Supervised Learning: Regression

1) Main objective of the analysis that specifies whether your model will be focused on prediction or interpretation.

The goal I was aiming to achieve was to build a model of best possible prediction capabilities.
Since there is a negative relation between interpretation and said prediction, interpretability might have had suffer.
Hence to above as the best model I have decised to choose the one which exhibits the lowest error along with lowest possible number of predictors used (curse of dimensionality)

2) Brief description of the data set you chose and a summary of its attributes.

The Ames Housing dataset was compiled by Dean De Cock for use in data science education.
[http://jse.amstat.org/v19n3/decock.pdf (http://jse.amstat.org/v19n3/decock.pdf)](http://jse.amstat.org/v19n3/decock.pdf)
Altogether the data set is made of 2920 rows and 81 columns.
The data set was already splitted into Train and Test sets since it comes directly from Kaggle Competition:
[https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview)](https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview)
Both the Test and Train data sets consists of 1460 rows each.
For the needs of Kaggle competition the Test split was stripped from "SalePrice" column.
Hence to above the performance of any model build on Ames Housing dataset is tested on a hold out split stored on a Kaggle server.
The data set exhibits 19 variables with NULL values needed to be dealt with.
The data set exhibits skew variables needed to be transformed.
The distribution of Target is positively skewed.

In [206]:

```
train = pd.read_csv('train.csv')
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
Id               1460 non-null int64
MSSubClass       1460 non-null int64
MSZoning         1460 non-null object
LotFrontage      1201 non-null float64
LotArea          1460 non-null int64
Street           1460 non-null object
Alley            91 non-null object
LotShape         1460 non-null object
LandContour      1460 non-null object
Utilities        1460 non-null object
LotConfig        1460 non-null object
LandSlope        1460 non-null object
Neighborhood     1460 non-null object
Condition1       1460 non-null object
Condition2       1460 non-null object
BldgType         1460 non-null object
HouseStyle       1460 non-null object
OverallQual      1460 non-null int64
OverallCond      1460 non-null int64
YearBuilt        1460 non-null int64
YearRemodAdd     1460 non-null int64
RoofStyle        1460 non-null object
RoofMatl         1460 non-null object
Exterior1st      1460 non-null object
Exterior2nd      1460 non-null object
MasVnrType       1452 non-null object
MasVnrArea       1452 non-null float64
ExterQual        1460 non-null object
ExterCond        1460 non-null object
Foundation       1460 non-null object
BsmtQual         1423 non-null object
BsmtCond         1423 non-null object
BsmtExposure     1422 non-null object
BsmtFinType1     1423 non-null object
BsmtFinSF1       1460 non-null int64
BsmtFinType2     1422 non-null object
BsmtFinSF2       1460 non-null int64
BsmtUnfSF        1460 non-null int64
TotalBsmtSF      1460 non-null int64
Heating          1460 non-null object
HeatingQC        1460 non-null object
CentralAir       1460 non-null object
Electrical       1459 non-null object
1stFlrSF         1460 non-null int64
2ndFlrSF         1460 non-null int64
LowQualFinSF     1460 non-null int64
GrLivArea        1460 non-null int64
BsmtFullBath     1460 non-null int64
BsmtHalfBath     1460 non-null int64
FullBath         1460 non-null int64
HalfBath         1460 non-null int64
BedroomAbvGr     1460 non-null int64
KitchenAbvGr     1460 non-null int64
KitchenQual      1460 non-null object
```

```
TotRmsAbvGrd     1460 non-null int64
Functional       1460 non-null object
Fireplaces       1460 non-null int64
FireplaceQu      770 non-null object
GarageType       1379 non-null object
GarageYrBlt      1379 non-null float64
GarageFinish     1379 non-null object
GarageCars       1460 non-null int64
GarageArea       1460 non-null int64
GarageQual       1379 non-null object
GarageCond       1379 non-null object
PavedDrive       1460 non-null object
WoodDeckSF       1460 non-null int64
OpenPorchSF      1460 non-null int64
EnclosedPorch    1460 non-null int64
3SsnPorch        1460 non-null int64
ScreenPorch      1460 non-null int64
PoolArea         1460 non-null int64
PoolQC           7 non-null object
Fence            281 non-null object
MiscFeature      54 non-null object
MiscVal          1460 non-null int64
MoSold           1460 non-null int64
YrSold           1460 non-null int64
SaleType         1460 non-null object
SaleCondition    1460 non-null object
SalePrice        1460 non-null int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

More details on GitHub (https://github.com/KonuTech/house-prices-advanced-regression-techniques/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning.ipynb)

3) Brief summary of data exploration and actions taken for data cleaning and feature engineering.

The first challenge of the Ames Housing data set was to deal with missing values.
The challenge has been tackled with a use of methods provided by scikit-learn imputation algorithms.
Variables with imputed values have been visualised before and after imputation to assess the meaningfullness of chosen methods.
For categorical variables the SimpleImputer have been chosen as adequate method of data imputation.
Appart from mentioned SimpleImputer in case of numerical values the IterativeImputer and KNNImputer were applied to compare distributions after and before imputation before jumping straight forward into development of a model.
Polynomial features and interactions have been derived for numeric columns to fullfil a need for Feature Engineering.
Above step was continued by computation of multiple variables expressing the sizes of deviations within each group of characteristics.
The deviation was defined as a substraction of a group mean from each one of indiviudual record being a member of that group, divided by a standard deviation of that group.
In case of variables showing skewness larger than 0.75 the np.Log1p transformation was applied to centre the distributions of variables towards it's modes.
The transformation returned natural logarithm of one plus the input.
Numeric variables were scaled with a use of Robust Scaler from scikit-learn.
Pandas get_dummies method was used to derive dummy variables from categorical variables.

More details on [GitHub (https://github.com/KonuTech/house-prices-advanced-regression-techniques/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning.ipynb)](https://github.com/KonuTech/house-prices-advanced-regression-techniques/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning.ipynb)

4) Summary of training at least three linear regression models which should be variations that cover using a simple linear regression as a baseline, adding polynomial effects, and using a regularization regression. Preferably, all use the same training and test splits, or the same cross-validation method.

Multiple models have been built to predict house prices.
For each one of algorithms one-hot encoding was used to transform character variables.
As an initial step of Feature Selection process Variance Threshold have been used to remove variables with little variance.
Feature Selection process was later on expanded by Univariate Feature Selection step and by Variance Inflation Factor cutoff what resulted in a reduction of inputs to 21.

Since the version of Ames Housing data set I used comes from Kaggle the accuracy of predictions for Test data set was assessed using hold out values of "SalePrice" stored on Kaggle server.
To compare models of competitors Root Mean Squared Logarithmic Error have been chosen as a performance measure. Lower the value of RMSLE the better.
Hence to above my model of:

- Linear Regression with Polynomials scored on test data set: 0.40747
- Linear Regression with Polynomials and LASSO Regularization scored: 0.40747
- Ridge Regression scored: 0.39873
- Gradient Boosted Regressor scored: 0.22508
- Random Forest scored: 0.20553

In case of classic Regressors the sums of coefficients (magnitute of coefficients) looked as follows:

- Linear Regression sum of coefficients: 103839.7599411368
  - number of coefficients not equal to zero: 21

- LASSO regularization sum of coefficients: 103834.67779647505
  - number of coefficients not equal to zero: 21

- Ridge Regression sum of coefficients: 100270.2512550827
  - number of coefficients not equal to zero: 21

More details on [GitHub (https://github.com/KonuTech/house-prices-advanced-regression-techniques/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning.ipynb)](https://github.com/KonuTech/house-prices-advanced-regression-techniques/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning.ipynb)

5) A paragraph explaining which of your regressions you recommend as a final model that best fits your needs in terms of accuracy and explainability.

Since major goal of analysis is prediction I recommend as final model the one with lowest Root Mean Squared Logarithmic Error on hold out Test data set. That is Random Forest model.

6) Summary Key Findings and Insights, which walks your reader through the main drivers of your model and insights from your data derived from your linear regression model.

From calculation of Feature Importance for each one of parameters per each one of models we can get into conclusion which variables have the strongest impact on house predictions.
By so called Majority Voting we can tell that:

- HouseStyle_DEVIATION_GrLivArea happen to be twice the most impactful predictor
    - Ground living area square feet in a relation to the style of house
- Electrical_DEVIATION_YearBuilt happend to be twice the second most impactful predictor
    - Year of built in a relation to electrical system
- HouseStyle_DEVIATION_TotalBsmtSF seems to be on third place
    - Total square feet of basement area in a relation to the style of house
- both ExterQual_DEVIATION_OverallQual and Exterior1st_DEVIATION_GrLivArea deserve to be mention here as well
    - Rates of the house in a relation to the quality of the material on the exterior

All of these variables were calculated during Feature Engineering process to include deviations of measures within all groups of all characteristics.
Deviations derived that way are expresing relations between numerical variables and groupings of cathegorical varibales.

Feature Importance of Ridge Regression

In [217]:

```
ridge_coefs_df = pd.DataFrame(dict(score=best_ridge.coef_, column=X_test.columns))
ridge_coefs_df.sort_values(['score'], ascending=False).head(10)
```

Out[217]:

	score	column
1	15681.591	Electrical_DEVIATION_YearBuilt
19	14357.359	HouseStyle_DEVIATION_TotalBsmtSF
14	11456.354	FireplaceQu_DEVIATION_TotalBsmtSF
16	9080.024	BsmtQual_DEVIATION_OverallQual
15	7905.632	LotShape_DEVIATION_GrLivArea
6	6133.478	ExterQual_DEVIATION_OverallQual
13	4592.795	Exterior1st_DEVIATION_GrLivArea
9	4489.103	FireplaceQu_DEVIATION_GrLivArea
17	3443.197	HouseStyle_DEVIATION_GrLivArea
10	2898.513	x23 x27

Feature Importance of Random Forest

In [218]:

```
pd.DataFrame(dict(score=best_forest.feature_importances_, column=X_test.columns)).sort_val
ues(['score'], ascending=False).head(10)
```

Out[218]:

	score	column
17	0.393	HouseStyle_DEVIATION_GrLivArea
1	0.166	Electrical_DEVIATION_YearBuilt
19	0.079	HouseStyle_DEVIATION_TotalBsmtSF
6	0.037	ExterQual_DEVIATION_OverallQual
16	0.035	BsmtQual_DEVIATION_OverallQual
18	0.034	HouseStyle_DEVIATION_1stFlrSF
13	0.031	Exterior1st_DEVIATION_GrLivArea
3	0.025	MasVnrType_DEVIATION_GrLivArea
14	0.024	FireplaceQu_DEVIATION_TotalBsmtSF
9	0.022	FireplaceQu_DEVIATION_GrLivArea

Feature Importance of Gradient Boosted Regressor

In [219]:

```
pd.DataFrame(dict(score=best_gbr.feature_importances_, column=X_test.columns)).sort_values
(['score'], ascending=False).head(10)
```

Out[219]:

	score	column
17	0.324	HouseStyle_DEVIATION_GrLivArea
1	0.153	Electrical_DEVIATION_YearBuilt
19	0.125	HouseStyle_DEVIATION_TotalBsmtSF
6	0.064	ExterQual_DEVIATION_OverallQual
13	0.048	Exterior1st_DEVIATION_GrLivArea
9	0.037	FireplaceQu_DEVIATION_GrLivArea
16	0.034	BsmtQual_DEVIATION_OverallQual
14	0.023	FireplaceQu_DEVIATION_TotalBsmtSF
18	0.022	HouseStyle_DEVIATION_1stFlrSF
10	0.020	x23 x27

More details on [GitHub (https://github.com/KonuTech/house-prices-advanced-regression-techniques/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning.ipynb)](https://github.com/KonuTech/house-prices-advanced-regression-techniques/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning.ipynb)

7) Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model adding specific data features to achieve a better explanation or a better prediction.

Random Forest and Gradient Boosted Regressor models already improved predictions of poorly performing linear regressors.
As next steps on the path of finding the best predictive model for house pricing I would try to:

- test more with Features Selection methods
- and/or test more with hyperparametrs tuning options
- and/or try diffrent algorithms like neural network/deep learning

In [ ]:

```

```