```
# This is a part of the code I worked on in a six-month AI learning program sponsored by the Ministry of Economics i
# The PDF is just the beginning of the project, and more analysis and functions are included in the latter. The whole
# https://github.com/rekkoba/ChicagoBoothApplication
```

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from xgboost import XGBClassifier
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error as MSE
```

Double-click (or enter) to edit

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
os.chdir('/content/drive/MyDrive/Python')
```

```
train = pd.read_csv("train.csv", sep=",")
test = pd.read_csv("test.csv", sep=",")
submission = pd.read_csv("sample_submit.csv", sep=",", index_col=0, header=None)
```

```
# train.isnull().sum()
```

```
# Remove data only from the train set in case of missing or outlier values
train = train.dropna(subset=['bathrooms', 'bedrooms', 'beds'])
```

```
train.describe()
```

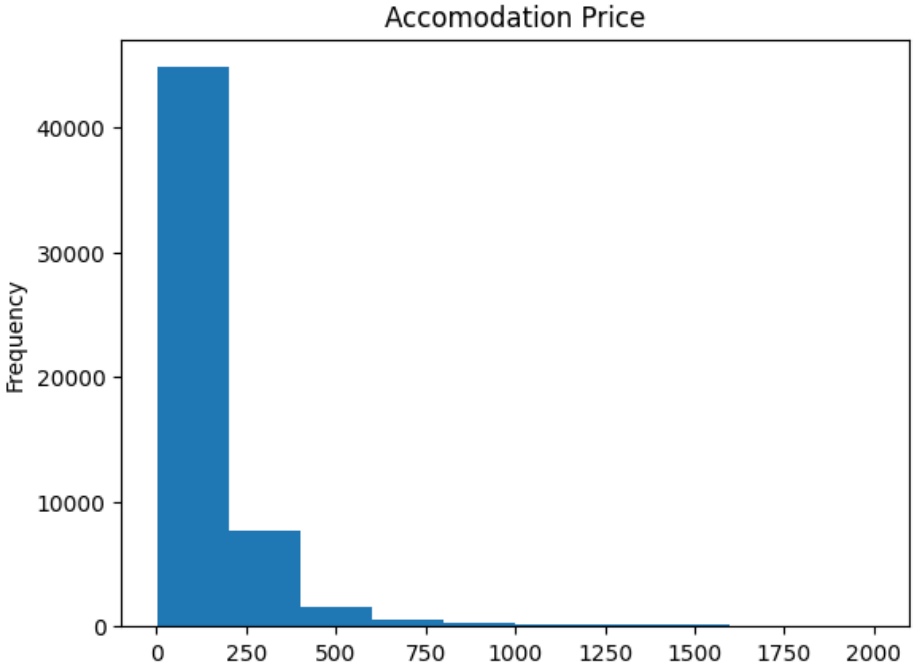
	id	accommodates	bathrooms	bedrooms	beds	latitude	longitude	numbe
count	55323.000000	55323.000000	55323.000000	55323.000000	55323.000000	55323.000000	55323.000000	
mean	27787.181588	3.157023	1.236800	1.265857	1.713248	38.451258	-92.343858	
std	16043.404061	2.154204	0.583499	0.850230	1.259037	3.081077	21.686759	
min	0.000000	1.000000	0.000000	0.000000	0.000000	33.338905	-122.511500	
25%	13892.500000	2.000000	1.000000	1.000000	1.000000	34.128248	-118.341932	
50%	27778.000000	2.000000	1.000000	1.000000	1.000000	40.663287	-76.995867	
75%	41681.500000	4.000000	1.000000	1.000000	2.000000	40.746292	-73.954703	
max	55582.000000	16.000000	8.000000	10.000000	18.000000	42.390437	-70.999166	

```
mean_val = train['review_scores_rating'].mean()
train['review_scores_rating'] = train['review_scores_rating'].fillna(mean_val)
test['review_scores_rating'] = test['review_scores_rating'].fillna(mean_val)
```

```
<ipython-input-9-9b546b6ef013>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-
  train['review_scores_rating'] = train['review_scores_rating'].fillna(mean_val)
```

```
y = train['y']
y.plot.hist(title='Accomodation Price')
plt.show()
```




```
print(y.value_counts().sort_index())
```

```
1.0       1
5.0       1
10.0     23
11.0      1
12.0      3
        ..
1938.0    1
1950.0    6
1980.0    1
1995.0    4
1999.0    3
Name: y, Length: 713, dtype: int64
```

```
before_rows = train.shape[0]
print(before_rows)

train = train[train['y'] >= 10]

after_rows = train.shape[0]
print(after_rows)
```

```
55323
55321
```

```
counts = train['room_type'].value_counts()
counts.plot.bar(title='Frequency of room_type')
plt.xlabel('room_type')
plt.ylabel('count')
plt.show()
```