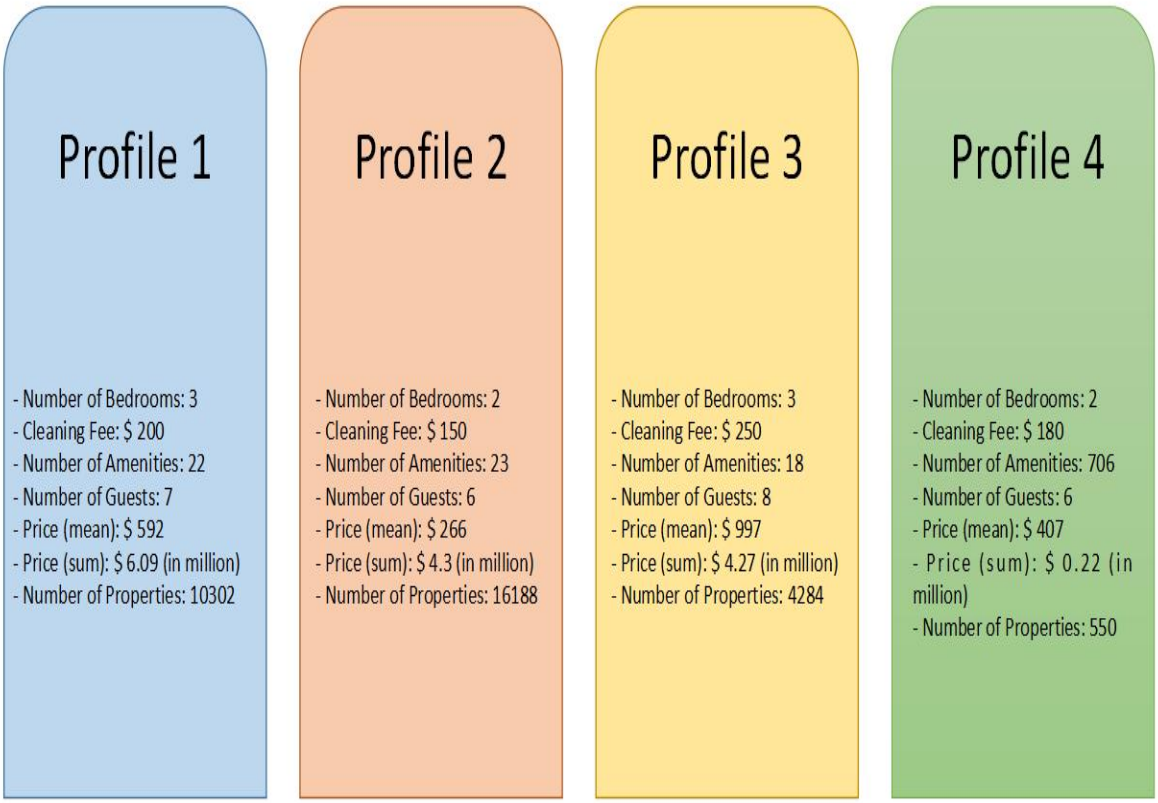
Contextualizing: the data used are from properties in the region of Itapema-SC and most of them refer to the month of December 2022.
They were collected from Airbnb and VivaReal to identify the profiles that generate the highest profitability in rent management and price forecasting to see if it is feasible to invest or not.

Some business questions were asked to draw my conclusions:
1) What is the best property profile to invest in the city?

To choose the best property profile I considered a large dataset from Airbnb and used some tools for Big Data and unsupervised Machine Learning methods to make a good decision. Going straight to the point, the image below helps understanding:

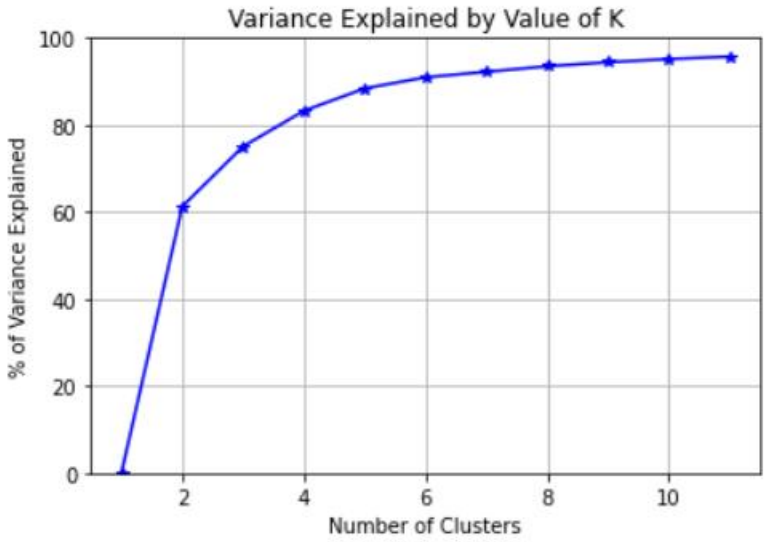


Based on the result that I obtained the best metric returned that the Profile 1. There were other criteria involved in the selection, these are just a summary, however for more details I will leave the scripts available in my repository and some one more print below with a brief explanation.

How did I find out the ideal amount of clusters based on the Euclidean distance with the KMeans algorithm?

```
k_range = range(1,12)
k_means_var = [KMeans(n_clusters = k).fit(pca) for k in k_range]
centroids = [X.cluster_centers_ for X in k_means_var]
k_euclid = [cdist(pca, cent, 'euclidean') for cent in centroids]
dist = [np.min(ke, axis=1) for ke in k_euclid]
sum_of_squares_intra_cluster = [sum(d**2) for d in dist]
total_sum = sum(pdist(pca) **2) / pca.shape[0]
sum_of_squares_inter_cluster = total_sum - sum_of_squares_intra_cluster

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(k_range, sum_of_squares_inter_cluster/total_sum * 100, 'b*-')
ax.set_ylim((0,100))
plt.grid(True)
plt.xlabel('Number of Clusters')
plt.ylabel('% of Variance Explained')
plt.title('Variance Explained by Value of K')
```

```
Text(0.5, 1.0, 'Variance Explained by Value of K')
```



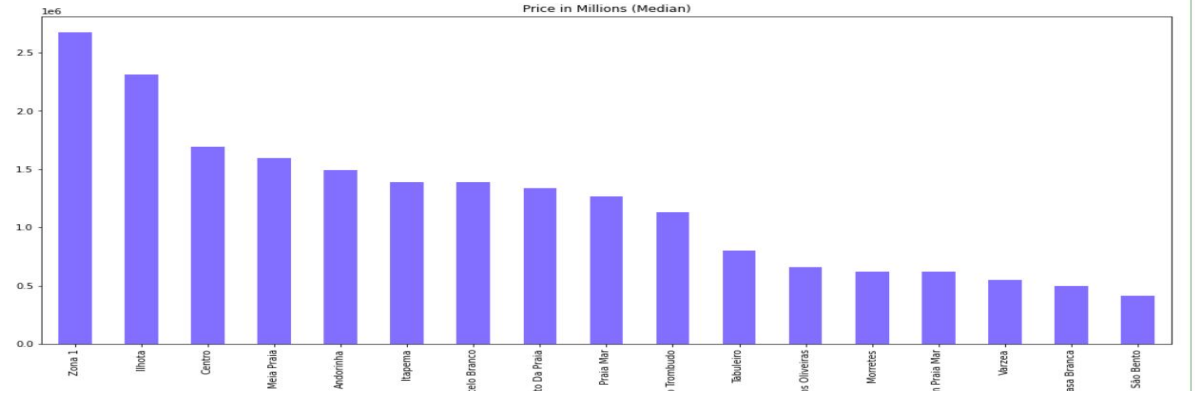
I used the metric silhouette_score to evaluate which model is better by changing the number of groupings. And another important thing was the issue of component division using the PCA (Principal Component Analysis) to reduce the dimensionality of the dataset; in addition to other ETL techniques.

2) Which is the best location in the city in terms of revenue?

The best location in Itapema is Meia Praia and to have considered this I took into account different measures of dispersions involving available-for-sale and short-term for profit properties. There are very different values when looking at total price, mean, median, skewness, among others to reduce bias in the data, but the region that had the best performance was this one.
For example of situations:

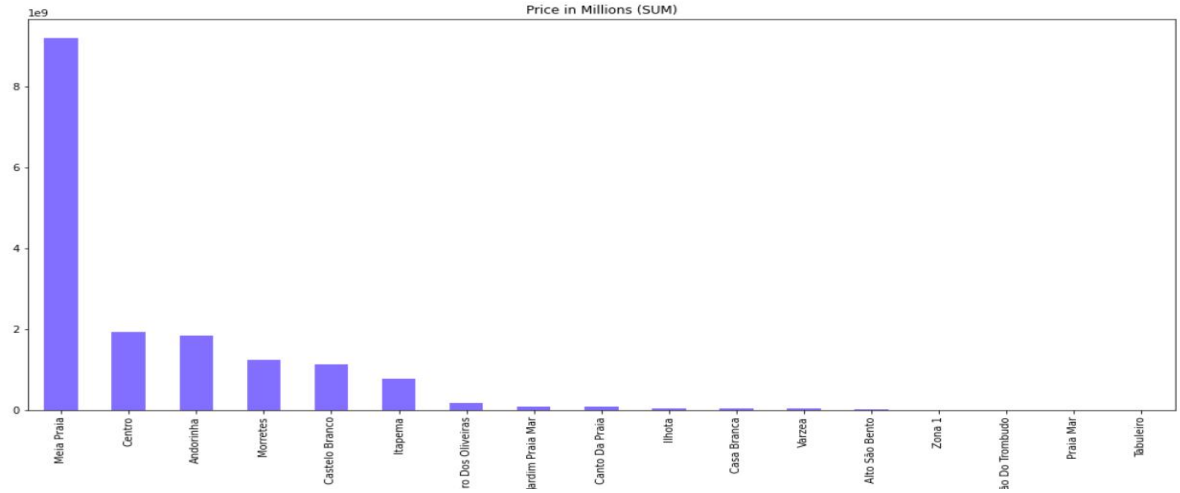
```
grouped_address = viva_real_itapema2.groupby('address_neighborhood')
plt.figure(1)
grouped_address['sale_price'].median().sort_values(ascending=False).plot.bar(figsize = (18,8),
                                                                              color = ['#836FFF'],
                                                                              title = 'Price in Millions (Median)')
```

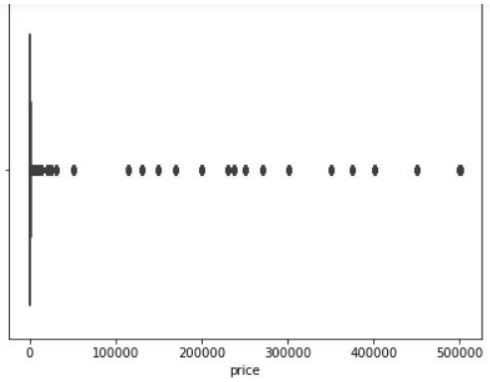
```
<AxesSubplot:title={'center':'Price in Millions (Median)'}, xlabel='address_neighborhood'>
```

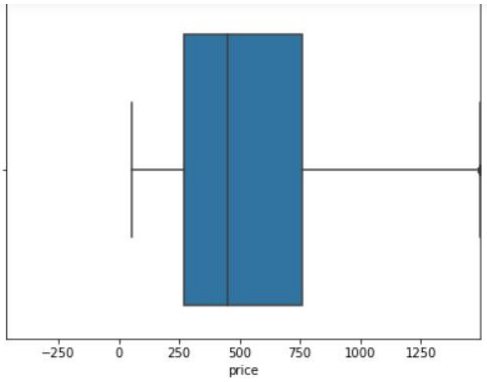


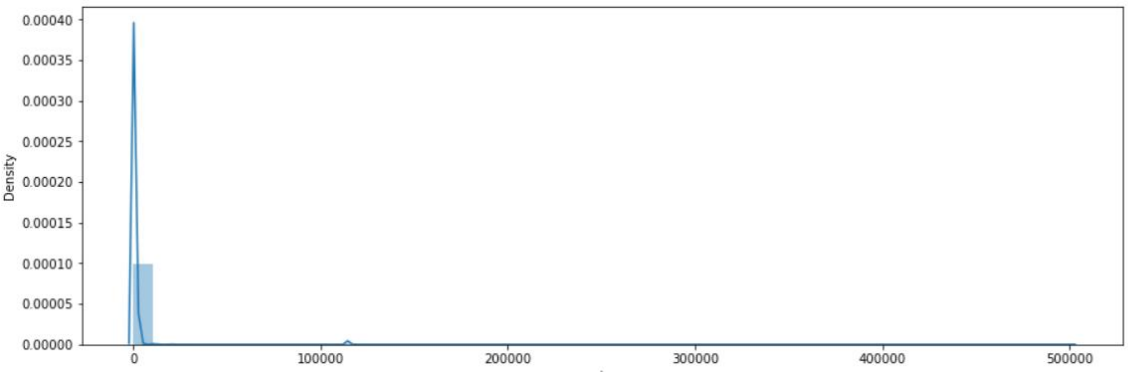

```
plt.figure(1)
grouped_address['sale_price'].sum().sort_values(ascending=False).plot.bar(figsize = (18,8),
                                                                          color = ['#836FFF'],
                                                                          title = 'Price in Millions (SUM)')
```

```
<AxesSubplot:title={'center':'Price in Millions (SUM)'}, xlabel='address_neighborhood'>
```









I even questioned some results, mainly because I was returning zoning (like “Zona 1”) in place of the neighborhood and when looking for more data I realized that I could pulverize these metrics.
In addition to removing outliers, I needed to make a feature selection to identify relevant samples with a good representation of the entire dataset.

3) What are the characteristics and reason for the best revenues in the city?

To answer this question I had to give up an excellent result of the Machine Learning models I trained so that there was a fair and coherent evaluation base on computational performance.
For example a model with too much overfitting:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1000)

for model_name, model in models.items():
    #train
    model.fit(X_train, y_train)

    #test
    predict = model.predict(X_test)
    print(avaliar_modelo(model_name, y_test, predict))
```

```
Model RandomForest:
R²:98.86%
RSME:29.14

Model LinearRegression:
R²:34.65%
RSME:220.18

Model ExtraTrees:
R²:99.30%
RSME:22.77
```

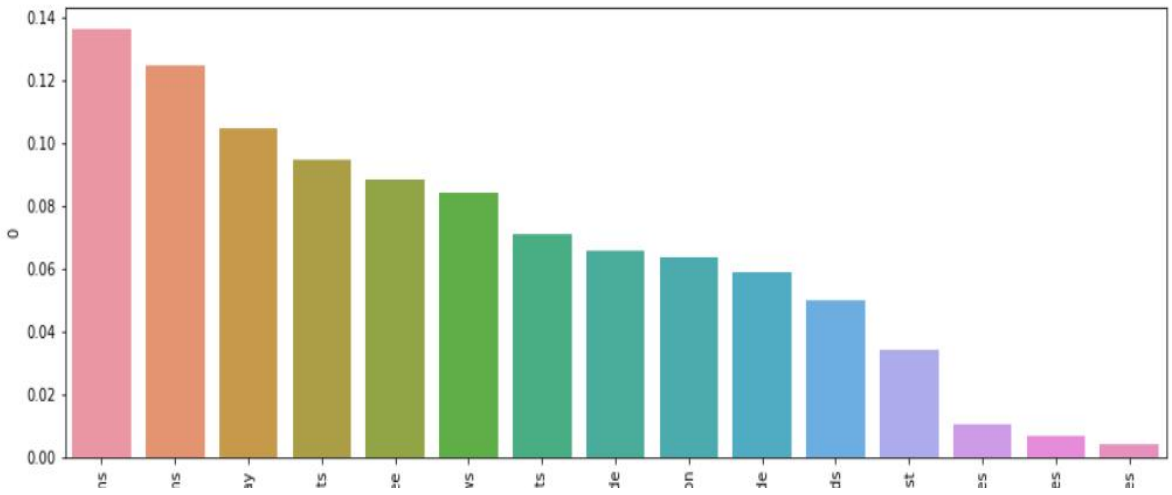
After removing outliers,I needed to increase the randomness of the data, cross-validate and slightly increase the sample to have a model with great computational processing speed and that meets what was proposed. The most realistic model has returned the following metrics (the higher R² is the betther, the lower RSM is the betther). The new result presented as the best model was RandomForest.
It could make more adjustments and tests, but the truth is that it needs to predict an adequate price and the current model is meeting the proposed one. As shown in the picture below (new results):

```
Model RandomForest:
R²:81.61%
RSME:104.87

Model LinearRegression:
R²:37.21%
RSME:193.78

Model ExtraTrees:
R²:81.58%
RSME:104.95
```

And using the hyperparameter feature_importance contained in the algorithm the model identified the main features responsible for making the revenue higher. See below:

```
importances = pd.DataFrame(model_et.feature_importances_, X_train.columns)
importances = importances.sort_values(by=0, ascending=False)
print(importances)
plt.figure(figsize=(15,5))
ax = sns.barplot(x=importances.index, y=importances[0])
ax.tick_params(axis='x', rotation=90)
```

```
                        0
number_of_bedrooms   0.14
number_of_bathrooms  0.13
minimum_stay         0.11
number_of_guests     0.09
cleaning_fee         0.09
number_of_reviews    0.08
n_comments           0.07
latitude             0.07
n_ad_description     0.06
longitude            0.06
number_of_beds       0.05
is_superhost         0.03
n_amenities          0.01
n_house_rules        0.01
n_safety_features    0.00
```



I am aware that the performance still has a lot to improve, but these features were the most important for the current model, based on available data.
It would be necessary to create an action plan, for example, to assess whether taking the variables (top 4, for example: number of bedrooms, number of bathrooms, minimum stay and number of guests) the revenue would increase or not and do an AB Test.

Ariel Carvalho Alves