Australian Housing Prices prediction

In [1]:
```
import numpy as np
import pandas as pd
import regex as re
import math

# Data Visualizations:
import matplotlib.pyplot as plt
import seaborn as sns

# Warnings:
import warnings
warnings.filterwarnings('ignore')

# Options:
pd.set_option('display.max_columns' , None)
pd.set_option('display.max_rows', None)

# Sklearn:
from sklearn.preprocessing import OrdinalEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

In [2]:
```
RE_data = pd.read_csv('E:\!DataScience_DSPP_JNTU\Assignments\ML_AL\JNTUH_ML_DL_assignment_2\RealEstateAU_1000_Samples.csv')
```

In [3]:
```
RE_data.head()
```

Out[3]:

	index	TID	breadcrumb	category_name	property_type	building_size	land_size	preferred_size	open_date	listing_agency	price	locatio
0	0	1350988	Buy>NT>DARWIN CITY	Real Estate & Property for sale in DARWIN CITY...	House	NaN	NaN	NaN	Added 2 hours ago	Professionals - DARWIN CITY	$435,000	
1	1	1350989	Buy>NT>DARWIN CITY	Real Estate & Property for sale in DARWIN CITY...	Apartment	171m²	NaN	171m²	Added 7 hours ago	Nick Mousellis Real Estate - Eview Group Member	Offers Over $320,000	
2	2	1350990	Buy>NT>DARWIN CITY	Real Estate & Property for sale in DARWIN CITY...	Unit	NaN	NaN	NaN	Added 22 hours ago	Habitat Real Estate - THE GARDENS	$310,000	
3	3	1350991	Buy>NT>DARWIN CITY	Real Estate & Property for sale in DARWIN CITY...	House	NaN	NaN	NaN	Added yesterday	Ray White - NIGHTCLIFF	$259,000	
4	4	1350992	Buy>NT>DARWIN CITY	Real Estate & Property for sale in DARWIN CITY...	Unit	201m²	NaN	201m²	Added yesterday	Carol Need Real Estate - Fannie Bay	$439,000	

In [4]:
```
RE_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   index            1000 non-null   int64
 1   TID              1000 non-null   int64
 2   breadcrumb       1000 non-null   object
 3   category_name    1000 non-null   object
 4   property_type    1000 non-null   object
 5   building_size    280 non-null    object
 6   land_size        533 non-null    object
 7   preferred_size   609 non-null    object
 8   open_date        302 non-null    object
 9   listing_agency   1000 non-null   object
 10  price            1000 non-null   object
 11  location_number  1000 non-null   int64
 12  location_type    1000 non-null   object
 13  location_name    1000 non-null   object
 14  address          988 non-null    object
 15  address_1        988 non-null    object
 16  city             1000 non-null   object
 17  state            1000 non-null   object
 18  zip_code         1000 non-null   int64
 19  phone            1000 non-null   object
 20  latitude         0 non-null      float64
 21  longitude        0 non-null      float64
 22  product_depth    1000 non-null   object
 23  bedroom_count    967 non-null    float64
 24  bathroom_count   967 non-null    float64
 25  parking_count    967 non-null    float64
 26  RunDate          1000 non-null   object
dtypes: float64(5), int64(4), object(18)
memory usage: 211.1+ KB
```

Data Processing

In [5]:
```
model_df = RE_data.drop(['index','TID','breadcrumb','category_name','open_date','listing_agency','location_number','location_typ
model_df.head()
```

Out[5]:

	property_type	land_size	preferred_size	price	city	product_depth	bedroom_count	bathroom_count	parking_count	RunDate
0	House	NaN	NaN	$435,000	Darwin City	premiere	2.0	1.0	1.0	2022-05-27 15:54:05
1	Apartment	NaN	171m²	Offers Over $320,000	Darwin City	premiere	3.0	2.0	2.0	2022-05-27 15:54:05
2	Unit	NaN	NaN	$310,000	Darwin City	premiere	2.0	1.0	1.0	2022-05-27 15:54:05
3	House	NaN	NaN	$259,000	Darwin City	premiere	1.0	1.0	0.0	2022-05-27 15:54:05
4	Unit	NaN	201m²	$439,000	Darwin City	premiere	3.0	2.0	2.0	2022-05-27 15:54:05

In [6]:
```
model_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   property_type   1000 non-null   object 
 1   land_size       533 non-null    object 
 2   preferred_size  609 non-null    object 
 3   price           1000 non-null   object 
 4   city            1000 non-null   object 
 5   product_depth   1000 non-null   object 
 6   bedroom_count   967 non-null    float64
 7   bathroom_count  967 non-null    float64
 8   parking_count   967 non-null    float64
 9   RunDate         1000 non-null   object 
dtypes: float64(3), object(7)
memory usage: 78.2+ KB
```

In [7]:
```
## Changing 'RunDate' as Index

model_df['RunDate'] = pd.to_datetime(model_df['RunDate'])
model_df.set_index('RunDate'  , inplace = True)
model_df.head()
```

Out[7]:

RunDate	property_type	land_size	preferred_size	price	city	product_depth	bedroom_count	bathroom_count	parking_count
2022-05-27 15:54:05	House	NaN	NaN	$435,000	Darwin City	premiere	2.0	1.0	1.0
2022-05-27 15:54:05	Apartment	NaN	171m²	Offers Over $320,000	Darwin City	premiere	3.0	2.0	2.0
2022-05-27 15:54:05	Unit	NaN	NaN	$310,000	Darwin City	premiere	2.0	1.0	1.0
2022-05-27 15:54:05	House	NaN	NaN	$259,000	Darwin City	premiere	1.0	1.0	0.0
2022-05-27 15:54:05	Unit	NaN	201m²	$439,000	Darwin City	premiere	3.0	2.0	2.0

In [8]:
```
# Identify duplicates:
def duplicated_func(df) :
    print(f"Total number of duplicated rows in data are: {df.duplicated().sum()}")

duplicated_func(model_df)
```

```
Total number of duplicated rows in data are: 108
```

In [9]:
```
#droping duplicate rows:
df = model_df.drop_duplicates(keep='first')

duplicated_func(df)
```

```
Total number of duplicated rows in data are: 0
```

In [10]:
```
# Identify missing values:
def missing_vals(df):
    for i in df :
        if df[i].isnull().sum() > 0 :
            print(f"{i} : {df[i].isnull().sum()} missing values out of {len(df[i])}")
missing_vals(df)
```

```
land_size : 391 missing values out of 892
preferred_size : 331 missing values out of 892
bedroom_count : 30 missing values out of 892
bathroom_count : 30 missing values out of 892
parking_count : 30 missing values out of 892
```

In [11]:
```
def cat_cols(df) :
    o = (df.dtypes == 'object')
    object_cols = o[o].index
    return object_cols

object_cols = cat_cols(df)
print(f"Categorical Columns are : {object_cols}")
```

```
Categorical Columns are : Index(['property_type', 'land_size', 'preferred_size', 'price', 'city',
       'product_depth'],
      dtype='object')
```

Data Analysis

In [12]:
```
df1 = df.copy()
```

In [13]:
```
# Figure out land_size column first
land_size_mode = df1['land_size'].mode()[0]
df1['land_size'].fillna(land_size_mode , inplace = True)
```

In [14]:
```
# Figure out preferred_size column first
preferred_size_mode = df1['preferred_size'].mode()[0]
df1['preferred_size'].fillna(preferred_size_mode , inplace = True)
```

In [15]:
```
# Figure out bedroom_count column first
bedroom_count_mode = df1['bedroom_count'].mode()[0]
df1['bedroom_count'].fillna(bedroom_count_mode , inplace = True)
```

In [16]:
```
# Figure out bathroom_count column first
bathroom_count_mode = df1['bathroom_count'].mode()[0]
df1['bathroom_count'].fillna(bathroom_count_mode , inplace = True)
```

In [17]:
```
# Figure out parking_count column first
parking_count_mode = df1['parking_count'].mode()[0]
```

```
df1['parking_count'].fillna(parking_count_mode , inplace = True)
```

In [18]:
```
missing_vals(df1)
```

In [19]:
```
df1.head()
```

Out[19]:

RunDate	property_type	land_size	preferred_size	price	city	product_depth	bedroom_count	bathroom_count	parking_count
2022-05-27 15:54:05	House	2.02ha	2.02ha	$435,000	Darwin City	premiere	2.0	1.0	1.0
2022-05-27 15:54:05	Apartment	2.02ha	171m²	Offers Over $320,000	Darwin City	premiere	3.0	2.0	2.0
2022-05-27 15:54:05	Unit	2.02ha	2.02ha	$310,000	Darwin City	premiere	2.0	1.0	1.0
2022-05-27 15:54:05	House	2.02ha	2.02ha	$259,000	Darwin City	premiere	1.0	1.0	0.0
2022-05-27 15:54:05	Unit	2.02ha	201m²	$439,000	Darwin City	premiere	3.0	2.0	2.0

In [20]:
```
# Land Size:

def convert_ha_to_sqm_for_land_size(df):
    modified_values = []

    for i in df['land_size']:
        if 'ha' in i:
            ha = float(i.strip('ha'))
            modified_values.append(ha * 10000)

        else:
            sq_num = re.findall(r'\d+',i)
            joined_sq_num = "".join(sq_num)
            modified_values.append(joined_sq_num)


    return modified_values
```

```
df1['land_size'] = convert_ha_to_sqm_for_land_size(df1)
```

In [21]:
```
# Preferred Size:

def convert_ha_to_sqm_for_preferred_size(df):
    modified_values = []

    for i in df['preferred_size']:
        if 'ha' in i:
            ha = float(i.strip('ha'))
            modified_values.append(ha * 10000)

        else:
            sq_num = re.findall(r'\d+',i)
            joined_sq_num = "".join(sq_num)
            modified_values.append(joined_sq_num)


    return modified_values

df1['preferred_size'] = convert_ha_to_sqm_for_preferred_size(df1)
```

In [22]:
```
df1['preferred_size'] = pd.to_numeric(df1['preferred_size'])
df1['land_size'] = pd.to_numeric(df1['land_size'])
```

In [23]:
```
df1.head()
```

Out[23]:

RunDate	property_type	land_size	preferred_size	price	city	product_depth	bedroom_count	bathroom_count	parking_count
2022-05-27 15:54:05	House	20200.0	20200.0	$435,000	Darwin City	premiere	2.0	1.0	1.0
2022-05-27 15:54:05	Apartment	20200.0	171.0	Offers Over $320,000	Darwin City	premiere	3.0	2.0	2.0
2022-05-27 15:54:05	Unit	20200.0	20200.0	$310,000	Darwin City	premiere	2.0	1.0	1.0
2022-05-27 15:54:05	House	20200.0	20200.0	$259,000	Darwin City	premiere	1.0	1.0	0.0
2022-05-27 15:54:05	Unit	20200.0	201.0	$439,000	Darwin City	premiere	3.0	2.0	2.0

In [24]:
```
df1['price'].unique()
```

Out[24]:
```
array(['$435,000', 'Offers Over $320,000', '$310,000', '$259,000',
       '$439,000', '$825,000', '$820,000', '$369 000', '$455,000',
       '$280,000', 'Openn Negotiation', 'PRICE GUIDE $439,000',
       '$775,000', '$625,000', 'Overs Over $599,000 Considered',
       '$490,000', '$337,500', 'FASTRAK', 'UNDER CONTRACT',
       'Offers Over $440,000', '$640,000', '$500,000', '$305,000 +',
       '$295,000 +', '$795,000', 'Offers Over $950,000',
       'Offers Over $485,000', '$250,000', '$549,000',
       'Offers over $299,000', '$395,000', '$475,000', 'Contact Agent',
       'OFFERS INVITED', '$465,000', 'Current Bid $600,000',
       'Offers Over $980,000', '$1,050,000',
       'UNDER CONTRACT... MORE PROPERTIES WANTED', '$450,000', '$749,000',
       '$289,000', 'Offers over $565,000', '$399,000',
       'Offers over $469,000', '$649,000',
       'Offers Over $499,000 Considered', '$400,000',
       'Offers Over $800,000', '$299,000', '$580,000',
       'Offers Over $375,000', '$489,000', 'OFFERS OVER $195,000',
       '$505,000', '$325,000', '$539,000 negotiable', '$180,000',
       '$145,000', '$215,000', '$865,000', '$369,000', '$350,000',
       'Offers over $539,000', '$209,000  Negoticate', '$420,000',
       '$510,000', '$535,000', '$335,000', 'Penthouse for $850,000',
       'Needs to GO! $510,000', '$1,300,000', 'Offers Over $580,000',
       '$470,000', '$519,000', '$219,000', '$235,000 negotiable',
       '$315,000', 'Offers over $450,000', 'Negotiable', '$380,000',
       '$275,000', 'PRICE GUIDE $1,070,000', '$270,000', '$1,100,000',
       'CONTACT AGENT', 'Price Guide $550,000', 'Offers over $868,000',
       '$539,000', 'From $165,000', 'Offers Over: $130,000', '$589,000',
       'Offers over $360,000', 'Offers over $430,000',
       'Offers over $839,000', '$147 000', 'PRICE GUIDE $795,000',
       'Offers Over $430,000', 'Open to offers!',
       '$555,000 Furnished Pool ,Gym LEASED: $640 per week',
       'Offers over $529,000', '$460,000', '$195,000',
       'Offers above $479,000', '$295,000', '$370,000 ono',
       'Motivated Seller- Offers over $475,000', 'Offers over $520,000',
       '$449,000', '$525,000', 'Offers over $500,000',
       'New Price $595,000', 'Offers Over $190,000', 'Under iContract',
       'Negotiable Above 1.5M', '$235,000+', '$389,000.00', '$370,000',
       '$1,950,000', '$175,000', '$520,000 O.N.O', '$329,000',
       'Offers above $559,000', 'Expressions of Interest', '$279,000',
       'From $500,000', '$1,250,000', '$800,000', '$280,000 to$300,000',
       'Invest Now $455,000', '$450,000 ONO', '$685,000', '$429,000',
       '$559,000', '$210,000', '$200,000-$250,000',
       'Offers over $300,000', '$499,000', 'P.O.A.', '$480,000',
       'Awaiting Price Guide', '$440,000', '$200,000 - $220,000',
```

```
'offers over $399,000', '$430,000', '$99,950',
'$900,000 negotiable', 'Price reduction! $895,000!', '$990,000',
'OFFERS OVER $610,000', '$1,199,000',
'Offers Over $665,000 Considered', 'PRICE GUIDE $575,000',
'Bidding from 700k', 'Bidding from $550,000', 'AUCTION',
'Offers over $1,200,000', 'FOR SALE', 'Offers over $365,000',
'OFFERS OVER $700,000', 'Offers Over: 565000', '$419,000',
'OFFERS OVER $330,000', '$560,000', '$460,000 +',
'PRICE GUIDE $290,000', 'PRICE GUIDE $570,000', '$569,000',
'$739,000', 'Offers Over $399,000', 'Auction', '$599,000',
'$660,000', 'Offers over $800,000', '$880,000',
'PRICE GUIDE $540,000', 'Offers Over $399,000 Considered',
'OFFERS OVER $370,000',
'Offers Over $500,000 - Offers by 6.30pm 22/6/22', 'OVER $450,000',
'Offers Over $470,000', '$612,300', '$592,000', '$614,000',
'$218,000', '$255,000', '$425,000', '$779,000', '$850,000',
'Offers under $340,000', '$570,000', 'Offers Over $285,000',
'$470,000 Negotiable', 'Under Offer', '$365,000', '$420,000 +',
'Price guide-$275,000-$325,000', '$269,000', '$750,000',
'Offers above $285,000', 'Expression of Interest', 'UNDER OFFER',
'$579,000', 'Expressions Of Interest', 'PRICE GUIDE $530,000',
'JUST LIKE THAT: UNDER CONTRACT IN 5 DAYS', 'PRICE GUIDE $750,000',
'$535,000 +', '$629,000', '$275,000 +', 'PROPERTY PREVIEW',
'Offers Over $550,000', '$639,000', 'Offers over $650,000',
'Under Contract', '$385,000', '$550,000', 'Offers over $549,000',
'Offer Over $300,000', '$387,000', '$590,000',
'Offers over $489,000', 'Offers over $569,000',
'Offers Over $435,000 Considered', '$1,650,000', '$293,000',
'$495,000', '$875,000 negotiable', 'Auction Action', '$240,000',
'Price Guide $570,000', 'Offers over $685,000',
'Offers over $635,000', 'Auction - Bidding from $700,000',
'AUCTION: Saturday 4th Jun @11am On-Site', '$529,000',
'Under  Contract', 'Current Bid $650,000', '$659,000',
'Price Guide High $500,000', 'Offers over $405,000',
'PRICE GUIDE $545,000',
'Auction - Wednesday 15th June 2022 at 5.30pm',
'OFFERS OVER $690,000', 'New to Market', 'PRICE GUIDE $1,200,000',
'PRICE GUIDE $510,000', 'Offers Over $750,000',
'PRICE GUIDE $490,000', 'Auction on site',
'Auction 8th June on site', '$320,000', 'Current Bid $490,000',
'$665,000', 'AUCTION - Bids from $630K', '$290,000',
'PRICE GUIDE $298,000', 'PRICE GUIDE $690,000',
'PRICE GUIDE $590,000', 'Current Bid - $510,000',
'Offers above $420,000', '$695,000', 'Offers over $285,000',
```

```
'Offers over $660,000 considered', '$685,000 ONO', '$950,000',
'$498,000', 'Current Bid $740,000',
'Price Guide: $700,000 - $720,000', '$485,000', '$665,000 ONO',
'Auction ON SITE', '$339,000', 'Offers above $399,000', '$735,000',
'$975,000', 'Auction Wednesday 1st of June 2022',
'Offers over $725,000', '$610,000', 'Current Bid $500,000',
'PRICE GUIDE $480,000', 'Price Guide: $520,000 - $550,000',
'Offers Over $620,000', '$1,490,000', '$292,000', '$520,000',
'Offers over 295,000', 'Offers over $820,000',
'Offers over $460,000', '$220,000', '$575,000', '$349,000',
'$389000', 'JUST LIKE THAT: UNDER CONTRACT PRIOR TO AUCTION',
'OFFERS OVER $760,000', '$630,000', '$300,000',
'Offers Over $300,000', 'Offers over $220,000',
'Offers over $799,000', 'PRICE GUIDE $525,000', '$860,000',
'Accepting Offers Between $275,000 - $295,000',
'Offers over $675,000', 'offers over $1,250,000', '$1,200,000',
'Offers over $850,000', 'PRICE GUIDE $735,000', '$355,700',
'$359,000', '$370,990', 'Offers over $700,000',
'$729,000 DHA lease $934 pwk', '$620,000',
'PRICE GUIDE $1,100,000', '$595,000', 'Offers Over $859,000',
'OFFERS OVER $800,000', '$450 000', "Offers from mid $800's",
'Mid $500,000 ono', 'Offers Over $600,000', 'Offers over $550,000',
'PRICE GUIDE $209,000', 'Offers over $370,000',
'Offers Over $290,000', 'Offers over $630,000', '$316,000',
'$115,000', '$1,400,000', '$418,000', 'OFFERS OVER $780,000',
'$460,000 ONO', '$410,000', '$819,000', 'REDUCED to SELL!!!',
'OFFERS OVER $695k', '$339,000 +', 'NEW PRICE $430,000',
'$540,000', 'PRICE GUIDE $830,000',
'JUST LIKE THAT: UNDER CONTRACT', '$375,000', '$515,000',
'Offers over $345,000', 'OFFERS OVER $505,000',
'$670,000 Negotiable', '$225,000', '$548,000',
'OFFERS OVER 399,000', 'PRICE GUIDE $445,000',
'$950000 or nearest Offer', 'PRICE GUIDE $646,000',
'$510,000 each unit', 'From $220,000', '$390,000',
'PRICE GUIDE $495,000', '$555,000', '$459,000',
'NEW PRICE $279,000', 'OFFERS OVER $530,000',
'Offers Over $530,000', '$429000', '$389,000', '$589,500',
'$165,000', '$355,000', 'Offers over $399,000',
'PRICE GUIDE $330,000', 'Offers invited $440,000',
'Price Guide: $650,000 - $700,000', '$479,000',
'Offers Over $775,000', 'PRICE GUIDE $329,000',
'Offers over $1.2m', '$899,000', 'Offers above $799,000',
'JUST LIKE THAT: UNDER CONTRACT IN 7 DAYS', 'NEW PRICE $639,000',
'$499,000 O.N.O', '$1,800,000', 'Offers welcome $695K',
```

```
'OVER $900,000', '$345,000 ONO', 'Offers above $859,000',
'Offers over $740,000', '$600,000', 'Offers Over $700,000',
'Offers Over $599,000', '$720,000', 'Range $600,000 - $650,000',
'Reduced Offers Over $330,000', 'Offers over $400,000',
'Offers Over $849,000', 'Offers over $679,000 considered',
'$560,000 +', '$319,000', 'OFFERS OVER $750,000', '$334,000',
'$460k NEGOTIABLE', '$545,000 ( over 1000sqm of land)',
'PRICE GUIDE $1,090,000', 'Offers Over $620,000 Considered',
'PRICE GUIDE $390,000', 'Overs over $320,000', '$895,000',
'PRICED TO SELL $719,000!', '$149,000', '$789,000', '$457,000',
'offers above $510,000', 'Open to Offers around $499,000',
'$650,000 +', '$830 000', 'Offers over $380,000', '$635,000 +',
'Offers Over $199,000', '$289,000 obo', '$499,000 +',
'Offers Over: $499,000', 'Offers over $480,000',
'Offers Over $339,000 Considered', 'Offers Over $340,000',
'OFFERS OVER $490,000', 'Offers Over $500,000 Considered',
'$462,500', 'All Reasonable Offers Considered',
'OFFERS OVER $550,000', '$740,000', 'PRICE GUIDE $895,000',
'Offers Over $255,000', 'High $400,000s', 'Offers Over $449,000',
'$780,000', '$260,000', '$525,000 - $535,000',
'Offers Over $765,000 Considered', '$849,000', '$770,000',
'Offers over $730,000', 'Offers Over $575,000',
'Offers Over $350,000', 'Offers Over $500,000 will be considered',
'Offers Over $780,000 Invited', 'O/O $750,000',
'Offers Over $295,000', 'From $370,000',
'MASSIVE PRICE REDUCTION!! NOW $277,500', 'Under contract',
'$650,000', 'Low - Mid $200,000', 'Offers Over $500,000',
'$570000 neg', 'High $500,000 ono', '$345,000 Negotiable',
'PRICE REDUCED - $475,000', '$360,000',
'$1,100,000 Duplex Investment', '$409,000', '$470,000 ONO',
'$283,000', 'Offers Over $699,000', '$390,000+', '$615,000',
'$645,000', '$675,000', 'OFFERS OVER $500,000',
'Offers Over $299,000', 'PRICE GUIDE $260,000', '$545,000',
'Offers over $385,000', 'Price Guide Low $700,000',
'OFFERS OVER $1,200,000', 'Offers invited $539,000',
'Offers over $600,000 Considered', 'Offers Over $499,000',
'$1.15m', '$699,000', 'PRICE GUIDE $395,000', 'PRICE DROP $645k',
'$3,990,000', 'Offers over $419,000', '$244,000',
'Offers Over $480,000', '$585,000 & $595,000', '$1,550,000',
'Offers Over $630,000', 'Price Guide Mid $300,000', '$469,000',
'$299,000 Negotiable', 'OFFERS OVER $415,000', '2 Residence',
'$601,000', '$655,000'], dtype=object)
```

In [25]:
```
price_list = []

def price(df):
    for i in df['price'] :
        price_num = re.findall(r'\d+', i)
        joined_price = "".join(price_num)
        if joined_price != '':
            price_list.append(joined_price)
        else :
            price_list.append(np.nan)

    return price_list


df1['price'] = price(df1)
```

In [26]:
```
df1['price'] = pd.to_numeric(df1['price'])
```

In [27]:
```
df1['price'].isnull().sum()
```

Out[27]:
```
196
```

In [28]:
```
price_mean = df1['price'].mean()
price_median = df1['price'].median()
#df1['price'] = df1['price'].fillna(price_mean)
df1['price'] = df1['price'].fillna(price_median)
```

In [29]:
```
object_cols = cat_cols(df1)
print(f"Categorical Columns are : {object_cols}")
```

```
Categorical Columns are : Index(['property_type', 'city', 'product_depth'], dtype='object')
```

In [30]:
```
df1[object_cols].head()
```

Out[30]:

RunDate	property_type	city	product_depth
2022-05-27 15:54:05	House	Darwin City	premiere
2022-05-27 15:54:05	Apartment	Darwin City	premiere
2022-05-27 15:54:05	Unit	Darwin City	premiere
2022-05-27 15:54:05	House	Darwin City	premiere
2022-05-27 15:54:05	Unit	Darwin City	premiere

In [31]:
```
def unique_vals_of_cat_cols(df) :
    o = (df.dtypes == 'object')
    object_cols = o[o].index

    for i in object_cols :
        print(f"{i} : {df[i].unique()}")

unique_vals_of_cat_cols(df1)
```

```
property_type : ['House' 'Apartment' 'Unit' 'Studio' 'Residential Land' 'Block Of Units'
 'Townhouse' 'Acreage' 'Duplex/Semi-detached' 'Other' 'Villa' 'Warehouse'
 'Lifestyle']
city : ['Darwin City' 'Leanyer' 'Stuart Park' 'Lyons' 'Durack' 'The Narrows'
 'Herbert' 'Nightcliff' 'Rapid Creek' 'Woodroffe' 'Driver' 'Humpty Doo'
 'Fannie Bay' 'Muirhead' 'Bellamack' 'Wanguri' 'Bakewell' 'Karama'
 'Coconut Grove' 'Jingili' 'Gunn' 'Moulden' 'Parap' 'Coolalinga' 'Marrara'
 'Woolner' 'Zuccoli' 'Rosebery' 'Anula' 'Virginia' 'Gray' 'Wagaman'
 'Farrar' 'Tiwi' 'Berry Springs' 'Malak' 'Bayview' 'Wulagi' 'Millner'
 'Larrakeyah' 'Howard Springs' 'Lee Point' 'Alawa' 'Johnston' 'Ludmilla'
 'Girraween' 'The Gardens' 'Bees Creek' 'Brinkin' 'Moil' 'Berrimah'
 'Knuckey Lagoon' 'Cullen Bay' 'Nakara' 'Rosebery Heights' 'Marlow Lagoon']
product_depth : ['premiere' 'midtier' 'feature' 'standard']
```

In [32]:
```
df2 = df1.copy()
```

In [33]:
```
ordinal_enc = OrdinalEncoder()
df2[object_cols] = ordinal_enc.fit_transform(df2[object_cols])
```

In [34]:
```
df2.head()
```

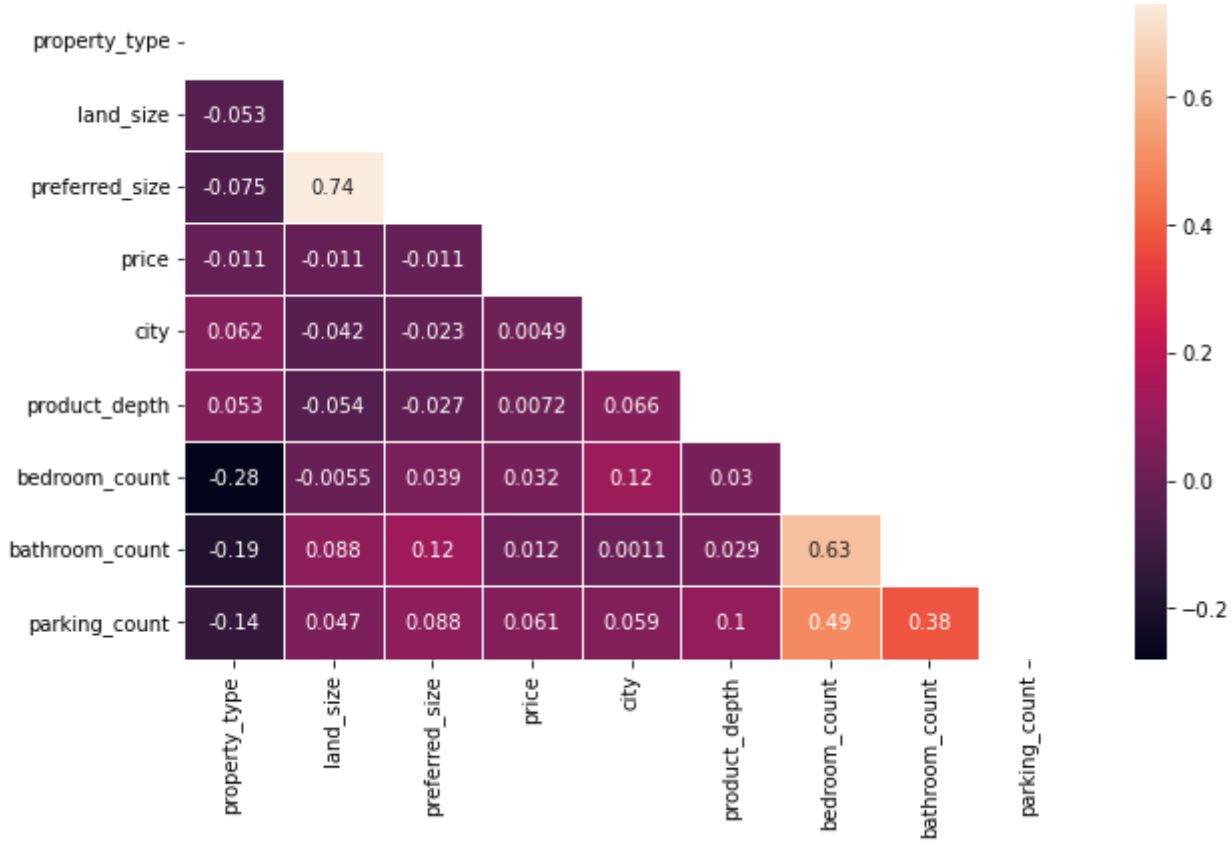
Out[34]:

RunDate	property_type	land_size	preferred_size	price	city	product_depth	bedroom_count	bathroom_count	parking_count
2022-05-27 15:54:05	4.0	20200.0	20200.0	435000.0	12.0	2.0	2.0	1.0	1.0
2022-05-27 15:54:05	1.0	20200.0	171.0	320000.0	12.0	2.0	3.0	2.0	2.0
2022-05-27 15:54:05	10.0	20200.0	20200.0	310000.0	12.0	2.0	2.0	1.0	1.0
2022-05-27 15:54:05	4.0	20200.0	20200.0	259000.0	12.0	2.0	1.0	1.0	0.0
2022-05-27 15:54:05	10.0	20200.0	201.0	439000.0	12.0	2.0	3.0	2.0	2.0

In [35]:
```
def corr(df) :
    correlation  = df.corr()
    mask = np.triu(np.ones_like(correlation , dtype = bool))

    plt.figure(figsize = (10,6))
    sns.heatmap(correlation , mask = mask , annot = True , linewidth = 1)
```

In [36]:
```
corr(df2)
```



In [37]:
```
features = df2.drop('price',axis = 1)
label = df2['price']
```

In [38]:
```
def splitted_data(features , label) :
    x_train,x_test,y_train,y_test = train_test_split(features, label , test_size = 0.3)

    print(f"Shape of x_train : {x_train.shape}")
    print(f"Shape of y_train : {y_train.shape}")
    print(f"Shape of x_test : {x_test.shape}")
    print(f"Shape of y_test: {y_test.shape}")
    return x_train,x_test,y_train,y_test

x_train,x_test,y_train,y_test = splitted_data(features , label)
```

```
Shape of x_train : (624, 8)
Shape of y_train : (624,)
Shape of x_test : (268, 8)
Shape of y_test: (268,)
```

In [39]:
```
pred_dict = {}

def model(modelname , x_train  , x_test , y_train , y_test) :

    model_build = modelname()
    model_build.fit(x_train,y_train)

    y_pred = model_build.predict(x_test)

    return y_pred

y_pred = model(LinearRegression, x_train,x_test,y_train,y_test)
pred_dict['linear_y_pred'] = y_pred
```

In [40]:
```
campare_df = pd.DataFrame()

campare_df['Y_TEST'] = y_test
campare_df['Y_PRED'] = y_pred

campare_df.head()
```

Out[40]:

RunDate	Y_TEST	Y_PRED
2022-05-27 15:54:05	489000.0	-4.344988e+10
2022-05-27 15:54:05	489000.0	7.827980e+10
2022-05-27 15:54:05	279000.0	3.551827e+10
2022-05-27 15:54:05	425000.0	4.718577e+10
2022-05-27 15:54:05	505000.0	3.931133e+10

In [41]:
```
def rmse_func(y_pred,y_test) :
    rmse = math.sqrt(mean_squared_error(y_test,y_pred))

    return rmse
```

```
rmse_func(y_pred,y_test)
```

Out[41]: 169092440760.59537

Conclusion

Linear Regression performed bad result, cause of missing values in the label column; and most of the values are outliers. If more data is fed to the training then may be chances are model will perform better.