In [1]:

```
import pandas as pd
import numpy as np
from sklearn import set_config
set_config(display='diagram')
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import make_pipeline
from sklearn.compose import ColumnTransformer
from sklearn.compose import make_column_selector as selector


ames = pd.read_csv('ames_clean.csv')

targetcol = 'SalePrice'
featurecol = ames.drop(columns=targetcol).select_dtypes(np.number).columns.values

target = ames[targetcol]
features = ames[['GrLivArea', 'YearBuilt']]
```

In [26]:

```
ames.columns
```

Out[26]:

```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

In [14]:

```
featurecol
```

Out[14]:

```
array(['Id', 'MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual',
       'OverallCond', 'YearBuilt', 'YearRemodAdd', 'MasVnrArea',
       'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF',
       '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath',
       'BsmtHalfBath', 'FullBath', 'HalfBath', 'BedroomAbvGr',
       'KitchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces', 'GarageYrBlt',
       'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal',
       'MoSold', 'YrSold'], dtype=object)
```

In [2]:

```
target
```

Out[2]:

```
0       208500
1       181500
2       223500
3       140000
4       250000
         ...
1455    175000
1456    210000
1457    266500
1458    142125
1459    147500
Name: SalePrice, Length: 1460, dtype: int64
```

In [3]:

```
features.head()
```

Out[3]:

	GrLivArea	YearBuilt
0	1710	2003
1	1262	1976
2	1786	2001
3	1717	1915
4	2198	2000

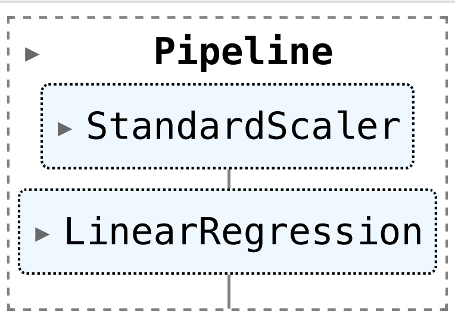
In [4]:

```
x_train, x_test, y_train, y_test = train_test_split(
    features,
    target,
    random_state=123
)
```

In [5]:

```
model = make_pipeline(StandardScaler(), LinearRegression())
model
```

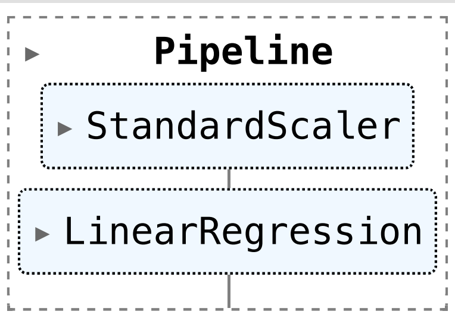
Out[5]:



In [6]:

```
model.fit(x_train, y_train)
```

Out[6]:



In [53]:

```
print('The models accuracy score is', (model.score(x_test, y_test)*100),'%')
```

```
The models accuracy score is 67.10437008209902 %
```

In [42]:

```
features2 = ames[['GrLivArea', 'YearBuilt', 'Neighborhood']]

numerical_column_selector = selector(dtype_exclude=object)
categorical_column_selector = selector(dtype_include=object)

numerical_columns = numerical_column_selector(features2)
categorical_columns = categorical_column_selector(features2)

x_train2, x_test2, y_train2, y_test2 = train_test_split(
    features2,
    target,
    random_state=123
)
```

In [9]:

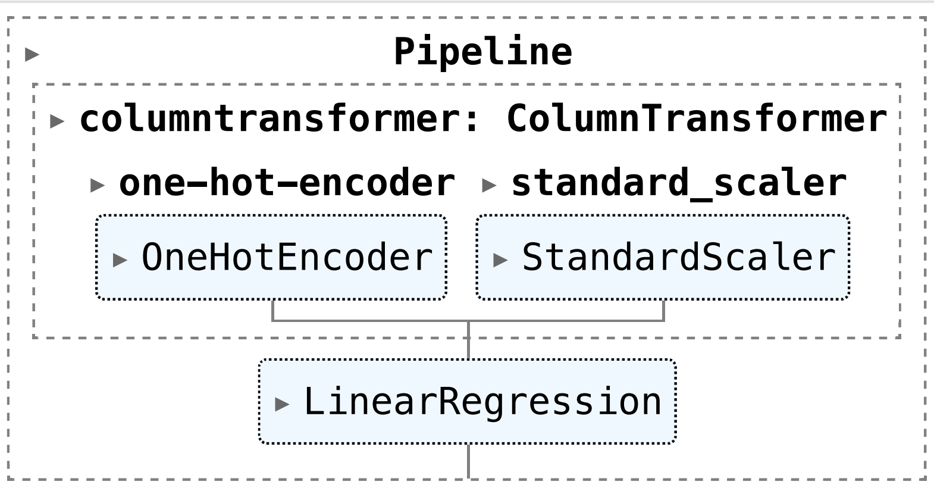
```
categorical_preprocessor = OneHotEncoder(handle_unknown = 'ignore')
numerical_preprocessor = StandardScaler()

preprocessor = ColumnTransformer([
    ('one-hot-encoder', categorical_preprocessor, categorical_columns),
    ('standard_scaler', numerical_preprocessor, numerical_columns)
])
```

In [10]:

```
model2 = make_pipeline(preprocessor, LinearRegression())
model2
```

Out[10]:



In [46]:

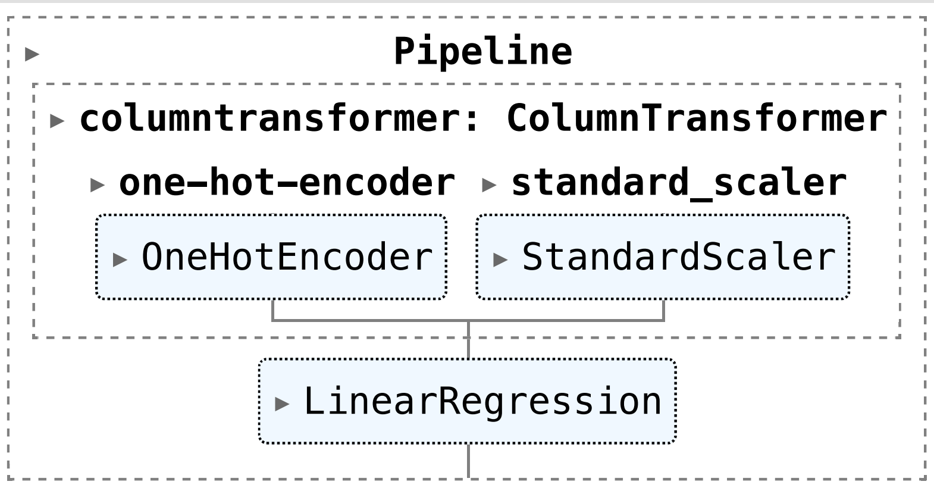
```
print(x_train2.columns)
```

```
Index(['GrLivArea', 'YearBuilt', 'Neighborhood'], dtype='object')
```

In [48]:

```
model2.fit(x_train2, y_train2)
```

Out[48]:



In [55]:

```
print('The models accuracy score is', (model2.score(x_test2, y_test2)*100),'%')
```

```
The models accuracy score is 72.12454069802112 %
```

In [ ]: