COIMBATORE INSTITUTE OF TECHNOLOGY

NAME **:** DEVA DHARSHINI D

ROLL NO **:** 1832015

Problem Statement 2:

Explore the given brazil house rent data set using EDA techniques visualize the results and build a suitable model to predict the house rent.

DESCRIPTION

Rent of a house increases or decreases depends on various factors like area, location ,facility, pet, safety and security, etc. A housing market can be understood as any market for properties which are negotiated either directly from their owners to buyers, or through the services of real estate brokers. People and companies are drawn to this market, which presents many profit opportunities that come from housing demands worldwide. These demands are influenced by several factors, such as demography, economy, and politics. For this reason, the analysis of such markets has been challenging data scientists and ML engineers around the world, as they must take into account a wide range of scientific fields, each one addressing different kinds of data, to come up with accurate results to customers and stakeholders.

DATA SET

City: City where the property is located
Area: Size of area
Rooms: Number of rooms
Bathroom: Number of bathrooms
Parking spaces: Number of parking spaces
Floor: Number of floors
Animal: If accept animals (1) or not (0)
Furniture: If it is furnished (1) or not (0)
Hoa: Homeowners association tax
Rent amount: Rent amount in Real (R$)
Property tax: Property tax in Real (R$)
Fire insurance: Fire insurance in Real (R$)
Total: Total amount in Real (R$)

Importing library’s

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
%matplotlib inline
sns.set_style('darkgrid')
```

Loading and Reading the Data

```
houses = pd.read_csv('houses_to_rent.csv')
houses.head()
```

	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa (R$)	rent amount (R$)	property tax (R$)	fire insurance (R$)	total (R$)
0	São Paulo	70	2	1	1	7	acept	furnished	2065	3300	211	42	5618
1	São Paulo	320	4	4	0	20	acept	not furnished	1200	4960	1750	63	7973
2	Porto Alegre	80	1	1	1	6	acept	not furnished	1000	2800	0	41	3841
3	Porto Alegre	51	2	1	0	2	acept	not furnished	270	1112	22	17	1421
4	São Paulo	25	1	1	0	1	not acept	not furnished	0	800	25	11	836

Exploring the data

```
houses.info()
```

```
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   city                 10692 non-null  object
 1   area                 10692 non-null  int64
 2   rooms                10692 non-null  int64
 3   bathroom             10692 non-null  int64
 4   parking spaces       10692 non-null  int64
 5   floor                10692 non-null  object
 6   animal               10692 non-null  object
 7   furniture            10692 non-null  object
 8   hoa (R$)             10692 non-null  int64
 9   rent amount (R$)     10692 non-null  int64
 10  property tax (R$)    10692 non-null  int64
 11  fire insurance (R$)  10692 non-null  int64
 12  total (R$)           10692 non-null  int64
dtypes: int64(9), object(4)
memory usage: 1.1+ MB
```

houses.describe().round(2)

	area	rooms	bathroom	parking spaces	hoa (R$)	rent amount (R$)	property tax (R$)	fire insurance (R$)	total (R$)
count	10692.00	10692.00	10692.00	10692.00	10692.00	10692.00	10692.00	10692.00	10692.00
mean	149.22	2.51	2.24	1.61	1174.02	3896.25	366.70	53.30	5490.49
std	537.02	1.17	1.41	1.59	15592.31	3408.55	3107.83	47.77	16484.73
min	11.00	1.00	1.00	0.00	0.00	450.00	0.00	3.00	499.00
25%	56.00	2.00	1.00	0.00	170.00	1530.00	38.00	21.00	2061.75
50%	90.00	2.00	2.00	1.00	560.00	2661.00	125.00	36.00	3581.50
75%	182.00	3.00	3.00	2.00	1237.50	5000.00	375.00	68.00	6768.00
max	46335.00	13.00	10.00	12.00	1117000.00	45000.00	313700.00	677.00	1120000.00

DATA PREPROCESSING

#change spaces for underscores

```
cols = houses.columns

cols = cols.map(lambda x: x.replace(' ','_') if isinstance(x, (str)) else x)

houses.columns = cols
```

#change the categorical variables

```
houses.animal.replace(['acept','not acept'],[1,0], inplace = True)

houses.furniture.replace(['furnished','not furnished'],[1,0], inplace = True)
```

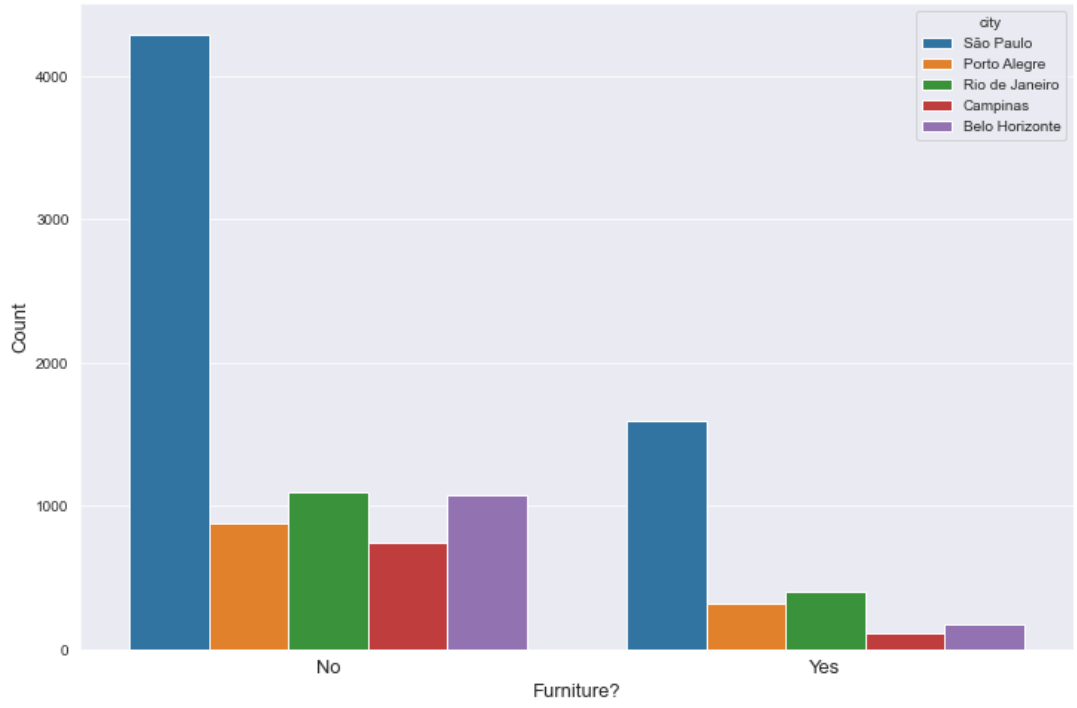
#change "$" for use queries

```
houses.rename(columns={'hoa_(R$)' : 'hoa',
'rent_amount_(R$)' : 'rent_amount',
'property_tax_(R$)' : 'property_tax',
'fire_insurance_(R$)' : 'fire_insurance',
'total_(R$)' : 'total'}, inplace = True)
```

EXPLORATING DATA ANALYSIS

```
ax = sns.countplot(houses['furniture'], hue = houses['city'])
ax.figure.set_size_inches(12, 8)
ax.set_xlabel('Furniture?', fontsize=13)
ax.set_ylabel('Count', fontsize=13)
ax.set_xticklabels(['No','Yes'], fontsize=13)
ax
```

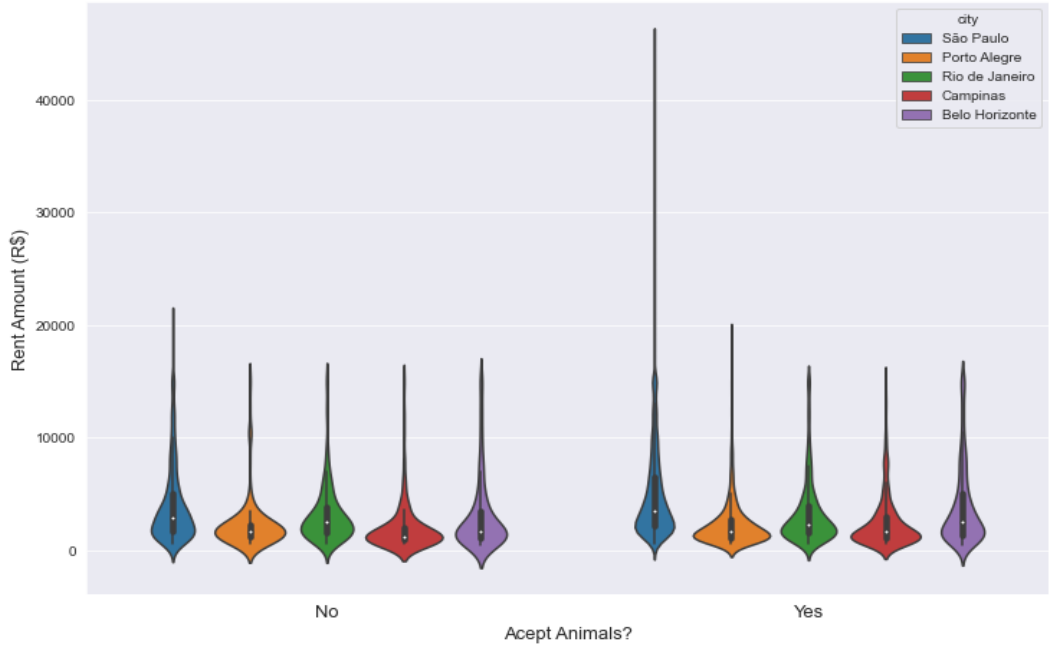
```
<matplotlib.axes._subplots.AxesSubplot at 0x2f02bbea190>
```



Sao Paulo has more information than other cities. In addition, it’s also possible to identify in the dataset more houses for rent that are not furnished.

```
ax = sns.violinplot(x ='animal', y ='rent_amount',
data = houses, hue ='city')
ax.figure.set_size_inches(12, 8)
ax.set_xlabel('Acept Animals?', fontsize=13)
ax.set_ylabel('Rent Amount (R$)', fontsize=13)
ax.set_xticklabels(['No','Yes'], fontsize=13)
ax
```

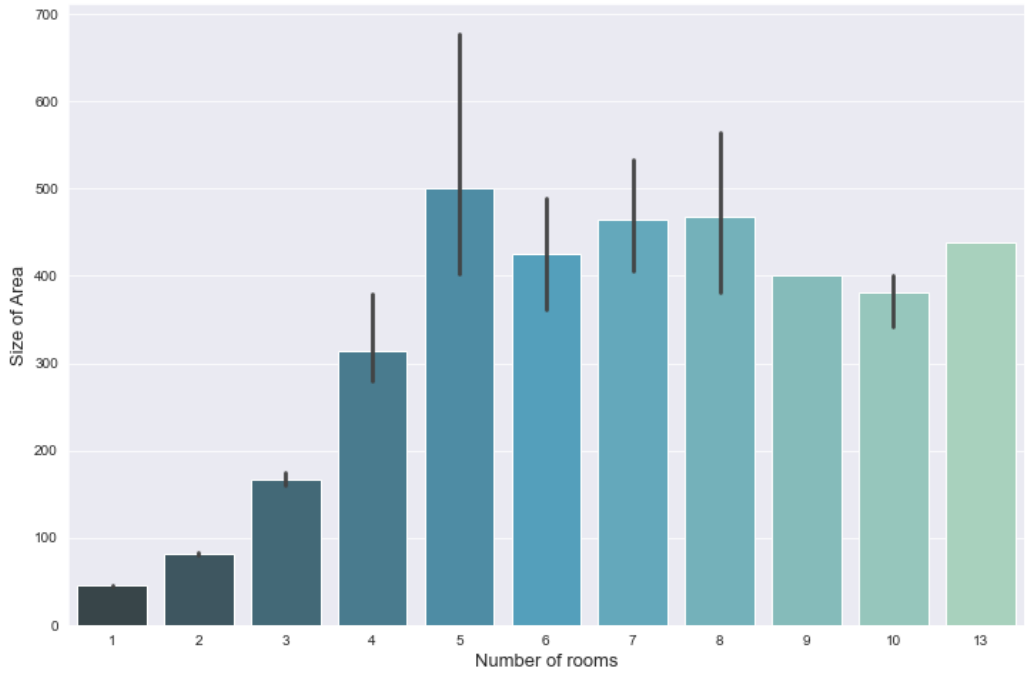
```
<matplotlib.axes._subplots.AxesSubplot at 0x2f02c5895b0>
```



In above graph I observed the biggest rent amount is from Sao Paulo and I can also see that animals (if accept or no) distribution is similar to each city. Besides that, it’s notorious that Porto Alegre and Campinas has a big density.

```
ax = sns.barplot(x='rooms', y='area', data = houses,
palette = 'GnBu_d')
ax.figure.set_size_inches(12, 8)
ax.set_xlabel('Number of rooms', fontsize=13)
ax.set_ylabel('Size of Area', fontsize=13)
```

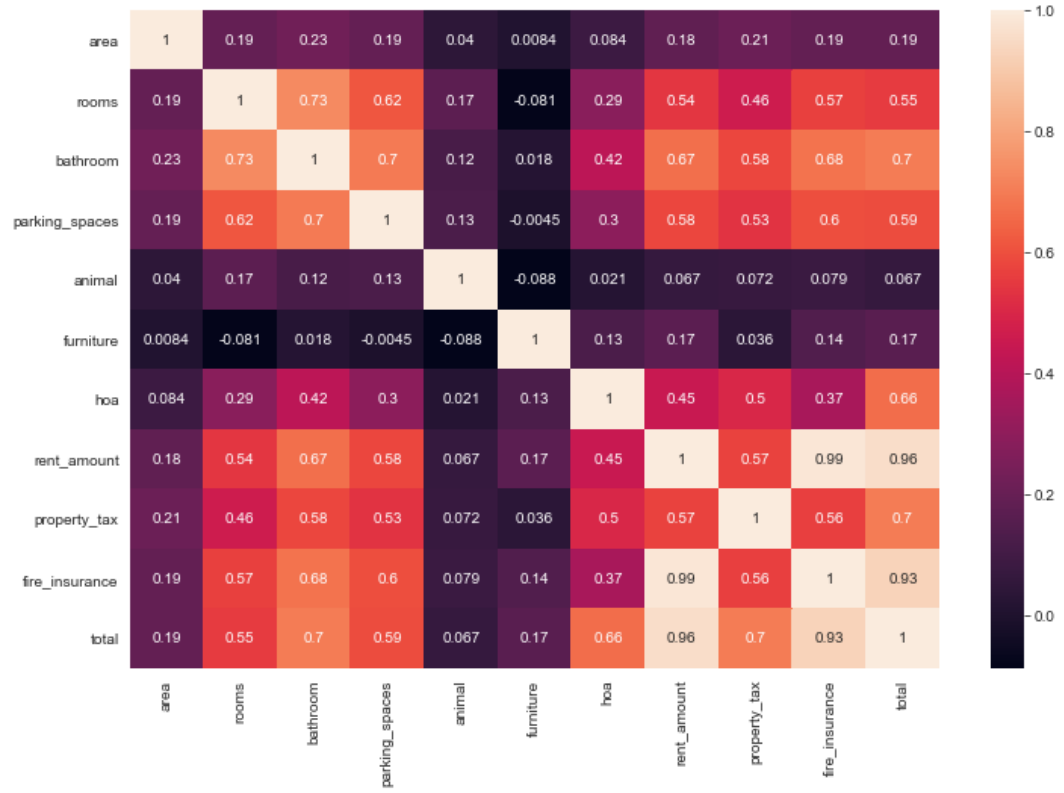
```
Text(0, 0.5, 'Size of Area')
```



This image shows the relationship between area and rooms. Until 5 rooms have a crescent and then stabilize with little variation but remember this data has influence of Outliers.

```
plt.figure(figsize=(12,8))
sns.heatmap(houses.corr(), annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2f02c41dfa0>
```



As you can see, not all features are correlated with the dependent variable rent_amount. Hence I will drop all other features apart from these. However this is not the end of the process. One of the assumptions of linear regression is that the independent variables need to be uncorrelated with each other. If these variables are correlated, then we need drop it. So let's check the correlation of selected features with each other. This can be done either by visually checking it from the above correlation matrix.

FITTING THE MODEL

#Splitting data

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state = 8)

reg = LinearRegression()

reg.fit(x_train, y_train)

predict = reg.predict(x_test)
```

#add a constant and looking the summary

```
x_train_constant = sm.add_constant(x_train)
```

```
model_sm = sm.OLS(y_train, x_train_constant, hascont = True).fit()

print(model_sm.summary())
```

#add a constant and looking the summary

```
x_train_constant = sm.add_constant(x_train)

model_sm = sm.OLS(y_train, x_train_constant, hascont = True).fit()

print(model_sm.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            rent_amount   R-squared:                       0.985
Model:                            OLS   Adj. R-squared:                  0.985
Method:                 Least Squares   F-statistic:                 1.588e+05
Date:                Sat, 06 Mar 2021   Prob (F-statistic):               0.00
Time:                        23:30:40   Log-Likelihood:                -55668.
No. Observations:                7474   AIC:                         1.113e+05
Df Residuals:                    7470   BIC:                         1.114e+05
Df Model:                           3
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
const            -17.4261      7.623     -2.286      0.022     -32.370      -2.482
hoa                0.3262      0.005     64.693      0.000       0.316       0.336
property_tax      -0.1154      0.011    -10.831      0.000      -0.136      -0.095
fire_insurance    68.4671      0.125    548.266      0.000      68.222      68.712
==============================================================================
Omnibus:                     3101.402   Durbin-Watson:                   2.039
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           298077.986
Skew:                           1.046   Prob(JB):                         0.00
Kurtosis:                      33.867   Cond. No.                     2.39e+03
==============================================================================
```

#looking the metrics

```
print('MAE: ', mean_absolute_error(y_test, predict).round(3))

print('RMSE: ', np.sqrt(mean_squared_error(y_test, predict)).round(3))

print('R2:', r2_score(y_test, predict).round(3))

metrics.append(np.sqrt(mean_squared_error(y_test, predict)))
```

```
MAE:  248.495
RMSE:  462.259
R2: 0.982
```

CONCLUSION

In this problem with seaborn and matplotlib to improve the understanding of some variables from the data, pandas to handle and analyzing some columns, and using statsmodels

to have a statistical comprehesion of the dataset and verifying if i can use the Linear Regression model.The value of r-square resembles 98% of accuracy score, which means that the model will perfectly predict the House rent. From my observation bathrooms, HOA property tax, fire insurance are the major factors that affects the house rent in Brazil. If this factors are high the rent of the house will be high, Otherwise the rent will decrease depends on the factors variation.