calculated_host_listings_count      3755 non-null int64
reviews_per_month                   3755 non-null float64
dtypes: float64(19), int64(10), object(19)
memory usage: 1.6+ MB
```

10. Vérifiez que toutes les variables numériques le sont effectivement bien. (N'oubliez pas de regarder y)

[ ]:

11. Faites votre dernière partie de preprocessing en dummyfiant les variables catégoriques

[ ]:

12. Faites maintenant un `train_test_split`

[ ]:

13. Normalisez `X_train` & `X_test`

[ ]:

```
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/data.py:645:
DataConversionWarning: Data with input dtype uint8, int64, float64 were all
converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
/usr/local/lib/python3.6/dist-packages/sklearn/base.py:464:
DataConversionWarning: Data with input dtype uint8, int64, float64 were all
converted to float64 by StandardScaler.
  return self.fit(X, **fit_params).transform(X)
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/data.py:645:
DataConversionWarning: Data with input dtype uint8, int64, float64 were all
converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
/usr/local/lib/python3.6/dist-packages/sklearn/base.py:464:
DataConversionWarning: Data with input dtype uint8, int64, float64 were all
converted to float64 by StandardScaler.
  return self.fit(X, **fit_params).transform(X)
```

14. Entrainez d'abord un modèle d'Adaboost standard et regardez votre score

[ ]:

```
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[ ]: 0.39617619127209813

15. Entrainez ensuite un modèle XGBoost et regardez votre score

[ ]:

[ ]: 0.6281307244443659

16. Par défaut, Adaboost prend des decision trees comme modèle a booster. Tentez de mettre une regression linéaire

[ ]:

```
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[ ]: -2.1943458198312368e+17

17. La régression linéaire n'était pas la meilleure idée mais peut être qu'on peut faire une grid_search sur le `learning rate` & `n_estimators`pour rattraper le score de XGBoost ?

[ ]:

```
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
```

```
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
```

```
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_search.py:841:
DeprecationWarning: The default of the `iid` parameter will change from True to
False in version 0.22 and will be removed in 0.24. This will change numeric
results when test-set sizes are unequal.
  DeprecationWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[ ]:
```
GridSearchCV(cv='warn', error_score='raise-deprecating',
       estimator=AdaBoostRegressor(base_estimator=LinearRegression(copy_X=True,
fit_intercept=True, n_jobs=None,
         normalize=False),
         learning_rate=1.0, loss='linear', n_estimators=50,
         random_state=None),
       fit_params=None, iid='warn', n_jobs=None,
       param_grid={'n_estimators': [40, 50, 70, 100, 150, 200], 'learning_rate':
[1.0, 0.9, 0.8, 0.7, 0.6, 0.5]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=0)
```

[ ]:

[ ]:
```
{'learning_rate': 1.0, 'n_estimators': 200}
```

[ ]:

```
/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py:761:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[ ]:
```
0.49078434974719043
```

[ ]: