```
import time
st_time = time.time()


import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import StandardScaler ,RobustScaler
from sklearn.pipeline import Pipeline, make_pipeline
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn import utils


df = pd.read_csv('/content/drive/MyDrive/hotel_100_2022.csv', encoding="ISO-8859-1")
df
```

	Hotel	Location	Country	Region	Company	Score	Rank	Rooms	
0	Rosewood Castiglion del Bosco	Montalcino	Italy	Europe	Massimo and Chiara Ferragamo	99.25	1	53	Cou
1	Grace Hotel	Santorini	Greece	Europe	Auberge Resorts Collection	99.22	2	20	
2	Waldorf Astoria Maldives Ithaafushi	Ithaafushi Island	Maldives	Southeast Asia	Hilton	99.11	3	119	
3	Pickering House Inn	Wolfeboro	United States	North America	Peter and Patty Cooke	98.95	4	10	E
4	One&Only Reethi Rah	North Malé Atoll	Maldives	Southeast Asia	Kerzner International	98.93	5	130	
...	...	...	...	...	...	...	...	...	
96	Casa San Agustín	Cartagena	Colombia	Latin America	Casa San Agustín	95.79	97	31	E
97	The Connaught	London	England	Europe	Maybourne Hotel Group	95.79	97	121	Conte
98	Wentworth Mansion	Charleston	United States	North America	Richard Widman	95.78	99	21	
99	Taj Lands End	Mumbai	India	Southeast Asia	Indian Hotels Company Limited	95.73	100	488	Conte

```
df.describe()
```

	Score	Rank	Rooms	Year	2021	Past_rank
count	101.000000	101.000000	101.000000	101.000000	101.000000	101.000000
mean	97.061980	50.752475	94.554455	1962.910891	0.247525	10.584158
std	0.955555	29.244967	108.379839	72.785177	0.433727	23.687662
min	95.730000	1.000000	6.000000	1592.000000	0.000000	0.000000
25%	96.360000	25.000000	32.000000	1928.000000	0.000000	0.000000
50%	96.800000	51.000000	63.000000	1996.000000	0.000000	0.000000
75%	97.600000	75.000000	116.000000	2011.000000	0.000000	0.000000
max	99.250000	100.000000	792.000000	2021.000000	1.000000	100.000000

```
df.isnull().sum()
```

```
Hotel        0
Location     0
Country      0
Region       0
Company      0
Score        0
Rank         0
Rooms        0
Theme        0
Year         0
2021         0
Past_rank    0
dtype: int64
```

```
df = df.drop(['Hotel','Rank','Past_rank','Company'], axis=1)
df
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Location  101 non-null    object 
 1   Country   101 non-null    object 
 2   Region    101 non-null    object 
 3   Score     101 non-null    float64
 4   Rooms     101 non-null    int64  
 5   Theme     101 non-null    object 
 6   Year      101 non-null    int64  
 7   2021      101 non-null    int64  
dtypes: float64(1), int64(3), object(4)
memory usage: 6.4+ KB
```

99 Mumbai India Southeast Asia 95.75 468 Contemporary 1999

```
df.isnull().any()
```

```
Location    False
Country     False
Region      False
Score       False
Rooms       False
Theme       False
Year        False
2021        False
dtype: bool
```

```
categorical_features=[feature for feature in df.columns if df[feature].dtypes=='O']
print('number of categorical variables:',len(categorical_features))
df[categorical_features].head()
```

```
number of categorical variables: 4
```

	Location	Country	Region	Theme
0	Montalcino	Italy	Europe	Countryside
1	Santorini	Greece	Europe	Coastal
2	Ithaafushi Island	Maldives	Southeast Asia	Island
3	Wolfeboro	United States	North America	Boutique
4	North Malé Atoll	Maldives	Southeast Asia	Island

```
y = df['Score']
y = pd.DataFrame(y)
y
```

	Score
0	99.25
1	99.22
2	99.11
3	98.95
4	98.93
...	...
96	95.79
97	95.79
98	95.78
99	95.73

```
x = df
x
```

	Location	Country	Region	Score	Rooms	Theme	Year	202
0	Montalcino	Italy	Europe	99.25	53	Countryside	2000	
1	Santorini	Greece	Europe	99.22	20	Coastal	2000	
2	Ithaafushi Island	Maldives	Southeast Asia	99.11	119	Island	2019	
3	Wolfeboro	United States	North America	98.95	10	Boutique	1813	
4	North Malé Atoll	Maldives	Southeast Asia	98.93	130	Island	2005	
...	...	...	...	...	...	...	...	
96	Cartagena	Colombia	Latin America	95.79	31	Boutique	1700	
97	London	England	Europe	95.79	121	Contemporary	1897	
98	Charleston	United States	North America	95.78	21	Manor	1886	
99	Mumbai	India	Southeast Asia	95.73	488	Contemporary	1999	
100	Hermanus	South Africa	Africa	95.73	11	Coastal	1952	

101 rows × 8 columns

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
x['Location'] = encoder.fit_transform(x['Location'])
x['Country'] = encoder.fit_transform(x['Country'])
x['Region'] = encoder.fit_transform(x['Region'])
x['Theme'] = encoder.fit_transform(x['Theme'])
```

```
x
```

	Location	Country	Region	Score	Rooms	Theme	Year	2021
0	46	14	3	99.25	53	4	2000	0
1	71	11	3	99.22	20	2	2000	1
2	29	16	8	99.11	119	5	2019	1
3	82	31	6	98.95	10	1	1813	1
4	56	16	8	98.93	130	5	2005	0
...	...	...	...	...	...	...	...	...
96	11	5	4	95.79	31	1	1700	0
97	40	8	3	95.79	121	3	1897	0
98	13	31	6	95.78	21	7	1886	0
99	49	12	8	95.73	488	3	1999	0
100	25	24	0	95.73	11	2	1952	0

101 rows × 8 columns

```
x = x.drop(['Theme','Region','Country','Location'],axis=1)
```

```
x
```

	Score	Rooms	Year	2021
0	99.25	53	2000	0
1	99.22	20	2000	1
2	99.11	119	2019	1
3	98.95	10	1813	1
4	98.93	130	2005	0
...	...	...	...	...
96	95.79	31	1700	0
97	95.79	121	1897	0
98	95.78	21	1886	0
99	95.73	488	1999	0
100	95.73	11	1952	0

101 rows × 4 columns

```
np.isfinite(x).sum()
```

```
Score     101
Rooms     101
```

```
     Year     101
     2021     101
     dtype: int64
```

```
np.isnan(x).sum()
```

```
     Score    0
     Rooms    0
     Year     0
     2021     0
     dtype: int64
```

```
# lab_enc = preprocessing.OneHotEncoder()
# x = lab_enc.fit_transform(x)
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.8,shuffle = True)
```

```
from sklearn.svm import SVR,LinearSVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
```

```
from sklearn.metrics import r2_score
from sklearn.preprocessing import StandardScaler ,RobustScaler
```

```
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

```
     ((80, 4), (21, 4), (80, 1), (21, 1))
```

```
def my_modelfit(my_model,my_x_train,my_y_train):
  my_model.fit(my_x_train, my_y_train)
```

```
def my_predict(my_model,my_x_test):
  y_pred = my_model.predict(my_x_test)
  return y_pred
```

```
def my_r2_score(my_y_test,my_y_pred):
  r2 = r2_score(my_y_test, my_y_pred)
  return r2
```

```
clf1 = SVR(kernel= 'linear',C=5)
clf2 = KNeighborsRegressor(n_neighbors=5)
clf3 = DecisionTreeRegressor()
clf4 = RandomForestRegressor()
clf5 = SVR(kernel= 'rbf',C=5)
clf5 = LinearSVR()
Sscaler = StandardScaler()
Rscaler = RobustScaler()
```

```
my_modelfit(clf1, x_train, y_train)
y_pred_res1 = my_predict(clf1,x_test)
print(my_r2_score(y_test,y_pred_res1))

my_modelfit(clf2, x_train, y_train)
y_pred_res2 = my_predict(clf2,x_test)
print(my_r2_score(y_test,y_pred_res2))

my_modelfit(clf3, x_train, y_train)
y_pred_res3 = my_predict(clf3,x_test)
print(my_r2_score(y_test,y_pred_res3))

my_modelfit(clf4, x_train, y_train)
y_pred_res4 = my_predict(clf4,x_test)
print(my_r2_score(y_test,y_pred_res4))

my_modelfit(clf5, x_train, y_train)
y_pred_res5 = my_predict(clf5,x_test)
print(my_r2_score(y_test,y_pred_res5))
```

```
0.9990892843490367
0.017108530757033336
0.9976605719108942
0.99751381950702
-12.528700907474681
```

```
end_time = time.time()
print(end_time-st_time)
```

```
0.9920618534088135
```

✓ 0 sn. tamamlanma zamanı: 22:00