Data Scientist Capstone

Kaggle competition titled House Prices - Advanced Regression Techniques

by Umrbek Allakulov

Abstract

This report is part of the data science capstone project as part of the Udacity Data Scientist Nanodegree curriculum. The project analyzes house prices in residential homes in Ames, Iowa using unsupervised and supervised machine learning models. It identifies the major clusters of houses and produces predictions of house sale prices on the training set as well as unseen data provided in the Kaggle completion.

TABLE OF CONTENTS

1 INTRODUCTION

This project sets out to analyze house prices in residential homes in Ames, Iowa using unsupervised and supervised machine learning models. For this purpose, the Ames Housing datasets are employed, which are provided as part of a Kaggle competition[1].

This report follows the CRoss Industry Standard Process for Data Mining (CRISP-DM), which is a most common methodology for data science projects. Namely, this report is organized around the following 6 phases included in CRISP-DM:

1. Business understanding
2. Data understanding
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment

2 BUSINESS UNDERSTANDING

This data science project aims to answer the following main questions:

1. What different segments of residential houses can be identified in Ames, Iowa?
2. How accurately can supervised machine learning models predict house prices in Ames, Iowa?
3. What are the top 5 important attributes that influence the house prices in Ames, Iowa?

To answer these questions, the following strategies are adopted:

1. To identify different segment of houses in the dataset, the K-Means algorithm is used. The algorithm partitions data into k mutually exclusive clusters by finding a partition in which the observations belonging to each cluster are as close to each other as possible, and simultaneously as far from objects in other clusters as possible[2].
2. Two different machine learning algorithms are trained and their performances are compared. The performance metrics used are cross-validated scores of root mean squared error and the R squared values on the training set. The former metric gives an indication of performance on unseen data, and is selected because of its advantage of penalizing large errors and representing the amount of error in the same unit as the predicted column making it easy to interpret. The latter metric, R squared, is used as a goodness-of-fit measure of the models on the training test in this analysis. It summarizes the percent of variation in the target variable that the regression model explains.
3. To answer the third question, the Mean Decrease in Impurity importance of each input variable is computed. The impurity importance for each variable is computed as the sum of all impurity decrease

[1] The Kaggle competition can be accessed through the following link: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview

[2] More information about the K Means algorithm can be found here: https://stanford.edu/~cpiech/cs221/handouts/kmeans.html

measures of all nodes in the forest at which a split on the variable has been conducted, normalized by the number of trees[3]. Variables that lead to larger decreases in impurity are considered more important.

3 DATA UNDERSTANDING

The Ames Housing dataset was compiled by Dean De Cock for use in data science education. The following are the datasets used in this project:

- train.csv - the training set
- test.csv - the test set
- data_description.txt - full description of each column, originally prepared by Dean De Cock but lightly edited to match the column names used.

The Ames Housing dataset contains 79 explanatory variables describing various aspect of residential homes in Ames, Iowa.

4 DATA PREPARATION

4.1 DATA DESCRIPTION

The attributes in the train and test datasets contain three data types, as shown in Table 1.

Type	Attribute
int64	35
float64	3
object	43

Table 1. Data types in the train and test sets

As can be seen in Figure 1, some attributes contain very high proportion of missing values. However, most attributes contain less than 6% values missing. The presence of missing values in a dataset can negatively affect the performance of a predictive model. Therefore, 5 attributes that contain more than 18% values missing are removed in the modelling phase of the project.

[3] For further reference, please see https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6198850/

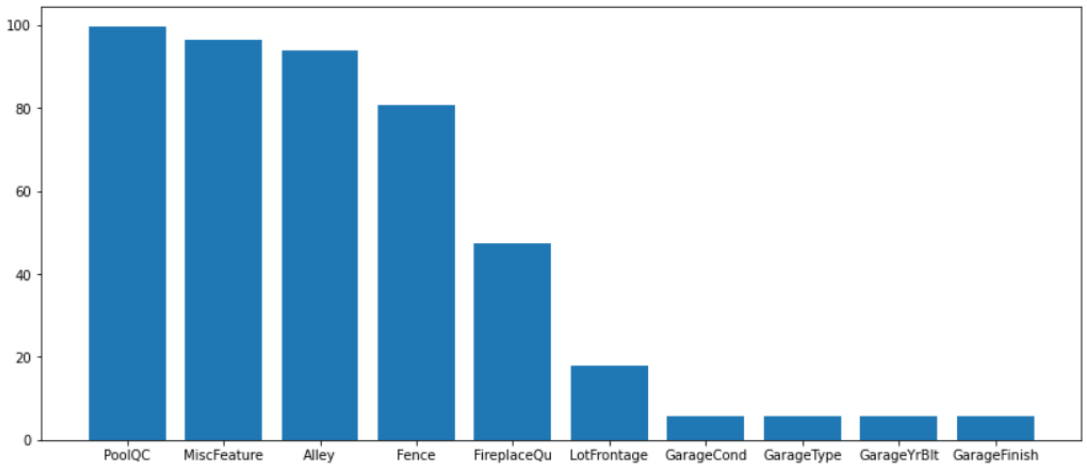

Figure 1.Top 10 attributes with the highest % of missing values

4.2 DESCRIPTIVE STATISTICS

As will be elaborate in chapter 5 of the report, the preprocessing of data is accomplished in two main parts: preprocessing of categorical attributes and numerical attributes. For consistency, the descriptive statistics are presented in the same manner.

4.2.1 Categorical attributes

Tables 2 and 3 depict the main characteristics of the categorical variables. The number of categories contained in these attributes range from 2 (binary variable) to 16. The rows on frequency indicate that some of the categorical attributes contain large numbers of missing values. The data preparation chapter will elaborate on the imputation method used to deal with the missing values.

	MSSubClass	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle
count	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460
unique	15	5	2	4	4	2	5	3	25	9	8	5	8
top	20	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm	Norm	1Fam	1Story
freq	536	1151	1454	925	1311	1459	1052	1382	225	1260	1445	1220	726

Table 2. Descriptive statistics of categorical variables - part 1

	RoofStyle	Exterior1st	Exterior2nd	MasVnrType	Foundation	Heating	Electrical	GarageType	SaleType	SaleCondition	CentralAir	RoofMatl
count	1460	1460	1460	1452	1460	1460	1459	1379	1460	1460	1460	1460
unique	6	15	16	4	6	6	5	6	9	6	2	8
top	Gable	VinylSd	VinylSd	None	PConc	GasA	SBrkr	Attchd	WD	Normal	Y	CompShg
freq	1141	515	504	864	647	1428	1334	870	1267	1198	1365	1434

Table 3. Descriptive statistics of categorical variables - part 2

4.2.2 Numerical attributes

Table 4 depicts descriptive statistics of the first batch of numerical attributes out of three. What stands out in Table 1 is that the residential houses included in the training dataset are generally large. The 50% percentile shows that an average house has an indoor area of 2556 square feet and contains two bathrooms. There are certainly some outliers – there is at least one house with 6 bathrooms. There is also a house with 215,395 sq feet, which is about 2.8 times bigger than a football pitch (76,900 sq feet).

	LotFrontage	Total_lot_area	MasVnrArea	GarageYrBlt	Total_area	Total_bathrooms	BsmtFullBath	BsmtHalfBath	TotalBsmtSF	BsmtUnfSF	BsmtFinSF1	Total_indoor_area
count	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00
mean	69.69	9922.43	102.25	1978.42	4537.36	2.21	0.42	0.06	1052.36	567.23	438.46	2556.04
std	21.05	7019.85	178.50	23.99	1364.97	0.78	0.52	0.24	415.17	442.32	433.15	773.91
min	21.00	1324.00	0.00	1900.00	668.00	1.00	0.00	0.00	0.00	0.00	0.00	334.00
25%	60.00	8043.75	0.00	1962.00	3552.00	2.00	0.00	0.00	795.00	222.50	0.00	2008.00
50%	69.31	9795.54	0.00	1978.11	4379.00	2.00	0.00	0.00	991.00	477.50	381.00	2472.00
75%	79.00	10752.50	162.25	2001.00	5312.25	2.50	1.00	0.00	1296.25	808.00	709.00	3000.75
max	313.00	215395.00	1600.00	2010.00	12161.00	6.00	3.00	2.00	3206.00	2336.00	2188.00	6872.00

Table 4. Descriptive statistics of numerical attributes - part 1

Table 5 further confirms the observations in the previous table. A house in the 50% percentile of the dataset features a garage for 2 cars. On average, the overall quality of houses is excellent (5 out of 5).

	BsmtFinSF2	Total_finished_basement	GarageArea	GarageCars	OverallQual	OverallCond	ExterQual	ExterCond	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1
count	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00
mean	46.68	485.14	471.79	1.77	5.09	4.58	2.39	2.08	2.56	2.01	0.65	2.56
std	161.52	454.75	212.17	0.75	1.38	1.11	0.57	0.35	0.68	0.28	1.04	2.07
min	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00
25%	0.00	0.00	329.50	1.00	4.00	4.00	2.00	2.00	2.00	2.00	0.00	0.00
50%	0.00	464.00	478.50	2.00	5.00	4.00	2.00	2.00	3.00	2.00	0.00	3.00
75%	0.00	789.00	576.00	2.00	6.00	5.00	3.00	2.00	3.00	2.00	1.00	5.00
max	1474.00	2306.00	1390.00	4.00	9.00	8.00	4.00	4.00	4.00	3.00	3.00	5.00

Table 5. Descriptive statistics of numerical attributes - part 2

The most important variable included in table 6 is the target variable, which is the sale price of houses. The mean sale price stands at USD 180,810. This is significantly higher than the median price of USD 163, 000.

	BsmtFinType2	HeatingQC	KitchenQual	Functional	GarageFinish	GarageQual	GarageCond	PavedDrive	SalePrice
count	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00	1456.00
mean	0.27	3.14	2.51	6.84	0.77	1.98	1.98	1.86	180810.06
std	0.87	0.96	0.66	0.67	0.81	0.24	0.23	0.50	79462.62
min	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	34900.00
25%	0.00	2.00	2.00	7.00	0.00	2.00	2.00	2.00	129900.00
50%	0.00	4.00	2.00	7.00	1.00	2.00	2.00	2.00	163000.00
75%	0.00	4.00	3.00	7.00	1.00	2.00	2.00	2.00	214000.00
max	5.00	4.00	4.00	7.00	2.00	4.00	4.00	2.00	755000.00

Table 6. Descriptive statistics of numerical attributes - part 3

In fact, by plotting the sale price against any of the area metrics it is easy to see there are two houses in the dataset that are extremely large and expensive. Figure 2 show this in the case of the total area.

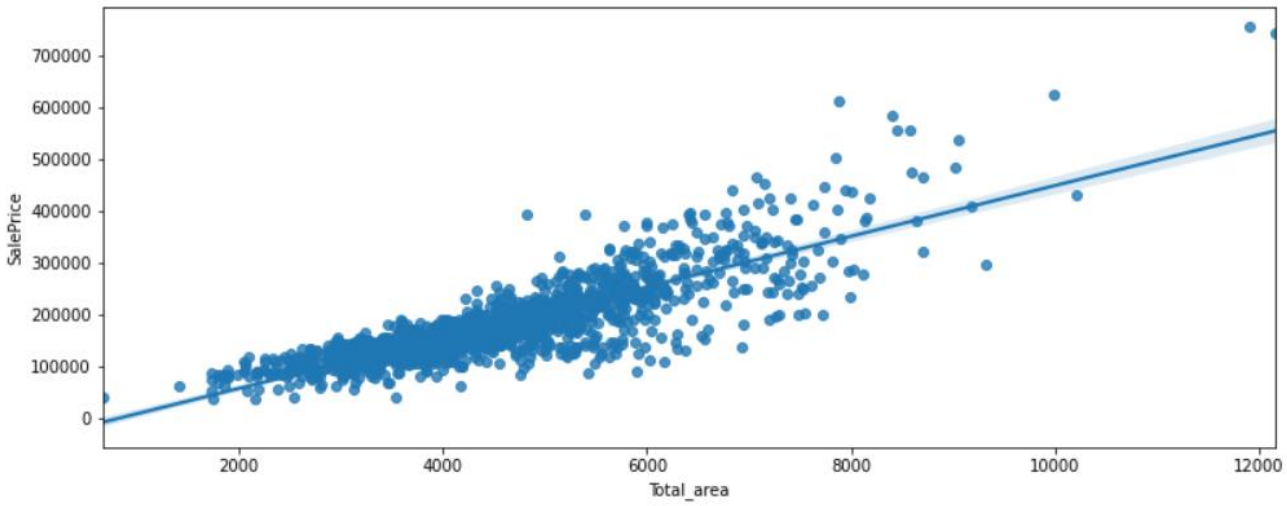

Figure 2. Linear relationship between the total area and price of houses

Next, the correlation matrix of the numerical attributes shows the correlation coefficients between these variables. Cells in red contain highly positive correlation coefficients whereas blue colors depict negative correlation coefficients. The higher the intensity of the colors, the stronger the correlation.

The last row is of especially high importance, since it shows the correlation values of different attributes with the sale price. As can be seen, some attributes are strongly correlated with the target variable. On the other hand, some attributes have zero correlation with the sale price, meaning they will likely offer very little explanatory power in our predictive model. It is also important to note that there are a few strong correlations within the explanatory variables, which may cause multicollinearity in regression models.

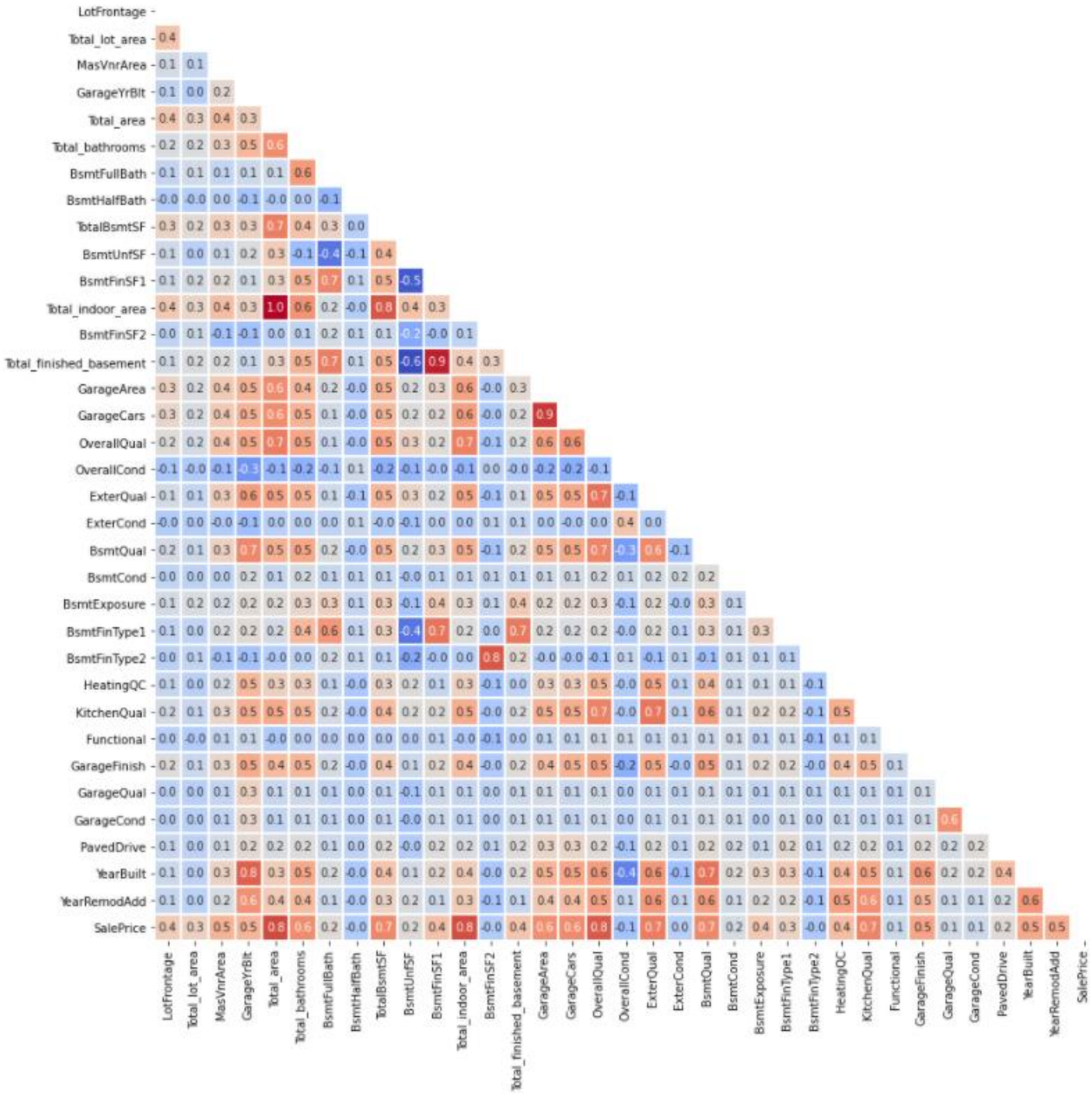

Figure 3. Correlation between numerical attributes

5 DATA PREPARATION

This chapter describes the various measures taken to prepare the data for modelling. The preprocessing of data is performed in two main parts: preprocessing of categorical attributes and numerical attributes.

Before these operations are carried out, a list of attributes containing a large proportion of missing values are identified and dropped.

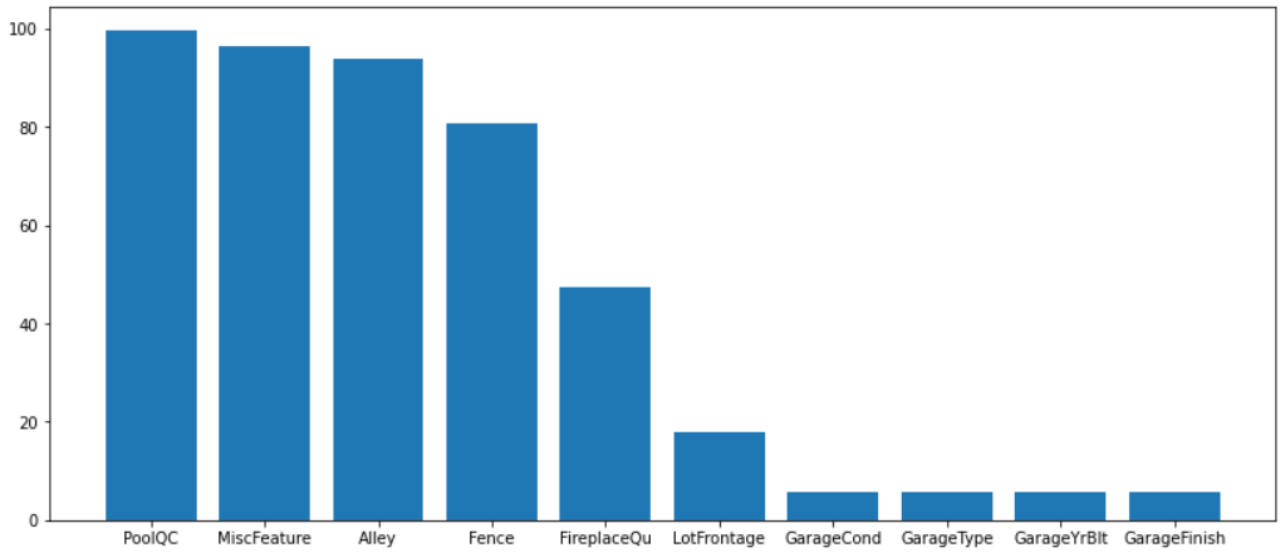

Figure 4. Attributes with the highest percentage of missing values

As can be seen in Figure 4, the following five attributes have very high proportion (more than 40%) of missing values and are thus dropped from the training and test sets:

- Alley
- FireplaceQu
- PoolQC
- Fence
- 'MiscFeature

5.1 CATEGORICAL ATTRIBUTES

To start with, the categorical attributes are imputed using the most frequent value strategy of SimpleImputer from the Scikit-learn library. Next, dummy variables are created using the get_dummies from Pandas library. The categories of the attributes are identified using a merged dataset of the train and test sets: this ensures the presence of the same set of categories in the both datasets.

5.2 NUMERICAL ATTRIBUTES

The numerical attributes are preprocessed separately for continuous and ordinal attributes, as these types of variables require different sets of methods.

5.2.1 Continuous attributes

Before delving into next steps, new features are created using the existing continuous attributes. The first sets of new attributes aggregate the areas of the different parts of the house. Altogether, eight such aggregated features are engineered. In addition, dummy variables are featured to signify whether a given attribute is present in the house (taking the value of 1) or absent (taking the value of 0).

After that, the continuous attributes are imputed using KNNImputer the Scikit-learn library. Similar to the imputation of the categorical variables, the imputer in this case is first trained on the merged dataset of the train and test sets.

5.2.2 Ordinal attributes

First, the ordinal attributes are imputed using the most frequent value strategy of SimpleImputer from the Scikit-learn library. Once again, the imputer is trained on the merged dataset of the train and test sets.

Next, ordinal encoding is applied to the ordinal attributes by using OrdinalEncoder. To achieve the desired order values, dictionaries of values are created and fit into OrdinalEncoder from the Scikit-learn library. To achieve this, two dictionaries are created. The first one distributes the names of the attributes across six categories, depending on which categories the attributes contain. The second dictionary orders the categories for the six types of attributes. The consistency of the order important, because ordinal numbers indicate position, or order of things or qualities. Next, OrdinalEncoder loops through each attribute in the first dictionary, picks up the relevant categories from the dictionary, and ordinal numbers by learning from the merged dataset of the train and test sets. Finally, the ordinal numbers are applied separately to appropriate attributes in the train and test sets.

6 MODELLING AND EVALUATION

6.1 UNSUPERVISED LEARNING

The objective of the unsupervised learning model in this project is to help identify distinguishing clusters of similar houses in the train dataset. To achieve this, first principal component analysis (PCA) is performed on the dataset. PCA is an unsupervised technique that helps reduce the dimensionality of high-dimensional datasets while preserving the original structure and relationships inherent to the original dataset. To choose the appropriate number of dimensions, PCA is applied on the dataset while fitting a Gradient Boosting for regression. This ML technique is used again later, as described in more detail in the next chapter.

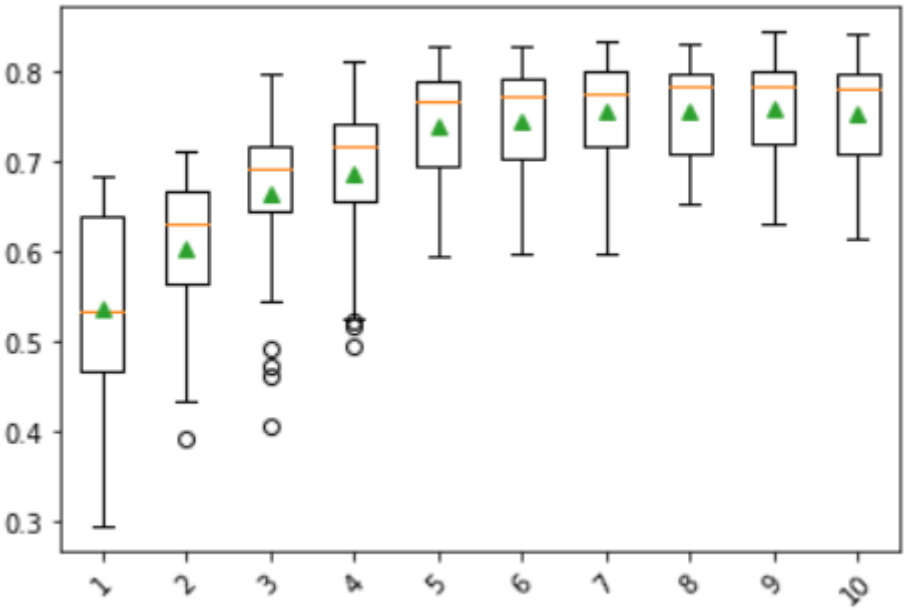

Figure 5. Number of components versus model accuracy, measured as r squared

As can be seen in Figure 5, the explained variance ratio starts to level off after five components and reaches a peak at eight components. Therefore, eight components are selected for the unsupervised machine learning model for clustering.

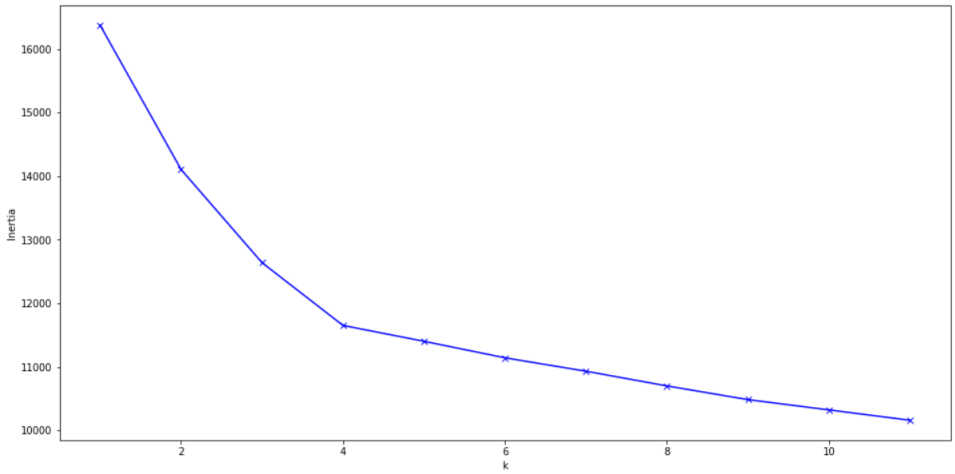

Figure 6. The Elbow Method showing the optimal value of k

For the cluster analysis, KMeans from the Scikit library is applied to the training dataset. To determine this optimal value of k, the Elbow Method used. As can be observed in Figure 6, inertia, which is the sum of squared distances of samples to their closest cluster center, first declines fast with additional clusters. However, its slop nearly plateaus starting at four clusters, which is determined as the optimal number of k.

The distribution of observations across the four clusters are outlined in Table 7.

Cluster	Number of observations
3	458
1	393
2	315
0	286

Table 7. Clusters and number of observations

A simple analysis of the clusters can be performed using the most correlated numerical attributes identified in the correlation matrix earlier. This is depicted in

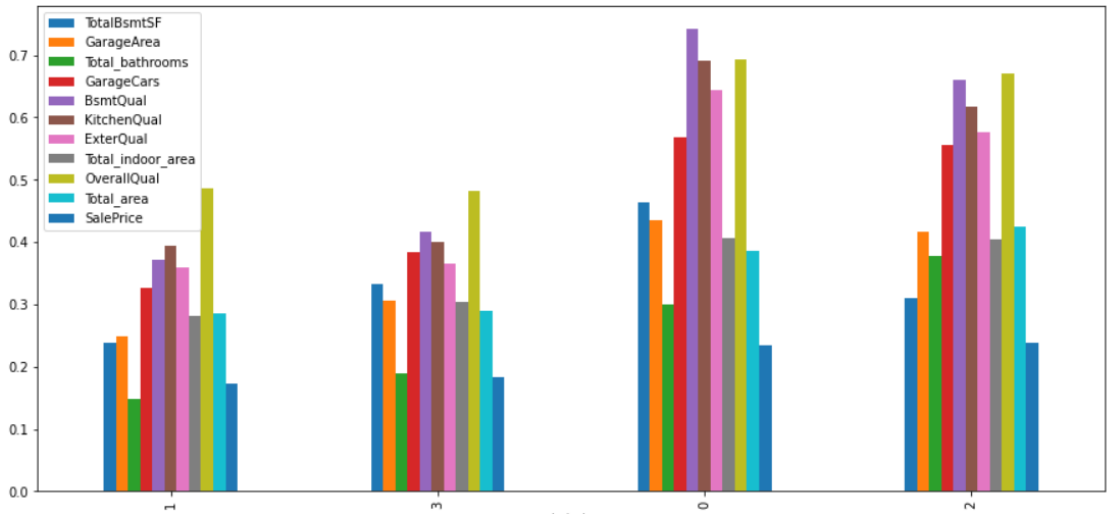

Figure 7. Differences among the four clusters

Overall, houses in clusters 0 and 2 tend to be more expensive, bigger, and better quality than houses in clusters 1 and 3. Additionally, the following observation can be made across these two groups of clusters:

1. More expensive clusters – 0 and 2
    a. Clusters 0 and 2 are close in price and total area
    b. Cluster 0: expensive segment, more bathrooms
    c. Cluster 2: expensive segment, large basement
1. Less expensive clusters - 1 and 3
    a. Clusters 1 and 3 are close in price and total area
    b. Cluster 3: cheaper segment, more basement and bathroom
    c. Cluster 1: cheaper segment, all attributes inferior to cluster 3

6.2 SUPERVISED LEARNING

For this segment of analysis, two different regression algorithms are employed using two different search strategies for fine-tuning their hyperparameters.

6.2.1 Gradient Boosting Regressor

The first algorithm selected for this project is Gradient Boosting for regression. The Scikit learn documentation describes the algorithm as follows:

"GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function."

The strategy for fine-tuning Gradient Boosting Regressor includes two phases. In the first phase, the number of features are reduced by wrapping Gradient Boosting Regressor with recursive feature elimination (RFE). Next, the selected features are used to train the Gradient Boosting Regressor model.

6.2.1.1 Feature selection

The process works as follows:

1. First, Gradient Boosting Regressor is trained on a predetermined number of features. In this case, the number of features tested included 100, 205, 208, 221

2. For each of the four iterations, RFE determines the importance of each feature based on root mean squared error (RMSE) scores, keeps the important features and and prunes the less important features
3. Next, RMSE scores of the each estimator with the different number of selected features are obtained. The number of features that results in the smallest RMSE is selected for further modelling.

As a result of the above process, 208 features is determined to be optimal for training the Gradient Boosting Regressor.

6.2.1.2 Tuning Gradient Boosting Regressor Hyperparameters with Bayes Search

To tune the hyperparameters of the Gradient Boosting Regressor, BayesSearchCV from Scikit-Optimize is selected. Compared to Scikit-Learn's GridSearchCV and RandomizedSearchCV, BayesSearchCV offers more efficiency and usually superior or comparable results. What makes BayesSearchCV efficient is that "not all parameter values are tried out, but rather a fixed number of parameter settings is sampled from the specified distributions"[4], as stated in its documentation. For this analysis, the number of parameter settings that are tried is set at 100. The resulting best estimator parameters are as follows:

- learning_rate=0.008862930368187014,
- max_depth=8,
- n_estimators=3038
- n_iter_no_change=10
- random_state=10
- subsample=0.32556277999210126
- tol=0.001

The model yields RMSE of 21990.30 and R-square value of 0.9821.

6.2.2 Extreme Gradient Boosting Regressor

The second algorithm selected for this project is Extreme Gradient Boosting (XGB) Regressor. XGB Regressor is similar to Gradient Boosting Regressor, and offers additional benefits such as more efficiency and accuracy[5].

In contrast to the previous modelling exercise in which the number of features was reduced, the search strategy for tuning the parameters of the XGB Regressor model employs the full set of features. The estimator that produces the lowest RMSE score has the following parameters:

- base_score=0.5
- booster='gbtree'
- colsample_bylevel=1
- colsample_bynode=1
- colsample_bytree=1
- eval_metric='mae'
- gamma=0,
- gpu_id=-1
- importance_type='gain'
- interaction_constraints=''

[4] More information is available here: https://scikit-optimize.github.io/stable/modules/generated/skopt.BayesSearchCV.html

[5] More information is available through the documentation of the Python API reference: https://xgboost.readthedocs.io/en/latest/python/python_api.html

- learning_rate=0.009355177335003302
- max_delta_step=0
- max_depth=4
- min_child_weight=1
- missing=nan
- monotone_constraints='()'
- n_estimators=4749
- n_jobs=8
- num_parallel_tree=1
- random_state=10
- reg_alpha=0
- reg_lambda=1
- scale_pos_weight=1
- seed=10,
- subsample=0.27990084664978787
- tree_method='exact'
- validate_parameters=1

This model yields a cross-validation RMSE score of 21068.05 and R-square value of 0.9965 on the training set. Both evaluation metrics represent an improvement over the performance of the first model (cross-validation RMSE of 21990.30 and R-square value of 0.9821 on the training set). The improvement brought about by the XGB Regressor estimator can be due to a combination of factors. Firstly, unlike the Gradient Boosting Regressor algorithm, XGB Regressor benefit from regularization – a technique used to help avoid overfitting. Also, XGB Regressor grows the tree up to max_depth indicated in the parameter space, unlike Gradient Boosting Regressor which stop as soon as a negative loss is encountered. Moreover, the search strategy for tuning the hyperparameters for the second model included all features. While an individual feature may not be important on its own, it may become significant in the algorithm in combination with other features.

6.2.3 Evaluation of feature importances

This section of the report estimates the influence of the features to predictions produced by the two models and search strategies outlined in the previous sections. Figure 8 and Figure 9 depict the Mean Decrease in Impurity for the top 20 most important features of the two models.

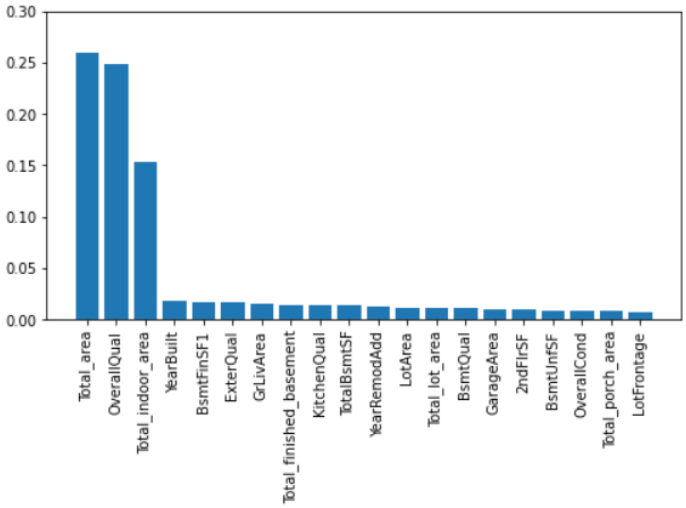

Figure 8. Feature importances of the GradientBoostingRegressor estimator

As can be seen in Figure 8, total area, overall quality, and the total indoor area rank among the most important features of the GradientBoostingRegressor estimator. All other features have significantly less importance to make predictions in this model.

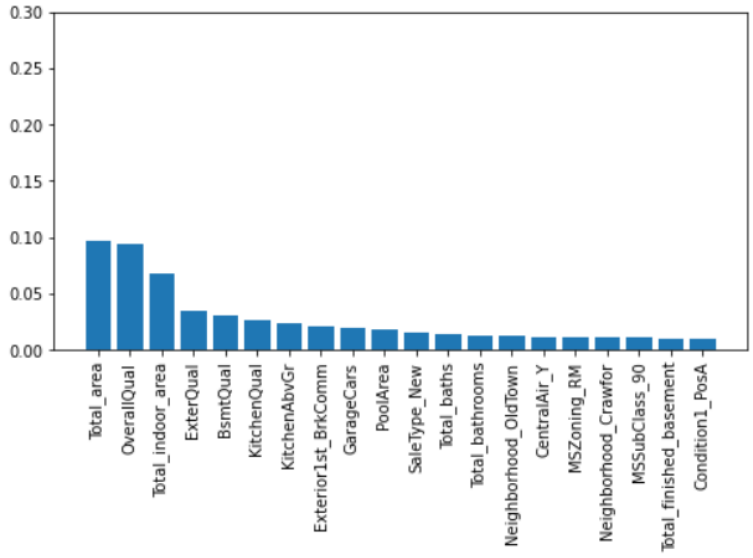

Figure 9. Feature importances of the XGBRegressor estimator

Similarly, the top three features from the previous model is shared by the XGBRegressor estimator. However, the relative important of the top three models is much smaller compared to the rest of the top 20 features. That could be partly due to the fact that the XGBRegressor estimator was trained on the full list of features, whereas GradientBoostingRegressor was trained on a subset of 208 features.

6.3 Areas for future improvement

While the two models and search strategies yield reasonably good models, additional improvements can be considered. Improvement in predictive performance may come a combination of different steps such as:

1. Testing other regression algorithms
2. Employing grid search cross validation (as opposed to Bayes optimization employed in this project) over larger parameter spaces
3. Improvements in feature engineering.

The last point is particularly interesting, because the two regression algorithms tested in the project assigned very different weights to the feature importances. It remains to be seen whether engineering of new features, handling of data types, and alternative imputation methods could further improve the predictive performance of the regression algorithms.

7 Deployment

The project repository is available on GitHub. It is structured around the following files:

Data:

|- train.csv
|- test.csv

|- data_description.txt

Notebook:
|- Udacity_capstone.ipynb

Documentation:

|- README.md
|- Udacity_Data_Scientist_Capstone.pdf

All the analysis can be performed by running the Jupyter Notebook titled Udacity_capstone.ipynb. Executing the codes in the notebook produces and exports two csv files for submission to the Kaggle competition:

1. submission_gbr.csv: predictions of the GradientBoostingRegressor estimator
2. submission_full.csv: predictions of the XGBRegressor estimator

8 CONCLUSION

This data science project sets out to analyze house prices in residential homes in Ames, Iowa using unsupervised and supervised machine learning models. In concluding, the main questions outlined at the beginning of the report are answered briefly using the analysis in the body of the report.

1. What different segments of residential houses can be identified in Ames, Iowa?

Four clusters of residential houses are identified. Two of them feature more expensive, bigger, and better quality houses. The remaining two are less expensive, small, and comparatively inferior quality houses.

2. How accurately can supervised machine learning models predict house prices in Ames, Iowa?

This project test two models. The best scoring model yielded and RMSE score of 21068.05 and R-square score of 0.9965. The model also performed relatively well on unseen data from the Kaggle competition. It scored 0.12724, which is the RMSE between the logarithm of the predicted value and the logarithm of the unseen sales price. At the time of writing, this score was ranked in top 25% among submissions by 4724 competing teams.

3. What are the top 5 important attributes that influence the house prices in Ames, Iowa?

According to the model with the best performance, the top 5 important attributes that influence house prices are as follows:

1. total area
2. overall quality
3. total indoor area
4. external quality
5. basement quality