Data Science Initiative
BROWN

Midterm Project Presentation

—— OYO rental price prediction in China

Brown University, Data Science Initiative,22fall

GitHub: https://github.com/AstrosiosaurQ7/data1030_mid_proj

Xiaoyan Liu

19/10/2022

Content

1 Introduction

2 EDA Exploratory Data Analysis

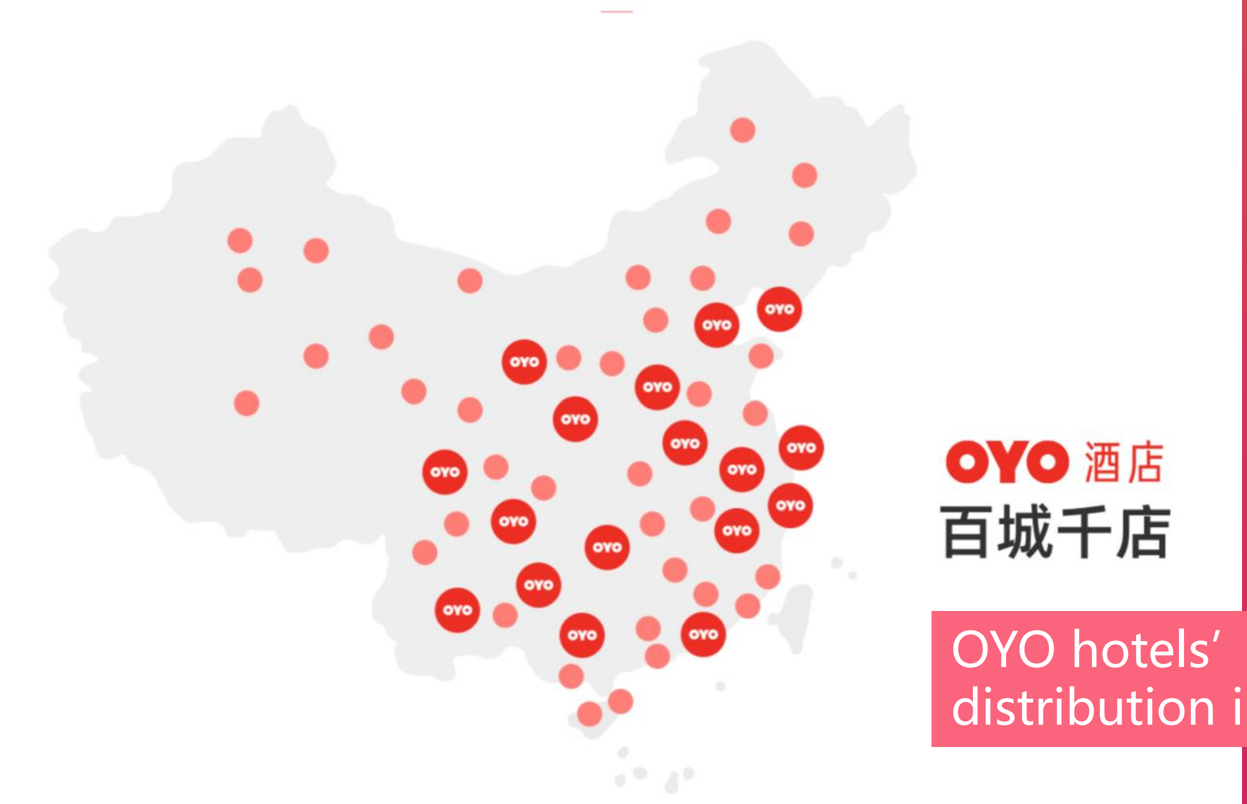
3 Splitting and preprocessing

01 Intro

OYO Intro

OYO Hotels & Homes

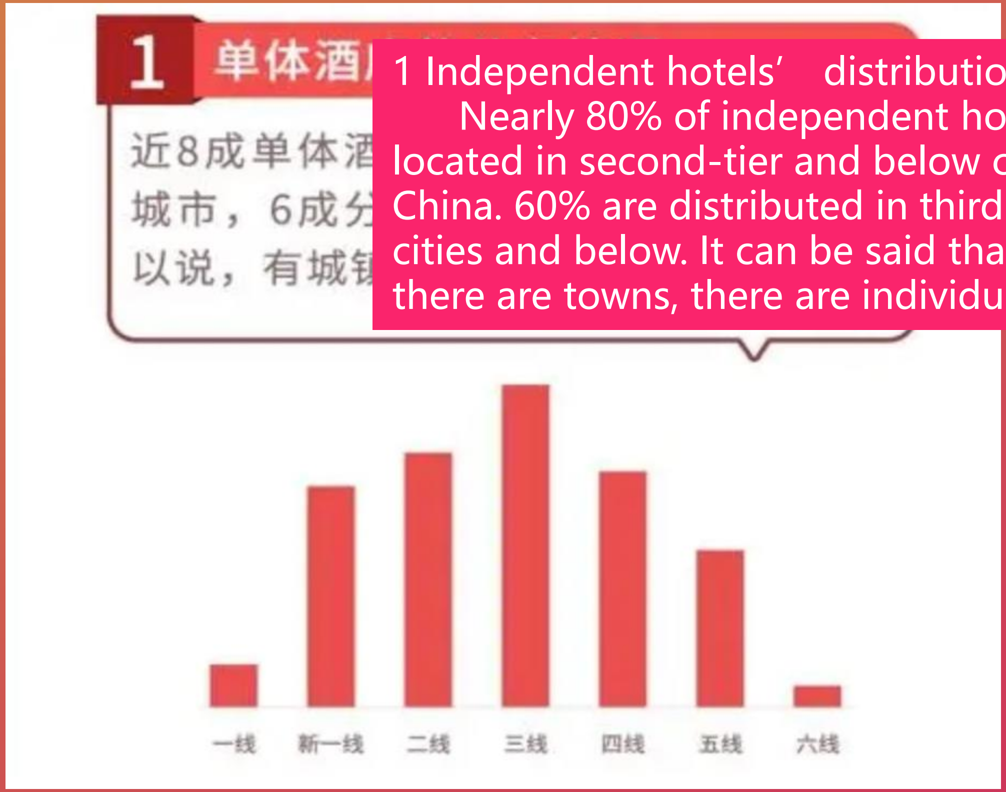
franchise operation



OYO hotels’ distribution in China

01 Intro

Problem +



1 Independent hotels’ distribution
Nearly 80% of independent hotels are located in second-tier and below cities in China. 60% are distributed in third-tier cities and below. It can be said that where there are towns, there are individual hotels.

<<Hate and Love of Chinese Independent Hotel Owners: Big Data Report of Chinese Independent Hotel Owners>>

——OYO Industry Analysis Report

- ➢ Feature matrix shape (5834,25)
- ➢ Target variable shape (5834,)

This report hopes to predict the rental price of OYO hotels according to the property type 、hotel location and so on.

01 Intro

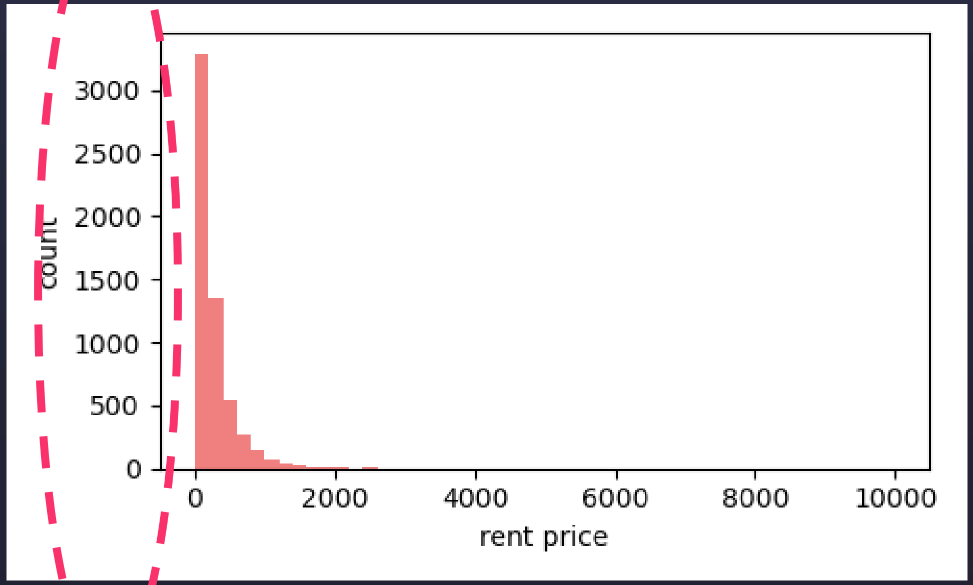
Problem

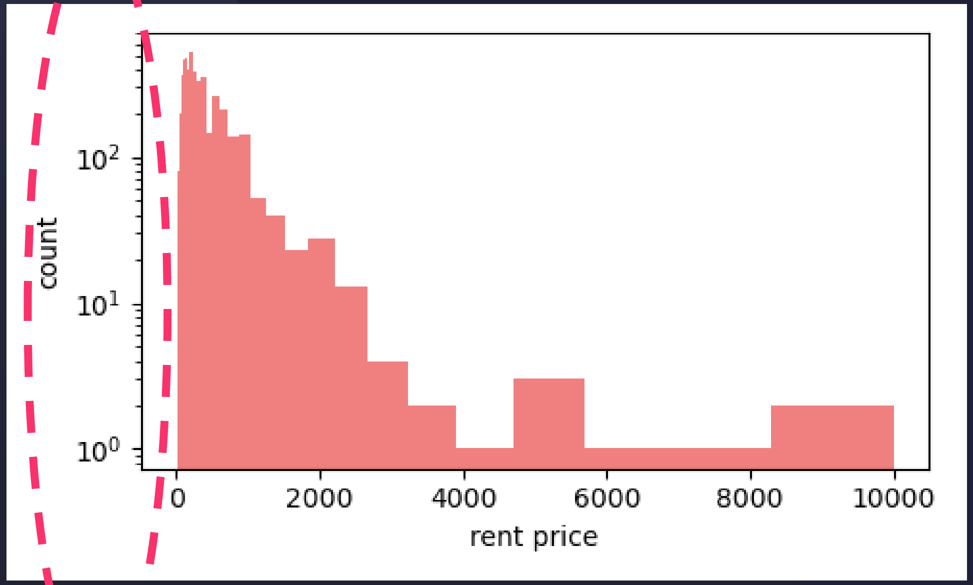
+

Target variable : rental price / $

Regression / Right-skewed

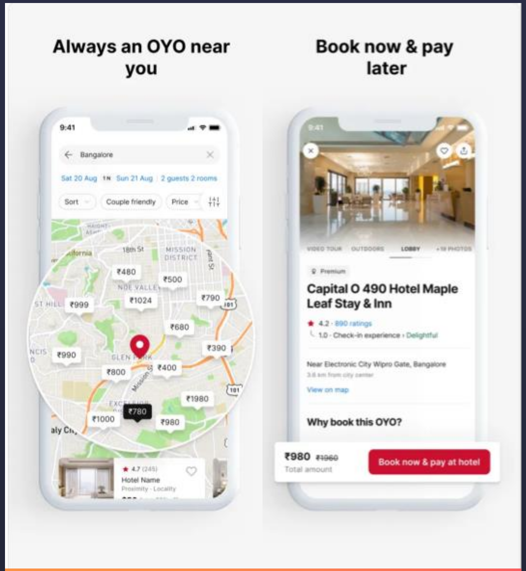
- Enlarge the coordinate axis by log function





01 Intro

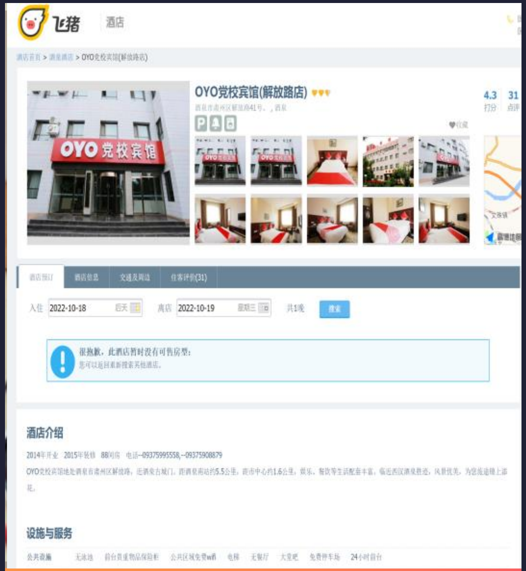
Data From +



OYO APP

XieCheng

MeiTuan



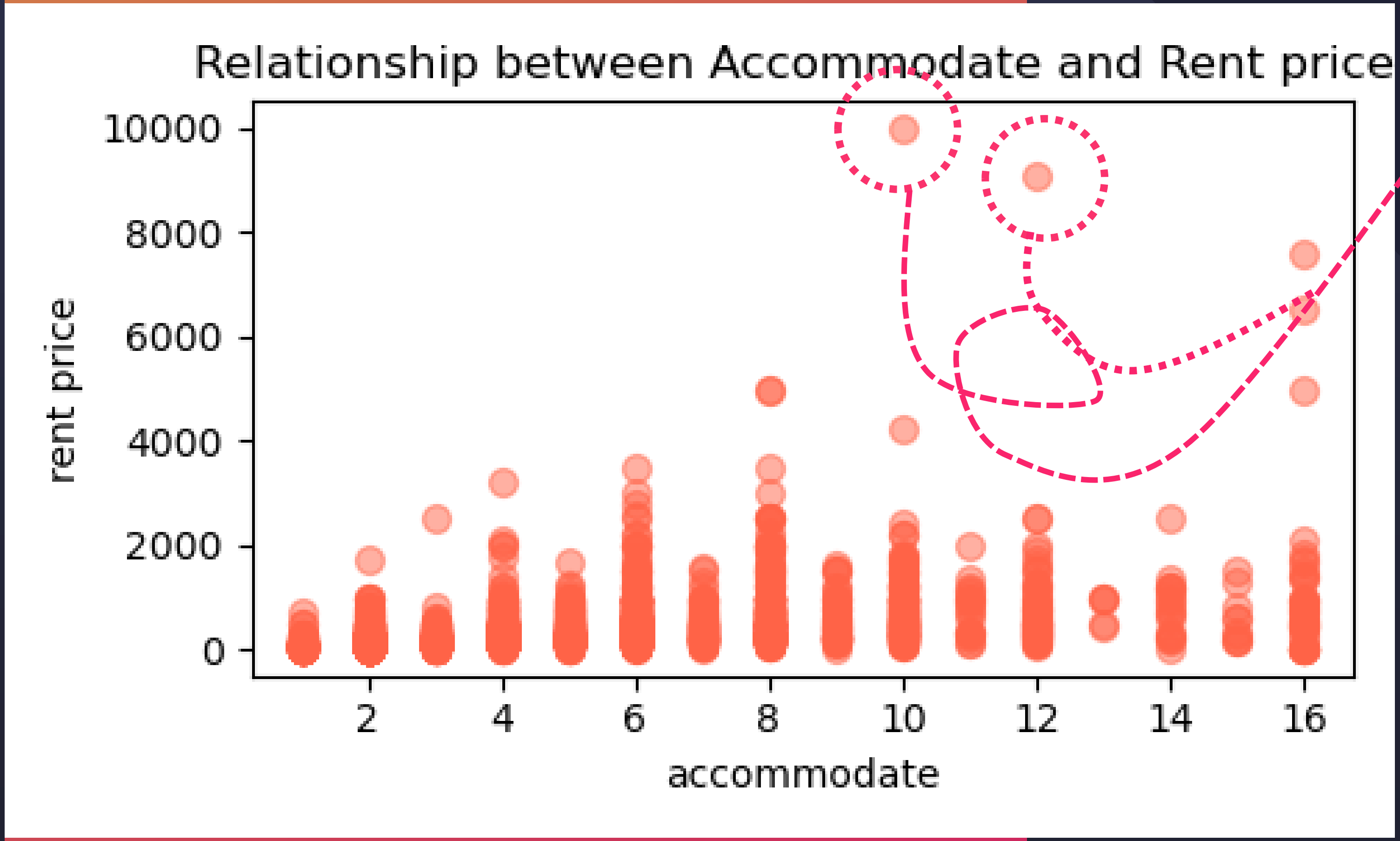
Fei zhu

OYO APP， Chinese OTA platform : XieCheng / MeiTuan/FeiZhu

Kaggle [Oyo Rental Price Prediction in China | Kaggle]

02 EDA Exploratory Data Analysis

Num classification



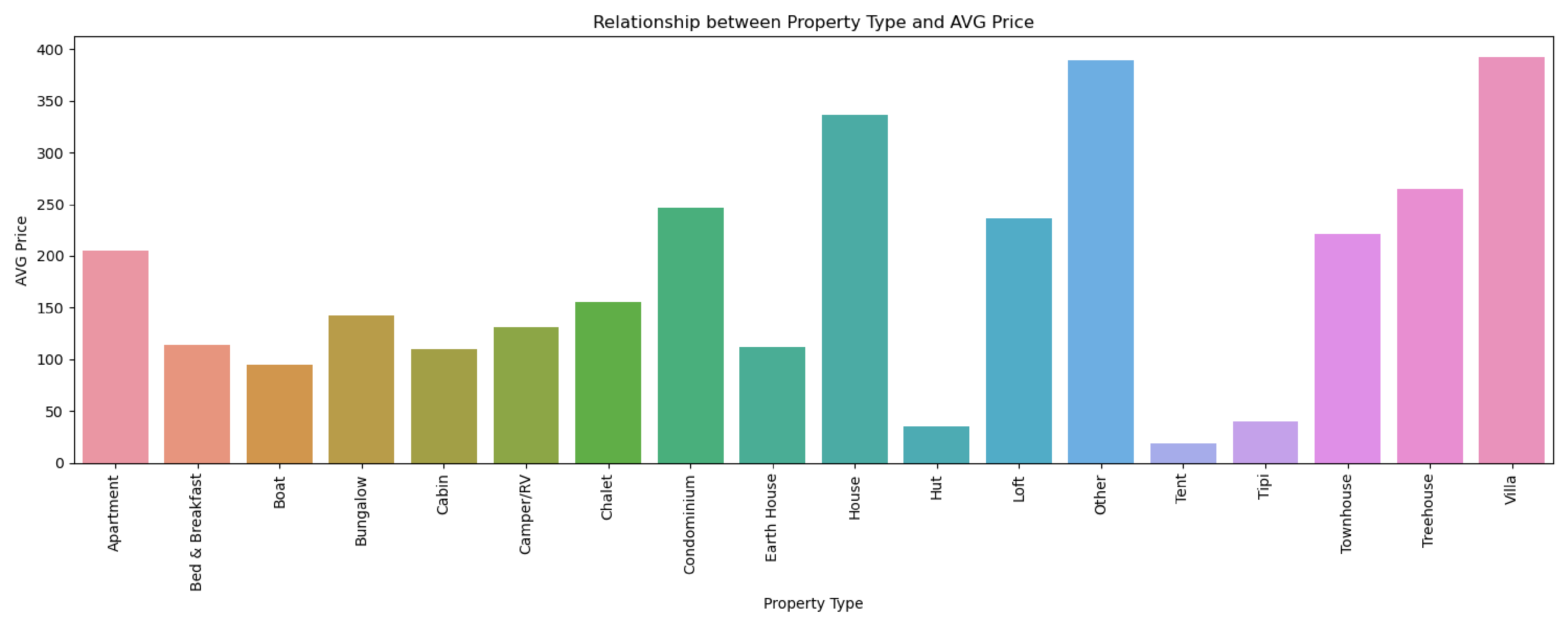
- Special point
- Scatter of Accommodate and Rental price
- Bar plot can't show unusually high prices.

02

EDA Exploratory Data Analysis

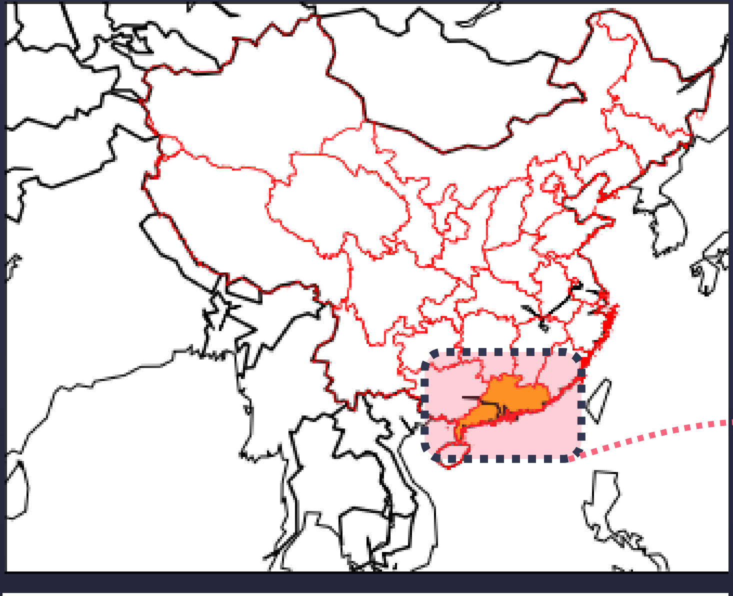
classification

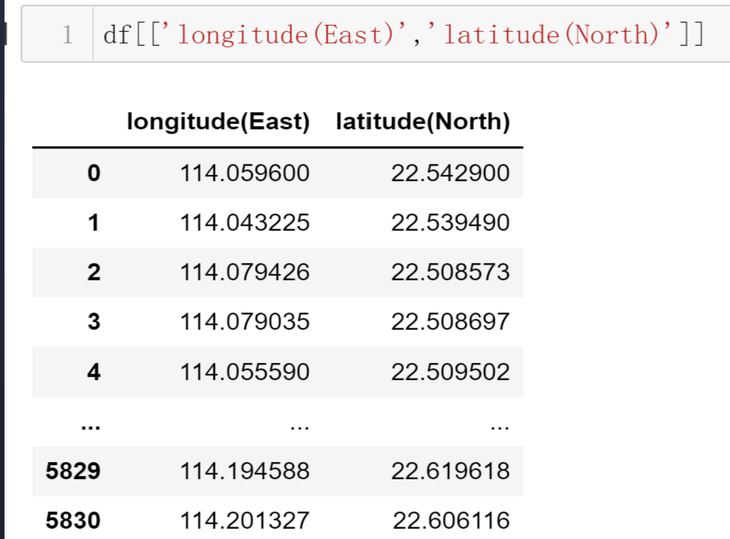
Property Type and AVG Price



02 EDA Exploratory Data Analysis

location

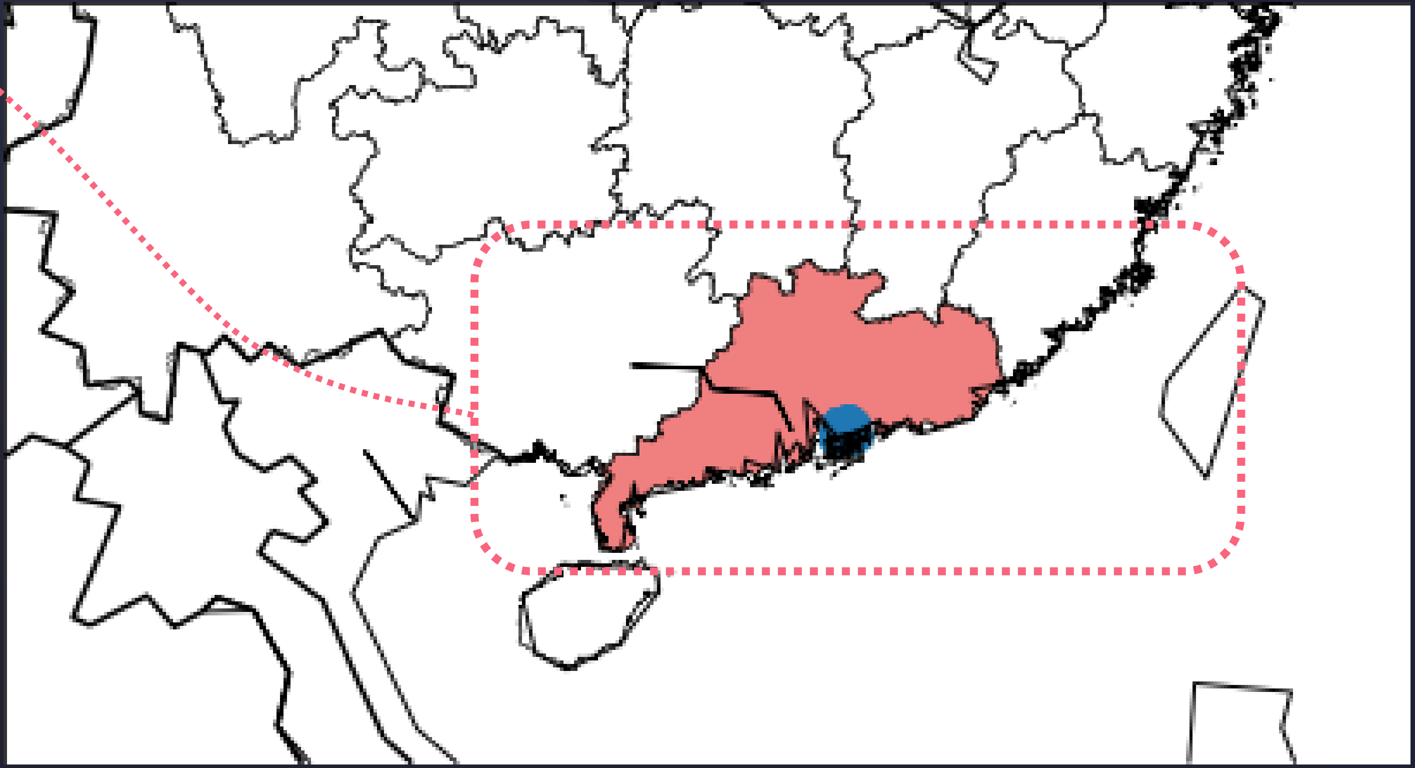




```
df[[’longitude(East)’,’latitude(North)’]]
```

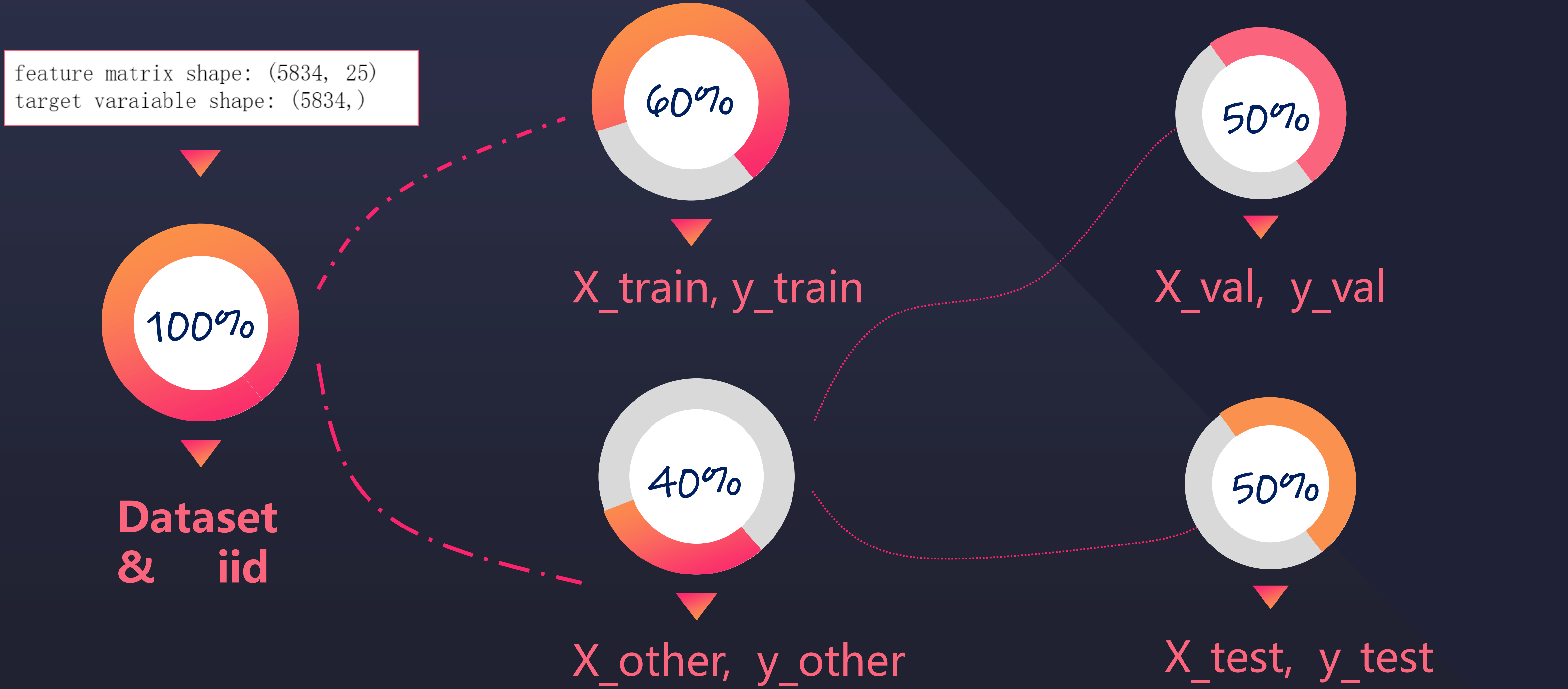
```
      longitude(East)  latitude(North)
   0       114.059600        22.542900
   1       114.043225        22.539490
   2       114.079426        22.508573
   3       114.079035        22.508697
   4       114.055590        22.509502
 ...              ...              ...
5829       114.194588        22.619618
5830       114.201327        22.606116
```

Most hotels are concentrated in the surrounding cities of Shenzhen and Hong Kong, which are near to the seaport and have developed tourism and economic industries.



03 Splitting and preprocessing

splitting +



```
feature matrix shape: (5834, 25)
target varaiable shape: (5834,)
```

Dataset
& iid

X_train, y_train

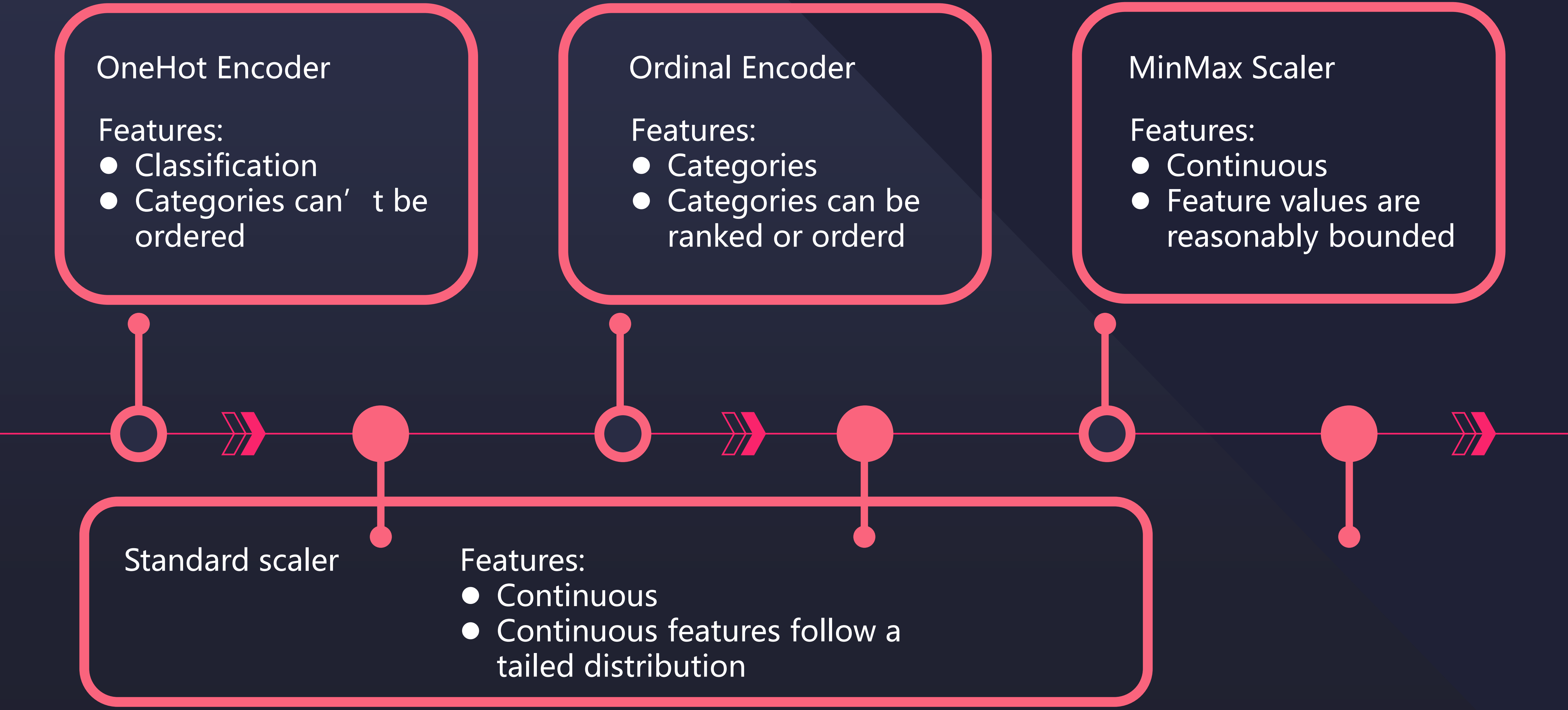
X_other, y_other

X_val, y_val

X_test, y_test

03 Splitting and preprocessing

preprocessing



OneHot Encoder

Features:

- Classification
- Categories can’ t be ordered

Ordinal Encoder

Features:

- Categories
- Categories can be ranked or orderd

MinMax Scaler

Features:

- Continuous
- Feature values are reasonably bounded

Standard scaler

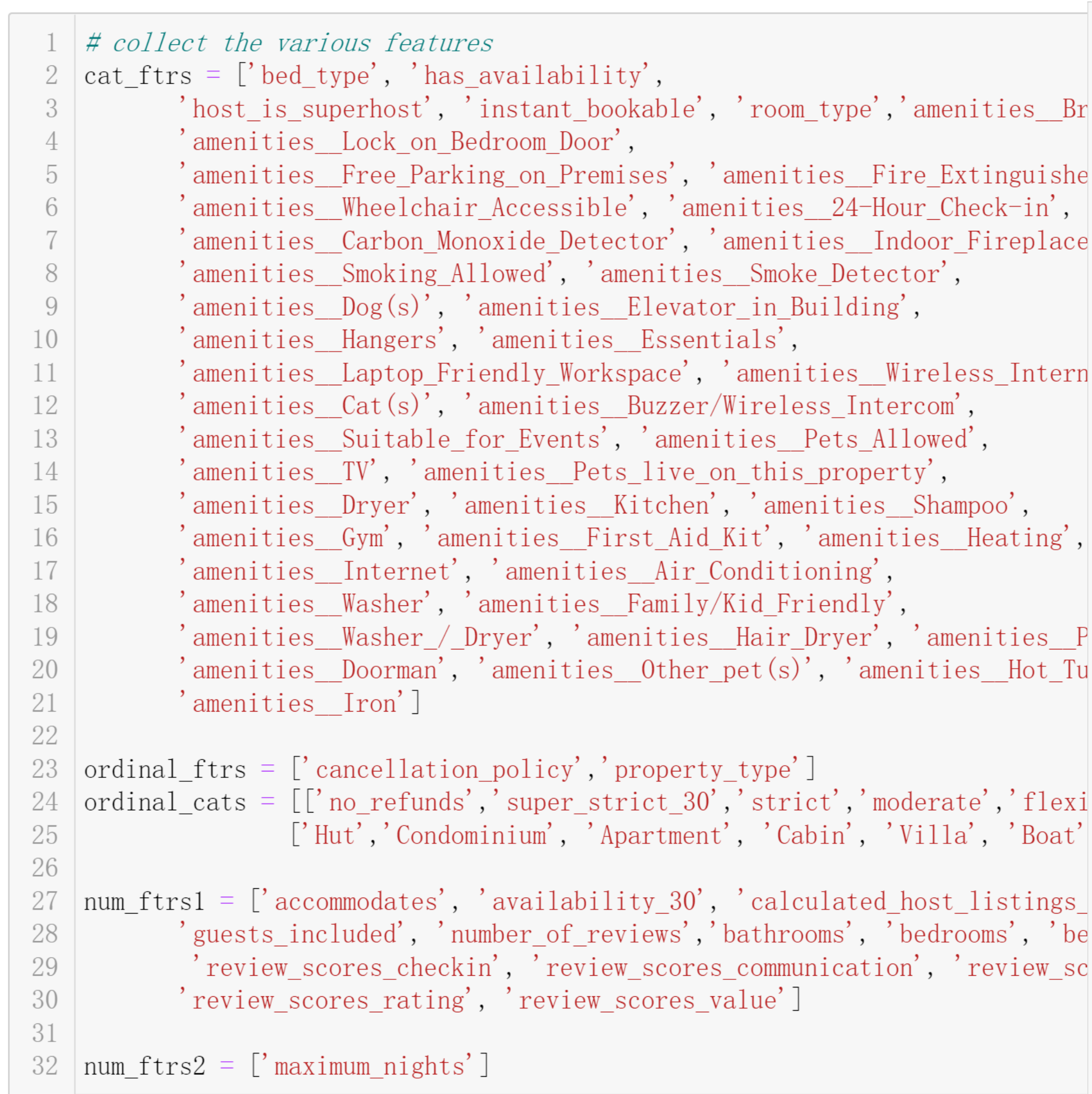
Features:

- Continuous
- Continuous features follow a tailed distribution

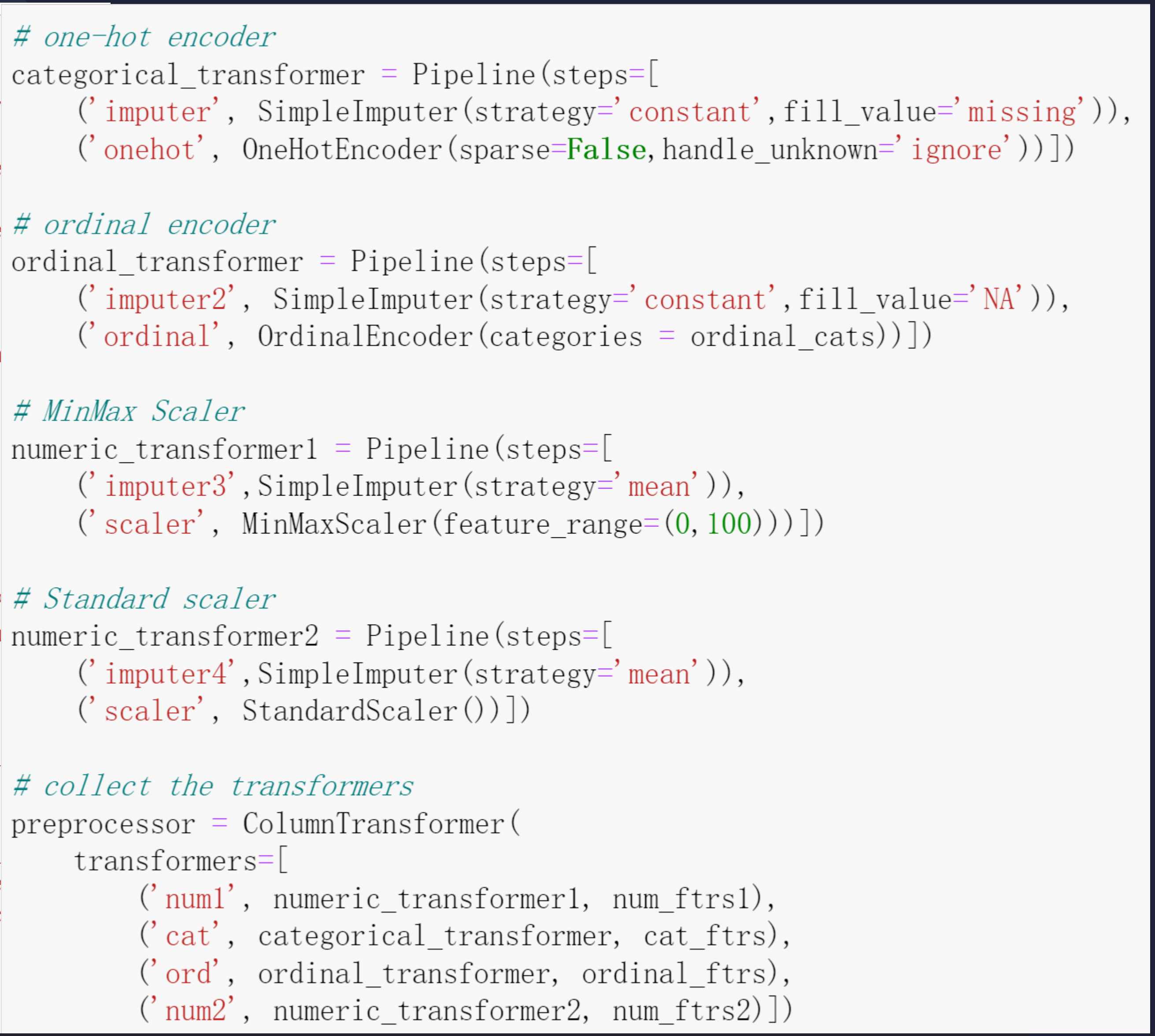
03

Splitting and preprocessing

preprocessing-backup



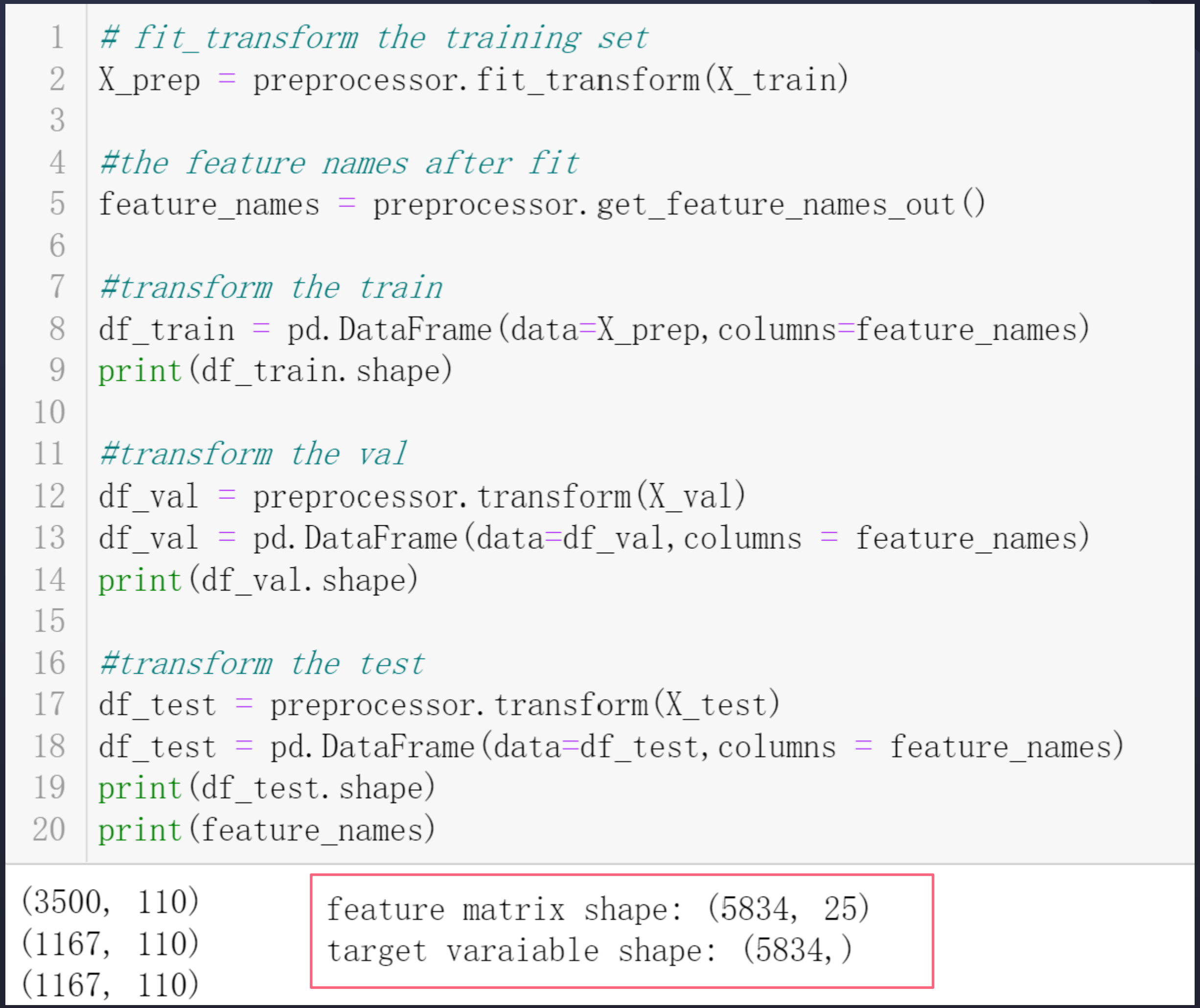
```
# collect the various features
cat_ftrs = ['bed_type', 'has_availability',
        'host_is_superhost', 'instant_bookable', 'room_type','amenities__Br
        'amenities__Lock_on_Bedroom_Door',
        'amenities__Free_Parking_on_Premises', 'amenities__Fire_Extinguishe
        'amenities__Wheelchair_Accessible', 'amenities__24-Hour_Check-in',
        'amenities__Carbon_Monoxide_Detector', 'amenities__Indoor_Fireplace
        'amenities__Smoking_Allowed', 'amenities__Smoke_Detector',
        'amenities__Dog(s)', 'amenities__Elevator_in_Building',
        'amenities__Hangers', 'amenities__Essentials',
        'amenities__Laptop_Friendly_Workspace', 'amenities__Wireless_Intern
        'amenities__Cat(s)', 'amenities__Buzzer/Wireless_Intercom',
        'amenities__Suitable_for_Events', 'amenities__Pets_Allowed',
        'amenities__TV', 'amenities__Pets_live_on_this_property',
        'amenities__Dryer', 'amenities__Kitchen', 'amenities__Shampoo',
        'amenities__Gym', 'amenities__First_Aid_Kit', 'amenities__Heating',
        'amenities__Internet', 'amenities__Air_Conditioning',
        'amenities__Washer', 'amenities__Family/Kid_Friendly',
        'amenities__Washer_/_Dryer', 'amenities__Hair_Dryer', 'amenities__P
        'amenities__Doorman', 'amenities__Other_pet(s)', 'amenities__Hot_Tu
        'amenities__Iron']

ordinal_ftrs = ['cancellation_policy','property_type']
ordinal_cats = [['no_refunds','super_strict_30','strict','moderate','flexi
                ['Hut','Condominium', 'Apartment', 'Cabin', 'Villa', 'Boat'

num_ftrs1 = ['accommodates', 'availability_30', 'calculated_host_listings_
        'guests_included', 'number_of_reviews','bathrooms', 'bedrooms', 'be
         'review_scores_checkin', 'review_scores_communication', 'review_sc
        'review_scores_rating', 'review_scores_value']

num_ftrs2 = ['maximum_nights']
```



```
# one-hot encoder
categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='constant',fill_value='missing')),
    ('onehot', OneHotEncoder(sparse=False,handle_unknown='ignore'))])

# ordinal encoder
ordinal_transformer = Pipeline(steps=[
    ('imputer2', SimpleImputer(strategy='constant',fill_value='NA')),
    ('ordinal', OrdinalEncoder(categories = ordinal_cats))])

# MinMax Scaler
numeric_transformer1 = Pipeline(steps=[
    ('imputer3',SimpleImputer(strategy='mean')),
    ('scaler', MinMaxScaler(feature_range=(0,100)))])

# Standard scaler
numeric_transformer2 = Pipeline(steps=[
    ('imputer4',SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())])

# collect the transformers
preprocessor = ColumnTransformer(
    transformers=[
        ('num1', numeric_transformer1, num_ftrs1),
        ('cat', categorical_transformer, cat_ftrs),
        ('ord', ordinal_transformer, ordinal_ftrs),
        ('num2', numeric_transformer2, num_ftrs2)])
```

03 Splitting and preprocessing

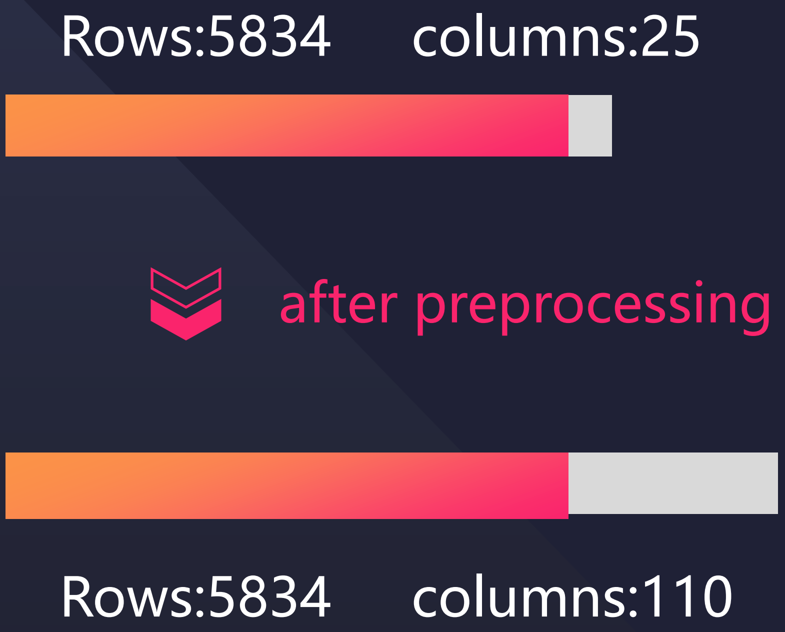
preprocessing +



```
# fit_transform the training set
X_prep = preprocessor.fit_transform(X_train)

#the feature names after fit
feature_names = preprocessor.get_feature_names_out()

#transform the train
df_train = pd.DataFrame(data=X_prep,columns=feature_names)
print(df_train.shape)

#transform the val
df_val = preprocessor.transform(X_val)
df_val = pd.DataFrame(data=df_val,columns = feature_names)
print(df_val.shape)

#transform the test
df_test = preprocessor.transform(X_test)
df_test = pd.DataFrame(data=df_test,columns = feature_names)
print(df_test.shape)
print(feature_names)
```

```
(3500, 110)
(1167, 110)
(1167, 110)
```

```
feature matrix shape: (5834, 25)
target varaiable shape: (5834,)
```



Rows:5834 columns:25

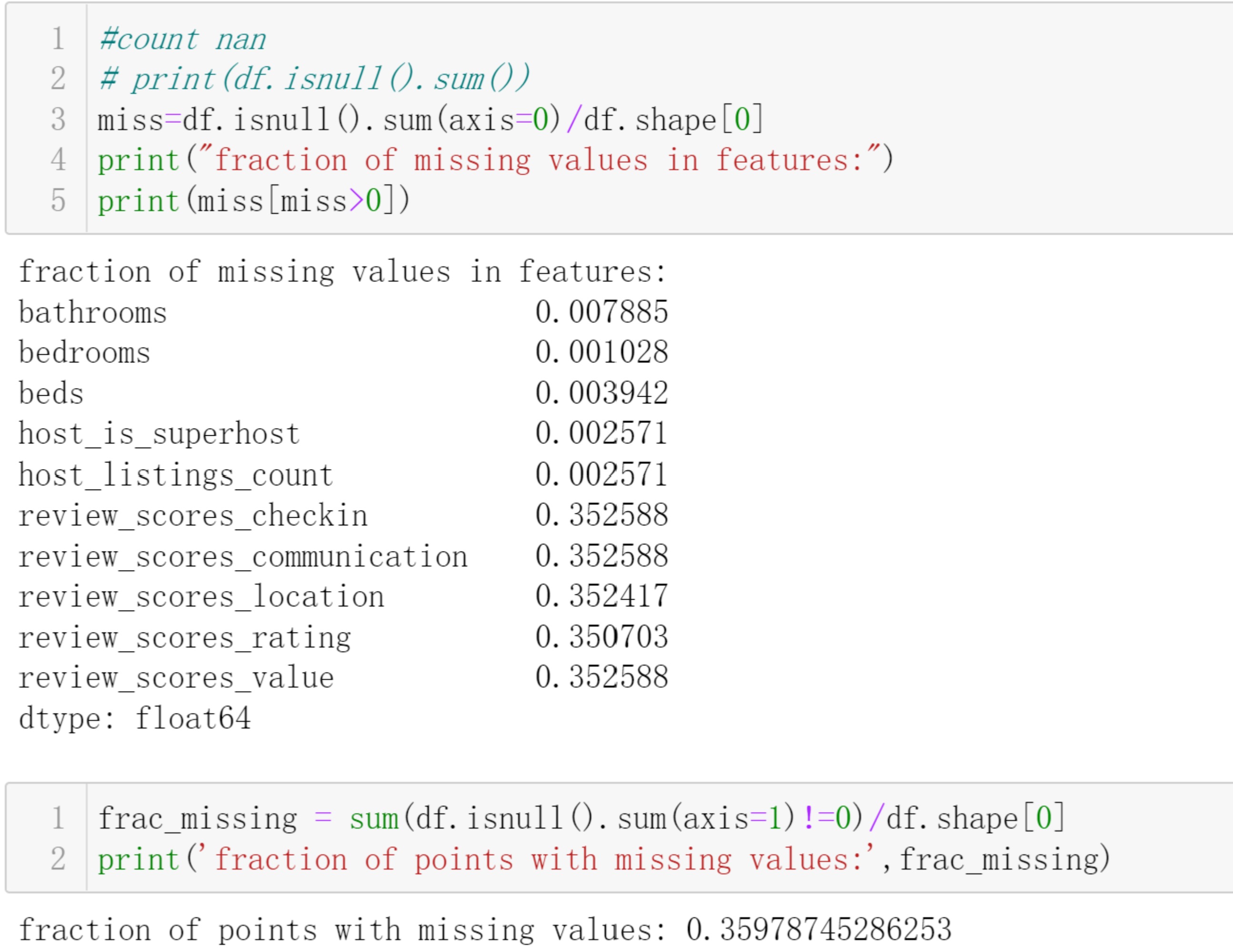
after preprocessing

Rows:5834 columns:110

03 Splitting and preprocessing

missing

+



```
#count nan
# print(df.isnull().sum())
miss=df.isnull().sum(axis=0)/df.shape[0]
print("fraction of missing values in features:")
print(miss[miss>0])
```

```
fraction of missing values in features:
bathrooms                      0.007885
bedrooms                       0.001028
beds                           0.003942
host_is_superhost              0.002571
host_listings_count            0.002571
review_scores_checkin          0.352588
review_scores_communication    0.352588
review_scores_location         0.352417
review_scores_rating           0.350703
review_scores_value            0.352588
dtype: float64
```

```
frac_missing = sum(df.isnull().sum(axis=1)!=0)/df.shape[0]
print('fraction of points with missing values:',frac_missing)
```

```
fraction of points with missing values: 0.35978745286253
```

Fraction of missing values in features:

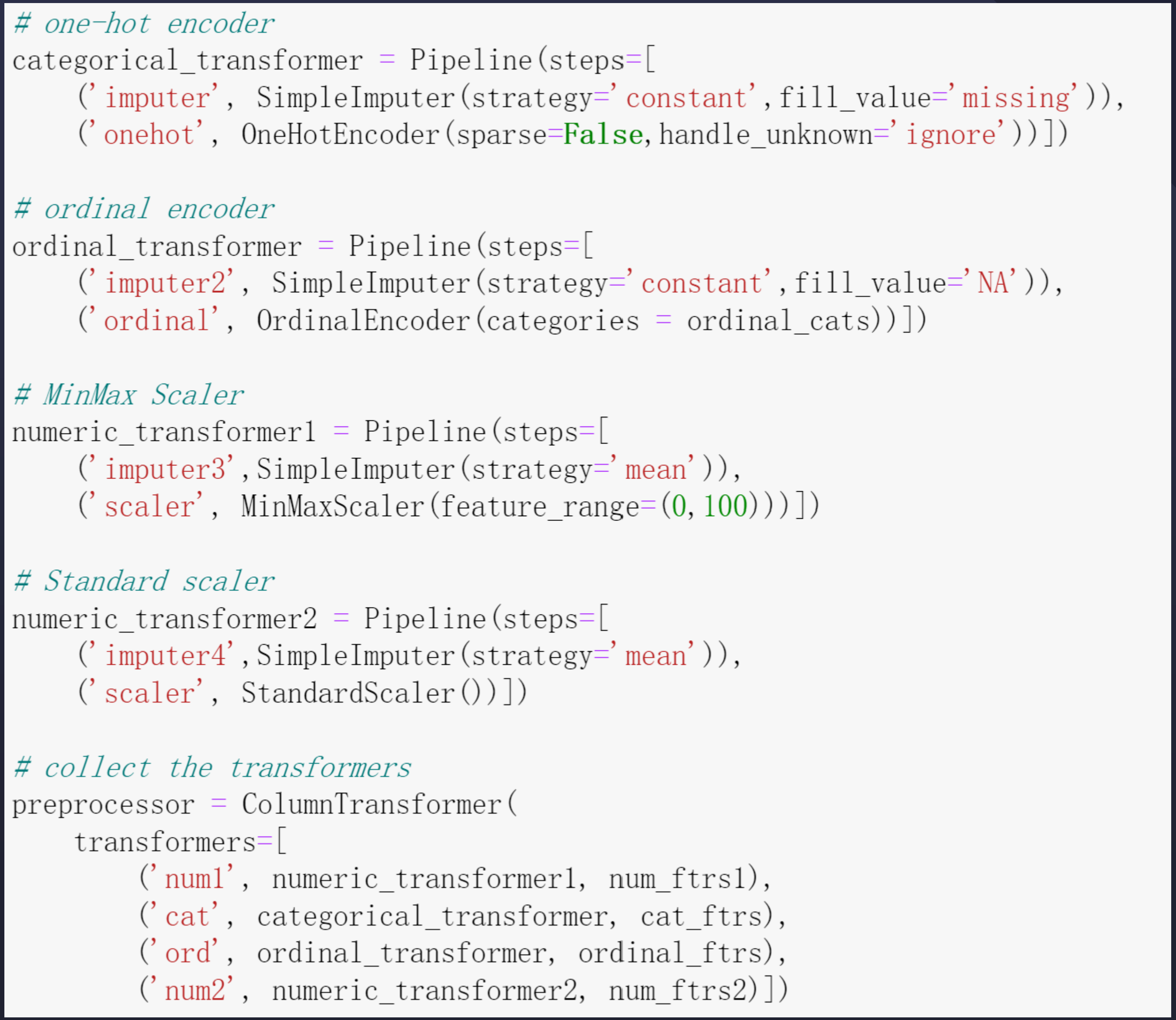
10 features

Fraction of points with missing values:

35.7%

03 Splitting and preprocessing

missing +



```
# one-hot encoder
categorical_transformer = Pipeline(steps=[
    (’imputer’, SimpleImputer(strategy=’constant’,fill_value=’missing’)),
    (’onehot’, OneHotEncoder(sparse=False,handle_unknown=’ignore’))])

# ordinal encoder
ordinal_transformer = Pipeline(steps=[
    (’imputer2’, SimpleImputer(strategy=’constant’,fill_value=’NA’)),
    (’ordinal’, OrdinalEncoder(categories = ordinal_cats))])

# MinMax Scaler
numeric_transformer1 = Pipeline(steps=[
    (’imputer3’,SimpleImputer(strategy=’mean’)),
    (’scaler’, MinMaxScaler(feature_range=(0, 100)))])

# Standard scaler
numeric_transformer2 = Pipeline(steps=[
    (’imputer4’,SimpleImputer(strategy=’mean’)),
    (’scaler’, StandardScaler())])

# collect the transformers
preprocessor = ColumnTransformer(
    transformers=[
        (’num1’, numeric_transformer1, num_ftrs1),
        (’cat’, categorical_transformer, cat_ftrs),
        (’ord’, ordinal_transformer, ordinal_ftrs),
        (’num2’, numeric_transformer2, num_ftrs2)])
```

Missing

NA

Mean

Mean

Data Science Initiative
BROWN

THANK YOU FOR WATCHING