



REDEFININDO AS LINHAS DE BAIRROS DE São Paulo

Eduardo Dadalto Câmara Gomes - AESP-19'

loft

O PROBLEMA

Case proposto pela Loft para redefinir as linhas de bairro da Cidade de São Paulo como são hoje.

DESAFIOS

O *dataset* fornecido possui dimensionalidade elevada e uma quantidade de dados da ordem de grandeza de 10^6 .

SOLUÇÃO PROPOSTA

Redução de dimensionalidade por *Principal Component Analysis (PCA)* e classificação dos novos bairros por Modelo Mistura de Gauss (GMM).

RESUMO

01 02

03 04

05 06

IMPLEMENTAÇÃO

Linguagem de programação *python* e bibliotecas populares como *pandas* e *sklearn*.

RESULTADOS

Matriz de correlação entre as variáveis propostas, classificação dos dados em *clusters* e análise de performance por *Bayesian information criterion* (BIC).

CONCLUSÃO

Aprendizado e sugestões para melhorias futuras.



O PROBLEMA

Redefinir as linhas de bairros da cidade de São Paulo, utilizando dados do IPTU de 1.4 milhões de propriedades na cidade.

Objetivo de criar novos bairros onde propriedades são similares entre si e diferentes de propriedades de outros bairros.

DESAFIOS

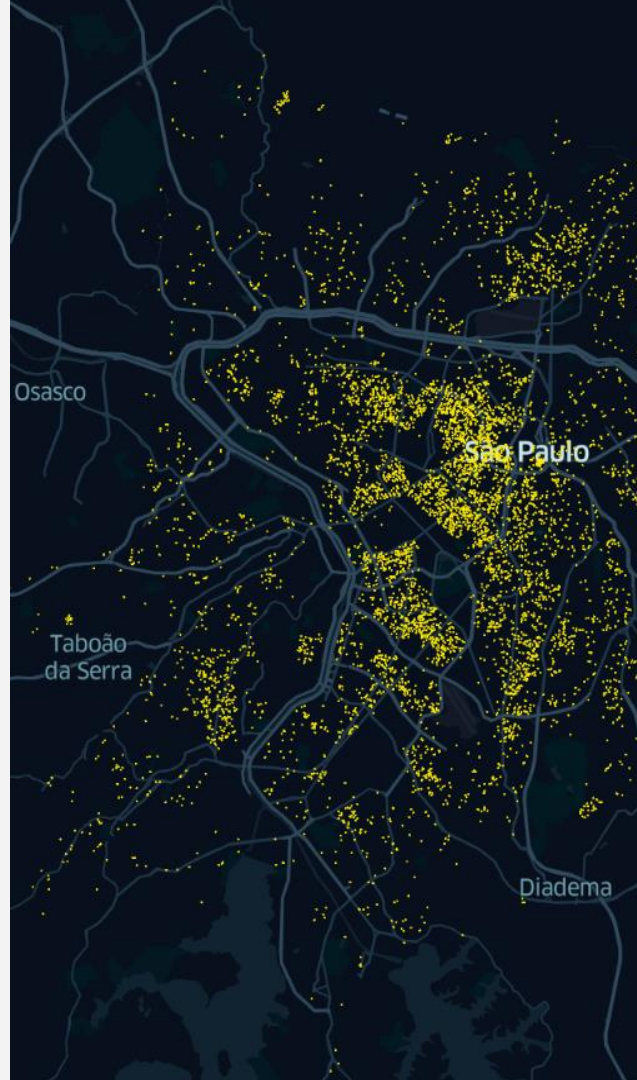
DIMENSIONALIDADE

O *dataset* original possuía 27 variáveis de 1,176,676 imóveis.

Para a classificação, muitas dessas variáveis **não possuem utilidade em termos de dinâmica de mercado**, como o número do contribuinte, o número do condomínio, o CEP, etc.

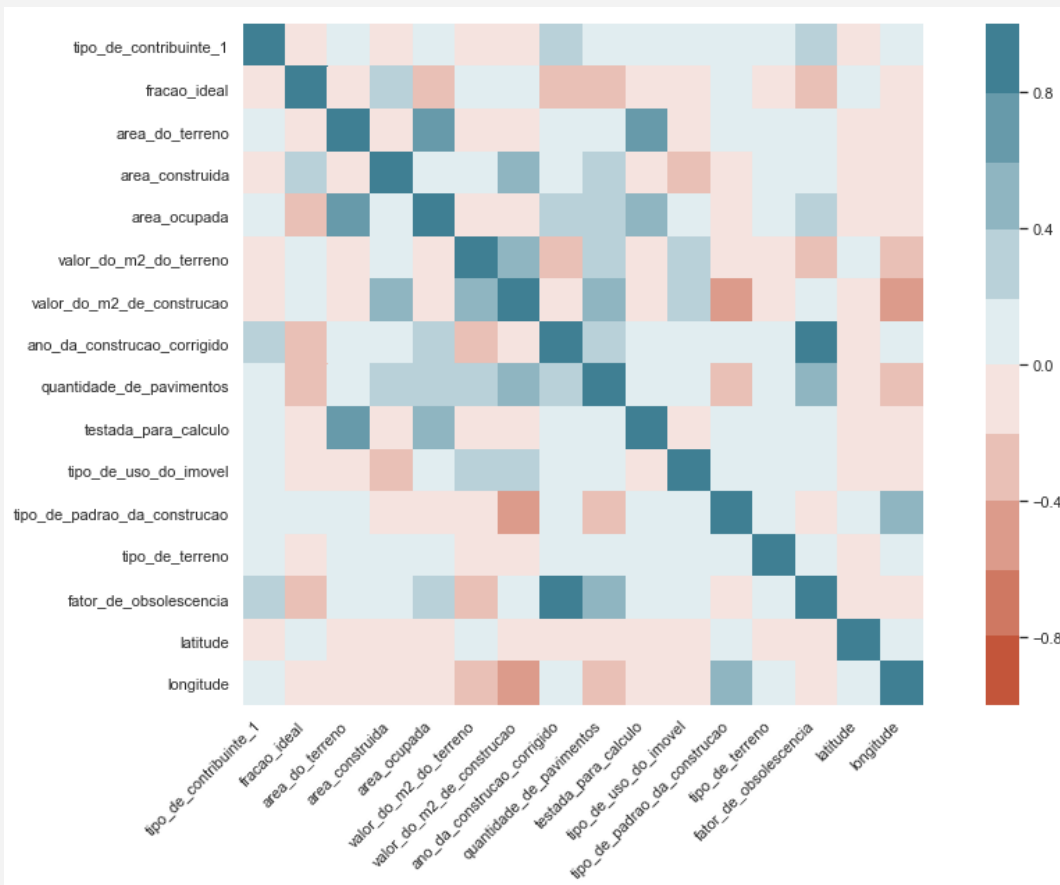
Sedo assim, restaram **16 variáveis** pertinentes para o cálculo e **1,172,180 imóveis** visto que as informações de alguns estavam incompletas.

Mesmo assim, a quantidade de dados é elevada para a aplicação de um algoritmo de classificação não supervisionado em uma **máquina pessoal em apenas 2 dias**.



SOLUÇÃO PROPOSTA

Redução de Dimensionalidade por Análise de Correlação



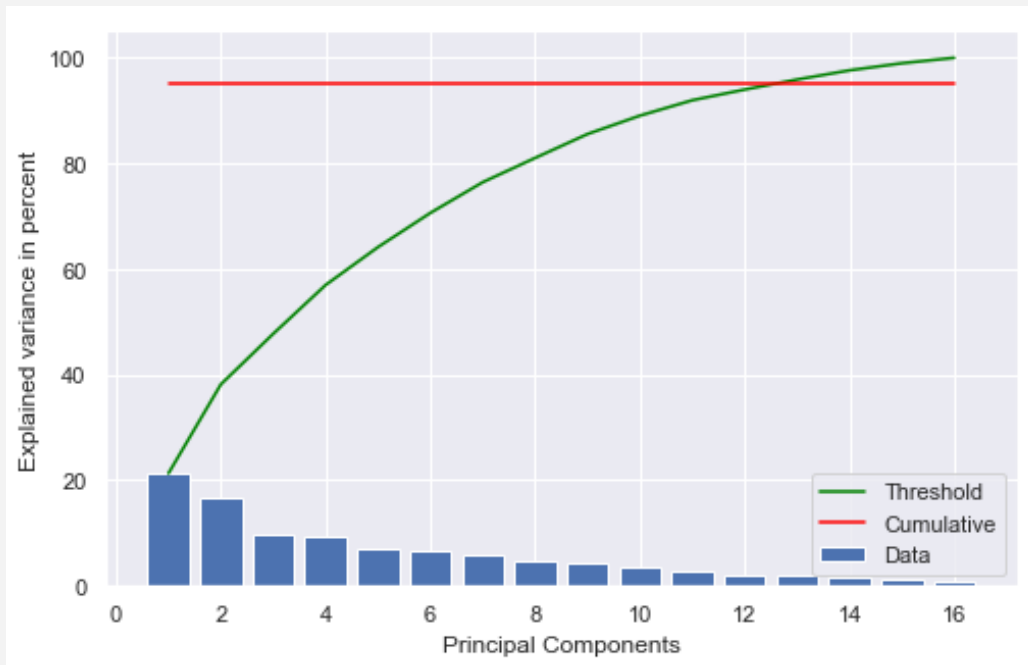
As variáveis foram padronizadas e suas correlações cruzadas calculadas.

Pelo gráfico nota-se uma **forte correlação** entre o fator de obsolescência e o ano da construção corrigido.

Optou-se por retirar o ano da construção corrigido, restando-se 15 variáveis.

SOLUÇÃO PROPOSTA

Redução de Dimensionalidade por Principal Component Analysis (PCA)



Pelo gráfico, foi escolhido manter 12 componentes principais.

Foram calculados os autovalores e os autovetores da matriz de correlação e as variáveis foram projetadas no espaço de componentes principais.

Um *Threshold* de 95% foi escolhido para determinar o número de variáveis necessárias para representar bem o conjunto.

SOLUÇÃO PROPOSTA

Modelo de Mistura de Gauss (GMM)

Finalmente, para **classificar** o conjunto de variáveis restantes em subconjuntos de máxima semelhança, foi escolhido utilizar um o **algoritmo de “clusterização” não supervisionado**.

Optou-se pelo algoritmo GMM por ter implementação prática e por ser mais versátil que o algoritmo padrão para esse tipo de problema, o K-Means.

A métrica utilizada foi a **euclidiana** no espaço de dimensão 12 composto pela projeção dos dados padronizados em 12 componentes principais.

$$D(x_i, x_{i'}) = ||x_i - x_{i'}|| = \sqrt{\sum_{j=1}^p (x_{ij} - x_{i'j})^2}$$

Para determinar o **número de “clusters” ideal**, foi utilizado o *Bayesian information criterion* (BIC) e escolhido o número de componentes no ponto de inflexão da medida.

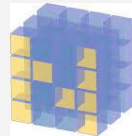
IMPLEMENTAÇÃO

O código completo está em anexo e disponível no repositório:



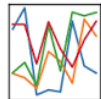
<https://github.com/edadaltocg/case-loft>

```
362 # Akaike information criterion (AIC) or
363 from sklearn.mixture import GaussianMixture
364 from sklearn import metrics
365
366 n = 12
367 data_PCA = pd.read_csv('case_zonas-valor
368 data_reduced = pd.read_csv('case_zonas-v
369 data_reduced = data_reduced.drop(columns
370 data_reduced_sampled = data_reduced.samp
371 data_reduced_sampled.to_csv('data-reduce
372 header = list(data_reduced.columns.value
373 data_reduced_values = data_reduced_sampl
374 scaler = preprocessing.StandardScaler()
375 data_standarized = scaler.fit_transform(
376 data_reduced_standarized = pd.DataFrame(
377
378 sklearn_pca = sklearnPCA(n_components=n)
379 data_PCA = sklearn_pca.fit_transform(dat
380 pca_header = ['PC_%s' % i for i in range(
381 data_PCA_sampled = pd.DataFrame(data_PCA
382
383 X = data_PCA_sampled
384 n_components = np.arange(140, 160, 1)
385 MC = 10
386 y_bic_means = np.empty([MC, n_components.
387 y_aic_means = np.empty([MC, n_components.
388
389 for i in np.arange(0, MC):
390     models = [GaussianMixture(n, covaria
391     y_bic_means[i, :] = np.array([m.bic(X
392     y_aic_means[i, :] = np.array([m.aic(X
393
394 plt.figure(figsize=(15, 8))
395 plt.plot(n_components, y_bic_means,
```



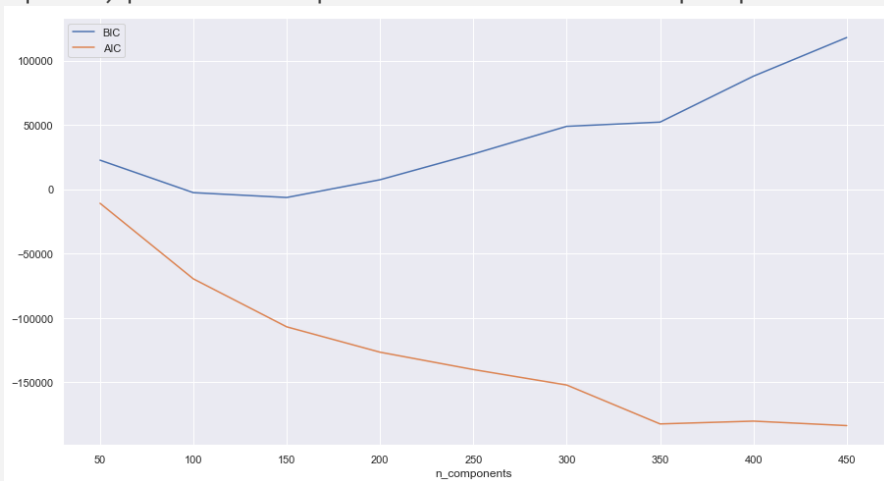
NumPy

pandas
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



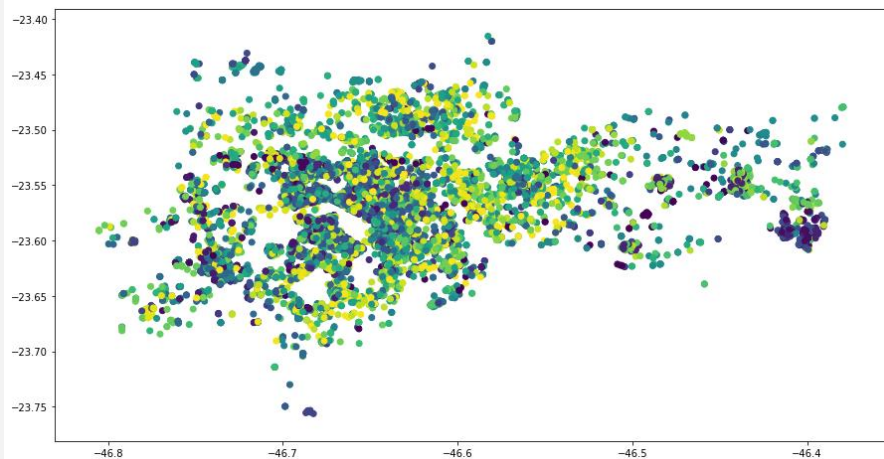
RESULTADOS

Resultado de uma simulação de Monte Carlo (número de simulações igual a 10 para cada ponto) para o GMM aplicado a 10% dos dados pós processados. Métrica BIC em azul.



O BIC mostra um número de clusters ideal por volta de 150 para o GMM.

Sendo assim, **optou-se por repartir São Paulo em 150 bairros**, ao invés de cerca de 800 bairros como atualmente.



Uma porção dos dados foi plotada e os novos bairros podem ser observados.


Para uma distribuição mais homogênea, sugere-se diminuir o número de variáveis.

CONCLUSÃO

Finalmente, gostaria de agradecer a Loft por lançar esse desafio super interessante à comunidade iteana e parabenizá-los por atacar de maneira inovadora uma questão tão sensíveis da sociedade como o mercado imobiliário.

dudu.dadalto@gmail.com

+55 27 99890-0453

 +33 07 69 66 05 97

Futuramente, pode-se testar outros algoritmos de clusterização, propor modelos com mais variáveis, otimizar os parâmetros do modelo atual ou realizar simulações com menos variáveis para encontrar um divisões de bairros menos heterogêneas.



This is where you give credit to the ones who are part of this project.

Did you like the resources on this template? Get them for free at our other websites.

Presentation template by [Slidesgo](#)

Icons by [Flaticon](#)

Infographics by [Freepik](#)

Images created by [Freepik](#)

Author introduction slide photo created by Freepik

Text & Image slide photo created by Freepik.com

CREDITS