Data Cleaning and Feature Extraction

In this section, I cleared outliars, handled missing data and get dummy variables of categorical data.

Notebook Structure

- **Exploring Data**
- **Clearing Outliers**
- **Handling Missing Data**
  - **Explore Missing Data**
  - **Filling Null values which has meaning**
  - **Filling with Mean**
  - **Dropping Some Missing Rows**
  - **Dropping Columns**
  - **Filling Missing with Feature Extraction**
- **Encoding Data**

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path
from src.utils import html_table

raw_data_dir = Path('../data/raw/')
interim_data_dir = Path('../data/interim/')
processed_data_dir = Path('../data/processed/')
file_name = 'train.csv'
file_path = raw_data_dir / file_name
```

Exploring Data

In this section, I inspect variables and check the most correlated continuous variables on the variable which will be predicted(DV). Our outcome variable is 'SalePrice'. We will basically try to estimate house prices.

In [2]:
```
df = pd.read_csv(file_path)
html_table(df.head())
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1460 non-null   int64
 1   MSSubClass     1460 non-null   int64
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          91 non-null     object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
 10  LotConfig      1460 non-null   object
 11  LandSlope      1460 non-null   object
 12  Neighborhood   1460 non-null   object
 13  Condition1     1460 non-null   object
 14  Condition2     1460 non-null   object
 15  BldgType       1460 non-null   object
 16  HouseStyle     1460 non-null   object
 17  OverallQual    1460 non-null   int64
 18  OverallCond    1460 non-null   int64
 19  YearBuilt      1460 non-null   int64
 20  YearRemodAdd   1460 non-null   int64
 21  RoofStyle      1460 non-null   object
 22  RoofMatl       1460 non-null   object
 23  Exterior1st    1460 non-null   object
 24  Exterior2nd    1460 non-null   object
 25  MasVnrType     1452 non-null   object
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object
 28  ExterCond      1460 non-null   object
 29  Foundation     1460 non-null   object
 30  BsmtQual       1423 non-null   object
 31  BsmtCond       1423 non-null   object
 32  BsmtExposure   1422 non-null   object
 33  BsmtFinType1   1423 non-null   object
 34  BsmtFinSF1     1460 non-null   int64
 35  BsmtFinType2   1422 non-null   object
 36  BsmtFinSF2     1460 non-null   int64
 37  BsmtUnfSF      1460 non-null   int64
 38  TotalBsmtSF    1460 non-null   int64
 39  Heating        1460 non-null   object
 40  HeatingQC      1460 non-null   object
 41  CentralAir     1460 non-null   object
 42  Electrical     1459 non-null   object
 43  1stFlrSF       1460 non-null   int64
 44  2ndFlrSF       1460 non-null   int64
 45  LowQualFinSF   1460 non-null   int64
 46  GrLivArea      1460 non-null   int64
 47  BsmtFullBath   1460 non-null   int64
 48  BsmtHalfBath   1460 non-null   int64
 49  FullBath       1460 non-null   int64
 50  HalfBath       1460 non-null   int64
 51  BedroomAbvGr   1460 non-null   int64
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

In [4]:
```
df.describe()
```

Out[4]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrA
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1452.000
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.685
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407	181.066
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.000
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	0.000
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	166.000
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000

8 rows × 38 columns

In [5]:
```
df.corr()['SalePrice'].sort_values(ascending=False)
```

```
C:\Users\omers\AppData\Local\Temp\ipykernel_21524\497439051.py:1: FutureWarning: The default value of numeri
c_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid col
umns or specify the value of numeric_only to silence this warning.
  df.corr()['SalePrice'].sort_values(ascending=False)
```

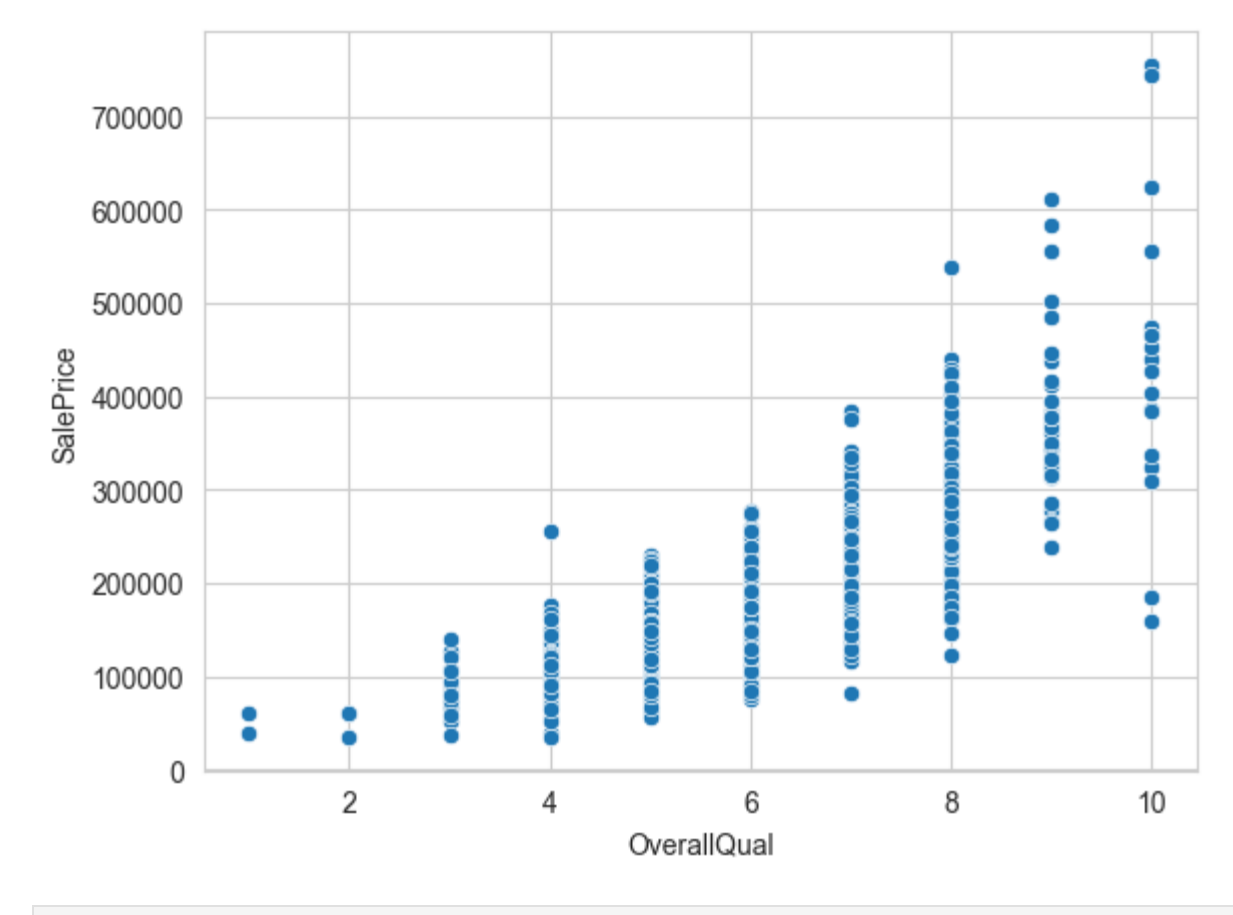
Out[5]:
```
SalePrice        1.000000
OverallQual      0.790982
GrLivArea        0.708624
GarageCars       0.640409
GarageArea       0.623431
TotalBsmtSF      0.613581
1stFlrSF         0.605852
FullBath         0.560664
TotRmsAbvGrd     0.533723
YearBuilt        0.522897
YearRemodAdd     0.507101
GarageYrBlt      0.486362
MasVnrArea       0.477493
Fireplaces       0.466929
BsmtFinSF1       0.386420
LotFrontage      0.351799
WoodDeckSF       0.324413
2ndFlrSF         0.319334
OpenPorchSF      0.315856
HalfBath         0.284108
LotArea          0.263843
BsmtFullBath     0.227122
BsmtUnfSF        0.214479
BedroomAbvGr     0.168213
ScreenPorch      0.111447
PoolArea         0.092404
MoSold           0.046432
3SsnPorch        0.044584
BsmtFinSF2      -0.011378
BsmtHalfBath    -0.016844
MiscVal         -0.021190
Id              -0.021917
LowQualFinSF    -0.025606
YrSold          -0.028923
OverallCond     -0.077856
MSSubClass      -0.084284
EnclosedPorch   -0.128578
KitchenAbvGr    -0.135907
Name: SalePrice, dtype: float64
```

Clearing Outliers

We will check outliers data points for SalePrices. We don't want to extreme values disrupt our models.

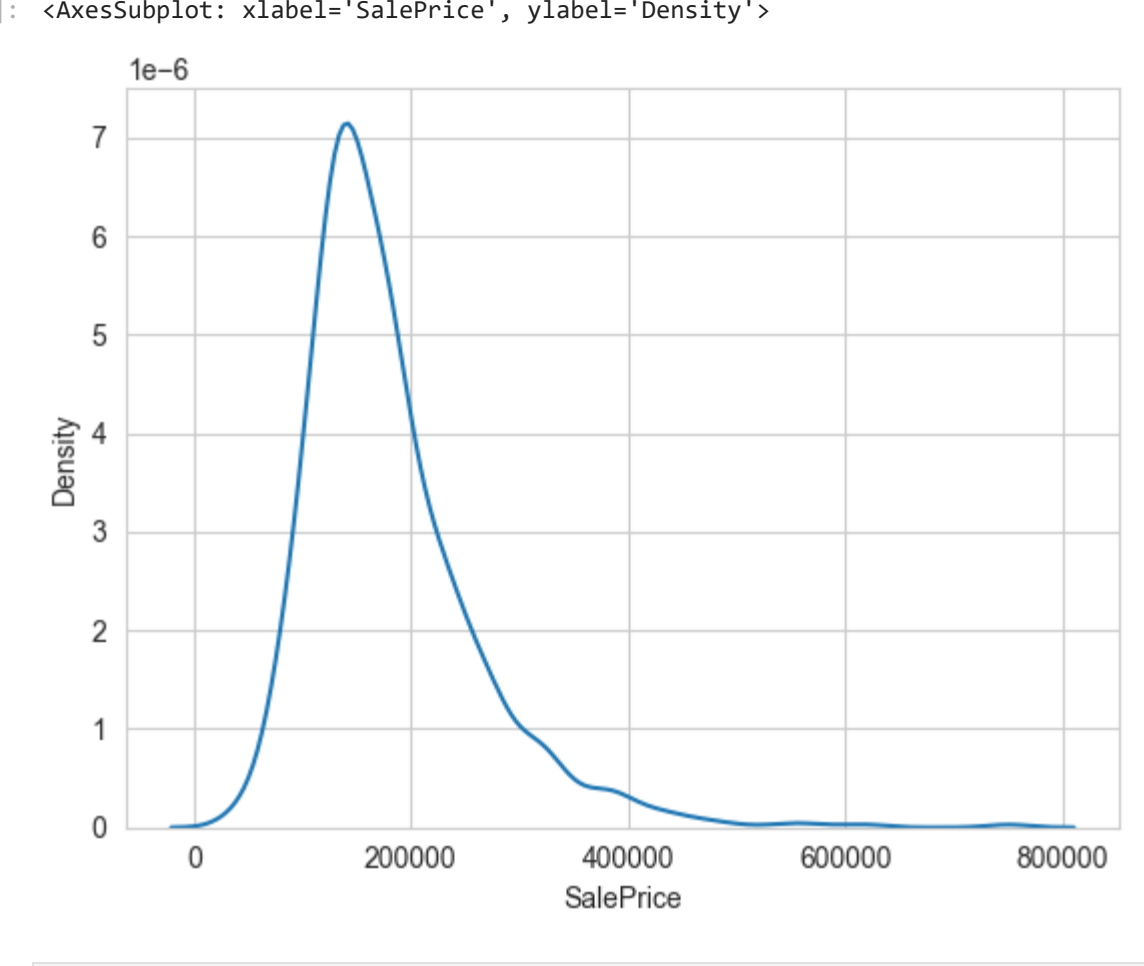
In [6]:
```
# check most correlated variable
sns.scatterplot(x='OverallQual', y='SalePrice', data=df)
```

Out[6]: `<AxesSubplot: xlabel='OverallQual', ylabel='SalePrice'>`



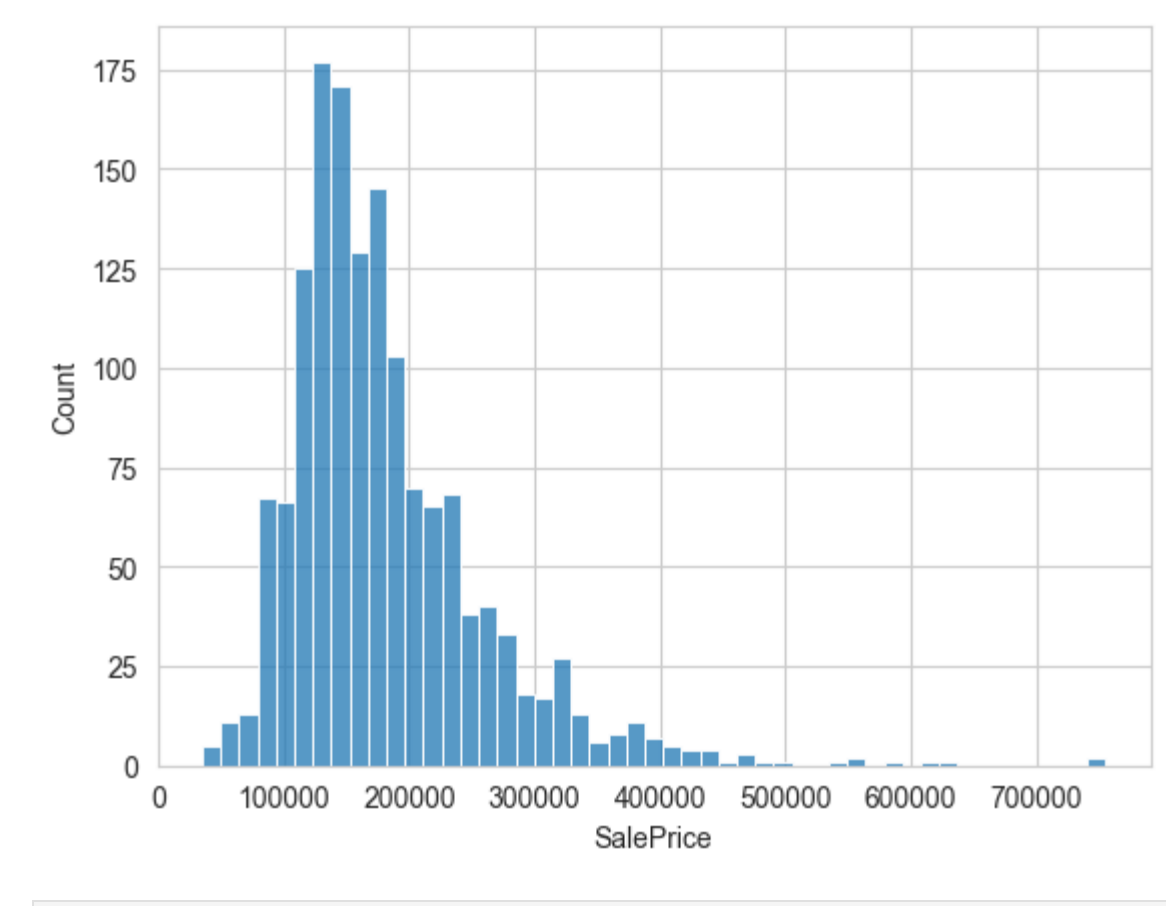
In [7]:
```
# check Sale Price normally distributed
sns.kdeplot(x='SalePrice', data=df)
```

Out[7]: `<AxesSubplot: xlabel='SalePrice', ylabel='Density'>`



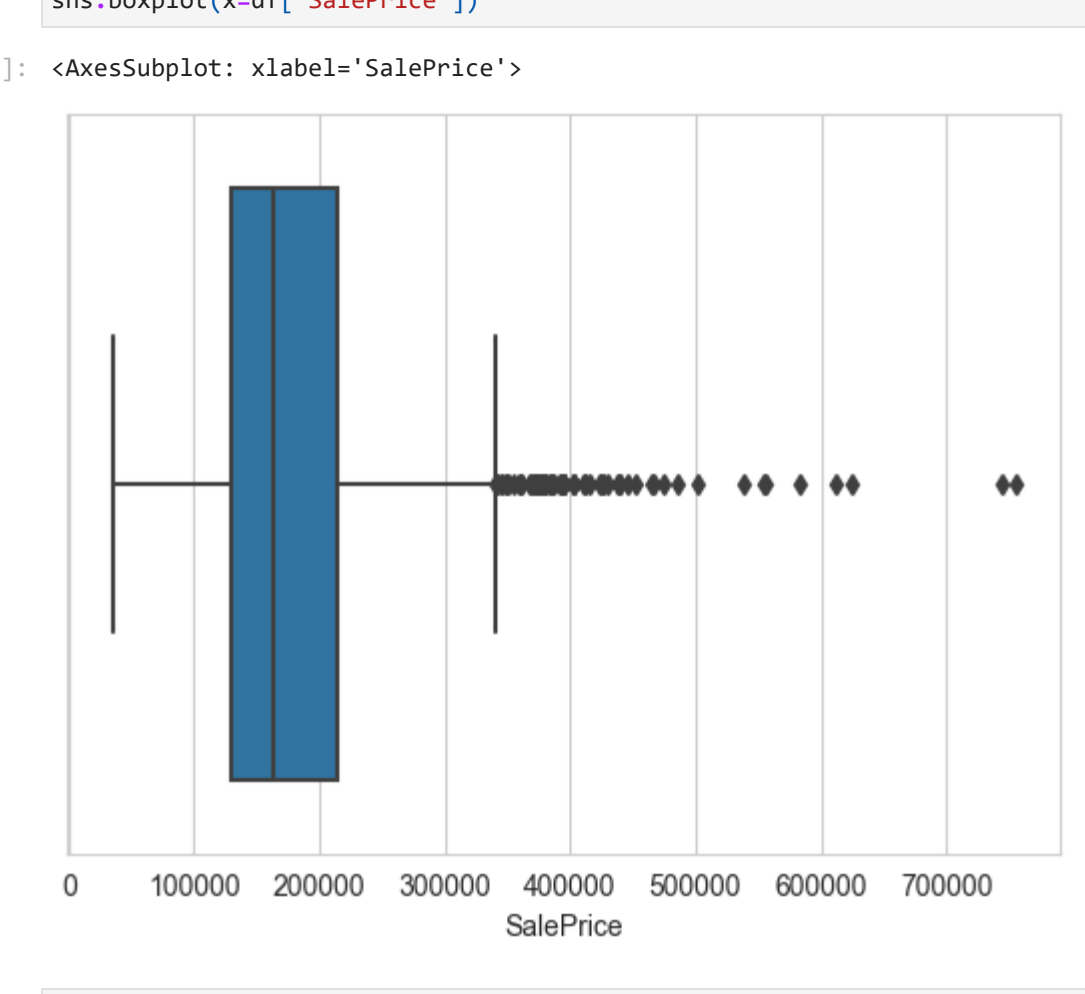
In [8]:
```
sns.histplot(x='SalePrice', data=df)
```

Out[8]: `<AxesSubplot: xlabel='SalePrice', ylabel='Count'>`



In [9]:
```
# Check IQR and outliers
sns.boxplot(x=df['SalePrice'])
```

Out[9]: `<AxesSubplot: xlabel='SalePrice'>`



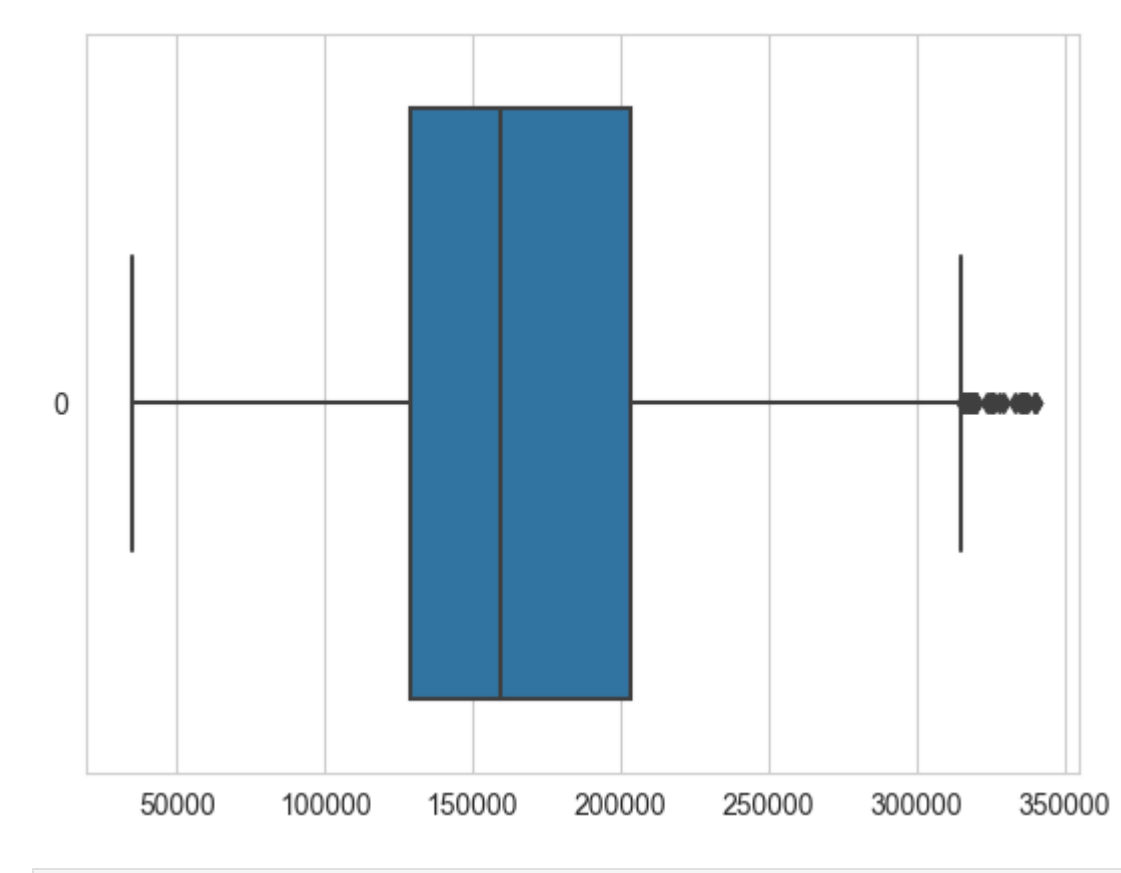
In [10]:
```
q1, q3 = np.quantile(df['SalePrice'], [0.25, 0.75])
iqr = q3 - q1
upper_limit = q3 + 1.5 * iqr
print(f'Quartiles for SalePrice:\n\tQ1: {q1}\n\tQ3: {q3}\
\n\tIQR: {iqr}\n\tUpper Limit: {upper_limit}')
```

```
Quartiles for SalePrice:
        Q1: 129975.0
        Q3: 214000.0
        IQR: 84025.0
        Upper Limit: 340037.5
```

In [11]:
```
sales_limit = df[df['SalePrice'] < upper_limit]['SalePrice']
print('All data size is: ', len(df), '\n', 'Without upper data size is: ',
      len(sales_limit))
sns.boxplot(sales_limit, orient='h')
```

```
All data size is:  1460
 Without upper data size is:  1399
```

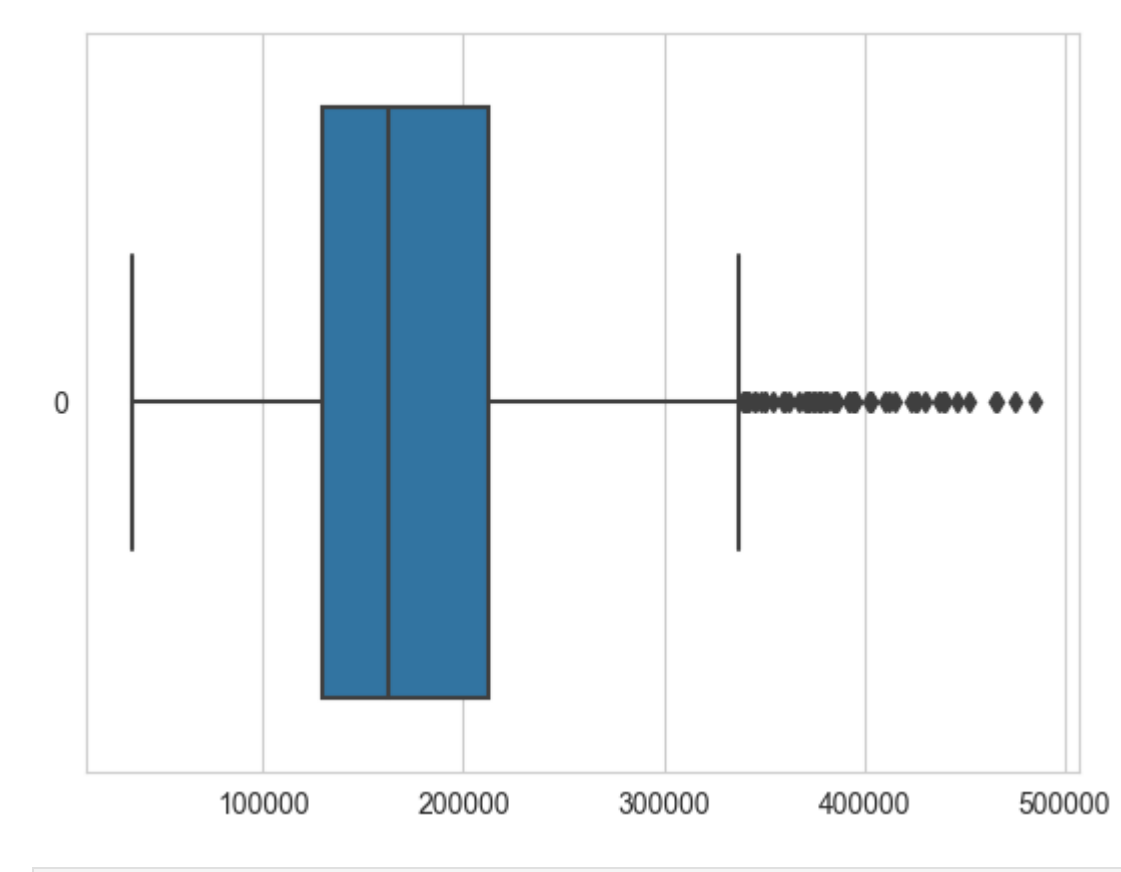
Out[11]: `<AxesSubplot: >`



In [12]:
```
sales_limit = df[df['SalePrice'] < 500000]['SalePrice']
print('All data size is: ', len(df), '\n', 'Without upper data size is: ',
      len(sales_limit))
sns.boxplot(sales_limit, orient='h')
```

```
All data size is:  1460
 Without upper data size is:  1451
```

Out[12]: `<AxesSubplot: >`



In [13]:
```
df = df[df['SalePrice'] < 500000]
```

Handling Missing Data

Explore Missing Data

In [14]:
```
df.isnull().sum().sort_values(ascending=False)
```

Out[14]:
```
PoolQC          1445
MiscFeature     1397
Alley           1360
Fence           1171
FireplaceQu      690
                ...
ExterQual          0
Exterior2nd        0
Exterior1st        0
RoofMatl           0
SalePrice          0
Length: 81, dtype: int64
```

In [15]:
```
# Helper functions to show missing data
def percent_missing(df: pd.DataFrame) -> pd.Series:
    '''

    Args:
        df (pandas.DataFrame): main dataset

    Returns:
        (pd.Series) percentage for every features with missing data
    '''
    percent_nan = df.isnull().sum().sort_values(ascending=False) / len(df) * 100
    return percent_nan[percent_nan > 0]


def report_missing(df: pd.DataFrame):
    '''Visualize and demonstrate missing data about dataframe.

    Args:
        df (pandas.DataFrame): main dataset

    '''
    percent_nan = percent_missing(df)

    null_count = df.isna().sum().sort_values(ascending=False)
    print(f'Total row of dataset is: {len(df)}')
    html_table(null_count[null_count > 0])
    if (len(percent_nan)):
        fig, ax = plt.subplots()
        ax.set_xticklabels(ax.get_xticks(), rotation=90)
        sns.barplot(x=percent_nan.index, y=percent_nan, ax=ax)
```

In [16]:
```
percent_nan = percent_missing(df)
percent_nan
```

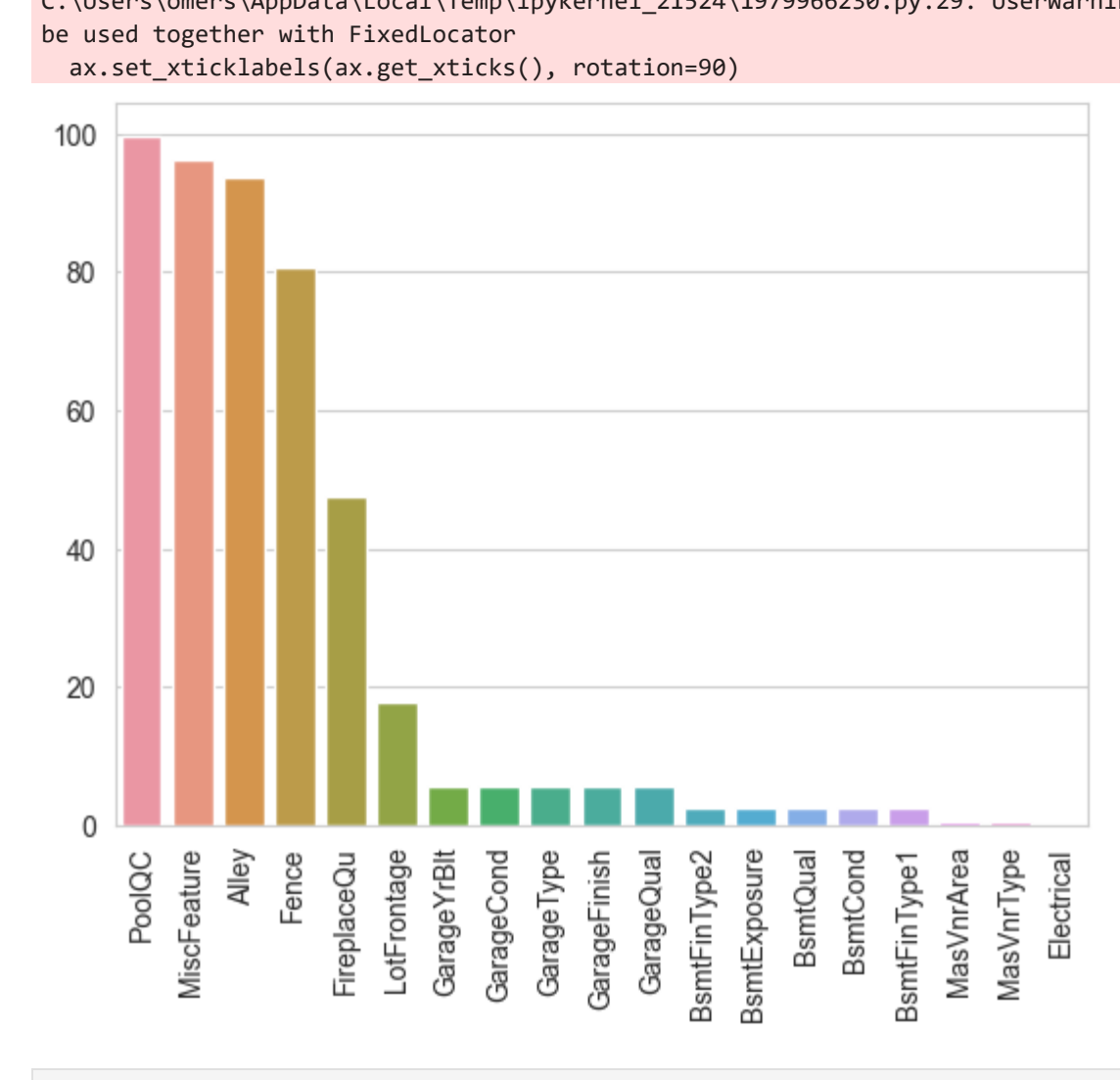
Out[16]:
```
PoolQC          99.586492
MiscFeature     96.278429
Alley           93.728463
Fence           80.702963
FireplaceQu     47.553411
LotFrontage     17.849759
GarageYrBlt      5.582357
GarageCond       5.582357
GarageType       5.582357
GarageFinish     5.582357
GarageQual       5.582357
BsmtFinType2     2.618884
BsmtExposure     2.618884
BsmtQual         2.549966
BsmtCond         2.549966
BsmtFinType1     2.549966
MasVnrArea       0.551344
MasVnrType       0.551344
Electrical       0.068918
dtype: float64
```

In [17]:
```
report_missing(df)
```

```
Total row of dataset is: 1451
```

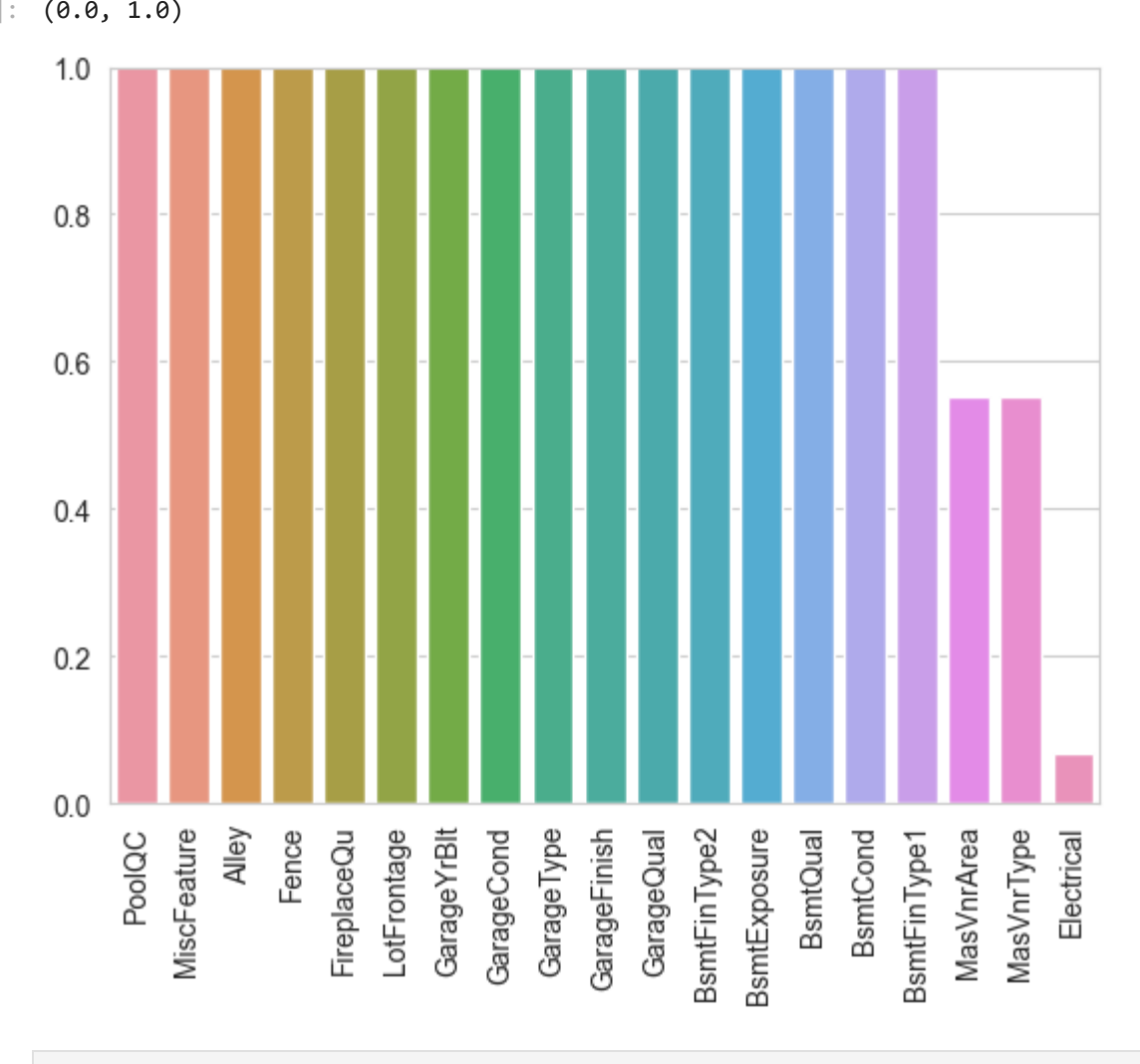
	0
PoolQC	1445
MiscFeature	1397
Alley	1360
Fence	1171
FireplaceQu	690
LotFrontage	259
GarageYrBlt	81
GarageCond	81
GarageType	81
GarageFinish	81
GarageQual	81
BsmtFinType2	38
BsmtExposure	38
BsmtQual	37
BsmtCond	37
BsmtFinType1	37
MasVnrArea	8
MasVnrType	8
Electrical	1

```
C:\Users\omers\AppData\Local\Temp\ipykernel_21524\1979966230.py:29: UserWarning: FixedFormatter should only
be used together with FixedLocator
  ax.set_xticklabels(ax.get_xticks(), rotation=90)
```



In [18]:
```
sns.barplot(x=percent_nan.index, y=percent_nan)
plt.xticks(rotation=90)
plt.ylim(0, 1)
```

Out[18]: `(0.0, 1.0)`



In [19]:
```
percent_nan[percent_nan < 1]
```

Out[19]:
```
MasVnrArea    0.551344
MasVnrType    0.551344
Electrical    0.068918
dtype: float64
```

In [20]:
```
def get_na(column, df=df):
    '''Gets rows with missing data in provided column'''
    return df[df[column].isna()]
```

In [22]:
```
for column in percent_nan[percent_nan < 1].index:
    print(f'Missings for: {column}, Percentage is: {percent_nan[column]}/100')
    html_table(get_na(column))
```

```
Missings for: MasVnrArea, Percentage is: 0.5513439007580978/100
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig
234	235	60	RL	NaN	7851	Pave	NaN	Reg	Lvl	AllPub	Inside
529	530	20	RL	NaN	32668	Pave	NaN	IR1	Lvl	AllPub	CulDSac
650	651	60	FV	65.0	8125	Pave	NaN	Reg	Lvl	AllPub	Inside
936	937	20	RL	67.0	10083	Pave	NaN	Reg	Lvl	AllPub	Inside
973	974	20	FV	95.0	11639	Pave	NaN	Reg	Lvl	AllPub	Corner
977	978	120	FV	35.0	4274	Pave	Pave	IR1	Lvl	AllPub	Inside
1243	1244	20	RL	107.0	13891	Pave	NaN	Reg	Lvl	AllPub	Inside
1278	1279	60	RL	75.0	9473	Pave	NaN	Reg	Lvl	AllPub	Inside

```
Missings for: MasVnrType, Percentage is: 0.5513439007580978/100
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig
234	235	60	RL	NaN	7851	Pave	NaN	Reg	Lvl	AllPub	Inside
529	530	20	RL	NaN	32668	Pave	NaN	IR1	Lvl	AllPub	CulDSac
650	651	60	FV	65.0	8125	Pave	NaN	Reg	Lvl	AllPub	Inside
936	937	20	RL	67.0	10083	Pave	NaN	Reg	Lvl	AllPub	Inside
973	974	20	FV	95.0	11639	Pave	NaN	Reg	Lvl	AllPub	Corner
977	978	120	FV	35.0	4274	Pave	Pave	IR1	Lvl	AllPub	Inside
1243	1244	20	RL	107.0	13891	Pave	NaN	Reg	Lvl	AllPub	Inside
1278	1279	60	RL	75.0	9473	Pave	NaN	Reg	Lvl	AllPub	Inside

```
Missings for: Electrical, Percentage is: 0.06891798759476223/100
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig
1379	1380	80	RL	73.0	9735	Pave	NaN	Reg	Lvl	AllPub	Inside

In [23]:
```
print('All missings are: \n\t', list(percent_nan.index))
```

```
All missings are:
         ['PoolQC', 'MiscFeature', 'Alley', 'Fence', 'FireplaceQu', 'LotFrontage', 'GarageYrBlt', 'GarageCon
d', 'GarageType', 'GarageFinish', 'GarageQual', 'BsmtFinType2', 'BsmtExposure', 'BsmtQual', 'BsmtCond', 'Bsm
tFinType1', 'MasVnrArea', 'MasVnrType', 'Electrical']
```

In [24]:
```
df[percent_nan.index].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1451 entries, 0 to 1459
Data columns (total 19 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   PoolQC        6 non-null      object
 1   MiscFeature   54 non-null     object
 2   Alley         91 non-null     object
 3   Fence         280 non-null    object
 4   FireplaceQu   761 non-null    object
 5   LotFrontage   1192 non-null   float64
 6   GarageYrBlt   1370 non-null   float64
 7   GarageCond    1370 non-null   object
 8   GarageType    1370 non-null   object
 9   GarageFinish  1370 non-null   object
 10  GarageQual    1370 non-null   object
 11  BsmtFinType2  1413 non-null   object
 12  BsmtExposure  1413 non-null   object
 13  BsmtQual      1414 non-null   object
 14  BsmtCond      1414 non-null   object
 15  BsmtFinType1  1414 non-null   object
 16  MasVnrArea    1443 non-null   float64
 17  MasVnrType    1443 non-null   object
 18  Electrical    1450 non-null   object
dtypes: float64(3), object(16)
memory usage: 226.7+ KB
```

Filling Null values which has meaning

Filling Categorical Variables In the description text we can see that 'Electrical', 'MasVnrArea', 'LotFrontage', 'GarageYrBlt' variables are categorical and has meaning for Null values. I filled missing with None value as string type.

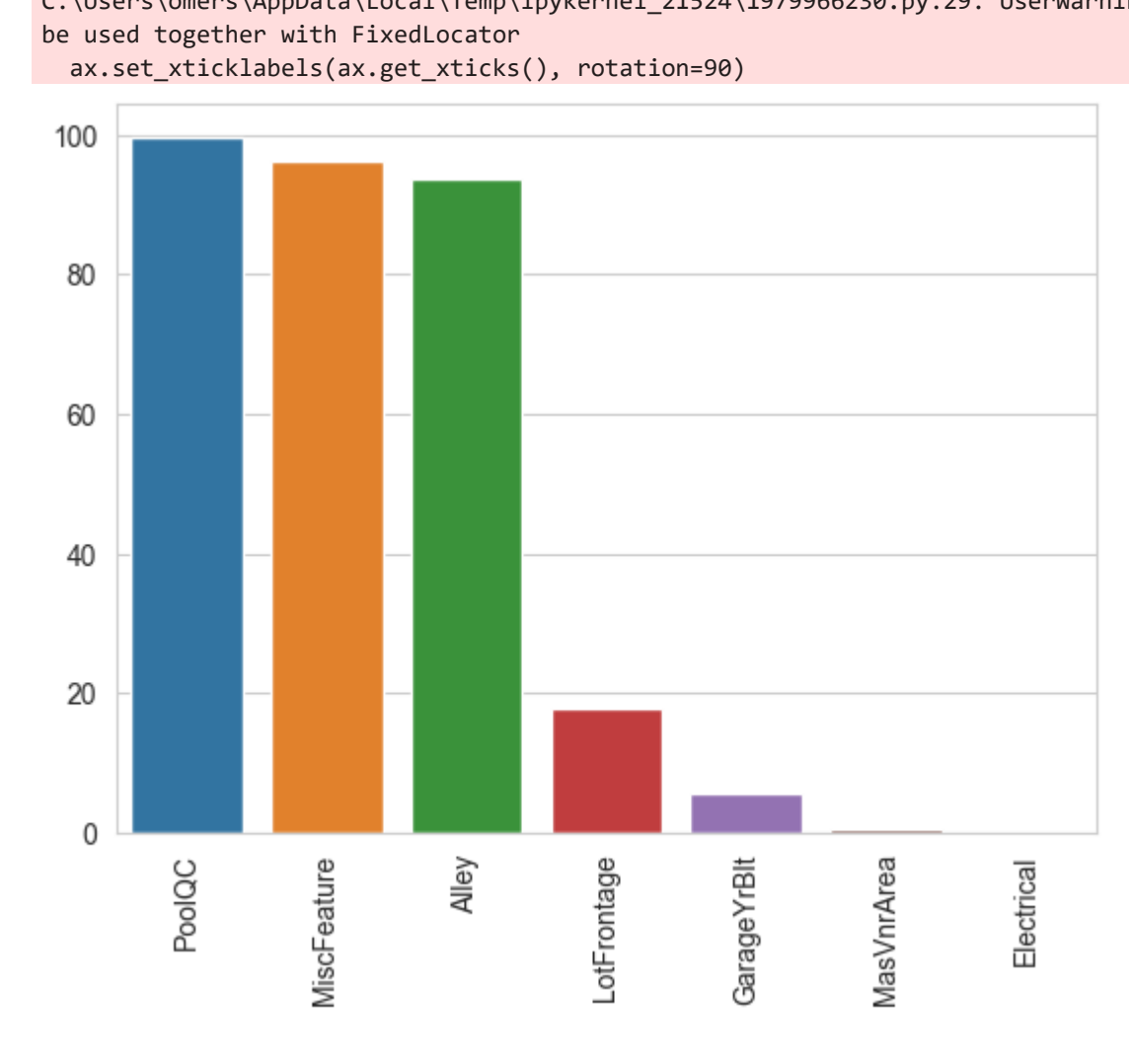
In [26]:
```
categoric_columns = ['MasVnrType', 'BsmtFinType1', 'BsmtCond', 'BsmtQual',
                     'BsmtExposure', 'BsmtFinType2', 'GarageQual',
                     'GarageFinish', 'GarageType', 'GarageCond', ]
categoric_big_columns = ['PoolQC', 'Fence', 'Alley', 'MiscFeature',
                         'FireplaceQu', ]
true_none_columns = ['Electrical', 'MasVnrArea', 'LotFrontage', 'GarageYrBlt']
df[categoric_columns] = df[categoric_columns].fillna('None')
```

In [27]:
```
df['Fence'] = df['Fence'].fillna('None')
df['FireplaceQu'] = df['FireplaceQu'].fillna('None')

percent_nan = percent_missing(df)

report_missing(df)
```

```
Total row of dataset is: 1451
```

	0
PoolQC	1445
MiscFeature	1397
Alley	1360
LotFrontage	259
GarageYrBlt	81
MasVnrArea	8
Electrical	1

```
C:\Users\omers\AppData\Local\Temp\ipykernel_21524\1979966230.py:29: UserWarning: FixedFormatter should only
be used together with FixedLocator
  ax.set_xticklabels(ax.get_xticks(), rotation=90)
```



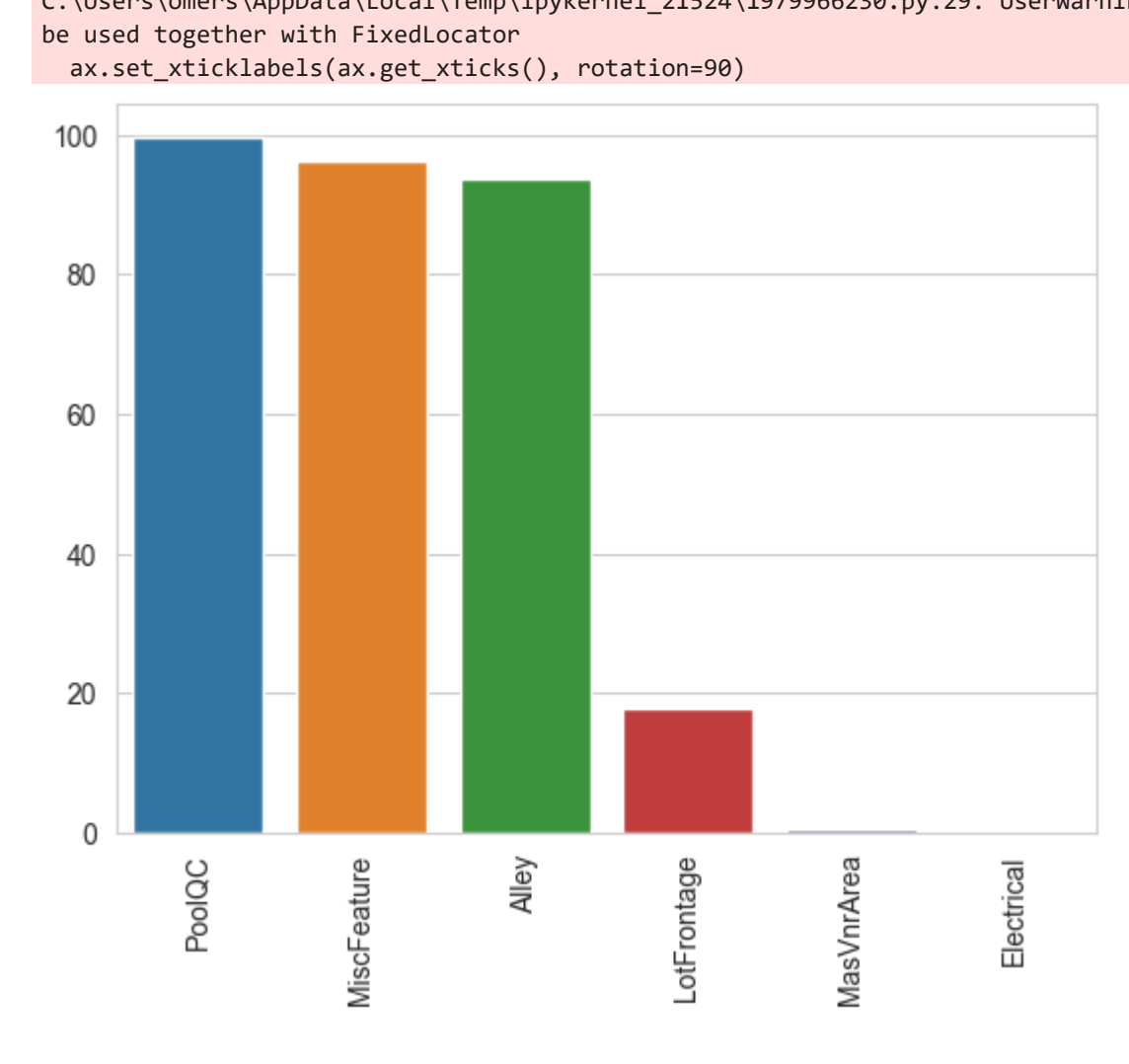
Filling with Mean

In [28]:
```
df['GarageYrBlt'] = df['GarageYrBlt'].fillna(df['GarageYrBlt'].mean())
report_missing(df)
```

```
Total row of dataset is: 1451
```

	0
PoolQC	1445
MiscFeature	1397
Alley	1360
LotFrontage	259
MasVnrArea	8
Electrical	1

```
C:\Users\omers\AppData\Local\Temp\ipykernel_21524\1979966230.py:29: UserWarning: FixedFormatter should only
be used together with FixedLocator
  ax.set_xticklabels(ax.get_xticks(), rotation=90)
```



Dropping Some Missing Rows

I dropped missing values with inpecting data and reading data description.

In [29]:
```
df[df['MasVnrArea'].isna()][['MasVnrArea', 'MasVnrType']]
```

Out[29]:

	MasVnrArea	MasVnrType
234	NaN	None
529	NaN	None
650	NaN	None
936	NaN	None
973	NaN	None
977	NaN	None
1243	NaN	None
1278	NaN	None

In [30]:
```
print(f'Before drop total row is: {len(df)}')
df.dropna(axis=0, subset=['Electrical', 'MasVnrArea'], inplace=True)
print(f'After drop total row is: {len(df)}')
```

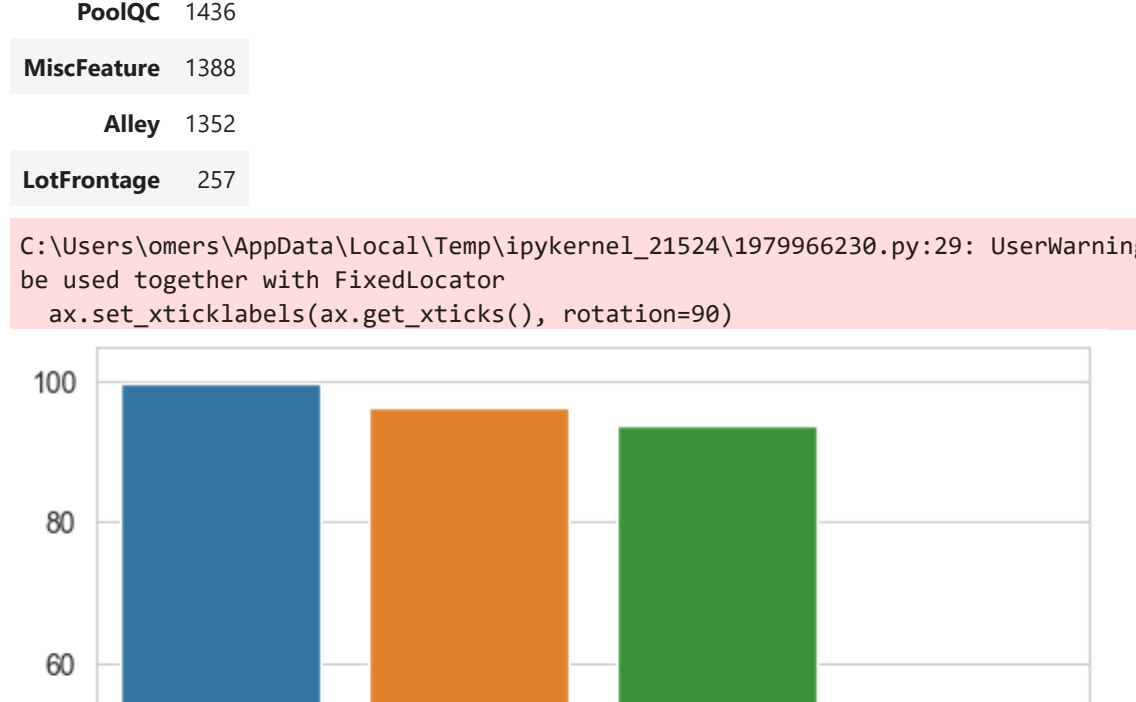
```
Before drop total row is: 1451
After drop total row is: 1442
```

In [31]:
```python
report_missing(df)
```

Total row of dataset is: 1442

	0
PoolQC	1436
MiscFeature	1388
Alley	1352
LotFrontage	257

```
C:\Users\omers\AppData\Local\Temp\ipykernel_21524\1979966230.py:29: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_xticklabels(ax.get_xticks(), rotation=90)
```



Dropping Columns

In [32]:
```python
df = df.drop(['PoolQC', 'MiscFeature', 'Alley'], axis=1)
```

In [33]:
```python
html_table(df['Fence'].value_counts())
html_table(df['FireplaceQu'].value_counts())
```

	Fence
None	1162
MnPrv	156
GdPrv	59
GdWo	54
MnWw	11

	FireplaceQu
None	685
Gd	372
TA	309
Fa	33
Ex	23
Po	20

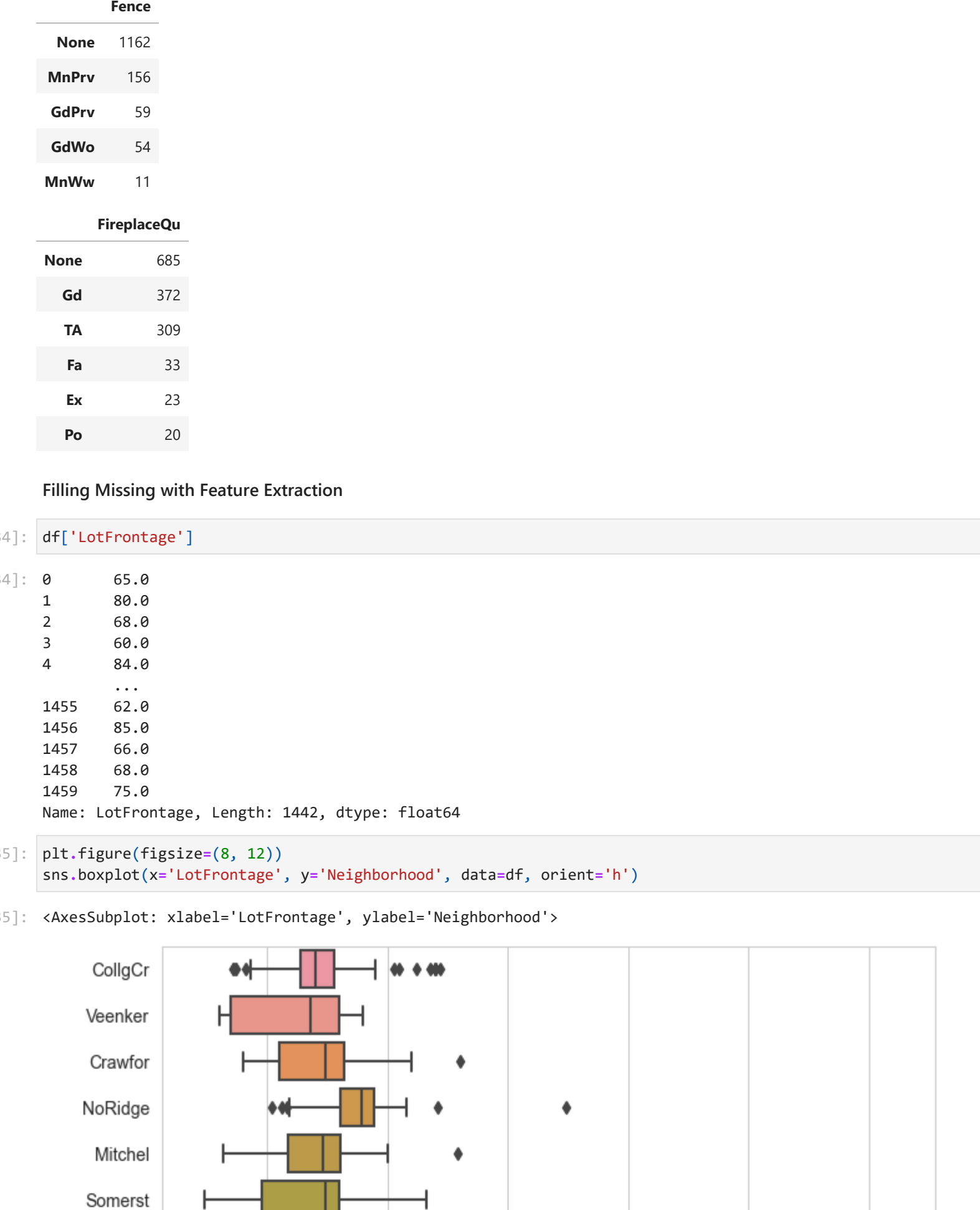
Filling Missing with Feature Extraction

In [34]:
```python
df['LotFrontage']
```

Out[34]:
```
0       65.0
1       80.0
2       68.0
3       60.0
4       84.0
        ...
1455    62.0
1456    85.0
1457    66.0
1458    68.0
1459    75.0
Name: LotFrontage, Length: 1442, dtype: float64
```

In [35]:
```python
plt.figure(figsize=(8, 12))
sns.boxplot(x='LotFrontage', y='Neighborhood', data=df, orient='h')
```

Out[35]:
```
<AxesSubplot: xlabel='LotFrontage', ylabel='Neighborhood'>
```



In [36]:
```python
df.groupby('Neighborhood')['LotFrontage'].mean()
```

Out[36]:
```
Neighborhood
Blmngtn    47.142857
Blueste    24.000000
BrDale     21.562500
BrkSide    57.509804
ClearCr    83.461538
CollgCr    71.656000
Crawfor    71.804878
Edwards    68.217391
Gilbert    79.877551
IDOTRR     62.500000
MeadowV    27.800000
Mitchel    70.083333
NAmes      76.462366
NPkVill    32.285714
NWAmes     81.288889
NoRidge    88.333333
NridgHt    80.611111
OldTown    62.788991
SWISU      58.913043
Sawyer     74.437500
SawyerW    71.591837
Somerst    64.653333
StoneBr    62.294118
Timber     80.379310
Veenker    59.714286
Name: LotFrontage, dtype: float64
```

In [37]:
```python
df['LotFrontage'] = df.groupby('Neighborhood')['LotFrontage'].transform(
    lambda value: value.fillna(value.mean()))
report_missing(df)
```

Total row of dataset is: 1442

	0

In [38]:
```python
# saving cleaned data before process
df.drop(axis=1, columns=['Id'], inplace=True)
df.to_csv(interim_data_dir / file_name, index=False)
```

Encoding Data

In this section, I implemented one hot encoding and got dummy variables for categorical data.

In [39]:
```python
with open('../references/data_description.txt', 'rt') as f:
    print(f.read())
```

```
MSSubClass: Identifies the type of dwelling involved in the sale.

        20      1-STORY 1946 & NEWER ALL STYLES
        30      1-STORY 1945 & OLDER
        40      1-STORY W/FINISHED ATTIC ALL AGES
        45      1-1/2 STORY - UNFINISHED ALL AGES
        50      1-1/2 STORY FINISHED ALL AGES
        60      2-STORY 1946 & NEWER
        70      2-STORY 1945 & OLDER
        75      2-1/2 STORY ALL AGES
        80      SPLIT OR MULTI-LEVEL
        85      SPLIT FOYER
        90      DUPLEX - ALL STYLES AND AGES
       120      1-STORY PUD (Planned Unit Development) - 1946 & NEWER
       150      1-1/2 STORY PUD - ALL AGES
       160      2-STORY PUD - 1946 & NEWER
       180      PUD - MULTILEVEL - INCL SPLIT LEV/FOYER
       190      2 FAMILY CONVERSION - ALL STYLES AND AGES

MSZoning: Identifies the general zoning classification of the sale.

       A        Agriculture
       C        Commercial
       FV       Floating Village Residential
       I        Industrial
       RH       Residential High Density
       RL       Residential Low Density
       RP       Residential Low Density Park
       RM       Residential Medium Density

LotFrontage: Linear feet of street connected to property

LotArea: Lot size in square feet

Street: Type of road access to property

       Grvl     Gravel
       Pave     Paved

Alley: Type of alley access to property

       Grvl     Gravel
       Pave     Paved
       NA       No alley access

LotShape: General shape of property

       Reg      Regular
       IR1      Slightly irregular
       IR2      Moderately Irregular
       IR3      Irregular

LandContour: Flatness of the property

       Lvl      Near Flat/Level
       Bnk      Banked - Quick and significant rise from street grade to building
       HLS      Hillside - Significant slope from side to side
       Low      Depression

Utilities: Type of utilities available

       AllPub   All public Utilities (E,G,W,& S)
       NoSewr   Electricity, Gas, and Water (Septic Tank)
       NoSeWa   Electricity and Gas Only
       ELO      Electricity only

LotConfig: Lot configuration

       Inside   Inside lot
       Corner   Corner lot
       CulDSac  Cul-de-sac
       FR2      Frontage on 2 sides of property
       FR3      Frontage on 3 sides of property

LandSlope: Slope of property

       Gtl      Gentle slope
       Mod      Moderate Slope
       Sev      Severe Slope

Neighborhood: Physical locations within Ames city limits

       Blmngtn  Bloomington Heights
       Blueste  Bluestem
       BrDale   Briardale
       BrkSide  Brookside
       ClearCr  Clear Creek
       CollgCr  College Creek
       Crawfor  Crawford
       Edwards  Edwards
       Gilbert  Gilbert
       IDOTRR   Iowa DOT and Rail Road
       MeadowV  Meadow Village
       Mitchel  Mitchell
       Names    North Ames
       NoRidge  Northridge
       NPkVill  Northpark Villa
       NridgHt  Northridge Heights
       NWAmes   Northwest Ames
       OldTown  Old Town
       SWISU    South & West of Iowa State University
       Sawyer   Sawyer
       SawyerW  Sawyer West
       Somerst  Somerset
       StoneBr  Stone Brook
       Timber   Timberland
       Veenker  Veenker

Condition1: Proximity to various conditions

       Artery   Adjacent to arterial street
       Feedr    Adjacent to feeder street
       Norm     Normal
       RRNn     Within 200' of North-South Railroad
       RRAn     Adjacent to North-South Railroad
       PosN     Near positive off-site feature--park, greenbelt, etc.
       PosA     Adjacent to postive off-site feature
       RRNe     Within 200' of East-West Railroad
       RRAe     Adjacent to East-West Railroad

Condition2: Proximity to various conditions (if more than one is present)

       Artery   Adjacent to arterial street
       Feedr    Adjacent to feeder street
       Norm     Normal
       RRNn     Within 200' of North-South Railroad
       RRAn     Adjacent to North-South Railroad
       PosN     Near positive off-site feature--park, greenbelt, etc.
       PosA     Adjacent to postive off-site feature
       RRNe     Within 200' of East-West Railroad
       RRAe     Adjacent to East-West Railroad

BldgType: Type of dwelling

       1Fam     Single-family Detached
       2FmCon   Two-family Conversion; originally built as one-family dwelling
       Duplx    Duplex
       TwnhsE   Townhouse End Unit
       TwnhsI   Townhouse Inside Unit

HouseStyle: Style of dwelling

       1Story   One story
       1.5Fin   One and one-half story: 2nd level finished
       1.5Unf   One and one-half story: 2nd level unfinished
       2Story   Two story
       2.5Fin   Two and one-half story: 2nd level finished
       2.5Unf   Two and one-half story: 2nd level unfinished
       SFoyer   Split Foyer
       SLvl     Split Level

OverallQual: Rates the overall material and finish of the house

       10       Very Excellent
       9        Excellent
       8        Very Good
       7        Good
       6        Above Average
       5        Average
       4        Below Average
       3        Fair
       2        Poor
       1        Very Poor

OverallCond: Rates the overall condition of the house

       10       Very Excellent
       9        Excellent
       8        Very Good
       7        Good
       6        Above Average
       5        Average
       4        Below Average
       3        Fair
       2        Poor
       1        Very Poor

YearBuilt: Original construction date

YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)

RoofStyle: Type of roof

       Flat     Flat
       Gable    Gable
       Gambrel  Gabrel (Barn)
       Hip      Hip
       Mansard  Mansard
       Shed     Shed

RoofMatl: Roof material

       ClyTile  Clay or Tile
       CompShg  Standard (Composite) Shingle
       Membran  Membrane
       Metal    Metal
       Roll     Roll
       Tar&Grv  Gravel & Tar
       WdShake  Wood Shakes
       WdShngl  Wood Shingles

Exterior1st: Exterior covering on house

       AsbShng  Asbestos Shingles
       AsphShn  Asphalt Shingles
       BrkComm  Brick Common
       BrkFace  Brick Face
       CBlock   Cinder Block
       CemntBd  Cement Board
       HdBoard  Hard Board
       ImStucc  Imitation Stucco
       MetalSd  Metal Siding
       Other    Other
       Plywood  Plywood
       PreCast  PreCast
       Stone    Stone
       Stucco   Stucco
       VinylSd  Vinyl Siding
       Wd Sdng  Wood Siding
       WdShing  Wood Shingles

Exterior2nd: Exterior covering on house (if more than one material)

       AsbShng  Asbestos Shingles
       AsphShn  Asphalt Shingles
       BrkComm  Brick Common
       BrkFace  Brick Face
       CBlock   Cinder Block
       CemntBd  Cement Board
       HdBoard  Hard Board
       ImStucc  Imitation Stucco
       MetalSd  Metal Siding
       Other    Other
       Plywood  Plywood
       PreCast  PreCast
       Stone    Stone
       Stucco   Stucco
       VinylSd  Vinyl Siding
       Wd Sdng  Wood Siding
       WdShing  Wood Shingles

MasVnrType: Masonry veneer type

       BrkCmn   Brick Common
       BrkFace  Brick Face
       CBlock   Cinder Block
       None     None
       Stone    Stone

MasVnrArea: Masonry veneer area in square feet

ExterQual: Evaluates the quality of the material on the exterior

       Ex       Excellent
       Gd       Good
       TA       Average/Typical
       Fa       Fair
       Po       Poor

ExterCond: Evaluates the present condition of the material on the exterior

       Ex       Excellent
       Gd       Good
       TA       Average/Typical
       Fa       Fair
       Po       Poor

Foundation: Type of foundation

       BrkTil   Brick & Tile
       CBlock   Cinder Block
       PConc    Poured Contrete
       Slab     Slab
       Stone    Stone
       Wood     Wood

BsmtQual: Evaluates the height of the basement

       Ex       Excellent (100+ inches)
       Gd       Good (90-99 inches)
       TA       Typical (80-89 inches)
       Fa       Fair (70-79 inches)
       Po       Poor (<70 inches
       NA       No Basement

BsmtCond: Evaluates the general condition of the basement

       Ex       Excellent
       Gd       Good
       TA       Typical - slight dampness allowed
       Fa       Fair - dampness or some cracking or settling
       Po       Poor - Severe cracking, settling, or wetness
       NA       No Basement

BsmtExposure: Refers to walkout or garden level walls

       Gd       Good Exposure
       Av       Average Exposure (split levels or foyers typically score average or above)
       Mn       Mimimum Exposure
       No       No Exposure
       NA       No Basement

BsmtFinType1: Rating of basement finished area

       GLQ      Good Living Quarters
       ALQ      Average Living Quarters
       BLQ      Below Average Living Quarters
       Rec      Average Rec Room
       LwQ      Low Quality
       Unf      Unfinshed
       NA       No Basement

BsmtFinSF1: Type 1 finished square feet

BsmtFinType2: Rating of basement finished area (if multiple types)

       GLQ      Good Living Quarters
       ALQ      Average Living Quarters
       BLQ      Below Average Living Quarters
       Rec      Average Rec Room
       LwQ      Low Quality
       Unf      Unfinshed
       NA       No Basement

BsmtFinSF2: Type 2 finished square feet

BsmtUnfSF: Unfinished square feet of basement area

TotalBsmtSF: Total square feet of basement area

Heating: Type of heating

       Floor    Floor Furnace
       GasA     Gas forced warm air furnace
       GasW     Gas hot water or steam heat
       Grav     Gravity furnace
       OthW     Hot water or steam heat other than gas
       Wall     Wall furnace

HeatingQC: Heating quality and condition

       Ex       Excellent
       Gd       Good
       TA       Average/Typical
       Fa       Fair
       Po       Poor

CentralAir: Central air conditioning

       N        No
       Y        Yes

Electrical: Electrical system

       SBrkr    Standard Circuit Breakers & Romex
       FuseA    Fuse Box over 60 AMP and all Romex wiring (Average)
       FuseF    60 AMP Fuse Box and mostly Romex wiring (Fair)
       FuseP    60 AMP Fuse Box and mostly knob & tube wiring (poor)
       Mix      Mixed

1stFlrSF: First Floor square feet

2ndFlrSF: Second floor square feet

LowQualFinSF: Low quality finished square feet (all floors)

GrLivArea: Above grade (ground) living area square feet

BsmtFullBath: Basement full bathrooms

BsmtHalfBath: Basement half bathrooms

FullBath: Full bathrooms above grade

HalfBath: Half baths above grade

Bedroom: Bedrooms above grade (does NOT include basement bedrooms)

Kitchen: Kitchens above grade

KitchenQual: Kitchen quality

       Ex       Excellent
       Gd       Good
       TA       Typical/Average
       Fa       Fair
       Po       Poor

TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)

Functional: Home functionality (Assume typical unless deductions are warranted)

       Typ      Typical Functionality
       Min1     Minor Deductions 1
       Min2     Minor Deductions 2
       Mod      Moderate Deductions
       Maj1     Major Deductions 1
       Maj2     Major Deductions 2
       Sev      Severely Damaged
       Sal      Salvage only

Fireplaces: Number of fireplaces

FireplaceQu: Fireplace quality

       Ex       Excellent - Exceptional Masonry Fireplace
       Gd       Good - Masonry Fireplace in main level
       TA       Average - Prefabricated Fireplace in main living area or Masonry Fireplace in basement
       Fa       Fair - Prefabricated Fireplace in basement
       Po       Poor - Ben Franklin Stove
       NA       No Fireplace

GarageType: Garage location

       2Types   More than one type of garage
       Attchd   Attached to home
       Basment  Basement Garage
       BuiltIn  Built-In (Garage part of house - typically has room above garage)
       CarPort  Car Port
       Detchd   Detached from home
       NA       No Garage

GarageYrBlt: Year garage was built

GarageFinish: Interior finish of the garage

       Fin      Finished
       RFn      Rough Finished
       Unf      Unfinished
       NA       No Garage

GarageCars: Size of garage in car capacity

GarageArea: Size of garage in square feet

GarageQual: Garage quality

       Ex       Excellent
       Gd       Good
       TA       Typical/Average
       Fa       Fair
       Po       Poor
       NA       No Garage

GarageCond: Garage condition

       Ex       Excellent
       Gd       Good
       TA       Typical/Average
       Fa       Fair
       Po       Poor
       NA       No Garage

PavedDrive: Paved driveway

       Y        Paved
       P        Partial Pavement
       N        Dirt/Gravel

WoodDeckSF: Wood deck area in square feet

OpenPorchSF: Open porch area in square feet

EnclosedPorch: Enclosed porch area in square feet

3SsnPorch: Three season porch area in square feet

ScreenPorch: Screen porch area in square feet

PoolArea: Pool area in square feet

PoolQC: Pool quality

       Ex       Excellent
       Gd       Good
       TA       Average/Typical
       Fa       Fair
       NA       No Pool

Fence: Fence quality

       GdPrv    Good Privacy
       MnPrv    Minimum Privacy
       GdWo     Good Wood
       MnWw     Minimum Wood/Wire
       NA       No Fence

MiscFeature: Miscellaneous feature not covered in other categories

       Elev     Elevator
       Gar2     2nd Garage (if not described in garage section)
       Othr     Other
       Shed     Shed (over 100 SF)
       TenC     Tennis Court
       NA       None

MiscVal: $Value of miscellaneous feature

MoSold: Month Sold (MM)

YrSold: Year Sold (YYYY)

SaleType: Type of sale

       WD       Warranty Deed - Conventional
       CWD      Warranty Deed - Cash
       VWD      Warranty Deed - VA Loan
       New      Home just constructed and sold
       COD      Court Officer Deed/Estate
       Con      Contract 15% Down payment regular terms
       ConLw    Contract Low Down payment and low interest
       ConLI    Contract Low Interest
       ConLD    Contract Low Down
       Oth      Other

SaleCondition: Condition of sale

       Normal   Normal Sale
       Abnorml  Abnormal Sale -  trade, foreclosure, short sale
       AdjLand  Adjoining Land Purchase
       Alloca   Allocation - two linked properties with separate deeds, typically condo with a garage unit

       Family   Sale between family members
       Partial  Home was not completed when last assessed (associated with New Homes)
```

In [40]:
```python
df.head()
```

Out[40]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	...	3SsnPorc
0	60	RL	65.0	8450	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	
1	20	RL	80.0	9600	Pave	Reg	Lvl	AllPub	FR2	Gtl	...	
2	60	RL	68.0	11250	Pave	IR1	Lvl	AllPub	Inside	Gtl	...	
3	70	RL	60.0	9550	Pave	IR1	Lvl	AllPub	Corner	Gtl	...	
4	60	RL	84.0	14260	Pave	IR1	Lvl	AllPub	FR2	Gtl	...	

5 rows × 77 columns

In [41]:
```python
df['MSSubClass'].head()
```

Out[41]:
```
0    60
1    20
2    60
3    70
4    60
Name: MSSubClass, dtype: int64
```

In [42]:
```python
df['MSSubClass'] = df['MSSubClass'].apply(str)
df['MSSubClass'].info()
```

```
<class 'pandas.core.series.Series'>
Int64Index: 1442 entries, 0 to 1459
Series name: MSSubClass
Non-Null Count  Dtype
--------------  -----
1442 non-null   object
dtypes: object(1)
memory usage: 54.8+ KB
```

In [43]:
```python
df.reset_index(drop=True, inplace=True)
```

In [44]:
```python
object_df = df.select_dtypes(include='object')
numeric_df = df.select_dtypes(exclude='object')
```

In [45]:
```python
from sklearn.preprocessing import OneHotEncoder

enc = OneHotEncoder(handle_unknown='ignore', sparse=False)
enc.fit(object_df)
```

Out[45]:
```
OneHotEncoder
OneHotEncoder(handle_unknown='ignore', sparse=False)
```

In [46]:
```python
df_objects_dummies = enc.transform(object_df)
df_objects_dummies[0:5, :]
```

Out[46]:
```
array([[0., 0., 0., ..., 0., 1., 0.],
       [0., 0., 0., ..., 0., 1., 0.],
       [0., 0., 0., ..., 0., 1., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 1., 0.]])
```

In [47]:
```python
df_encoded = pd.concat((numeric_df, pd.DataFrame(df_objects_dummies)), axis=1)
df_encoded.info()
df_encoded.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1442 entries, 0 to 1441
Columns: 305 entries, LotFrontage to 268
dtypes: float64(272), int64(33)
memory usage: 3.4 MB
```

Out[47]:

	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnf
0	65.0	8450	7	5	2003	2003	196.0	706	0	15
1	80.0	9600	6	8	1976	1976	0.0	978	0	28
2	68.0	11250	7	5	2001	2002	162.0	486	0	43
3	60.0	9550	7	5	1915	1970	0.0	216	0	54
4	84.0	14260	8	5	2000	2000	350.0	655	0	49

5 rows × 305 columns

In [48]:
```python
df_encoded.to_csv(processed_data_dir / file_name, index=False)
```

In [49]:
```python
def build_features(df: pd.DataFrame, enc: OneHotEncoder) -> pd.DataFrame:
    '''Process new data to enter model without scaling
    Operations -> Handle missings, apply one hot encoding on data.

    Args:
        df (pandas.DataFrame): df to procces
        enc (sklearn.preprocessing.OneHotEncoder): One hot encoder fitted on train set

    Returns:
        (pd.DataFrame): Dataset for models without scaling
    '''
    print(f'Before drop total row is: {len(df)}')
    df.dropna(axis=0, subset=['Electrical', 'MasVnrArea'], inplace=True)
    df['GarageYrBlt'] = df['GarageYrBlt'].fillna(df['GarageYrBlt'].mean())
    df = df.drop(['PoolQC', 'MiscFeature', 'Alley'], axis=1)
    df['Fence'] = df['Fence'].fillna('None')
    df['FireplaceQu'] = df['FireplaceQu'].fillna('None')
    df['LotFrontage'] = df.groupby('Neighborhood')['LotFrontage'].transform(
        lambda value: value.fillna(value.mean()))
    df.drop(axis=1, columns=['Id'], inplace=True)
    nullable_cols = ['GarageFinish', 'GarageQual', 'GarageCond', 'GarageType',
                     'BsmtCond',
                     'BsmtExposure', 'BsmtQual', 'BsmtFinType1', 'BsmtFinType2']
    fill_cols = [col for col in df.columns if col not in nullable_cols]
    df.dropna(axis=0, subset=fill_cols, inplace=True)
    print(f'After drop total row is: {len(df)}')

    df.reset_index(inplace=True, drop=True)
    df['MSSubClass'] = df['MSSubClass'].apply(str)
    object_df = df.select_dtypes(include='object')
    numeric_df = df.select_dtypes(exclude='object')
    df_objects_dummies = enc.transform(object_df)
    df_encoded = pd.concat((numeric_df, pd.DataFrame(df_objects_dummies)),
                           axis=1)
    print(df_encoded.shape)
    assert df_encoded.shape[1] == 304  # 305-1 sale price not included
    return df_encoded
```

In [50]:
```python
df_new = pd.read_csv(raw_data_dir / 'test.csv')
df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1459 entries, 0 to 1458
Data columns (total 80 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1459 non-null   int64
 1   MSSubClass     1459 non-null   int64
 2   MSZoning       1455 non-null   object
 3   LotFrontage    1232 non-null   float64
 4   LotArea        1459 non-null   int64
 5   Street         1459 non-null   object
 6   Alley          107 non-null    object
 7   LotShape       1459 non-null   object
 8   LandContour    1459 non-null   object
 9   Utilities      1457 non-null   object
 10  LotConfig      1459 non-null   object
 11  LandSlope      1459 non-null   object
 12  Neighborhood   1459 non-null   object
 13  Condition1     1459 non-null   object
 14  Condition2     1459 non-null   object
 15  BldgType       1459 non-null   object
 16  HouseStyle     1459 non-null   object
 17  OverallQual    1459 non-null   int64
 18  OverallCond    1459 non-null   int64
 19  YearBuilt      1459 non-null   int64
 20  YearRemodAdd   1459 non-null   int64
 21  RoofStyle      1459 non-null   object
 22  RoofMatl       1459 non-null   object
 23  Exterior1st    1458 non-null   object
 24  Exterior2nd    1458 non-null   object
 25  MasVnrType     1443 non-null   object
 26  MasVnrArea     1444 non-null   float64
 27  ExterQual      1459 non-null   object
 28  ExterCond      1459 non-null   object
 29  Foundation     1459 non-null   object
 30  BsmtQual       1415 non-null   object
 31  BsmtCond       1414 non-null   object
 32  BsmtExposure   1415 non-null   object
 33  BsmtFinType1   1417 non-null   object
 34  BsmtFinSF1     1458 non-null   float64
 35  BsmtFinType2   1417 non-null   object
 36  BsmtFinSF2     1458 non-null   float64
 37  BsmtUnfSF      1458 non-null   float64
 38  TotalBsmtSF    1458 non-null   float64
 39  Heating        1459 non-null   object
 40  HeatingQC      1459 non-null   object
 41  CentralAir     1459 non-null   object
 42  Electrical     1459 non-null   object
 43  1stFlrSF       1459 non-null   int64
 44  2ndFlrSF       1459 non-null   int64
 45  LowQualFinSF   1459 non-null   int64
 46  GrLivArea      1459 non-null   int64
 47  BsmtFullBath   1457 non-null   float64
 48  BsmtHalfBath   1457 non-null   float64
 49  FullBath       1459 non-null   int64
 50  HalfBath       1459 non-null   int64
 51  BedroomAbvGr   1459 non-null   int64
 52  KitchenAbvGr   1459 non-null   int64
 53  KitchenQual    1458 non-null   object
 54  TotRmsAbvGrd   1459 non-null   int64
 55  Functional     1457 non-null   object
 56  Fireplaces     1459 non-null   int64
 57  FireplaceQu    729 non-null    object
 58  GarageType     1383 non-null   object
 59  GarageYrBlt    1381 non-null   float64
 60  GarageFinish   1381 non-null   object
 61  GarageCars     1458 non-null   float64
 62  GarageArea     1458 non-null   float64
 63  GarageQual     1381 non-null   object
 64  GarageCond     1381 non-null   object
 65  PavedDrive     1459 non-null   object
 66  WoodDeckSF     1459 non-null   int64
 67  OpenPorchSF    1459 non-null   int64
 68  EnclosedPorch  1459 non-null   int64
 69  3SsnPorch      1459 non-null   int64
 70  ScreenPorch    1459 non-null   int64
 71  PoolArea       1459 non-null   int64
 72  PoolQC         3 non-null      object
 73  Fence          290 non-null    object
 74  MiscFeature    51 non-null     object
 75  MiscVal        1459 non-null   int64
 76  MoSold         1459 non-null   int64
 77  YrSold         1459 non-null   int64
 78  SaleType       1458 non-null   object
 79  SaleCondition  1459 non-null   object
dtypes: float64(11), int64(26), object(43)
memory usage: 912.0+ KB
```

In [51]:
```python
df_new_encoded = build_features(df_new, enc)
```

```
Before drop total row is: 1459
After drop total row is: 1431
(1431, 304)
```

In [52]:
```python
report_missing(df_new_encoded)
```

```
Total row of dataset is: 1431
```

0

In [53]:
```python
df_new_encoded.to_csv(processed_data_dir / 'test.csv', index=False)
```

In [54]:
```python
import joblib

joblib.dump(enc, '../models/featurebuild/0.1-onehotencoder.joblib')
```

Out[54]: ['../models/featurebuild/0.1-onehotencoder.joblib']