In [3]:
```
import numpy as np
import pandas as pd
from sklearn.pipeline import make_pipeline, Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
```

In [2]:
```
df = pd.read_csv('train.csv')
```

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

Out[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	Sa
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	Normal	2
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	Normal	1
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	Normal	2
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	Abnorml	1
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	Normal	2

5 rows × 81 columns

In [4]:
```
df.columns
```

Out[4]:
```
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 [5]:
```
select_df = df[['MSSubClass',   'MSZoning',     'LotFrontage',  'LotArea',       'Street', 'LotShape',   'LandContour',
          'Utilities', 'MiscVal',         'MoSold',       'YrSold',       'SaleType', 'SalePrice']].dropna()
```

In [6]:
```
from numpy.lib.function_base import select
X  = pd.get_dummies(select_df.drop('SalePrice', axis=1))
y = select_df.SalePrice
```

In [7]:
```
X.head()
```

Out[7]:

	MSSubClass	LotFrontage	LotArea	MiscVal	MoSold	YrSold	MSZoning_C (all)	MSZoning_FV	MSZoning_RH	MSZoning_RL	...	Utilities_AllPub	SaleType_COD	SaleType_CWD	SaleType_Con	SaleType_
0	60	65.0	8450	0	2	2008	0	0	0	1	...	1	0	0	0	
1	20	80.0	9600	0	5	2007	0	0	0	1	...	1	0	0	0	
2	60	68.0	11250	0	9	2008	0	0	0	1	...	1	0	0	0	
3	70	60.0	9550	0	2	2006	0	0	0	1	...	1	0	0	0	
4	60	84.0	14260	0	12	2008	0	0	0	1	...	1	0	0	0	

5 rows × 31 columns

In [8]:
```
X.shape
```

Out[8]:
```
(1201, 31)
```

In [9]:
```
pipeline = make_pipeline(StandardScaler(), RandomForestRegressor())
```

In [10]:
```
pipeline.fit(X, y)
```

Out[10]:
```
Pipeline(steps=[('standardscaler', StandardScaler()),
                ('randomforestregressor', RandomForestRegressor())])
```

In [11]:
```
pipeline.predict(X)
```

Out[11]:
```
array([202752.  , 164191.  , 217367.5 , ..., 230036.03, 145643.75,
       151905.5 ])
```

In [12]:
```
import pickle
```

In [13]:
```
with open('pipelinemodel.pkl', 'wb') as f:
  pickle.dump(pipeline,f)
```

In [14]:
```
with open('pipelinemodel.pkl', 'rb') as f:
  reloaded_model = pickle.load(f)
```

In [15]:
```
reloaded_model
```

Out[15]:
```
Pipeline(steps=[('standardscaler', StandardScaler()),
                ('randomforestregressor', RandomForestRegressor())])
```

In [16]:
```
reloaded_model.named_steps
```

Out[16]:
```
{'standardscaler': StandardScaler(),
 'randomforestregressor': RandomForestRegressor()}
```

In [17]:
```
reloaded_model.steps[1][1].predict(X)
```

Out[17]:
```
C:\Users\steph\anaconda3\lib\site-packages\sklearn\base.py:413: UserWarning: X has feature names, but RandomForestRegressor was fitted without feature names
  warnings.warn(
array([352391.25, 352391.25, 367792.25, ..., 352091.25, 352391.25,
       352391.25])
```

Using the Pipeline Class

In [18]:
```
#with Pipeline Class
custom_pipeline = Pipeline([('Scaling', StandardScaler()), ('rfmodel', RandomForestRegressor())])
```

In [19]:
```
custom_pipeline
```

Out[19]:
```
Pipeline(steps=[('scaling', StandardScaler()),
                ('rfmodel', RandomForestRegressor())])
```

In [20]:
```
#with the make_pipeline class
make_pipeline_model = make_pipeline(StandardScaler(), RandomForestRegressor())
```

In [21]:
```
make_pipeline_model
```

Out[21]:
```
Pipeline(steps=[('standardscaler', StandardScaler()),
                ('randomforestregressor', RandomForestRegressor())])
```

Column transformers

In [22]:
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
```

In [23]:
```
X.select_dtypes('object')
```

Out[23]:

0
1
2
3
4
...
1455
1456
1457
1458
1459

1201 rows × 0 columns

In [24]:
```
X.select_dtypes('object').columns
```

Out[24]:
```
Index([], dtype='object')
```

In [25]:
```
#What the Numeric Features
numeric_features = X.select_dtypes(exclude='object').columns
numeric_features
```

Out[25]:
```
Index(['MSSubClass', 'LotFrontage', 'LotArea', 'MiscVal', 'MoSold', 'YrSold',
       'MSZoning_C (all)', 'MSZoning_FV', 'MSZoning_RH', 'MSZoning_RL',
       'MSZoning_RM', 'Street_Grvl', 'Street_Pave', 'LotShape_IR1',
       'LotShape_IR2', 'LotShape_IR3', 'LotShape_Reg', 'LandContour_Bnk',
       'LandContour_HLS', 'LandContour_Low', 'LandContour_Lvl',
       'Utilities_AllPub', 'SaleType_COD', 'SaleType_CWD', 'SaleType_Con',
       'SaleType_ConLD', 'SaleType_ConLI', 'SaleType_ConLw', 'SaleType_New',
       'SaleType_Oth', 'SaleType_WD'],
      dtype='object')
```

In [26]:
```
numeric_pipeline = Pipeline([('scaler', StandardScaler())])
```

In [27]:
```
#Categorical features
categorical_features = X.select_dtypes('object').columns
categorical_features
```

Out[27]:
```
Index([], dtype='object')
```

In [28]:
```
categorical_pipeline = Pipeline([('onehot', OneHotEncoder())])
```

In [29]:
```
preprocessor = ColumnTransformer([('num', numeric_pipeline, numeric_features),
                                  ('cat', categorical_pipeline, categorical_features)])
```

In [30]:
```
preprocessor
```

Out[30]:
```
ColumnTransformer(transformers=[('num',
                                 Pipeline(steps=[('scaler', StandardScaler())]),
                                 Index(['MSSubClass', 'LotFrontage', 'LotArea', 'MiscVal', 'MoSold', 'YrSold',
       'MSZoning_C (all)', 'MSZoning_FV', 'MSZoning_RH', 'MSZoning_RL',
       'MSZoning_RM', 'Street_Grvl', 'Street_Pave', 'LotShape_IR1',
       'LotShape_IR2', 'LotShape_IR3', 'LotShape_Reg', 'LandContour_Bnk',
       'LandContour_HLS', 'LandContour_Low', 'LandContour_Lvl',
       'Utilities_AllPub', 'SaleType_COD', 'SaleType_CWD', 'SaleType_Con',
       'SaleType_ConLD', 'SaleType_ConLI', 'SaleType_ConLw', 'SaleType_New',
       'SaleType_Oth', 'SaleType_WD'],
      dtype='object')),
                                ('cat',
                                 Pipeline(steps=[('onehot', OneHotEncoder())]),
                                 Index([], dtype='object'))])
```

In [31]:
```
ml_pipeline = Pipeline([('preprocessor', preprocessor), ('classifier', RandomForestRegressor())])
```

In [32]:
```
ml_pipeline
```

Out[32]:
```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(transformers=[('num',
                                                  Pipeline(steps=[('scaler',
                                                                   StandardScaler())]),
                                                  Index(['MSSubClass', 'LotFrontage', 'LotArea', 'MiscVal', 'MoSold', 'YrSold',
       'MSZoning_C (all)', 'MSZoning_FV', 'MSZoning_RH', 'MSZoning_RL',
       'MSZoning_RM', 'Street_Grvl', 'Street_Pave', 'LotShape_IR1',
       'LotShape_IR2', 'LotShape_IR3', 'LotShape_Reg', 'LandContour_Bnk',
       'LandContour_HLS', 'LandContour_Low', 'LandContour_Lvl',
       'Utilities_AllPub', 'SaleType_COD', 'SaleType_CWD', 'SaleType_Con',
       'SaleType_ConLD', 'SaleType_ConLI', 'SaleType_ConLw', 'SaleType_New',
       'SaleType_Oth', 'SaleType_WD'],
      dtype='object')),
                                                 ('cat',
                                                  Pipeline(steps=[('onehot',
                                                                   OneHotEncoder())]),
                                                  Index([], dtype='object'))])),
                ('classifier', RandomForestRegressor())])
```

In [33]:
```
ml_pipeline.fit(X,y)
```

Out[33]:
```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(transformers=[('num',
                                                  Pipeline(steps=[('scaler',
                                                                   StandardScaler())]),
                                                  Index(['MSSubClass', 'LotFrontage', 'LotArea', 'MiscVal', 'MoSold', 'YrSold',
       'MSZoning_C (all)', 'MSZoning_FV', 'MSZoning_RH', 'MSZoning_RL',
       'MSZoning_RM', 'Street_Grvl', 'Street_Pave', 'LotShape_IR1',
       'LotShape_IR2', 'LotShape_IR3', 'LotShape_Reg', 'LandContour_Bnk',
       'LandContour_HLS', 'LandContour_Low', 'LandContour_Lvl',
       'Utilities_AllPub', 'SaleType_COD', 'SaleType_CWD', 'SaleType_Con',
       'SaleType_ConLD', 'SaleType_ConLI', 'SaleType_ConLw', 'SaleType_New',
       'SaleType_Oth', 'SaleType_WD'],
      dtype='object')),
                                                 ('cat',
                                                  Pipeline(steps=[('onehot',
                                                                   OneHotEncoder())]),
                                                  Index([], dtype='object'))])),
                ('classifier', RandomForestRegressor())])
```

In [34]:
```
ml_pipeline.predict(X)
```

Out[34]:
```
array([204270.5 , 161552.  , 219448.36, ..., 229845.5 , 149212.25,
       151057.5 ])
```

In [35]:
```
with open('columntransformermodel.pkl', 'wb') as f:
  pickle.dump(ml_pipeline,f)
```

In [36]:
```
with open('columntransformermodel.pkl', 'rb') as f:
  reloaded_ml_pipeline = pickle.load(f)
```

In [37]:
```
reloaded_ml_pipeline
```

Out[37]:
```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(transformers=[('num',
                                                  Pipeline(steps=[('scaler',
                                                                   StandardScaler())]),
                                                  Index(['MSSubClass', 'LotFrontage', 'LotArea', 'MiscVal', 'MoSold', 'YrSold',
       'MSZoning_C (all)', 'MSZoning_FV', 'MSZoning_RH', 'MSZoning_RL',
       'MSZoning_RM', 'Street_Grvl', 'Street_Pave', 'LotShape_IR1',
       'LotShape_IR2', 'LotShape_IR3', 'LotShape_Reg', 'LandContour_Bnk',
       'LandContour_HLS', 'LandContour_Low', 'LandContour_Lvl',
       'Utilities_AllPub', 'SaleType_COD', 'SaleType_CWD', 'SaleType_Con',
       'SaleType_ConLD', 'SaleType_ConLI', 'SaleType_ConLw', 'SaleType_New',
       'SaleType_Oth', 'SaleType_WD'],
      dtype='object')),
                                                 ('cat',
                                                  Pipeline(steps=[('onehot',
                                                                   OneHotEncoder())]),
                                                  Index([], dtype='object'))])),
                ('classifier', RandomForestRegressor())])
```

In [ ]:
```
```