

In [1]:

```
import os
import warnings
from pathlib import Path

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from IPython.display import display
from pandas.api.types import CategoricalDtype

from category_encoders import MEstimateEncoder
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.feature_selection import mutual_info_regression
from sklearn.model_selection import KFold, cross_val_score
from xgboost import XGBRegressor

# Set Matplotlib defaults
plt.style.use("seaborn-whitegrid")
plt.rc("figure", autolayout=True)
```



```
plt.style.use("seaborn-whitegrid")
plt.rc("figure", autolayout=True)
plt.rc(
    "axes",
    labelweight="bold",
    labelsize="large",
    titleweight="bold",
    titlesize=14,
    titlepad=10,
)

# Mute warnings
warnings.filterwarnings('ignore')
```

/tmp/ipykernel_20/676408276.py:21: MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated since 3.6, as they no longer correspond to the styles shipped by seaborn. However, they will remain available as 'seaborn-v0_8-`<style>`'. Alternatively, directly use the seaborn API instead.

```
plt.style.use("seaborn-whitegrid")
```

Data Preprocessing

Before we can do any feature engineering, we need to *preprocess* the data to get it in a form suitable for analysis. The data we used in the course was a bit simpler than the competition data. For the *Ames* competition dataset, we'll need to:

- **Load** the data from CSV files
- **Clean** the data to fix any errors or inconsistencies
- **Encode** the statistical data type (numeric, categorical)
- **Impute** any missing values

We'll wrap all these steps up in a function, which will make easy for you to get a fresh dataframe whenever you need. After reading the CSV file, we'll apply three preprocessing steps, `clean`, `encode`, and `impute`, and then create the data splits: one (`df_train`) for training the model, and one (`df_test`) for making the predictions that you'll submit to the competition for scoring on the leaderboard.

In [2]:

```
def load_data():  
    # Read data  
    data_dir = Path("../input/house-prices-advanced-regression-techniques/")  
    df_train = pd.read_csv(data_dir / "train.csv", index_col="Id")  
    df_test = pd.read_csv(data_dir / "test.csv", index_col="Id")  
    # Merge the splits so we can process them together  
    df = pd.concat([df_train, df_test])  
    # Preprocessing  
    df = clean(df)  
    df = encode(df)  
    df = impute(df)  
    # Reform splits  
    df_train = df.loc[df_train.index, :]  
    df_test = df.loc[df_test.index, :]  
    return df_train, df_test
```