Rodolfo José de Araujo Bugarin

ZAP Data Science Challenge

São Paulo
2019

SUMÁRIO

1. INTRODUÇÃO

O mercado imobiliário vem passando por rápidas mudanças decorrentes de avanços tecnológicos imensos. Vendedores, corretores e compradores passaram a utilizar plataformas digitais para fazer seus negócios. Isto criou uma grande base de dados nas maiores empresas que possuem plataformas digitais que atendem este setor. Com o desenvolvimento de algoritmos de inteligência artificial que usam modelos estatísticos (tradicionais, *machine learning* e *deep learning*) o setor imobiliário passa a oferecer serviços que facilitam a busca de imóveis através da recomendação de preços, análise de característicos dos imóveis, compreensão de indicadores sociais que afetam a tomada de decisão de quem compra e de quem vende.

O presente trabalho é construído para responder a proposta do "Data Science Challenge" do Grupo ZAP: estimar preço de venda de apartamentos com base em uma amostra de anúncios do site da empresa. Nas próximas seções apresentamos:

- ★ Descrição da Base de Dados Original
- ★ Análise Exploratória de Dados
- ★ Modelagem Estatística
- ★ Modelo(s) Proposto(s)
- ★ Solução Desenvolvida
- ★ Arquitetura Proposta
- ★ Conclusões e Recomendações

2. DESCRIÇÃO DA BASE DE DADOS ORIGINAL

A base de dados original está no formato JSON. Esta base foi convertida para um Dataframe que contém 133.964 regisros com 37 colunas que detalham cada anúncio, conforme definição abaixo[1].

- address_city: cidade do imóvel.
- address_country: país do imóvel.
- address_district: distrito do imóvel.
- address_geolocation_location_lat: geolocalização da latitude do imóvel.
- address_geolocation_location_lon: geolocalização da longitude do imóvel.
- address_geolocation_precision: precisão da geolocalização do imóvel.
- address_locationid: id da localização do imóvel.
- address_neighborhood: bairro do imóvel.
- address_state: estado federativo do imóvel.
- address_street: nome da rua do imóvel.
- address_streetnumber: número do imóvel na rua.
- address_unitnumber: complemento do imóvel.
- address_zipcode: CEP do imóvel.
- address_zone: zona do imóvel.
- bathrooms: número de banheiros.
- bedrooms: número de quartos.
- createdat: timestamp da criação do anúncio.
- description: descrição do anúncio.
- id: identificador único do anúncio.
- images: lista de links de imagens do anúncio.
- listingstatus: status do anúncio (ativo ou inativo).
- owner: indicador se o anunciante é proprietário do imóvel.
- parkingspaces: número de vagas de garagem.
- pricinginfos_businesstype: tipo do anúncio (venda, aluguel ou ambos).
- pricinginfos_monthlycondofee: valor da taxa do condomínio.
- pricinginfos_period: período do preço de aluguel (diário, mensal ou anual).
- pricinginfos_price: preço de venda.
- pricinginfos_rentaltotalprice: preço do aluguel.
- pricinginfos_yearlyiptu: preço do IPTU.
- publicationType: tipo de publicação do anúncio (normal ou premium).
- publisherId: identificador do anunciante
- suites: número de suítes do imóvel.
- title: título do anúncio

[1] A descrição baseou-se na documentação fornecida pelo ZAP. As diferenças para a descrição original foram feitas pelo autor deste trabalho.

- totalareas: área total do imóvel em metros quadrados (m²).
- unittypes: tipo do imóvel (apartamento, casa, comercial, etc).
- updatedat: timestamp do último update.
- usableareas: tamanho do imóvel em metros quadrados (m²).

3. ANÁLISE EXPLORATÓRIA

3.1 Base de Dados Completa

A nossa análise exploratória inicial utiliza os 133.964 registos do dataset source-4-ds-train que inclui anúncios de Venda e de Aluguel de imóveis conforme Tabela 1 de frequência abaixo:

Tabela 1 - Distribuição dos Tipos de Negócio

pricinginfos_businesstype	*Quantidade*	*Frequência*
RENTAL	28.632	21,4%
SALE	105.332	78,6%
Total	133.964	100%

Fonte: Grupo ZAP. Tabulação própria.

Este dataset possui as informações de País, Estado e Cidade, conforme podemos observar respectivamente nas Tabela 2, Tabela 3 e Tabela 4.

Tabela 2- Distribuição por País

address_country	*Quantidade*	*Frequência*
"vazio"	120.112	89,66%
BR	13.852	10,34%
Total	133.964	100,00%

Fonte: Grupo ZAP. Tabulação própria.

Tabela 3- Distribuição por Estado

address_country	*Quantidade*	*Frequência*
"vazio"	3	0,00%
SP	246	0,18%
Santa Catarina	1	0,00%
São Paulo	133.714	99,81%
Total	133.964	100,00%

Fonte: Grupo ZAP. Tabulação própria.

Tabela 4- Distribuição por Cidade

address_city	*Quantidade*	*Frequência*
São Paulo	133.964	100%
Total	133.964	100%

Fonte: Grupo ZAP. Tabulação própria.

Constatamos que temos anúncios localizados somente na cidade de São Paulo, conforme pode ser visto na Tabela 4

Entretanto as variáveis de País e Estado apresentam inconsistências. Assumimos como premissa forte que a informação da cidade está correta e que 100% dos anúncios são localizados no Brasil, Estado de São Paulo.

A variável explicada (target) deste trabalho é o preço de venda do imóvel. A Tabela 5 apresenta a análise estatística descritiva básica da variável "pricinginfos_price" que, por definição, contém o preço de venda.

Tabela 5- Análise Descritiva de "Preço de Venda"

Pricinginfos_price	
Total de Imóveis	133.964
Média	663.748,35
Desvio Padrão	1.317.731,68
Mínimo	70.00
Percentil 25%	175,000.00
Percentil 50%	371,000.00
Percentil 75%	700,000.00
Máximo	84,000,000.00

Fonte: Grupo ZAP. Tabulação própria.

Observando os dados acima, vemos uma variação muito grande para os valores presentes no campo "pricinginfos_price", indo do mínimo de 70 reais ao máximo de 84 milhões de reais. Para investigarmos esta variação no preço de venda dos imóveis, comparamos o conteúdo dos campos (Tabela 6) de alguns anúncios e a análise estatística dos campos de "Preço de Venda" e "Preço de Aluguel" somente para os anúncios de Aluguel (Tabela 7).

Tabela 6- “Preço de Venda” x “Preço de Aluguel” de alguns anúncios

pricinginfos_businesstype = 'RENTAL'		
ID do Anúncio	*Preço de Venda (pricinginfos_price)*	*Preço de Aluguel (pricinginfos_ rentaltotalprice)*
e7e0b554ac	24,929.00	29,829.00
4d96835e38	1,889.00	2,450.00
a50840b3a5	3,849.00	3,849.00
f1087bf3f8	7,699.00	7,699.00
7c15d854de	7,000.00	7,000.00

Fonte: Grupo ZAP. Tabulação própria.

Tabela 7- Análise Descritiva de “Preço de Venda” x “Preço de Aluguel”

pricinginfos_businesstype = 'RENTAL'		
	Preço de Venda (pricinginfos_price)	Preço de Aluguel (pricinginfos_ rentaltotalprice)
Total de Imóveis	28,632.00	28,619.00
Média	9,602.49	11,058.42
Desvio Padrão	74,950.25	77,137.33
Mínimo	70.00	70.00
Percentil 25%	1,750.00	2,310.00
Percentil 50%	3,423.00	4,130.00
Percentil 75%	8,049.00	9,170.00
Máximo	11,900,000.00	11,900,000.00

Fonte: Grupo ZAP. Tabulação própria.

Ao verificarmos a análise estatística e compararmos alguns dos registros de “Aluguel” nas duas tabelas acima, constamos uma forte evidência que o conteúdo do campo “pricinginfos_price” é similar ao conteúdo do campo “pricinginfos_rentaltotalprice” quando o anúncio é de aluguel. Isto nos leva inferir que os dois campos estão preenchidos com o valor do aluguel quando a anúncio é uma oferta de aluguel.

Refazemos então a análise estatística descritiva básica da variável “pricinginfos_price” considerando somente os anúncios de “Venda” na Tabela 8.

Tabela 8 - Análise Descritiva de "Preço de Venda"

pricinginfos_businesstype = SALE	
Pricinginfos_price	
Total de Imóveis	105.332
Média	841.562,35
Desvio Padrão	1.434.908,50
Mínimo	7.000,00
Percentil 25%	296.800,00
Percentil 50%	482.299,00
Percentil 75%	886.900,00
Máximo	84.000.000,00

Fonte: Grupo ZAP. Tabulação própria.

Com este subconjunto de dados de anúncios do tipo "Venda" o "Preço de Venda" mínimo saiu de 70 reais para 7 mil reais. Para este mesmo subconjunto o campo de "Preço de Aluguel" está vazio em 105.237 registos e possui o valor zero em 95 registros.

Adicionalmente, o *dataset* completo tem ofertas de imóvel que vão desde a venda ou aluguel de terrenos até prédios inteiros, entre outros tipos de imóvel conforme pode ser visto na Tabela 9:

Tabela 9- Distribuição dos "Tipos de Imóvel" Ofertados

unittypes	*Quantidade*	*Frequência*
APARTMENT	72.241	53,93%
BUSINESS	663	0,49%
CLINIC	33	0,02%
COMMERCIAL_ALLOTMENT_LAND	807	0,60%
COMMERCIAL_BUILDING	85	0,06%
COMMERCIAL_PROPERTY	5.871	4,38%
CONDOMINIUM	4.015	3,00%
COUNTRY_HOUSE	9	0,01%
FARM	10	0,01%
FLAT	7.661	5,72%
HOME	9.030	6,74%
KITNET	522	0,39%
OFFICE	7.714	5,76%
PENTHOUSE	2.772	2,07%
RESIDENTIAL_ALLOTMENT_LAND	1.430	1,07%
RESIDENTIAL_BUILDING	142	0,11%
SHED_DEPOSIT_WAREHOUSE	2.197	1,64%
STORE	694	0,52%
TWO_STORY_HOUSE	18.068	13,49%
Total	133.964	100,00%

Fonte: Grupo ZAP. Tabulação própria.

3.2 Base de Dados de Anúncios de Venda de Apartamentos

A partir deste ponto faremos a análise do conjunto de registos de anúncios de venda de apartamentos. Desta forma, filtramos o *dataset* original para anúncios de "Venda" de "Apartamentos" o que resulta em 64.146 registros[2]. E reavaliamos a análise estatística descritiva básica da nossa variável explicada "Preço de Venda" para este novo conjunto de dados conforme pode ser visto na Tabela 10:

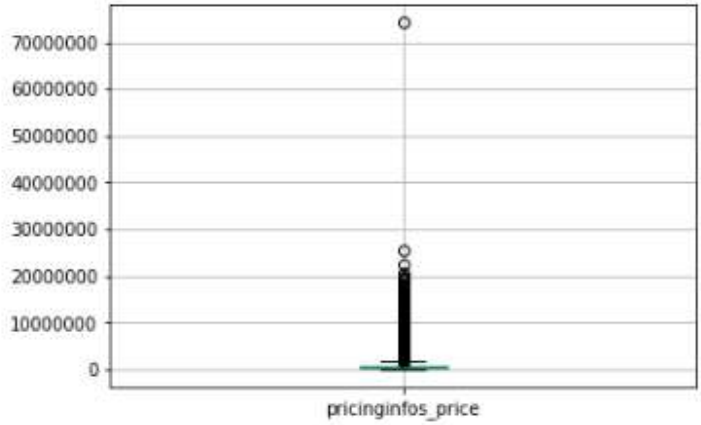
Tabela 10- Análise Descritiva de "Preço de Venda" de Apartamentos

Pricinginfos_price	
Total de Imóveis	64.146
Média	763.357,45
Desvio Padrão	1.047.635,18
Mínimo	10.500,00
Percentil 25%	273.000,00
Percentil 50%	465.499,00
Percentil 75%	858.585,00
Máximo	74.200.000,00

Fonte: Grupo ZAP. Tabulação própria.

Observando os dados acima, vemos uma variação para os valores presentes no campo "pricinginfos_price", indo do mínimo de 10 mil e quinhentos reais ao máximo de 74 milhões de reais. No Box Plot de "Preço de Venda (Gráfico 1) vemos muitos outliers com um destaque para este imóvel de 74 milhões de reais. Conferimos este anúncio na base de dados e é um apartamento à venda na Alameda Campinas; existem 97 outros anúncios de venda nesta rua e este imóvel de preço máximo não tem características que o distinguem dos demais 96.

Gráfico 1- Box Plot – Preço de Venda

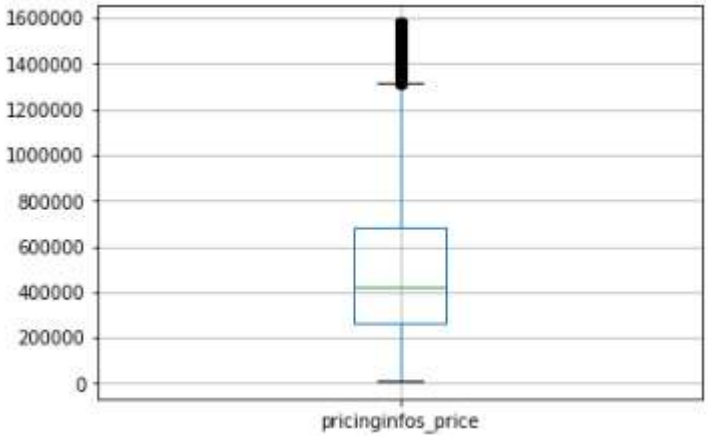


Fonte: Grupo ZAP. Tabulação própria.

[2] O filtro no dataset para selecionar somente os apartamentos à venda foi aplicado com as seguintes condições: "pricinginfos_businesstype" = "SALE" e "unittypes" = "APARTMENT".

Refizemos o Box Plot do Preço de Venda excluindo o conjunto de imóveis listados entre os 10% mais caros (Gráfico 2) e ainda observamos muitos valores aberrantes (outliers) com grande afastamento acima dos demais preços de venda dos imóveis.

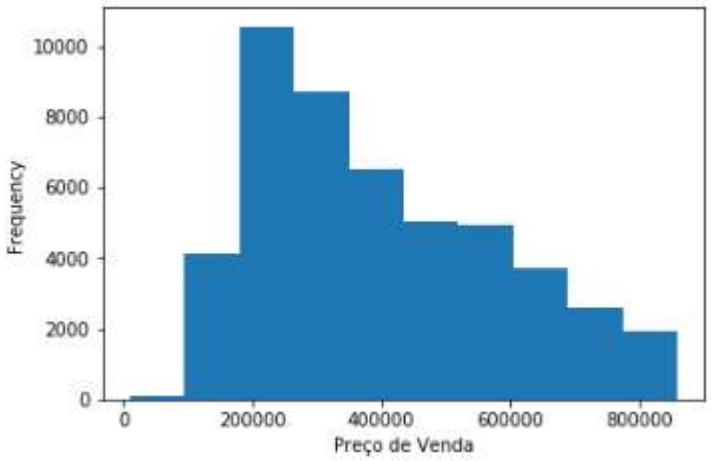
Gráfico 2- Box Plot – Preço de Venda (<= 90%)



Fonte: Grupo ZAP. Tabulação própria.

Analisando o gráfico histograma (Gráfico 3) somente para o subconjunto dos imóveis cujos preços de venda estão até o 3º quartil[3]. Observamos uma grande concentração de imóveis com preço de venda 200 mil e 400 mil reais.

Gráfico 3 - Box Plot – Preço de Venda (<= 75%)



Fonte: Grupo ZAP. Tabulação própria.

[3] O corte foi feito para os preços de venda até o terceiro quartil (75%) pois o gráfico histograma completo ficava com uma única barra dado os outliers mais aberrantes da amostra, impossibilitando a análise do gráfico. Com a limitação no 3º quartil foi possível realizar tal análise.

Retomando a análise das variáveis de endereço, constata-se que a informação de Distrito é quase que completamente vazia, com apenas 3 registros indicando o distrito Jaraguá. Por sua vez a informação de Bairro possui inconsistências, com diferentes grafias (com e sem acento, com e sem nome de Zona, etc. – exemplos: Vila Regente Feijo x Vila Regente Feijó, Vila Diva x Vila Diva (Zona Leste) x Vila Diva (Zona Norte)). No total temos anúncios de apartamentos à venda em 1.131 bairros diferentes. Para efeitos de ilustração, vemos os 10 bairros com mais apartamentos à venda na Tabela 11.

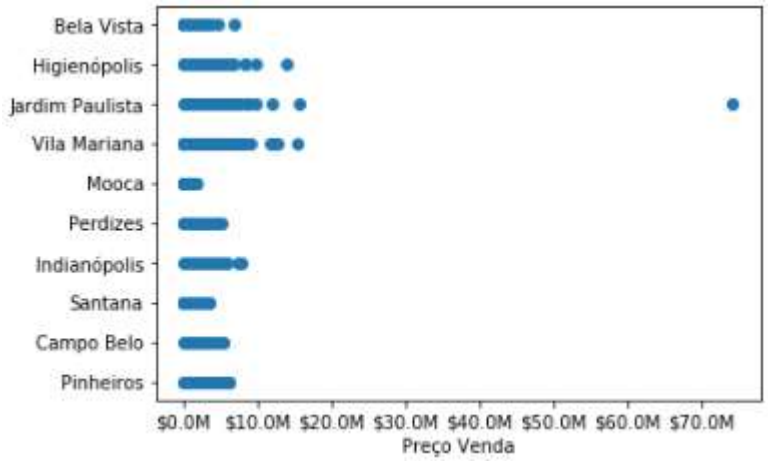
Tabela 11- Top 10 Bairros com Mais Imóveis à Venda

address_neighborhood	*Quantidade*
Higienópolis	1.761
Jardim Paulista	1.697
Santana	1.665
Vila Mariana	1.604
Pinheiros	1.586
Mooca	1.394
Bela Vista	1.196
Indianópolis	1.151
Perdizes	1.104
Campo Belo	1.102

Fonte: Grupo ZAP. Tabulação própria.

Nestes "Top 10" bairros com mais imóveis à venda temos a seguinte distribuição de preços de Venda (Gráfico 4):

Gráfico 4 – Top 10 Bairros x Preço de Venda



Fonte: Grupo ZAP. Tabulação própria.

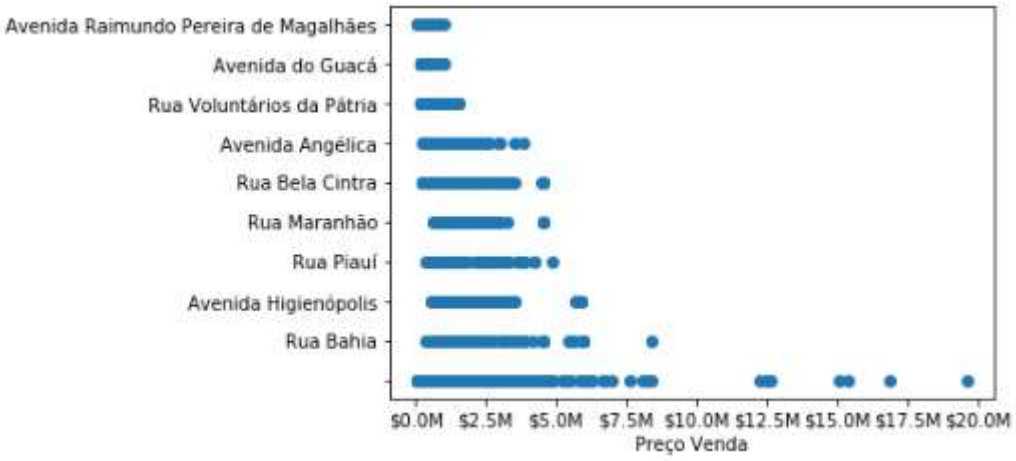
Da mesma forma, a informação da Rua do Imóvel também possui inconsistências de grafia e, adicionalmente, 2.899 ruas não preenchidas (vide Tabela 12). No total temos anúncios de apartamentos à venda em 4.592 ruas diferentes. Na Tabela 12 temos os 10 bairros com mais apartamentos à venda. E no Gráfico 5 temos a distribuição de preços de venda para estas 10 ruas.

Tabela 12- Top 10 Ruas com Mais Imóveis à Venda

address_neighborhood	*Quantidade*
"vazio"	2.899
Avenida Raimundo Pereira de Magalhães	304
Rua Voluntários da Pátria	296
Rua Maranhão	286
Rua Bela Cintra	265
Rua Piauí	247
Avenida Higienópolis	231
Rua Bahia	228
Avenida do Guacá	225
Avenida Angélica	219

Fonte: Grupo ZAP. Tabulação própria.

Gráfico 5 – Top 10 Ruas x Preço de Venda



Fonte: Grupo ZAP. Tabulação própria.

Finalizando a análise das variáveis de endereço relevantes nota-se que a maior parte (88%) dos imóveis não possui a informação da Zona preenchida (regiões - Tabela 13). Entretanto temos essa informação disponível no campo "id da localização do imóvel" para a maioria dos registros (Tabela 14). Após tratamento[4], constatamos que a Zona Sul é a região que tem mais apartamentos à venda em São Paulo (39%), o que é coerente com o fato que a Zona Sul é a maior região de São Paulo e concentra a maioria dos imóveis (Tabela 15).

Tabela 13 - Distribuição das Zonas (regiões) de São Paulo

address_zone	*Quantidade*	*Frequência*
"vazio"	56.488	88,1%
Centro	1.564	2,4%
Zona Leste	1.185	1,8%
Zona Norte	1.336	2,1%
Zona Oeste	1.039	1,6%
Zona Sul	2.534	4,0%
Total	64.146	100,0%

Fonte: Grupo ZAP. Tabulação própria.

Tabela 14- Id da localização do imóvel (exemplos)

address_locationid
BR>Sao Paulo>NULL>Sao Paulo>Zona Sul>Vila Olimpia
BR>Sao Paulo>NULL>Sao Paulo>Zona Sul>Paraiso
BR>Sao Paulo>NULL>Sao Paulo>Zona Oeste>Pinheiros
BR>Sao Paulo>NULL>Sao Paulo>Centro>Aclimacao
BR>Sao Paulo>NULL>Sao Paulo>Zona Sul>Morumbi
BR>Sao Paulo>NULL>Sao Paulo>Zona Sul>Vila Olimpia

Fonte: Grupo ZAP. Tabulação própria.

Tabela 15- Distribuição das Zonas (regiões) de São Paulo (após tratamento)

Zona	*Quantidade*	*Frequência*
"vazio"	38	0,1%
Zona Centro	6.621	10,3%
Zona Leste	11.458	17,9%
Zona Norte	11.401	17,8%
Zona Oeste	9.480	14,8%
Zona Sul	25.148	39,2%
Total	64.146	100,0%

Fonte: Grupo ZAP. Tabulação própria.

[4] Criamos a variável "Zona" pela combinação de parte do conteúdo da variável "*address_locationid*" e da variável "*address_zone*".

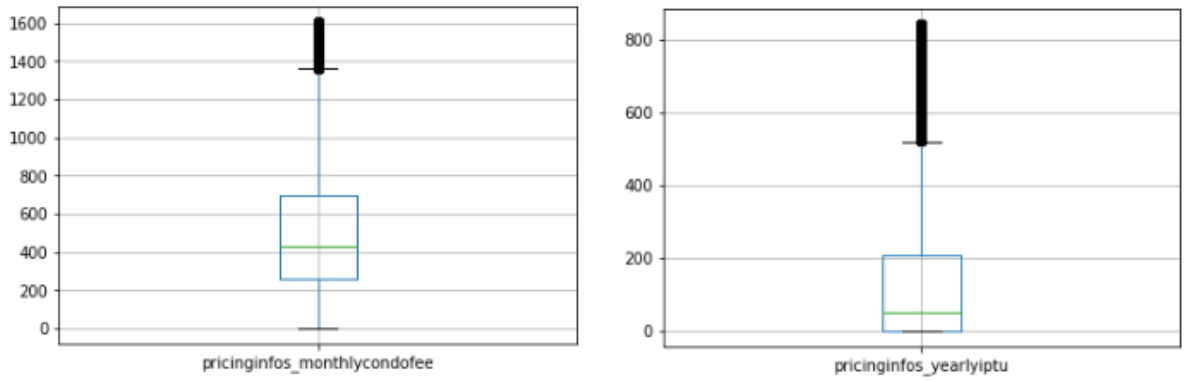
A análise estatística descritiva básica da “Taxa de Condomínio” e do “IPTU” pode ser vista na Tabela 16. Da mesma forma que o “Preço de Venda”, estes custos do imóvel também apresentam uma grande variação por causa de valores muito aberrantes (outliers). No Gráfico 6 e Gráfico 7 temos respectivamente o Box Plot e o Histograma sem os 10% maiores outliers destas variáveis de custo.

Tabela 16- Análise Descritiva dos Custos Condomínio e IPTU

	pricinginfos_ monthlycondofee	*pricinginfos_ yearlyiptu*
Total de Imóveis	60.516	54.559
Média	2.020,19	933,61
Desvio Padrão	107.501,41	56.983,66
Mínimo	0,00	0.00
Percentil 25%	280,00	0,00
Percentil 50%	482,00	71,00
Percentil 75%	886,25	301,00
Máximo	24.430.000,00	10.367.000,00

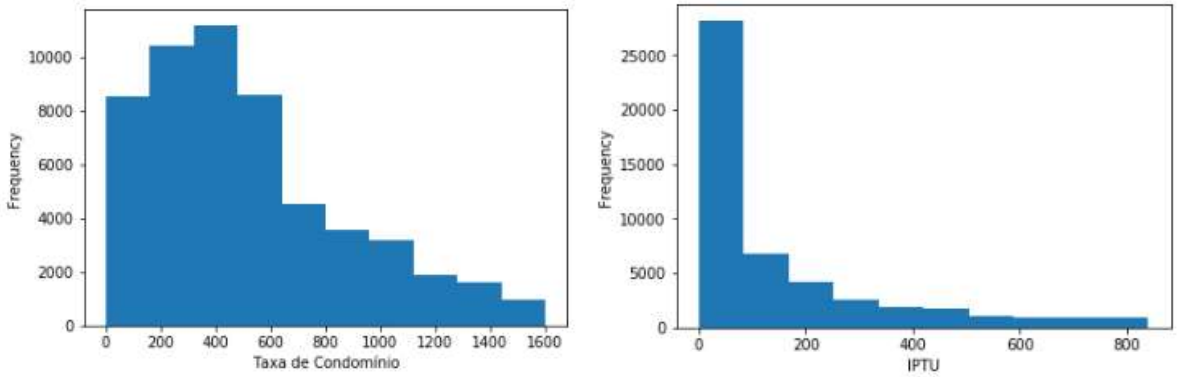
Fonte: Grupo ZAP. Tabulação própria.

Gráfico 6 – Box Plot – “Condomínio”e IPTU



Fonte: Grupo ZAP. Tabulação própria.

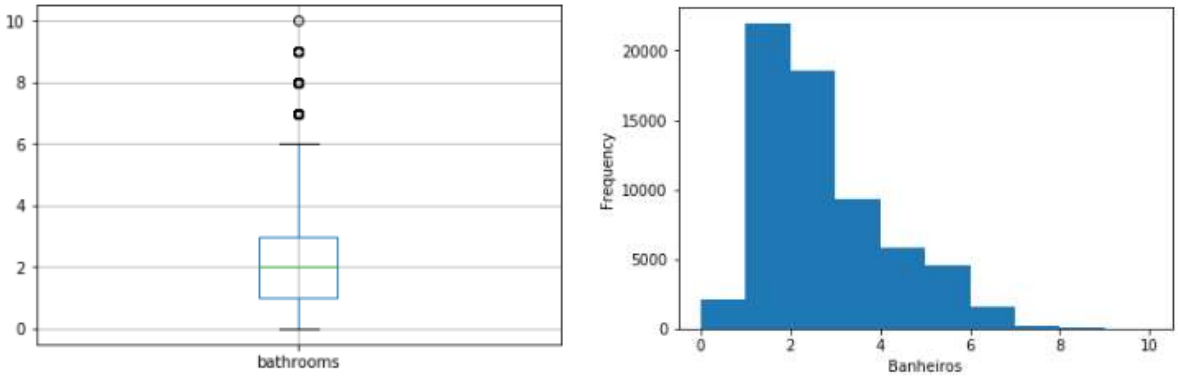
Gráfico 7 – Histograma – “Condomínio”e IPTU



Fonte: Grupo ZAP. Tabulação própria.

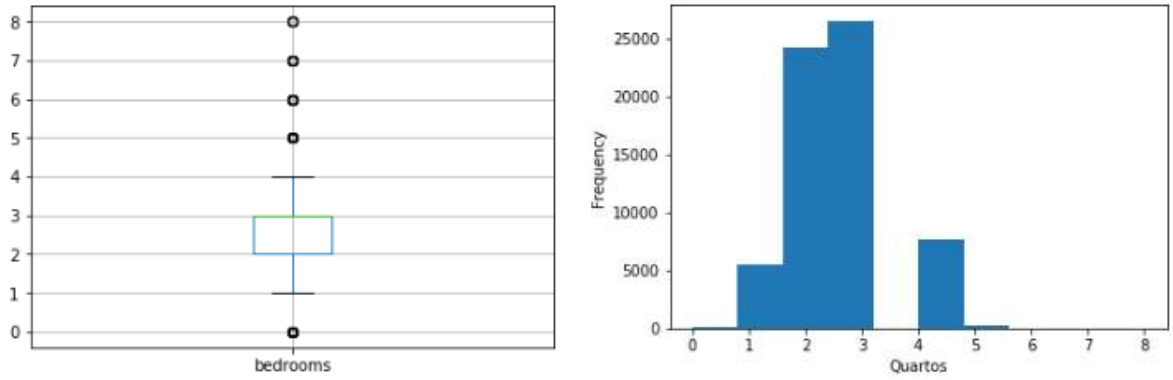
Quanto às características dos imóveis, também constatamos muitos outliers nas quantidades de banheiros, quartos, suítes, vagas de estacionamento, tamanho e área total. Nos gráficos abaixo temos o Box Plot com até a quantidade 10 de cada variável, bem como o Histograma sem os 10% maiores outliers destas variáveis.

Gráfico 8 – Box Plot e Histograma – "Quantidade de Banheiros"



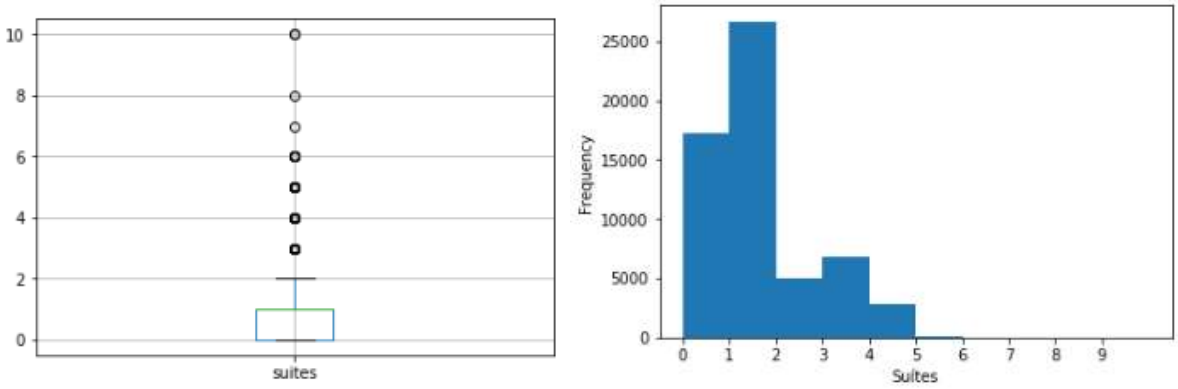
Fonte: Grupo ZAP. Tabulação própria.

Gráfico 9 – Box Plot e Histograma – "Quantidade de Quartos"



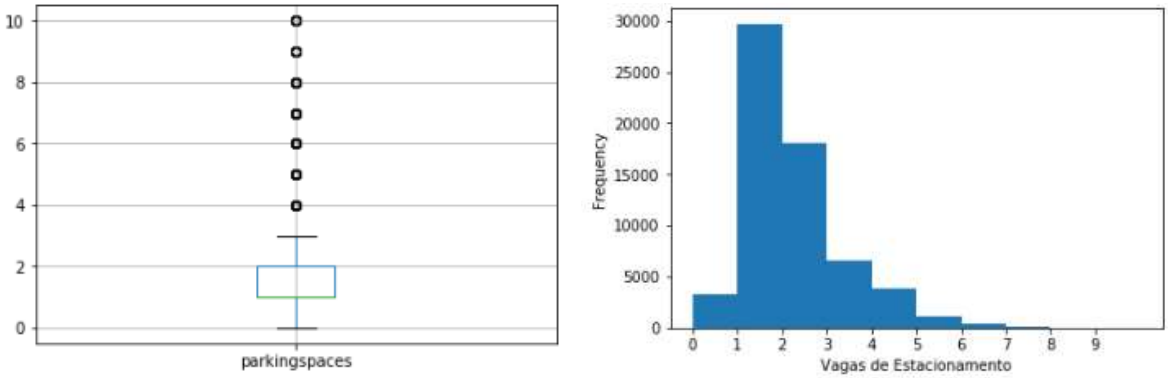
Fonte: Grupo ZAP. Tabulação própria.

Gráfico 10 – Box Plot e Histograma – "Quantidade de Suítes"



Fonte: Grupo ZAP. Tabulação própria.

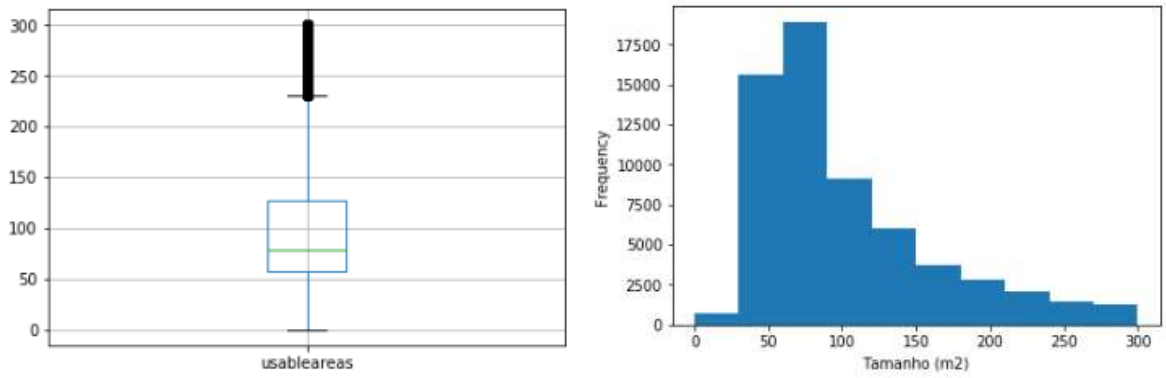
Gráfico 11– Box Plot e Histograma – "Quantidade de Vagas de Estacionamento"



Fonte: Grupo ZAP. Tabulação própria.

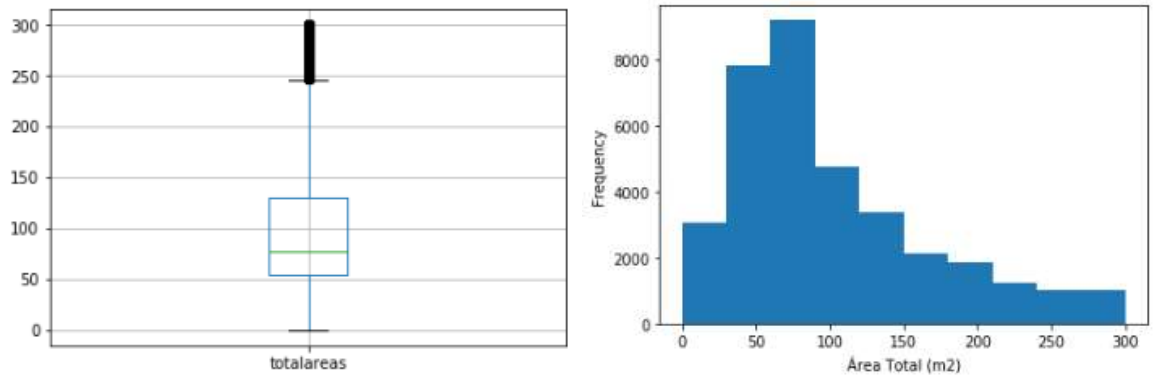
E para finalizarem nossa análise das variáveis quantitativas, também constatamos muitos outliers nas metragens dos imóveis: tamanho do apartamento e área total. Nos Gráfico 12 e Gráfico 13 abaixo temos o Box Plot Histograma com até 300 metros quadrados:

Gráfico 12– Box Plot e Histograma – "Tamanho do Apartamento"



Fonte: Grupo ZAP. Tabulação própria.

Gráfico 13– Box Plot e Histograma – "Área Total"



Fonte: Grupo ZAP. Tabulação própria.

E para finalizarem nossa análise das variáveis categóricas temos as seguintes tabelas de frequência para o “Tipo De Publicação Do Anúncio” (Tabela 17) e o “Indicador se o Anunciante é Proprietário Do Imóvel” (Tabela 18). Constatamos que a maioria dos anúncios foram publicados por terceiros (possivelmente corretores de imóveis ou imobiliárias) e não são “premium”.

Tabela 17- Distribuição do “Tipo De Publicação Do Anúncio”

publicationtype	*Quantidade*	*Frequência*
PREMIUM	1.019	2%
STANDARD	63.127	98%
Total	64.146	100,0%

Fonte: Grupo ZAP. Tabulação própria.

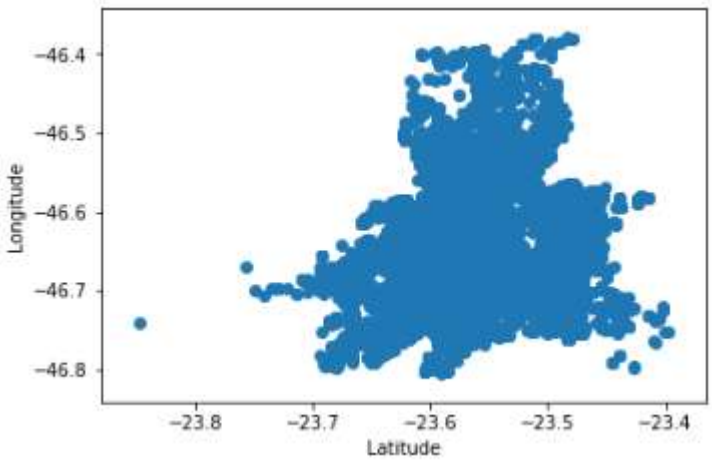
Tabela 18- Distribuição “Se Anunciante é Proprietário”

owner	*Quantidade*	*Frequência*
FALSE	64.129	99,97%
TRUE	17	0,03%
Total	64.146	100,0%

Fonte: Grupo ZAP. Tabulação própria.

Por fim temos a geolocalização do imóvel, definido por suas coordenas Latitude e Latitude. Podemos perceber uma grande concentração e imóveis aproximadamente entre as coordenadas -23.6,-46.7 e -23.5,-46.5.

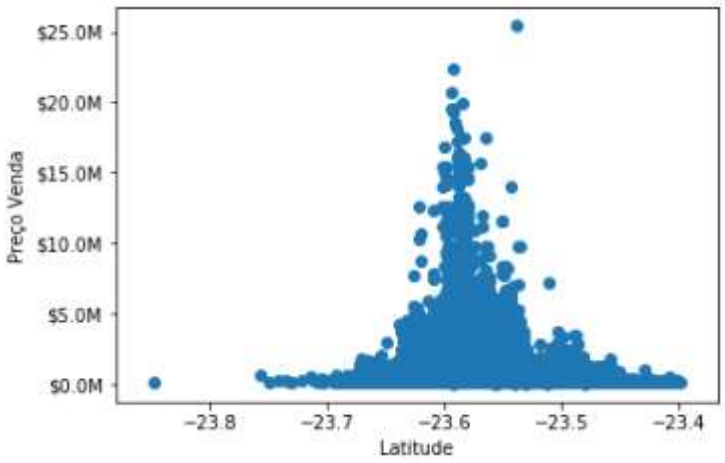
Gráfico 14–Apartamentos à Venda por Coordenadas



Fonte: Grupo ZAP. Tabulação própria.

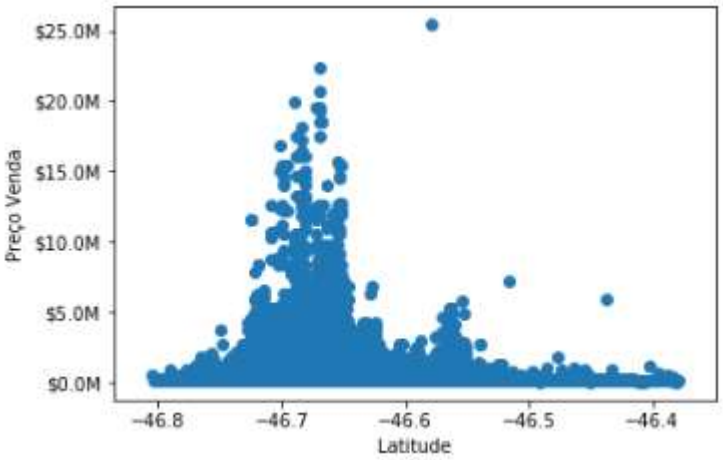
Também comparamos as coordenas com o Preço de Venda[5] dos imóveis, onde podemos constatar que os imóveis mais caros concentram-se em sua maioria nas coordenadas da região da Avenida Paulista (-23.57 -46.65) e da Cidade Jardim (-23.59, -46.70).

Gráfico 15 –"Preço de Venda" por "Latitude"



Fonte: Grupo ZAP. Tabulação própria.

Gráfico 16–"Preço de Venda" por "Longitude"



Fonte: Grupo ZAP. Tabulação própria.

[5] Por questões de visualização, o maior preço de venda (de 74 milhões e reais) foi retirado dos dados que geraram o gráfico.

4. MODELAGEM ESTATÍSTICA

4.1 Geolocalização

Um dos fatores mais importantes para definir o preço de venda de um imóvel e a sua localização. O dataset utilizado possui dois conjuntos de dado que podem ser usados para definir sua localização e que tem correlação entre si:

- Os dados de endereço (País, Cidade, Estado, Rua, Número e complemento).
- A geolocalização (latitude e longitude).

Os dados de endereço bairro e rua possuem muitas inconsistências, conforme explicado na análise exploratória acima. Desta forma optamos por utilizar as coordenadas latitude e longitude para definir a localização em nossos modelos. Como estes são campos numéricos, mas não são valores escalares, criamos a nova variável (escalar) "*Distancia*" para definir a distância de cada imóvel para uma coordenada de referência. Esta coordenada de referência foi calculada com base na média da amostra dos dados utilizados para treino dos modelos.

4.2 Dados Selecionados

Após a análise exploratória e o tratamento dos dados, selecionamos os seguintes atributos como conjunto inicial de dados[6] para o treino dos modelos estatísticos: *bathrooms, bedrooms, parkingspaces, pricinginfos_businesstype, pricinginfos_monthlycondofee, pricinginfos_rentaltotalprice, pricinginfos_yearlyiptu, publicationType, suítes, totalareas, usableareas, Zona* e *Distancia*.

Os seguintes atributos **não foram** utilizados na modelagem para estimativa de "Preço de Venda" por serem variáveis de controle, descrição[7], excesso de inconsistências, valor único, sem preenchimento, imagens[8] ou por terem sido substituídas por outro atributo após tratamento:

- address_city: valor único (São Paulo).
- address_country: premissa de valor único (Brasil).
- address_district: sem preenchimento.

[6] As variáveis categóricas foram convertidas em Dummies.

[7] Por simplificação não utilizamos algoritmos de linguagem natural para tentar extrair informações que pudessem ser utilizadas na modelagem estatística proposta. Muitas destas informações já estavam presentes em outras variáveis, tais como metragem, quantidade de quartos, etc.

[8] Também por simplificação não utilizamos algoritmos de tratamento de imagem para tentar identificar características físicas do imóvel que pudessem influenciar a decisão de compra (exemplo: estado de conservação do imóvel).

- address_geolocation_location_lat: utilizada para o tratamento de Distância.
- address_geolocation_location_lon: geolocalização da longitude do Distância.
- address_geolocation_precision: variável de controle.
- address_locationid: utilizada para o tratamento de Zona.
- address_neighborhood: inconsistente e substituída por latitude e longitude.
- address_state: premissa de valor único (São Paulo).
- address_street: inconsistente e substituída por latitude e longitude.
- address_streetnumber: substituída por latitude e longitude.
- address_unitnumber: substituída por latitude e longitude.
- address_zipcode: substituída por latitude e longitude.
- address_zone: utilizada para o tratamento de Zona.
- createdat: variável de controle.
- description: variável de descrição.
- id: variável de controle.
- images: variável com link para imagens.
- listingstatus: valor único (ACTIVE).
- owner: variável de controle.
- pricinginfos_period: sem preenchimento
- pricinginfos_price: é a própria variável target.
- publisherId: variável de controle.
- title: variável de descrição.
- unittypes: valor único (APARTMENT).
- updatedat: variável de controle.

4.3 Modelos Treinados

Utilizamos 80% do dataset source-4-ds-train para treinar os diferentes modelos abaixo, reservando 20% para aferir a precisão destes modelos.

4.3.1 Regressão Linear

Utilizamos a abordagem de Regressão Linear em uma primeira execução com toda as variáveis iniciais. Aplicando o modelo gerado nos 20% reservados para testarmos a estimativa do "Preço de Venda", obtivemos uma precisão de R^2 de 71,19% com Erro Quadrático Médio de $291{,}884 \times 10^{9.}$

Em seguida selecionamos as variáveis pelo método "*Backward Elimination*" com pvalor máximo de 0,05 e obtivemos o resultado abaixo:

OLS Regression Results

Dep. Variable:	Preco_Venda	**R-squared:**	0,621
Model:	OLS	**Adj. R-squared:**	0,621
Method:	Least Squares	**F-statistic:**	7008
Date:	Thu, 05 Sep 2019	**Prob (F-statistic):**	0.00
Time:	01:41:31	**Log-Likelihood:**	-759.750
No. Observations:	51316	**AIC:**	1.520.000
Df Residuals:	51303	**BIC:**	1.520.000
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	763.600	2.874,448	265,666	0,000	758.000	769.000
bathrooms	-30.150	4.700,550	-6,413	0,000	-39.400	-20.900
bedrooms	-137.600	4.131,314	-33,315	0,000	-146.000	-13.000
parkingspaces	165.600	5.116,738	32,361	0,000	156.000	176.000
pricinginfos_monthlycondofee	68.410	3.268,814	20,928	0,000	6.200	74.800
pricinginfos_yearlyiptu	12.650	2.959,907	4,273	0,000	68.460	18.400
suites	22.980	5.082,565	4,521	0,000	1.300	32.900
usableareas	716.500	5.280,541	135,680	0,000	706.000	727.000
Zona_Centro	-57.500	3.417,695	-16,823	0,000	-64.200	-50.800
Zona_Leste	-36.230	3.387,067	-10,697	0,000	-42.900	-29.600
Zona_Norte	-36.280	3.345,059	-10,844	0,000	-42.800	-29.700
Zona_Oeste	-62.415	3.131,176	-1,993	0,046	-12.400	-1.043
Distancia	-100.600	3.683,689	-27,298	0,000	-108.000	-93.300

Omnibus:	138.125,043	**Durbin-Watson:**	1,998
Prob(Omnibus):	0,000	**Jarque-Bera (JB):**	20.767.335.602,387
Skew:	32,184	**Prob(JB):**	0,000
Kurtosis:	3.118,853	**Cond. No.**	43,559

4.3.2 Árvore de Decisão

Realizamos vários testes com diferentes parâmetros com a abordagem "Árvore de Decisão" e obtivemos os seguintes resultados para R^2 e Erro Quadrático Médio (MSE):

Tabela 19- Resultados de "Árvore de Decisão"

Parâmetros	R^2	MSE
max_depth=20, min_samples_split=50	85,53%	$146{,}605 \times 10^9$
max_depth=30, min_samples_split=50	85,47%	$147{,}156 \times 10^9$
max_depth=50, min_samples_split=20	85,49%	$146{,}086 \times 10^9$

4.3.3 Random Forest

Realizamos vários testes com diferentes parâmetros com a abordagem "*Random Forest*" e obtivemos os seguintes resultados para R^2 e Erro Quadrático Médio (MSE):

Tabela 20 - Resultados de "*Random Forest*"

Parâmetros	R^2	MSE
n_estimators=50	89,36%	$107{,}743 \times 10^9$
n_estimators=100, max_depth=20	89,71%	$104{,}250 \times 10^9$
n_estimators=200, max_depth=30	89,79%	$103{,}386 \times 10^9$
n_estimators=200, max_depth=None, min_samples_split=20	89,02%	$111{,}241 \times 10^9$

4.3.4 Redes Neurais

Realizamos vários testes com diferentes parâmetros com a abordagem "Redes Neurais" e obtivemos os seguintes resultados para R^2 e Erro Quadrático Médio (MSE):

Tabela 21- Resultados de "Redes Neurais"

Parâmetros	R^2	MSE
max_iter=1300, activation=relu	76,60%	$237{,}004\times 10^9$
hidden_layer_sizes=(17,17), max_iter=200, activation='relu'	76,17%	$241{,}394\times 10^9$
hidden_layer_sizes=(18,18), max_iter=3000, activation='relu'	81,12%	$191{,}200\times 10^9$
hidden_layer_sizes=(18,18,18), max_iter=2000, activation='relu'	81,12%	$191{,}226\times 10^9$

4.3.5 Boosting

Realizamos vários testes com diferentes parâmetros com a abordagem "*Boosting*" e obtivemos os seguintes resultados para R^2 e Erro Quadrático Médio (MSE):

Tabela 22 - Resultados de "*Boosting*"

Parâmetros	R^2	MSE
n_estimators=50, 'min_samples_split': 50	56,59%	439,628× 10^9
n_estimators: 100, max_depth: 20, min_samples_split: 50, learning_rate: 0.01	76,28%	240,262× 10^9
n_estimators: 300, max_depth: 30, min_samples_split: 50, learning_rate: 0.01	89,55%	105,844× 10^9
n_estimators: 500, max_depth: 30, min_samples_split: 50, learning_rate: 0.01	89,97%	101,635× 10^9

Adicionalmente fizemos várias execuções através de GridSearchCV para validarmos diferentes parâmetros com a abordagem "*Boosting*" e obtivemos os seguintes resultados:

1ª execução:

tuned_parameters = {'n_estimators': [100, 200, 500], 'max_depth' : [20, 30], 'min_samples_split': [50], 'learning_rate': [0.01], 'loss': ['ls']}
Média (+/- Desvio Padrao) for {Hiperaramétros}
0.628 (+/-0.168) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 20, 'min_samples_split': 50, 'n_estimators': 100}
0.659 (+/-0.303) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 20, 'min_samples_split': 50, 'n_estimators': 200}
0.610 (+/-0.473) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 20, 'min_samples_split': 50, 'n_estimators': 500}
0.623 (+/-0.172) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 100}
0.659 (+/-0.304) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 200}
0.603 (+/-0.486) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 500}

Com a execução acima, obtivemos o melhor ajuste com estes parâmetros:

Parâmetros	R^2	MSE
'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 200	87,70%	124,575× 10^9

2ª execução:

tuned_parameters = {'n_estimators': [100, 200, 500], 'max_depth' : [30, 50], 'min_samples_split': [50], 'learning_rate': [0.001, 0.01, 0.1], 'loss': ['ls']}
Média (+/- Desvio Padrao) for {Hiperaramétros}
0.143 (+/-0.038) for {'learning_rate': 0.001, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 100}
0.260 (+/-0.067) for {'learning_rate': 0.001, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 200}
0.483 (+/-0.114) for {'learning_rate': 0.001, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 500}
0.143 (+/-0.038) for {'learning_rate': 0.001, 'loss': 'ls', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 100}
0.260 (+/-0.067) for {'learning_rate': 0.001, 'loss': 'ls', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 200}
0.483 (+/-0.114) for {'learning_rate': 0.001, 'loss': 'ls', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 500}
0.623 (+/-0.173) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 100}

0.662 (+/-0.297) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 200}
0.603 (+/-0.486) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 500}
0.623 (+/-0.172) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 100}
0.663 (+/-0.294) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 200}
0.600 (+/-0.498) for {'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 500}
0.563 (+/-0.507) for {'learning_rate': 0.1, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 100}
0.565 (+/-0.467) for {'learning_rate': 0.1, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 200}
0.544 (+/-0.515) for {'learning_rate': 0.1, 'loss': 'ls', 'max_depth': 30, 'min_samples_split': 50, 'n_estimators': 500}
0.584 (+/-0.538) for {'learning_rate': 0.1, 'loss': 'ls', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 100}
0.567 (+/-0.527) for {'learning_rate': 0.1, 'loss': 'ls', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 200}
0.585 (+/-0.452) for {'learning_rate': 0.1, 'loss': 'ls', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 500}

Com a execução acima, obtivemos o melhor ajuste com estes parâmetros:

Parâmetros	R^2	MSE
'learning_rate': 0.01, 'loss': 'ls', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 200	87,73%	$124{,}266 \times 10^9$

3ª execução:

tuned_parameters = {'n_estimators': [50, 150, 200], 'max_depth' : [50], 'min_samples_split': [50], 'learning_rate': [0.01], 'loss': ['lad']}
Média (+/- Desvio Padrao) for {Hiperaramétros}
0.230 (+/-0.064) for {'learning_rate': 0.01, 'loss': 'lad', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 50}
0.581 (+/-0.160) for {'learning_rate': 0.01, 'loss': 'lad', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 150}
0.666 (+/-0.177) for {'learning_rate': 0.01, 'loss': 'lad', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 200}

Com a execução acima, obtivemos o melhor ajuste com estes parâmetros:

Parâmetros	R^2	MSE
'learning_rate': 0.01, 'loss': 'lad', 'max_depth': 50, 'min_samples_split': 50, 'n_estimators': 200	74,89%	$254{,}271 \times 10^9$

5. MODELO(S) PROPOSTO(S)

De acordo com os resultados dos testes efetuados com os diferentes modelos e hiperparâmetros, os melhores modelos foram:

- ★ *Boosting* ➔ **R^2 de 89,97%** e **MSE de 101,635 × 10^9** com os seguintes hiperparâmetros: 500 estágios de boosting (n_estimators), 30 profundidade máxima (max_depth), 40 de quantidade mínima para split de cada nó interno (min_samples_split) e taxa de aprendizado de contribuição de cada árvore (learning_rate): 0,01.
- ★ *Random Forrest* ➔ **R^2 de 89,79%** e **MSE de 103,386 × 10^9** com os seguintes hiperparâmetros: 200 árvores na floresta (n_estimators) e 30 profundidade máxima de cada árvore (max_depth).

Pergunta do Desafio 1:

"*Você utilizaria a métrica escolhida para seleção de modelo também para comunicar os resultados para usuários e stakeholders internos? Em caso negativo, qual outra métrica você utilizaria nesse caso?*"

A escolha de um modelo depende tanto da precisão deste modelo quanto da cultura dos Stakeholders em compreendê-lo. As métricas de precisão escolhidas foram o R^2 e o Erro Quadrático Médio (MSE), que são medidas para a avaliação de modelos baseados em Regressão (como todos os utilizados neste projeto). Com R^2 temos o indicador de quanto as variáveis explicativas conseguem explicar a nossa variável target, com o MSE temos a medida de risco de quanto os valores estimados se distanciam (variam) da realidade.

O modelo escolhido para conclusão deste projeto foi o "*Boosting*", por ser o que apresenta o maior R^2 e o menor Erro Quadrático Médio (MSE). Entretanto, caso a organização tenha uma melhor compreensão de uma árvore de decisão, poderíamos escolher o modelo "*Random Forrest"* que tem métricas próximas.

Apesar do modelo escolhido ter uma precisão razoável, com o R^2 quase 90%, ainda apresenta um risco (MSE) extremante alto. Assim, recomendamos que antes de apresentar esses resultados para usuários e Stakeholerds deve-se proceder com testes exaustivos dos hiperparâmetros dos modelos estudados (veja seção 8.1) com o objetivo de melhorar a métrica MSE.

6. SOLUÇÃO DESENVOLVIDA

A solução desenvolvida por este trabalho foi toda implementada em Python através do Jupyter e tem como premissa forte que os *datasets* de entrada ("*source-4-ds-train.json*" e "*source-4-ds-test.json*") contém os registros de uma única cidade. Os programas componentes desta solução são:

6.1 Componentes

- ★ "*Converter JSON.ipynb*" ➔ programa para converter os arquivos JSON (*"source-4-ds-train.json"* e *"source-4-ds-test.json"*) em Dataframes, normalizando seus dados semiestruturados em uma tabela plana.
- ★ "*Analise Exploratoria - Base Treino Completa.ipynb*" ➔ Programa para Análise Exploratória do Dataset "*source-4-ds-train.json"* completo com <u>todos os imóveis</u>. Com este programa fizemos a análise exploratória inicial conforme detalhada na seção 3.1.
- ★ "*Analise Exploratoria - Base Treino Venda Apto.ipynb*" ➔ Programa para Análise Exploratória do Dataset source-4-ds-train.json somente para o conjunto de dados referente aos <u>apartamentos à venda</u>. Com este programa fizemos a análise exploratória conforme detalhada na seção 3.2.
- ★ "*Preparar Base de Treino.ipynb*" ➔ Programa para preparar a base de treino do conjunto de dados referente aos <u>apartamentos à venda.</u> Com este programa fazemos o "train_test_split" para separar e salvar (no formato pickle) as amostras que vão ser utilizadas pelo programa que irá treinar e testar o modelo. Este também faz todo o tratamento de dados, a saber:
    - o Seleção dos imóveis do tipo de unidade "apartamento" e tipo de negócio "venda".
    - o Criação da coluna "Zona" com base na combinação dos campos "*address_locationid*" e "*address_zone*" (vide página 12).
    - o Conversão de variáveis categóricas em *dummies*, com a remoção de caracteres especiais e espaços.
    - o Correção de *Missing* e *Outliers*.
    - o Criação da coluna que mede a distância entre a geolocalização de referência (mediana das coordenadas dos imóveis à venda na cidade) e cada imóvel listado.

- Remoção de todas as variáveis que não são utilizadas pelos métodos estatístico escolhidos.

Arquivos gerados (somente para registros de "apartamentos à venda"):

- df_X_features.pickle ➔ contém o dataframe com todos os registros de todas as variáveis explicativas antes de fazer o Split entre treino e teste e antes do tratamento de *Missing* e *Outliers*.
- df_y_label.pickle ➔ contém a série de todos os registros da variável target antes de fazer o Split entre treino e teste.
- df_X_train_features.pickle ➔ contém o dataframe com os registros de "treino" de todas as variáveis explicativas.
- df_X_test_features.pickle ➔ contém o dataframe com os registros de "teste" de todas as variáveis explicativas.
- df_y_train_label.pickle ➔ contém a série com os registros de "treino" da variável target.
- df_y_test_label.pickle ➔ contém a série com os registros de "teste" da variável target.
- df_medidas.pickle ➔ contém o dataframe com os parâmetros que são utilizados para tratamento de *Missing* (mediana das variáveis de treino), *Outliers* (percentil 99.9) e as coordendas de Geolocalização (mediana de latitude e longitude de um conjunto de dados).

★ "*Treinar o Modelo.ipynb*" ➔ Programa para treinar o modelo do zap com hiperparâmetros iniciais de vários métodos estatísticos. Todos os modelos gerados são salvos no formato pickle, a saber:

- modelo_lr.pickle ➔ contém o modelo de Regressão Linear gerado.
- modelo_dr.pickle ➔ contém o modelo de Árvore de Decisão gerado.
- 'modelo_rf.pickle ➔ contém o modelo de *Random Forrest* gerado.
- modelo_clf.pickle ➔ contém o modelo de *Boosting* gerado.
- modelo_mlp.pickle ➔ contém o modelo de *Redes Neurais "Multi-layer Perceptron regressor"* gerado.

★ "*Treinar Boosting e Hiperametros.ipynb*" ➔ Programa para Treinar o Modelo ***Boosting*** com diferentes hiperparâmetros usando o recurso *GridSearchCV* com *K-Fold Cross-Validation*. Para a execução deste programa ocorrer no mesmo momento que o programa "*Treinar o Modelo.ipynb*" é necessário que sejam

executados em diretórios diferentes (esta configuração de ambiente não fez parte desta entrega). O arquivo gerado é salvo no formato pickle:

- modelo_clf.pickle ➔ o melhor modelo de *Boosting* gerado.

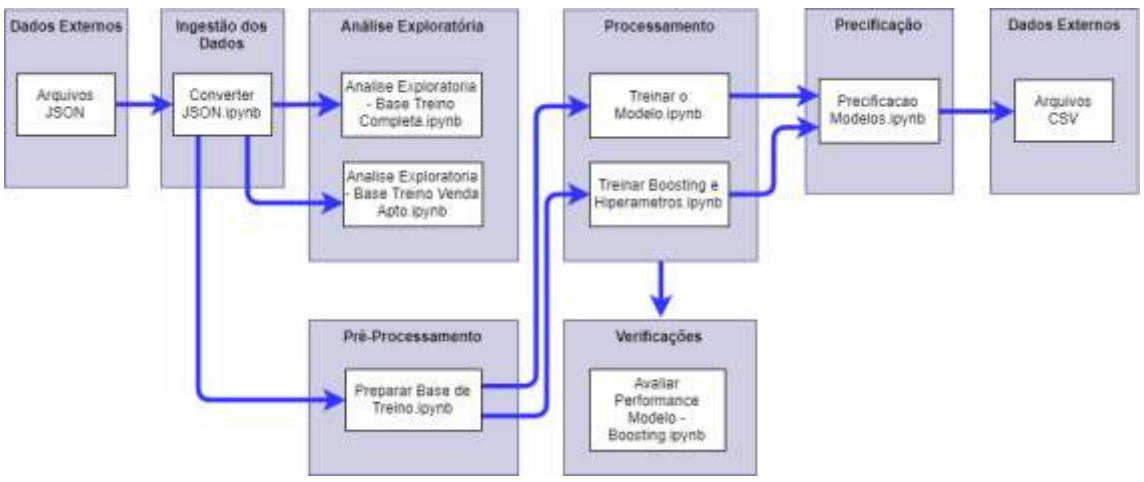
★ "*Precificacao Modelos.ipynb*" ➔ Programa para precificar o dataset "*source-4-ds-test.json*". Utiliza como entrada todos os modelos gerados para cada um dos métodos estatísticos. E gera como saída a precificação dos imóveis listados neste dataset para cada um dos modelos, salvos no formato CSV (o arquivo inclui os headers (id e price), os preços estão com decimal no formato americano e os campos estão separados por vírgula):

- predictions_lr.csv ➔ contém a precificação dos imóveis baseada em Regressão Linear.
- predictions_dr.csv ➔ contém a precificação base dos imóveis baseada ado em Árvore de Decisão.
- predictions_rf.csv ➔ contém a precificação base dos imóveis baseada ado em *Random Forrest*.
- predictions_clf.csv ➔ contém a precificação base dos imóveis baseada ado em *Boosting*.
- predictions_mlp.csv ➔ contém a precificação base dos imóveis baseada ado em *Redes Neurais "Multi-layer Perceptron regressor"*.

★ "*Avaliar Performance Modelo* - Boosting.ipynb" ➔ Programa para avaliar os seguintes itens do Modelo escolhido (*Boosting*):

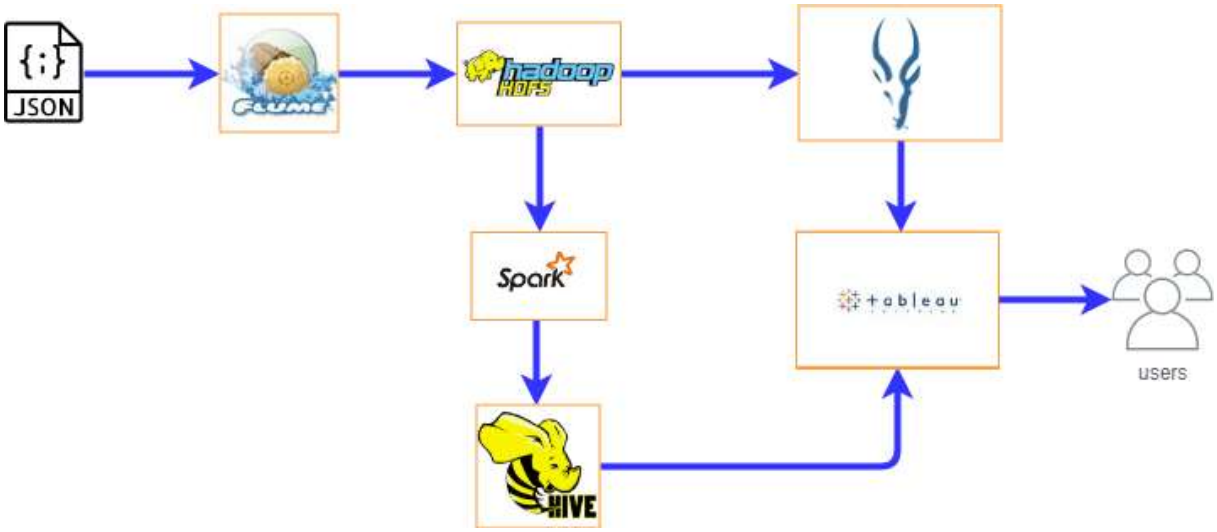
- Verificar a performance da Estimativa por Bairro (vide seção 8.3).
- Identificar as *features* mais importantes do modelo (vice seção 8.4).

6.2 Fluxo Desenvolvido



7. ARQUITETURA PROPOSTA

Para esta solução ser colocada em produção, propomos a utilização do ecossistema Hadoop combinado com o software de visualização Tableau:



- ★ Apache Flume ➔ Componente responsável pela ingestão dos arquivos JSON para o Data Lake no Hadoop Apache Hadoop (HDFS)
- ★ Apache Hadoop (HDFS) ➔ Data Lake para manter armazendo os arquivos JSON.
- ★ Apache Spark ➔ Os componentes entregues como parte deste projeto devem ser adaptados e configurados para serem utilizados no Apache Spark e para armazenar os resultados no Apache Hive.
- ★ Apache Impala ➔ Componente opcional que pode ser utilizado para obter informações detalhados dos dados armazenados no HDFS e consolidação no Tableau.
- ★ Tableau ➔ Componente que consolidará os dados dos modelos e precificações para disponibilizar aos *Stakeholders* e usuários internos.

8. CONCLUSÕES E RECOMENDAÇÕES

8.1 Modelos

Como mencionado na seção MODELO(S) PROPOSTO(S), o modelo escolhido tem uma precisão razoável, com o R^2 quase 90%, mas ainda apresenta um risco (MSE) extremante alto. Entretanto ressalvo que os testes de hiperparâmetros não foram exaustivos, bem como algumas variáveis que poderiam melhorar a precisão foram descartadas no processo de análise exploratória e preparação dos dados. Por isto, recomenda-se a revisão da modelagem dos dados e testes exaustivos de hiperparâmetros. Entre as variáveis que poderiam ser revistas temos, por exemplo:

- Bairro → tratar as inconsistências no nome dos bairros.
- Rua → tratar as inconsistências no nome dos bairras.
- Imagens dos imóveis → utilizar algoritmos de tratamento de imagem para tentar identificar características físicas do imóvel que pudessem influenciar a decisão de compra (exemplo: estado de conservação do imóvel).

Adicionalmente os hiperparâmetros dos modelos devem ser exaustivamente testados com os recursos de *K-Fold Cross-Validation* e *GridSearchCV*, até que cada um destes modelos chegue no ponto de *overfitting*. Buscar-se-á então os modelos mais eficientes, com a combinação de hiperparâmetros que não tenho atingido o ponto de *overfitting*.

8.2 Latitude e Longitude x Bairros e Ruas

A inclusão das variáveis bairros e ruas em um modelo acrescenta a complexidade que diferentes cidades têm diferentes bairros e ruas. Mesmo que duas ou mais cidades tenham bairros e ruas com o mesmo o nome, isto não implica na mesma localização; pelo contrário, são locais diferentes com o mesmo nome. Como na prática o conjunto de bairros e ruas de uma cidade é diferente de outra cidade, e estas variáveis precisam ser tratadas como dummies, temos que necessariamente implementar um modelo diferente para cada cidade.

Adicionalmente, todos os bairros de uma cidade deveriam estar presentes, tanto no conjunto de dados que irá gerar o modelo, quanto no conjunto de dados que será precificado. Como isto na realidade é impraticável, a solução tem que incluir um algoritmo incluir todas as colunas *dummies* de bairros tanto no *dataset* que será treinado, quanto nos dados que serão precificados.

Pelas ressalvas acima que o autor deste projeto optou por utilizar as coordenadas de latitude e longitude, permitindo simplificar a solução entrega.

8.3 Melhor Performance do Modelo

Pergunta do Desafio 2:

"*Em quais bairros ou em quais faixas de preço o seu modelo performa melhor?*"

O modelo escolhido tem melhor performance na estimava do "Preço de Venda" nos bairros listados na Tabela 23. A medida de performance foi calculada com base na méida da diferença relativa do preço estimado para o preço venda, com os imóveis agrupados por bairro.

Tabela 23- Performance da Estimativa por Bairro (TOP 10)

address_neighborhood	Diferença de Estimativa	Diferença Absoluta
Parque Edu Chaves	-0,03%	0,03%
Pirajussara	0,07%	0,07%
Vila Sonia	-0,07%	0,07%
Pinheiros	-0,11%	0,11%
Chácara Califórnia	-0,11%	0,11%
Planalto Paulista	0,12%	0,12%
Jardim Célia	0,13%	0,13%
Chácara Inglesa	-0,16%	0,16%
Jardim das Perdizes	0,16%	0,16%
Vila Regente Feijó	-0,16%	0,16%

Fonte: Autor. Tabulação própria.

8.4 As 3 Variáveis TOP

Pergunta do Desafio 3:

"*Se você tivesse que estimar o valor dos imóveis com apenas 3 campos, quais seriam eles?*"

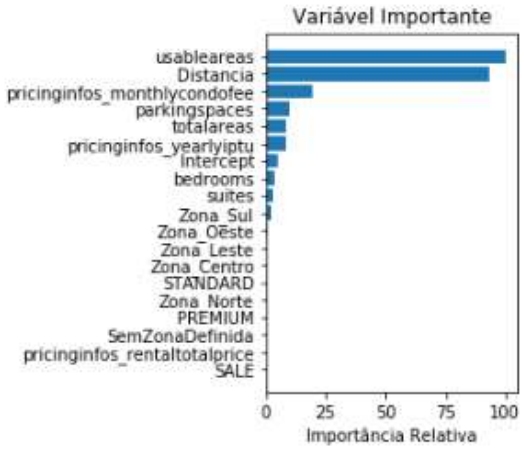
Ao avaliarmos a importância de cada variável do modelo utilizado, temos a definição dos três campos que utilizaríamos: tamanho do imóvel, distância para a coordenada de referência (vetor de localização do imóvel) e taxa de condomínio. Na Tabela 24 podemos ver o valor absoluto da importância das variáveis mais importantes, com destaque para o tamanho do imóvel e para a localização (distância). No Gráfico 17 fica ainda mais evidente a importância relativa destas duas variáveis. Tal conclusão do modelo é coerente com regra de mercado que o preço de um imóvel de definido principalmente por seu tamanho e localização.

Tabela 24 – Importância das Variáveis Explicativas (TOP 5)

Atributos	Importância Absoluta
Tamanho do Imóvel (usableareas)	39,2%
Vetor de localização (Distancia)	36,6%
Taxa de Condomínio (pricinginfos_monthlycondofee)	7,5%
Vagas de Estacionamento (parkingspaces)	3,9%
Área Total (totalareas)	3,4%

Fonte: Autor. Tabulação própria.

Gráfico 17– Importância das Variáveis Explicativas



Fonte: Autor. Tabulação própria.

---- The End ---