



```
In [200]: model_xgb = xgb.XGBRegressor(n_estimators=360, max_depth=2, learning_rate=0.1)
model_xgb.fit(X_train.loc[:, 'MSSubClass': 'SaleCondition_Partial'], y)
```

```
Out[200]: XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
                        colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
                        early_stopping_rounds=None, enable_categorical=False,
                        eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
                        importance_type=None, interaction_constraints='',
                        learning_rate=0.1, max_bin=256, max_cat_to_onehot=4,
                        max_delta_step=0, max_depth=2, max_leaves=0, min_child_weight=1,
                        missing=nan, monotone_constraints='()', n_estimators=360, n_jobs=0,
                        num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
                        reg_lambda=1, ...)
```

```
In [204]: model_lasso = LassoCV(alphas = [0.0005]).fit(X_train.loc[:, 'MSSubClass': 'SaleCondition_Partial'], y)
xgb_preds = np.expm1(model_xgb.predict(X_test))
lasso_preds = np.expm1(model_lasso.predict(X_test))
```

```
In [205]: predictions = pd.DataFrame({"xgb": xgb_preds, "lasso": lasso_preds})
predictions.plot(x = "xgb", y = "lasso", kind = "scatter")
```

```
Out[205]: <AxesSubplot:xlabel='xgb', ylabel='lasso'>
```

