Data Wrangling

Overview

This section describes the various data cleaning and data wrangling methods applied to the "house-prices-advanced-regression-techniques" data.

Summary Files

This dataset is split into two separate files:train and test sets. One file has a missing target value like "SalePrice" so for analysis purposes files were merged into comprehensive data frames.

```
# Import Train and Test Data
file =  '/Users/egoretc/Desktop/'
df3 = pd.read_csv(file + 'test.csv', error_bad_lines=False)
train = '/Users/egoretc/Desktop/'
df1=  pd.read_csv(test + 'train.csv', error_bad_lines=False)
combined = pd.concat([df1, df3],sort=True)
```

original version of the dataset
https://www.kaggle.com/c/house-prices-advanced-regression-techniques.

Duplicates and Missing Values

So observing the dataset quality I discover that it has so many missing values, excluding categorical values. Tested melting, pivoting, unstacking and keeping the original. For easy navigation through my dataset I performed sorting of column names in an alphabetical order.

2ndFlrSF 1460 non-null int64
3SsnPorch 1460 non-null int64
Alley 91 non-null object
If you notice , the number of total values of full columns differ . 1460-91 means I have 1369 missing values for "Alley".

Also, assert pd.notnull(df3).all().all() says it too

Next, I performed a method to fill missing values in a required column and replaced null values with the most common values of the same column. As a result, I am getting these numbers

```
LotArea       1460 non-null int64
Street        1460 non-null object
Alley         1460 non-null object
```

See, Alley is full now

For numerical missing values I am not going to drop them to avoid dropping valuable information but I will fill them with “None” values representing that this row has no data for this column.

Testing for duplicates responds like no duplicates.

Outliers and handling them

Detecting outliers, I performed a test for SalePrices.

```
df3.SalePrice.quantile([0.25,0.5,0.75])
#IQR = 75th percentile-25th percentile
IQR = 214000.0-129975.0
1.5*IQR
#Any number less than this is a suspected outlier.
163000.0-126037.5
#Any number less than this is a suspected outlier.
163000.0+126037.5
```

The boundaries are found so I can remove the data that is out of these boundaries.

```
df4 = df3[df3['SalePrice'] > 36962.5]
df4 = df3[df3['SalePrice'] < 289037.5 ]
```

This is my approach at this point.