1. Total SF – 1stFloorSF + 2ndFloorSF + BasementSF
   * total\_bldg\_sf = cleaned\_df[["totalbsmtsf", "1stflrsf", "2ndflrsf"]].sum(axis = "columns")
2. Total Baths – FullBath + HalfBath/2 + BsmtFullPath + BsmtHalfPath/2
   * total\_baths = cleaned\_df["fullbath"]  + cleaned\_df["halfbath"]/ 2 + cleaned\_df["bsmtfullbath"] + cleaned\_df["bsmthalfbath"] / 2
3. Outside SF
   * Combination of outside square footage
   * outside\_sf = cleaned\_df[["wooddecksf", "openporchsf", "3ssnporch", "screenporch"]].sum(axis="columns")
4. Abnormal Sale
   * Does sale condition = Abnormal (e.g., it was a foreclosure)
   * abnormal\_sale = (cleaned\_df["salecondition"] *==* "Abnorml").map({False: 0, True: 1}).fillna(0)
5. Top 3 Neighborhood (by median sale value/boxplot inspection)
   * top\_3\_nbrhd = cleaned\_df["neighborhood"].isin(["NridgHt", "NoRidge", "StoneBr"]).map({False: 0, True: 1})
6. Bottom 5 Neighborhood (by median sale value/boxplot inspection)
   * btm\_5\_nbrhd = cleaned\_df["neighborhood"].isin(["MeadowV", "IDOTRR", "BrDale", "OldTown", "Edwards"]).map({False: 0, True: 1})
7. OverallQualityAdjustment
   * Anything rated in 1, 2 or 3 -> 0
   * So scale is 0, 1, 2, 3, 4, 5, 6, 7 [0 is the “combination” of values in 1, 2, and 3 because those houses displayed medium values]
   * overall\_qual\_adj = cleaned\_df["overallqual"].apply(*lambda* x: 0 *if* x *<=*3 *else* x - 3)
8. Good Amenities Count
   * High ratings re: fireplace, basement, kitchen and heating + the basement has a GLQ (good living quarters)
   * good\_ament\_ct = pd.concat([excl\_heating, excl\_bsmt, ktch\_dummies, bsmt\_gd\_lvg, good\_frplc]
     1. Excellent, Good or Typical fireplace good\_frplc = cleaned\_df["fireplacequ"].isin(["Ex", "Gd", "TA"]).map({False: 0, True: 1})
     2. ktch\_groups = cleaned\_df["kitchenqual"].map({"TA": "ktch\_okay", "Fa": "ktch\_okay", "Gd": "ktch\_good", "Ex": "kitch\_topnotch"})
        1. ktch\_dummies = pd.get\_dummies(data=ktch\_groups).drop("ktch\_okay", axis="columns")
     3. excl\_bsmt = cleaned\_df["bsmtqual"].isin(["Ex"]).map({False: 0, True:1})
     4. bsmt\_gd\_lvg = (cleaned\_df["bsmtfintype1"].isin(["GLQ"]) | cleaned\_df["bsmtfintype2"].isin(["GLQ"])).map({False: 0, True:1})
     5. excl\_heating = cleaned\_df["heatingqc"].isin(["Ex"]).map({False: 0, True:1})
9. Bad Amenities Count
   * No Central AC, No Fireplace, or Bad Electrical
   * bad\_ament\_ct = pd.concat([no\_central\_ac, no\_fireplace, bad\_electrical], axis="columns").sum(axis="columns")
   * no\_fireplace = cleaned\_df["fireplaces"] *==* 0
     1. no\_fireplace = no\_fireplace.map({False: 0, True: 1})
   * no\_central\_ac = cleaned\_df['centralair'].isin(['N']).map({False: 0, True:1})
   * bad\_electrical = cleaned\_df['electrical'].isin(['Mix', 'FuseP', 'FuseF', 'FuseA']).map({False: 0, True:1})
10. New Home
    * Homes that are less than 5 years difference between year sold and year built
    * homeage = cleaned\_df["yrsold"] - cleaned\_df["yearbuilt"]
    * newHome = homeage *<* 5
    * newHome = newHome.map({False: 0, True: 1})
11. Remodel\_Age
    * Age of hold relative to last remodel (or year built if not remodeled)
    * remodelage = cleaned\_df["yrsold"] - cleaned\_df["yearremodadd"]

***Import/short data prep steps***

*import* janitor *as* jn

*import* pandas *as* pd

*import* numpy *as* np

%pylab inline

import\_list = ['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Alley','LotShape', 'Neighborhood', 'Condition1',

              'Condition2', 'BldgType', 'HouseStyle','OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',

              'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea', 'BsmtQual', 'BsmtCond', 'BsmtExposure',

              'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2','BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'CentralAir',

              'Electrical', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath',

              'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd', 'Fireplaces',

              'FireplaceQu', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch','ScreenPorch', 'YrSold',

              'SaleType', 'SaleCondition','Electrical',"HeatingQC","Fireplaces", "FireplaceQu", "BsmtQual", "BsmtFinType1",

              "BsmtFinType2", 'SalePrice']

initial\_df = pd.read\_csv(file\_path, header=0, usecols=import\_list, index\_col=0)

cleaned\_df = jn.clean\_names(initial\_df) # converts columns names to be lowercase

cleaned\_df = cleaned\_df.drop([1299, 524], axis="rows") *# remove outliers based on Pre\_process draft based upon Cook's distance > 3 times*

*# mean absolute average*

***Example of how combined***

y = cleaned\_df["saleprice"]

X = pd.concat([total\_bldg\_sf, total\_baths, good\_ament\_ct, btm\_5\_nbrhd, newHome, overall\_qual\_adj, bad\_ament\_ct, abnormal\_sale, outside\_sf],

    axis="columns")

X.columns = ["total\_bldg\_sf", "total\_baths", "good\_ament\_ct", "btm\_5\_nbrhd", "newHome", "overall\_qual\_adj", "bad\_ament\_ct", "abnormal\_sale", "outside\_sf"]

**Lasso Results:** Descending order of importance; home\_age & MSZoning\_RM not in forward/back top 10

|  |  |  |
| --- | --- | --- |
| Rank | Lasso Features | Description |
| 1 | OverallQuall |  |
| 2 | total\_sqft | data['TotalBsmtSF'] + data['1stFlrSF'] + data['2ndFlrSF'] |
| 3 | GarageCars |  |
| 3 | total\_bath | data['total\_bath'] = data['BsmtFullBath'] + data['FullBath'] + (data['BsmtHalfBath'] \* 0.5) + (data['HalfBath'] \* 0.5) |
| 4 | Fireplaces | ##has fireplace (y = 1 / n = 0)  data['Fireplaces'] = data['Fireplaces'].apply(lambda x: 0 if x == 0 else 1) |
| 4 | home\_age | data['YrSold'] - data['YearBuilt'] |
| 5 | MSZoning\_RM | OHE MSZoning |

**Forward/Backward Selection:** Differences are total\_bath vs LotFrontage

|  |  |
| --- | --- |
| Backward AIC = -4642.23 | |
| Feature (In order of importance) | Description |
| LotFrontage | data['LotFrontage'] = data['LotFrontage'].fillna(data['LotFrontage'].median()) |
| LotArea |  |
| BldgType | ##makes building type binary (single fam or not)  data['BldgType'] = data['BldgType'].apply(lambda x: 0 if x == '1Fam' else 1) |
| OverallQual |  |
| OverallCond |  |
| BedroomAbvGr |  |
| Fireplaces | ##has fireplace (y = 1 / n = 0)  data['Fireplaces'] = data['Fireplaces'].apply(lambda x: 0 if x == 0 else 1) |
| GarageCars |  |
| total\_sqft | data['TotalBsmtSF'] + data['1stFlrSF'] + data['2ndFlrSF'] |
| Deck\_sqft   * This is different than Jason’s | data['deck\_sqft'] = data['WoodDeckSF'] + data['OpenPorchSF'] + \  data['EnclosedPorch'] + data['3SsnPorch'] + data['ScreenPorch'] |

|  |  |
| --- | --- |
| Backward BIC = -4463.32 | |
| Feature (In order of importance) | Description |
| LotArea |  |
| BldgType | ##makes building type binary (single fam or not)  data['BldgType'] = data['BldgType'].apply(lambda x: 0 if x == '1Fam' else 1) |
| OverallQual |  |
| OverallCond |  |
| BedroomAbvGr |  |
| Fireplaces | ##has fireplace (y = 1 / n = 0)  data['Fireplaces'] = data['Fireplaces'].apply(lambda x: 0 if x == 0 else 1) |
| GarageCars |  |
| total\_sqft | data['TotalBsmtSF'] + data['1stFlrSF'] + data['2ndFlrSF'] |
| Deck\_sqft  This is different than Jason’s | data['deck\_sqft'] = data['WoodDeckSF'] + data['OpenPorchSF'] + \  data['EnclosedPorch'] + data['3SsnPorch'] + data['ScreenPorch'] |
| Total\_bath | data['total\_bath'] = data['BsmtFullBath'] + data['FullBath'] + (data['BsmtHalfBath'] \* 0.5) + (data['HalfBath'] \* 0.5) |