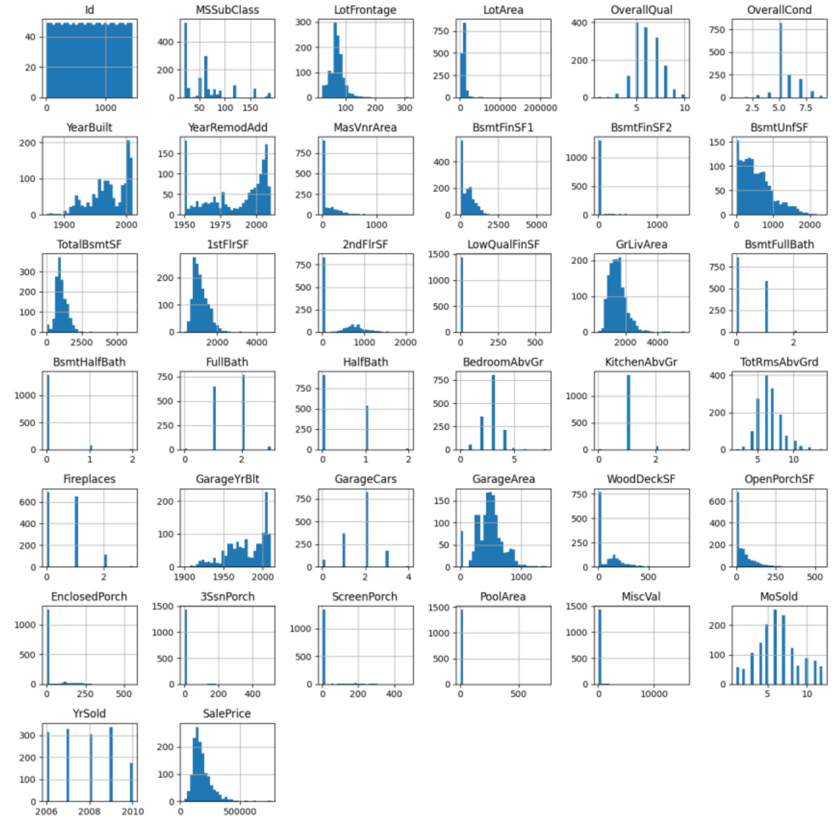
Predicting House Prices using Regression Models

Newton Tran

Problem Overview

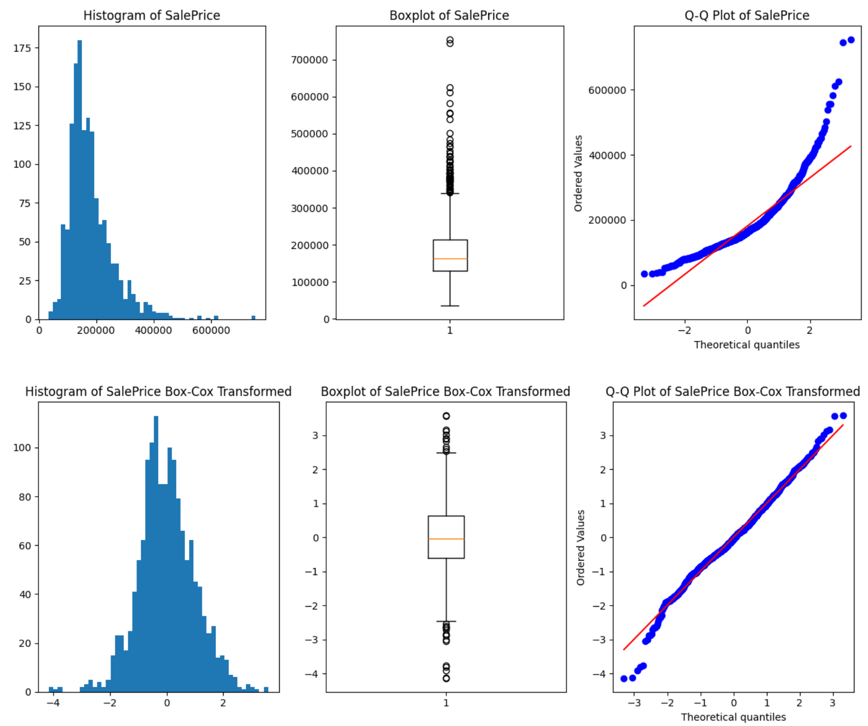
- Goal: Predict house prices using regression analysis on the Ames, Iowa housing dataset
- Dataset: 1460 training entries, 1459 test entries, 80 features (numerical + categorical)
- Target variable: `SalePrice`

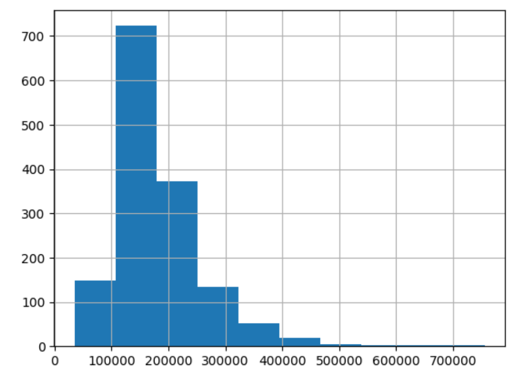
	column_name	percent_missing	num_missing	nunique_vals
0	PoolQC	99.520548	1453	3
1	MiscFeature	96.301370	1406	4
2	Alley	93.767123	1369	2
3	Fence	80.753425	1179	4
4	MasVnrType	59.726027	872	3
5	FireplaceQu	47.260274	690	5
6	LotFrontage	17.739726	259	110
7	GarageQual	5.547945	81	5
8	GarageFinish	5.547945	81	3
9	GarageType	5.547945	81	6
10	GarageYrBlt	5.547945	81	97
11	GarageCond	5.547945	81	5
12	BsmtFinType2	2.602740	38	6
13	BsmtExposure	2.602740	38	4
14	BsmtCond	2.534247	37	4
15	BsmtQual	2.534247	37	4
16	BsmtFinType1	2.534247	37	6
17	MasVnrArea	0.547945	8	327
18	Electrical	0.068493	1	5

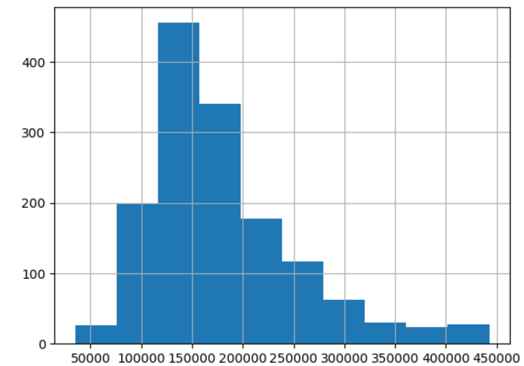


Exploratory Data Analysis

- Analyzed missing values, distributions, skewness, and outliers

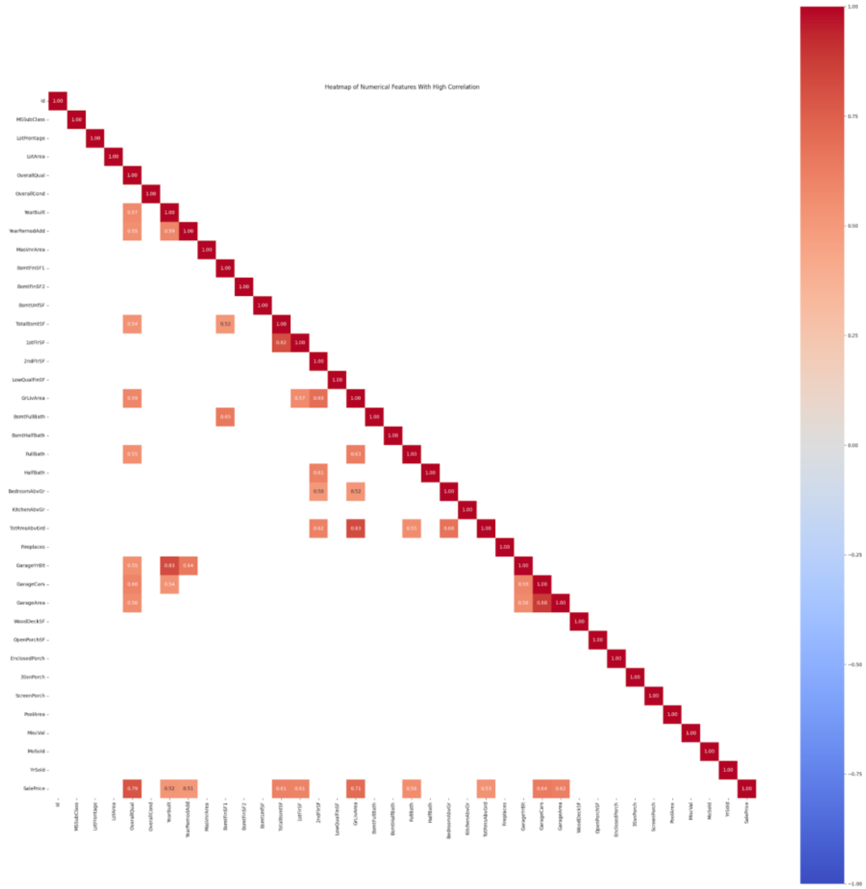






Exploratory Data Analysis

- Applied Box-Cox transformation and winsorization of top 1% observations ($442,567.01) to normalize `SalePrice`



Exploratory Data Analysis

- Identified multicollinearity (e.g., `GarageCars` vs `GarageArea`)
- Features that had a correlation value of 0.5 or above were utilized for additional feature engineering

Feature Engineering & Preprocessing

- KNN imputation (numerical), mode imputation (categorical)
- Log, ratio, and interaction features engineered
  - `LogMasVnrArea = log(MasVnrArea)`
  - `TotalBsmtFinSF  =  BsmtFinSF1 + BsmtFinSF2`
  - `Qual_GrLivArea = GrLivArea * OverallQual`
  - `TotalFlrSF = 1stFlrSF + 2ndFlrSF`
- Cyclic features encoded using sine/cosine
  - $MoSold_{sin} = \sin(2\pi \frac{MoSold}{12})$
  - $MoSold_{cos} = \cos(2\pi \frac{MoSold}{12})$

Feature Engineering & Preprocessing

- For categorical features, one-hot or ordinal encoding were used
- Some instances where a specific value was present in the train dataset but not in test dataset or vice versa
  - The value counts were consolidated on a numerical threshold
- Final dataset: 214 features after robust scaling and encoding

Exterior2nd	count	count
VinylSd	504.0	510.0
MetalSd	214.0	233.0
HdBoard	207.0	199.0
Wd Sdng	197.0	194.0
Plywood	142.0	128.0
CmentBd	60.0	66.0
Wd Shng	38.0	43.0
Stucco	26.0	21.0
BrkFace	25.0	22.0
AsbShng	20.0	18.0
ImStucc	10.0	5.0
Brk Cmn	7.0	15.0
Stone	5.0	1.0
AsphShn	3.0	1.0
Other	1.0	NaN
CBlock	1.0	2.0
Unknown	NaN	1.0

Exterior2nd	count	count
VinylSd	504	510
MetalSd	214	233
HdBoard	207	199
Wd Sdng	197	194
Other	196	195
Plywood	142	128

model	n_features	rmse_train($)	rmse_test($)	rmsle_train($)	rmsle_test($)	mae_train	mae_test	smape_train(%)	smape_test(%)	r2_train	r2_test	adj_r2_train	adj_r2_test
OLS	214	17260.097670	21200.826980	0.093021	0.123098	11675.981744	13842.418235	6.573991	8.162320	0.944531	0.925752	0.932075	0.719402
OLS with RFECV	173	17253.661299	20844.878324	0.093414	0.122896	11694.336993	13860.929174	6.601785	8.212344	0.944572	0.928225	0.934925	0.822995
Lasso	111	18777.862165	19297.480236	0.099270	0.119870	12173.831811	13022.480073	6.834922	7.846995	0.934347	0.938485	0.927445	0.900552
Ridge	214	18593.118489	19412.295526	0.098059	0.122021	12032.588980	12950.687849	6.738632	7.872308	0.935632	0.937751	0.921178	0.764748
ElasticNet	148	19159.335432	19426.978901	0.100306	0.122443	12284.675113	12981.932436	6.878788	7.885443	0.931652	0.937657	0.921725	0.873134
XGBRegressor	214	5335.859406	21382.802742	0.028545	0.127756	3862.321984	14280.964199	2.178442	8.507660	0.994699	0.924472	0.993508	0.714564
Random Forest	214	9519.239294	23618.655157	0.050147	0.146770	6096.598688	15970.619125	3.412393	9.473615	0.983128	0.907852	0.979339	0.651752
Theil–Sen	214	26261.157485	21208.019187	0.110922	0.129634	12507.467349	13820.665961	6.810716	8.299240	0.871592	0.925702	0.842757	0.719212

Models & Evaluation

- Training data was 80/20 train-test split
- Eight total models were developed
  - Linear Models: OLS, OLS with RFECV, Lasso, Ridge, ElasticNet, Theil-Sen
  - Tree-Based Models: Random Forest, XGBoost
- Best Model: Lasso Regression
  - Test RMSE: $19,297 | Adjusted R^2: 0.901

Key Observations

- Lasso Regression generalizes best with regularization
- Both tree-based models XGBoost and Random Forest overfit
- Engineered interaction features improved results
- Box-Cox and winsorization improved overall model predictive performance

Conclusion & Future Work

- Lasso captured strong linear trends in `SalePrice`
  - Also helped achieve our goal of obtaining an interpretable model
- Explore non-linear models (e.g., CatBoost, LightGBM)
- Experiment with more feature engineering strategies
- Apply dimensionality reduction for generalization

References

- Real-Life Applications of Correlation and Regression. (2024, April 10). GeeksforGeeks. https://www.geeksforgeeks.org/real-life-applications-of-correlation-and-regression/
- Regression analysis. (2019, March 22). Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Regression_analysis
- Gupta, M. (2018, September 13). ML | Linear Regression - GeeksforGeeks. GeeksforGeeks. https://www.geeksforgeeks.org/ml-linear-regression/
- GeeksforGeeks. (2024, August 13). Handling Missing Data with KNN Imputer. GeeksforGeeks; GeeksforGeeks. https://www.geeksforgeeks.org/handling-missing-data-with-knn-imputer/
- Regression Analysis: Understanding the Why? | GEOG 586: Geographic Information Analysis. (n.d.). Www.e-Education.psu.edu. https://www.e-education.psu.edu/geog586/node/624