index

January 24, 2022

1 Dealing with Categorical Variables - Lab

1.1 Introduction

In this lab, you'll explore the Ames Housing dataset for categorical variables, and you'll transform your data so you'll be able to use categorical data as predictors!

1.2 Objectives

You will be able to: * Determine whether variables are categorical or continuous * Use one hot encoding to create dummy variables * Describe why dummy variables are necessary

1.3 Importing the Ames Housing dataset

Let's start by importing the Ames Housing dataset from `ames.csv` into a pandas dataframe using pandas `read_csv()`

[12]:
```
# Import your data
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline

df = pd.read_csv("ames.csv")
```

Now look at the first five rows of `ames`:

[2]:
```
# Inspect the first few rows
df.head()
```

[2]:
```
   Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
0   1          60       RL         65.0     8450   Pave   NaN      Reg
1   2          20       RL         80.0     9600   Pave   NaN      Reg
2   3          60       RL         68.0    11250   Pave   NaN      IR1
3   4          70       RL         60.0     9550   Pave   NaN      IR1
4   5          60       RL         84.0    14260   Pave   NaN      IR1

  LandContour Utilities  … PoolArea PoolQC Fence MiscFeature MiscVal MoSold  \
0         Lvl    AllPub  …        0    NaN   NaN         NaN       0      2
```

```
1          Lvl    AllPub  ...        0    NaN   NaN          NaN       0      5
2          Lvl    AllPub  ...        0    NaN   NaN          NaN       0      9
3          Lvl    AllPub  ...        0    NaN   NaN          NaN       0      2
4          Lvl    AllPub  ...        0    NaN   NaN          NaN       0     12

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500
3   2006        WD        Abnorml     140000
4   2008        WD         Normal     250000

[5 rows x 81 columns]
```

1.4 Variable Descriptions

Look in `data_description.txt` for a full description of all variables.

A preview of some of the columns:

LotArea: Size of the lot in square feet

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
```

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
```

KitchenQual: Kitchen quality

```
Ex   Excellent
Gd   Good
TA   Typical/Average
```

```
Fa   Fair
Po   Poor
```

YrSold: Year Sold (YYYY)

SalePrice: Sale price of the house in dollars

Let's inspect all features using `.describe()` and `.info()`

[3]:
```python
# Use .describe()
df.describe()
```

[3]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	\
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	
std	421.610009	42.300571	24.284752	9981.264932	1.382997	
min	1.000000	20.000000	21.000000	1300.000000	1.000000	
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	

	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	…	\
count	1460.000000	1460.000000	1460.000000	1452.000000	1460.000000	…	
mean	5.575342	1971.267808	1984.865753	103.685262	443.639726	…	
std	1.112799	30.202904	20.645407	181.066207	456.098091	…	
min	1.000000	1872.000000	1950.000000	0.000000	0.000000	…	
25%	5.000000	1954.000000	1967.000000	0.000000	0.000000	…	
50%	5.000000	1973.000000	1994.000000	0.000000	383.500000	…	
75%	6.000000	2000.000000	2004.000000	166.000000	712.250000	…	
max	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	…	

	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SsnPorch	ScreenPorch	\
count	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	
mean	94.244521	46.660274	21.954110	3.409589	15.060959	
std	125.338794	66.256028	61.119149	29.317331	55.757415	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	25.000000	0.000000	0.000000	0.000000	
75%	168.000000	68.000000	0.000000	0.000000	0.000000	
max	857.000000	547.000000	552.000000	508.000000	480.000000	

	PoolArea	MiscVal	MoSold	YrSold	SalePrice
count	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000
mean	2.758904	43.489041	6.321918	2007.815753	180921.195890
std	40.177307	496.123024	2.703626	1.328095	79442.502883
min	0.000000	0.000000	1.000000	2006.000000	34900.000000
25%	0.000000	0.000000	5.000000	2007.000000	129975.000000
50%	0.000000	0.000000	6.000000	2008.000000	163000.000000

```
75%       0.000000      0.000000     8.000000  2009.000000  214000.000000
max     738.000000  15500.000000    12.000000  2010.000000  755000.000000

[8 rows x 38 columns]
```

[4]:
```python
# Use .info()
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
```

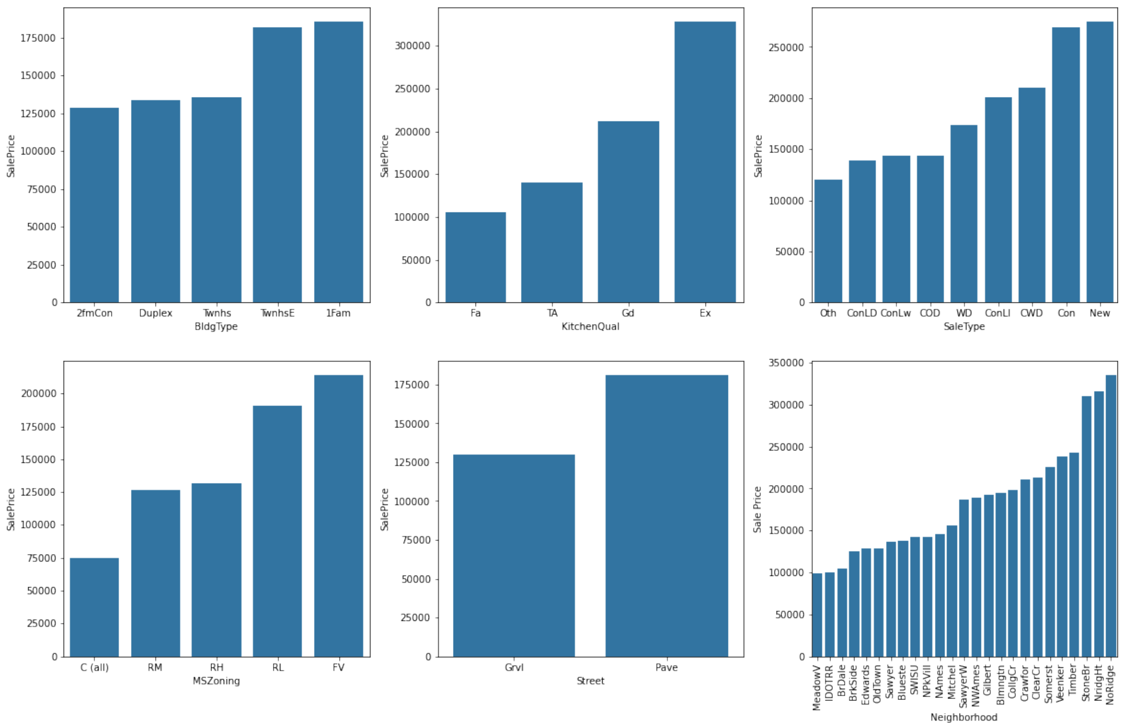
```
 34  BsmtFinSF1     1460 non-null   int64
 35  BsmtFinType2   1422 non-null   object
 36  BsmtFinSF2     1460 non-null   int64
 37  BsmtUnfSF      1460 non-null   int64
 38  TotalBsmtSF    1460 non-null   int64
 39  Heating        1460 non-null   object
 40  HeatingQC      1460 non-null   object
 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
```

1.4.1 Plot Categorical Variables

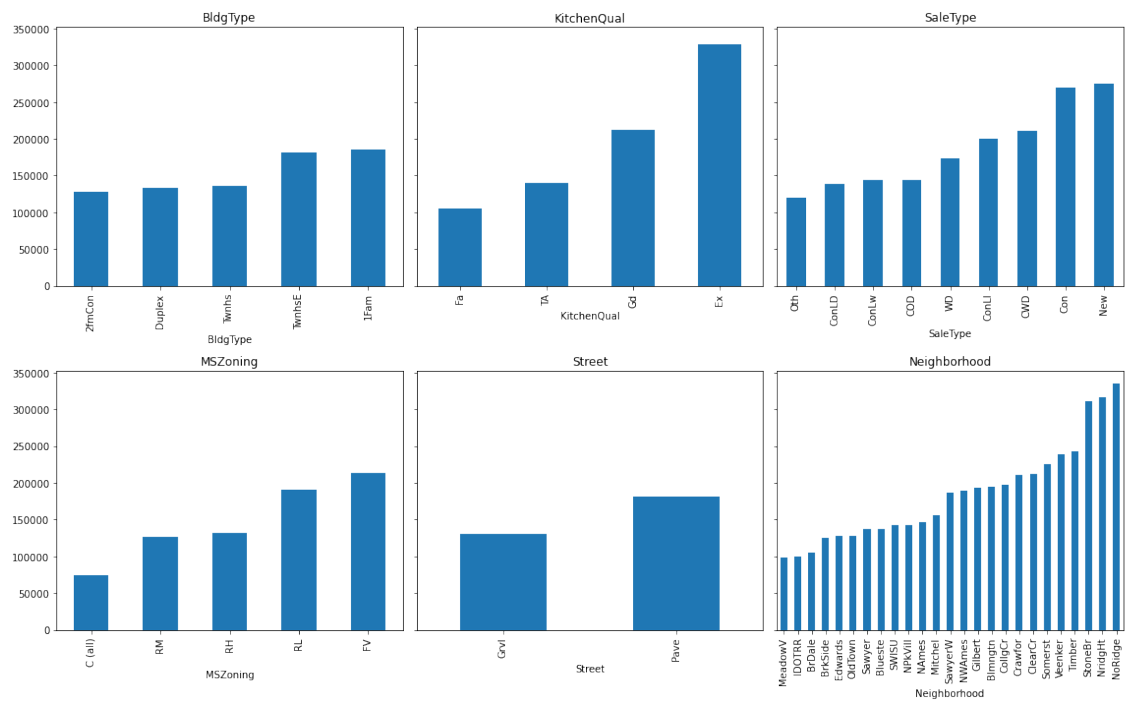
Now, pick 6 categorical variables and plot them against SalePrice with a bar graph for each variable. All 6 bar graphs should be on the same figure.

[5]:
```python
## import matplotlib.pyplot as plt
%matplotlib inline

to_plot = ['BldgType', 'KitchenQual', 'SaleType', 'MSZoning',
           'Street', 'Neighborhood']


fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(16,10))#, sharey=True)
fig.tight_layout(w_pad = 4 , h_pad = 4)
for i,item in enumerate(to_plot):
    grouped = df.groupby(item).SalePrice.mean()
    g = grouped.to_frame()
    g.reset_index(inplace = True)
#     g.sort_values(by= "SalePrice",ascending=False, axis = 1)
    ax = axes[i//3][i%3]
    sns.barplot(x = g[item], y = g["SalePrice"], ax = ax,
                order=g.sort_values("SalePrice")[item],
                color = "tab:blue")
    plt.xlabel(f"{item}")
    plt.xticks(rotation=90)
    plt.ylabel("Sale Price")
#     plt.show()

# Create bar plots
```



[7]:

```python
### From gitHub Solution

import matplotlib.pyplot as plt
%matplotlib inline

fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(16,10), sharey=True)

categoricals = ['BldgType', 'KitchenQual', 'SaleType', 'MSZoning', 'Street', 'Neighborhood']

for col, ax in zip(categoricals, axes.flatten()):
    (df.groupby(col)                  # group values together by column of interest
         .mean()['SalePrice']        # take the mean of the saleprice for each group
         .sort_values()              # sort the groups in ascending order
         .plot
         .bar(ax=ax))                # create a bar graph on the ax

    ax.set_title(col)                # Make the title the name of the column

fig.tight_layout()
```



1.5 Create dummy variables

Create dummy variables for the six categorical features you chose remembering to drop the first. Drop the categorical columns that you used, concat the dummy columns to our continuous variables and asign it to a new variable `ames_preprocessed`

[13]:
```
# Create dummy variables for your six categorical features
df_dummies = pd.get_dummies(df[to_plot], prefix = to_plot, drop_first = True)
```

[14]:
```
df.drop(to_plot, inplace = True, axis = 1)
```

[15]:
```
new_df = pd.concat([df_dummies, df], axis = 1)
new_df.head()
```

[15]:

	BldgType_2fmCon	BldgType_Duplex	BldgType_Twnhs	BldgType_TwnhsE	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	KitchenQual_Fa	KitchenQual_Gd	KitchenQual_TA	SaleType_CWD	SaleType_Con	\
0	0	1	0	0	0	
1	0	0	1	0	0	
2	0	1	0	0	0	

```
3                 0                 1                 0              0              0
4                 0                 1                 0              0              0

   SaleType_ConLD  …  ScreenPorch  PoolArea  PoolQC  Fence  MiscFeature  \
0               0  …            0         0     NaN    NaN          NaN
1               0  …            0         0     NaN    NaN          NaN
2               0  …            0         0     NaN    NaN          NaN
3               0  …            0         0     NaN    NaN          NaN
4               0  …            0         0     NaN    NaN          NaN

   MiscVal  MoSold  YrSold  SaleCondition  SalePrice
0        0       2    2008         Normal     208500
1        0       5    2007         Normal     181500
2        0       9    2008         Normal     223500
3        0       2    2006        Abnorml     140000
4        0      12    2008         Normal     250000

[5 rows x 119 columns]
```

1.6 Summary

In this lab, you practiced your knowledge of categorical variables on the Ames Housing dataset! Specifically, you practiced distinguishing continuous and categorical data. You then created dummy variables using one hot encoding.