desafio_grupo_zap

July 26, 2019

1 Desafio Grupo ZAP - 07/2019

1.1 Objetivo

- Um anúncio no portal é composto por diversas características do imóvel, como tamanho, número de quartos, etc. Uma das principais características do imóvel é o seu preço de venda, identificado pelo campo price que, por sua vez, está dentro do campo princingInfos.
- O objetivo é criar uma maneira automática de estimar um preço de venda para os apartamentos.

1.2 Entregas

- Arquivo csv com os preços dos anúncios com os campos: [id, price]
- Explicação da solução
- Código fonte da solução
- Respostas (com insumos para suporte) para as seguintes questões:
  - 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?
  - Em quais bairros ou em quais faixas de preço o seu modelo performa melhor?
  - Se você tivesse que estimar o valor dos imóveis com apenas 3 campos, quais seriam eles?
  - Como você vislumbra colocar a sua solução em produção?

2 Processo que será seguido

```
In [1]: from IPython.display import Image
        Image(url = 'images/processo.png')
```

```
Out[1]: <IPython.core.display.Image object>
```

3 Importando as bibliotecas que serão utilizadas neste projeto

```
In [2]: # Importando bibliotecas que serao utilizadas neste projeto
        import pandas as pd
        import json
```

```
from pandas.io.json import json_normalize
import numpy as np
import seaborn as sns
import itertools
import matplotlib.pyplot as plt
%matplotlib inline

# Models
from xgboost import XGBRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from mlxtend.regressor import StackingCVRegressor
from lightgbm import LGBMRegressor
import lightgbm as lgb
import xgboost as XGB
from sklearn.cluster import KMeans

# Stats
from scipy import stats
from scipy.stats import skew, norm
from scipy.stats import randint as sp_randint
from scipy.stats import uniform as sp_uniform

# Misc
from sklearn import preprocessing
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import RobustScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold, cross_val_score
from sklearn.model_selection import RandomizedSearchCV
from functools import partial
from hyperopt import fmin, hp, tpe, Trials, space_eval
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.decomposition import PCA
from sklearn.pipeline import make_pipeline
from IPython.display import Image

#!pip install folium
import folium
from folium import plugins

#!pip install pandasql
import pandasql as ps

# Ignore useless warnings
import warnings
```

```
warnings.filterwarnings(action="ignore")
pd.options.display.max_seq_items = 8000
pd.options.display.max_rows = 8000
pd.set_option('display.max_columns', None)

import pickle
import datetime
import re
import gc
```

```
/anaconda3/lib/python3.7/site-packages/lightgbm/__init__.py:46: UserWarning: Starting from vers
This means that in case of installing LightGBM from PyPI via the ``pip install lightgbm`` comma
Instead of that, you need to install the OpenMP library, which is required for running LightGBM
You can install the OpenMP library by the following command: ``brew install libomp``.
  "You can install the OpenMP library by the following command: ``brew install libomp``.", User
```

4 Extraindo e Carregando os Dados

```
In [3]: with open('data/source-4-ds-train.json') as file_data:
            data = file_data.readlines()

        train = json_normalize([json.loads(d) for d in data])
        train.head(3)
```

```
Out[3]:   address.city address.country address.district  \
        0    São Paulo              BR
        1    São Paulo              BR
        2    São Paulo

           address.geoLocation.location.lat  address.geoLocation.location.lon  \
        0                         -23.612923                         -46.614222
        1                         -23.643962                         -46.593475
        2                         -23.568559                         -46.647452

          address.geoLocation.precision  \
        0                       ROOFTOP
        1           RANGE_INTERPOLATED
        2                       ROOFTOP

                                       address.locationId  \
        0  BR>Sao Paulo>NULL>Sao Paulo>Zona Sul>Jardim da...
        1  BR>Sao Paulo>NULL>Sao Paulo>Zona Sul>Vila Liviero
        2  BR>Sao Paulo>NULL>Sao Paulo>Centro>Cerqueira C...

                  address.neighborhood address.state      address.street  \
        0              Jardim da Saúde     São Paulo  Rua Juvenal Galeno
        1  Vila Santa Teresa (Zona Sul)     São Paulo         Rua Juruaba
```

```
2                   Bela Vista     São Paulo    Avenida Paulista

  address.streetNumber address.unitNumber address.zipCode address.zone  \
0                   53                           04290030     Zona Sul
1                   16                           04187320     Zona Sul
2                  402                           01311000

   bathrooms  bedrooms                createdAt  \
0        3.0       4.0     2017-02-07T13:21:40Z
1        2.0       3.0     2016-03-21T18:35:17Z
2        4.0       0.0  2018-12-18T23:47:03.425Z

                                        description          id  \
0  04 dorms sendo 01 suíte e closet, sala de esta...  787c7bd19d
1  03 dorms sendo 01 suíte, sala, sala de jantar,...  4d68c0cdbe
2  Andar com 395,70mš de área útil, 04 wcs, 05 va...  e7e0b554ac

                                              images listingStatus  owner  \
0  [https://s3-sa-east-1.amazonaws.com/vr.images...        ACTIVE  False
1  [https://s3-sa-east-1.amazonaws.com/vr.images...        ACTIVE  False
2  [http://static.nidoimovel.com.br/d3d9446802a44...        ACTIVE  False

   parkingSpaces pricingInfos.businessType  pricingInfos.monthlyCondoFee  \
0            6.0                     SALE                           NaN
1            2.0                     SALE                           NaN
2            5.0                   RENTAL                        4900.0

  pricingInfos.period  pricingInfos.price  pricingInfos.rentalTotalPrice  \
0                 NaN              700000                           NaN
1                 NaN              336000                           NaN
2             MONTHLY               24929                       29829.0

   pricingInfos.yearlyIptu publicationType publisherId  suites  \
0                      NaN        STANDARD  f4603b2b52     1.0
1                      NaN        STANDARD  f4603b2b52     1.0
2                   4040.0        STANDARD  501f6d5e94     0.0

                                    title  totalAreas  \
0   PRÓXIMO A AVENIDA PRESIDENTE TANCREDO NEVES       388.0
1      PRÓXIMO A FACULDADE UNIP CAMPUS ANCHIETA       129.0
2  Excelente Conjunto Comercial na Av. Paulista         NaN

             unitTypes                 updatedAt  usableAreas
0      TWO_STORY_HOUSE  2018-12-06T19:27:12.623Z        388.0
1                 HOME  2018-12-12T13:17:23.547Z        129.0
2  COMMERCIAL_PROPERTY                       NaN        396.0
```

In [4]:
```
with open('data/source-4-ds-test.json') as file_data:
```

```
    data = file_data.readlines()

test = json_normalize([json.loads(d) for d in data])
test.head(3)
```

Out[4]:

```
  address.city address.country address.district  \
0    São Paulo              BR
1    São Paulo              BR
2    São Paulo

   address.geoLocation.location.lat  address.geoLocation.location.lon  \
0                         -23.557225                         -46.662765
1                         -23.592852                         -46.581879
2                         -23.493609                         -46.638456

  address.geoLocation.precision  \
0              GEOMETRIC_CENTER
1                       ROOFTOP
2                       ROOFTOP

                                address.locationId address.neighborhood  \
0     BR>Sao Paulo>NULL>Sao Paulo>Centro>Consolacao           Consolação
1  BR>Sao Paulo>NULL>Sao Paulo>Zona Leste>Quinta ...   Quinta da Paineira
2  BR>Sao Paulo>NULL>Sao Paulo>Zona Norte>Santa T...         Chora Menino

  address.state              address.street address.streetNumber  \
0    São Paulo             Rua Bela Cintra
1    São Paulo  Rua Bruno Cavalcanti Feder                  100
2    São Paulo              Rua Copacabana                  313

  address.unitNumber address.zipCode address.zone  bathrooms  bedrooms  \
0                          01415000       Centro        1.0         1
1                          03152155   Zona Leste        0.0         2
2                          02461000                     3.0         3

                 createdAt  \
0     2015-10-20T20:52:41Z
1  2018-07-31T06:10:07.427Z
2  2018-01-25T13:57:14.203Z

                                        description          id  \
0  Apartamentos de 1 dormitório na Rua Bela Cintr...  89224365f8
1  Ótima localização, próximo ao shopping Central...  363731333f
2  Apartamento maravilhoso com ampla sala ( abriu...  6e6283378a

                                              images listingStatus  owner  \
0  [https://s3-sa-east-1.amazonaws.com/vr.images...        ACTIVE  False
1  [http://images.ingaiasites.com.br/AolwiwJLLpET...        ACTIVE  False
```

```
2  [https://ssl-w08cnn0135.websiteseguro.com/mira...        ACTIVE  False

   parkingSpaces pricingInfos.businessType  pricingInfos.monthlyCondoFee  \
0            1.0                      SALE                           NaN
1            1.0                      SALE                           0.0
2            2.0                      SALE                         686.0

  pricingInfos.period pricingInfos.price  pricingInfos.rentalTotalPrice  \
0                 NaN               None                           NaN
1                 NaN               None                           NaN
2                 NaN               None                           NaN

   pricingInfos.yearlyIptu publicationType publisherId  suites  \
0                      NaN        STANDARD  967d57ce20     0.0
1                      0.0        STANDARD  bddebf057a     0.0
2                      NaN        STANDARD  d7190e8f4c     1.0

                                          title  totalAreas  unitTypes  \
0                          Apartamento Bela Cintra        47.0  APARTMENT
1  Apartamento residencial à venda, Quinta da Pai...        55.0  APARTMENT
2     Apartamento em Santa Terezinha - São Paulo, SP         NaN  APARTMENT

                  updatedAt  usableAreas
0  2018-11-08T15:02:53.953Z         47.0
1  2018-11-08T16:10:49.374Z         55.0
2  2019-02-12T18:29:26.933Z         92.0
```

```
In [5]: # Configurando a coluna id como index no dataset
        train.set_index('id',inplace=True)
        test.set_index('id',inplace=True)
```

Analisando o dataframe, visualizo apenas a coluna imagem aninhada. Json_normalize nos dá algumas dicas sobre como nivelar os dados semi-estruturados. Por enquanto vou remover essa coluna do dataset pra analise do modelo

```
In [6]: # Removendo algumas colunas do dataset que identifiquei como não necessárias para o mo
        train.drop(columns = ["images",
                              "address.unitNumber",
                              "address.streetNumber",
                              "description",
                              "publisherId",
                              "title",
                              "updatedAt",
                              "address.street"
                             ], inplace = True)

        test.drop(columns = ["images",
                             "address.unitNumber",
```

```
                                "address.streetNumber",
                                "description",
                                "publisherId",
                                "title",
                                "updatedAt",
                                "address.street",
                                "pricingInfos.price"
                               ], inplace = True)
```

```
In [7]: train.shape, test.shape
```

```
Out[7]: ((133964, 28), (16036, 27))
```

```
In [8]: train.dtypes
```

```
Out[8]: address.city                          object
        address.country                       object
        address.district                      object
        address.geoLocation.location.lat     float64
        address.geoLocation.location.lon     float64
        address.geoLocation.precision         object
        address.locationId                    object
        address.neighborhood                  object
        address.state                         object
        address.zipCode                       object
        address.zone                          object
        bathrooms                            float64
        bedrooms                             float64
        createdAt                             object
        listingStatus                         object
        owner                                   bool
        parkingSpaces                        float64
        pricingInfos.businessType             object
        pricingInfos.monthlyCondoFee         float64
        pricingInfos.period                   object
        pricingInfos.price                     int64
        pricingInfos.rentalTotalPrice        float64
        pricingInfos.yearlyIptu              float64
        publicationType                       object
        suites                               float64
        totalAreas                           float64
        unitTypes                             object
        usableAreas                          float64
        dtype: object
```

```
In [9]: # Renomeando as colunas para facilitar as analises posteriores
        train = train.rename(columns={"address.city": "city",
                                      "address.country": "country",
                                      "address.district": "district",
```

```
                                    "address.geoLocation.location.lat": "latitude",
                                    "address.geoLocation.location.lon": "longitude",
                                    "address.geoLocation.precision": "precision",
                                    "address.locationId": "locationId",
                                    "address.neighborhood": "neighborhood",
                                    "address.state": "state",
                                    "address.zipCode": "zipCode",
                                    "address.zone": "zone",
                                    "pricingInfos.businessType": "businessType",
                                    "pricingInfos.monthlyCondoFee": "monthlyCondoFee",
                                    "pricingInfos.period": "period",
                                    "pricingInfos.price": "price",
                                    "pricingInfos.rentalTotalPrice": "rentalTotalPrice",
                                    "pricingInfos.yearlyIptu": "yearlyIptu"})

# Renomeando as colunas para facilitar as analises posteriores
test = test.rename(columns={"address.city": "city",
                            "address.country": "country",
                            "address.district": "district",
                            "address.geoLocation.location.lat": "latitude",
                            "address.geoLocation.location.lon": "longitude",
                            "address.geoLocation.precision": "precision",
                            "address.locationId": "locationId",
                            "address.neighborhood": "neighborhood",
                            "address.state": "state",
                            "address.zipCode": "zipCode",
                            "address.zone": "zone",
                            "pricingInfos.businessType": "businessType",
                            "pricingInfos.monthlyCondoFee": "monthlyCondoFee",
                            "pricingInfos.period": "period",
                            "pricingInfos.rentalTotalPrice": "rentalTotalPrice",
                            "pricingInfos.yearlyIptu": "yearlyIptu"})
```

5 Análise Exploratória de Dados

```
In [10]: # Visualizando as primeiras 5 linhas do dataset de treino
         train.head()
```

```
Out[10]:                 city country district   latitude  longitude  \
         id
         787c7bd19d  São Paulo      BR          -23.612923 -46.614222
         4d68c0cdbe  São Paulo      BR          -23.643962 -46.593475
         e7e0b554ac  São Paulo                  -23.568559 -46.647452
         6654d93423  São Paulo                  -23.598023 -46.682781
         9ffaf676ae  São Paulo                  -23.582497 -46.653027

                              precision  \
```

```
id
787c7bd19d             ROOFTOP
4d68c0cdbe  RANGE_INTERPOLATED
e7e0b554ac             ROOFTOP
6654d93423  RANGE_INTERPOLATED
9ffaf676ae  RANGE_INTERPOLATED

                                                 locationId  \
id
787c7bd19d  BR>Sao Paulo>NULL>Sao Paulo>Zona Sul>Jardim da...
4d68c0cdbe  BR>Sao Paulo>NULL>Sao Paulo>Zona Sul>Vila Liviero
e7e0b554ac  BR>Sao Paulo>NULL>Sao Paulo>Centro>Cerqueira C...
6654d93423  BR>Sao Paulo>NULL>Sao Paulo>Zona Sul>Vila Olimpia
9ffaf676ae       BR>Sao Paulo>NULL>Sao Paulo>Zona Sul>Paraiso

                         neighborhood      state   zipCode      zone  \
id
787c7bd19d               Jardim da Saúde  São Paulo  04290030  Zona Sul
4d68c0cdbe  Vila Santa Teresa (Zona Sul)  São Paulo  04187320  Zona Sul
e7e0b554ac                   Bela Vista  São Paulo  01311000
6654d93423                 Vila Olímpia  São Paulo  04550004
9ffaf676ae                      Paraíso  São Paulo  04005030

            bathrooms  bedrooms                 createdAt listingStatus  \
id
787c7bd19d        3.0       4.0      2017-02-07T13:21:40Z        ACTIVE
4d68c0cdbe        2.0       3.0      2016-03-21T18:35:17Z        ACTIVE
e7e0b554ac        4.0       0.0  2018-12-18T23:47:03.425Z        ACTIVE
6654d93423        2.0       3.0  2018-10-26T16:18:28.915Z        ACTIVE
9ffaf676ae        5.0       4.0  2018-12-14T18:06:51.342Z        ACTIVE

            owner  parkingSpaces businessType  monthlyCondoFee   period  \
id
787c7bd19d  False            6.0         SALE              NaN      NaN
4d68c0cdbe  False            2.0         SALE              NaN      NaN
e7e0b554ac  False            5.0       RENTAL           4900.0  MONTHLY
6654d93423  False            2.0         SALE            686.0      NaN
9ffaf676ae  False            5.0         SALE           6230.0      NaN

              price  rentalTotalPrice  yearlyIptu publicationType  suites  \
id
787c7bd19d   700000               NaN         NaN        STANDARD     1.0
4d68c0cdbe   336000               NaN         NaN        STANDARD     1.0
e7e0b554ac    24929           29829.0      4040.0        STANDARD     0.0
6654d93423   739643               NaN      1610.0        STANDARD     1.0
9ffaf676ae  7520099               NaN     18900.0        STANDARD     4.0

            totalAreas            unitTypes  usableAreas
```

```
id
787c7bd19d       388.0      TWO_STORY_HOUSE        388.0
4d68c0cdbe       129.0                 HOME        129.0
e7e0b554ac         NaN  COMMERCIAL_PROPERTY        396.0
6654d93423        80.0            APARTMENT         80.0
9ffaf676ae       332.0            APARTMENT       3322.0
```

In [11]:
```
# Sumário estatístico
train.describe()
```

Out[11]:

	latitude	longitude	bathrooms	bedrooms	\
count	133953.000000	133953.000000	133051.000000	130945.000000	
mean	-23.554263	-46.643395	2.375683	2.327023	
std	0.165147	0.318494	2.146044	2.140123	
min	-23.848153	-46.820973	0.000000	0.000000	
25%	-23.594475	-46.684151	1.000000	2.000000	
50%	-23.558990	-46.654071	2.000000	2.000000	
75%	-23.527634	-46.607885	3.000000	3.000000	
max	0.000000	0.000000	200.000000	600.000000	

	parkingSpaces	monthlyCondoFee	price	rentalTotalPrice	\
count	129539.000000	1.171270e+05	1.339640e+05	2.871400e+04	
mean	2.443187	1.507679e+03	6.637484e+05	1.102183e+04	
std	5.251624	7.795406e+04	1.317732e+06	7.701223e+04	
min	0.000000	0.000000e+00	7.000000e+01	0.000000e+00	
25%	1.000000	0.000000e+00	1.750000e+05	2.310000e+03	
50%	2.000000	3.990000e+02	3.710000e+05	4.111500e+03	
75%	3.000000	8.610000e+02	7.000000e+05	9.100000e+03	
max	589.000000	2.443000e+07	8.400000e+07	1.190000e+07	

	yearlyIptu	suites	totalAreas	usableAreas
count	1.146120e+05	120347.000000	9.113200e+04	1.332110e+05
mean	4.182520e+03	1.106534	1.327059e+04	2.204832e+02
std	8.501091e+05	1.218938	3.682658e+06	5.713502e+03
min	0.000000e+00	0.000000	0.000000e+00	0.000000e+00
25%	0.000000e+00	0.000000	6.000000e+01	6.000000e+01
50%	8.400000e+01	1.000000	1.200000e+02	1.050000e+02
75%	3.640000e+02	2.000000	2.500000e+02	2.000000e+02
max	2.830242e+08	80.000000	1.111111e+09	2.025000e+06

Realizando alguns tratamentos e split de features antes de verificar valores nulos

In [12]:
```
# Preenchendo valores em branco (espaco) por valores nulos para serem tratados
train = train.replace(r'^\s*$', np.nan, regex=True)
test = test.replace(r'^\s*$', np.nan, regex=True)
```

In [13]:
```
# Alterando o valor None da variavel zipCode para '00000000'
train['zipCode'] = train['zipCode'].replace('None', '00000000', regex=True)
test['zipCode'] = test['zipCode'].replace('None', '00000000', regex=True)
```