Data Science project: House price analysis and prediction (April 2020)

A. Dechappe (B00377869), and Z. Timsans (B00223724)

***Abstract*—This is a data science project for predicting House price using regression techniques.**

***Index Terms*—data science, regression, machine learning.**

I. INTRODUCTION

In this data science project, we are going to address the problem of predicting house prices, based on multiple describing variables. The data for this project is obtained from *Kaggle* - popular online data science community resource. The particular problem is directed to the community in a form of competition. This was one of the main reasons project authors decided to undertake this particular data science problem. From the personal standpoint of view, this project will not only help practice and develop skills in data mining and visualisation but also give an insight into how properties are rated and how the price is estimated by owners and agencies.

II. DATASET

"House prices" data set[1] is a type of structured and captured quantitative data. It had been split into two Comma-Separated Values (CSV) files - train.csv and test.csv. For the following analysis, train.csv will be used. The test.csv will be used for making predictions for submission in *Kaggle*. Furthermore, the splitting of data will be reviewed in the Methodology section of this document. For feature explanation, please consult the Code.

To choose the appropriate variables for the analysis and prediction it is necessary to understand the problem and information available. Analysis data frame consists of 1460 rows and 81 columns. Exploratory process is time consuming but is very important to go through each column (feature) and examine its type and descriptive power for our problem.

A. Exploratory analysis

Not all variables are major indicators of the final price of the house but many of them do influence negotiations when purchasing a particular property. The initial selection of features is based on common sense and some previous knowledge on the subject field. Data could be split into the following groups to paint the picture of available information:
1) Numeric or categorical types. It is helpful when doing feature engineering. Numeric or categorical types. It is helpful when doing feature engineering.
2) Selecting features that have the most impact on the final price based on common sense, statistical relationships, and correlations.
3) Based on exploration, give a verdict whether to keep the feature or drop it.

We start by creating several feature aggregation and visualisation functions for each type of data: categorical, numerical, and for both. The *Numpy* and Pandas libraries help us manipulate data in order to successfully plot features using visualisation libraries, such as *MatPlotLib*, *Seaborn* and *Ploty*.

A first and most important feature to explore is the prediction target SalePrice. In fig. 1 we can observe the price distribution on histogram which indicates that most houses are priced at around $150,000.

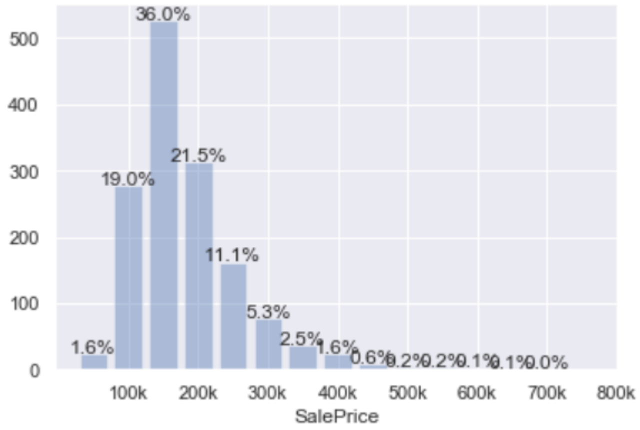
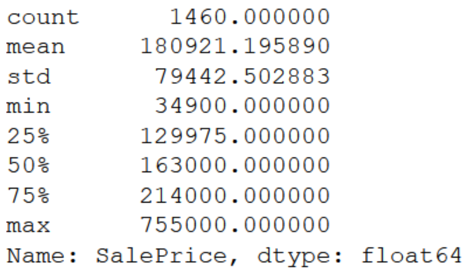


Figure 1. Histogram of SalePrice distribution.



```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

Figure 2. The statistical description of SalePrice.

It can be observed and also calculated that the SalePrice has highly positive skewness of 1.882876 and high kurtosis of 6.536282. Positive skewness means that the mean (180921) is greater than the median (163000) which can also be observed

[1] House Prices: Advanced Regression Techniques

in fig. 2. That leads to two conclusions: 1) houses are being sold below average value, and 2) data will have to be transformed in order to reduce skewness. Rule of thumb for normal skewness is from negative to positive 0.5. While high kurtosis (aka, Leptokurtic) means that the data is distributed around the peak value which signals potential outliers. Therefore, the SalePrice feature should be normalized to near a normal distribution (kurtosis score of less than 3).

Furthermore, SalePrice has been analysed against other numeric and categorical features. To simplify the process features are analysed based on rough categories of the property such as Lot, Basement, Above the ground, Garage, Condition, Other. Each of the mentioned categories has features describing size, quality, type, count, etc. Since there are 80 features available for exploration, this document would be too long to illustrate all of them. Some of the more important features that were uncovered are Overall quality (fig. 3.) that has a scale of 10 for quality measurements. This feature is also highly correlated with SalePrice and has a decent distribution curve. From the box whisker plot, we can observe that there are some outliers present but not too critical.

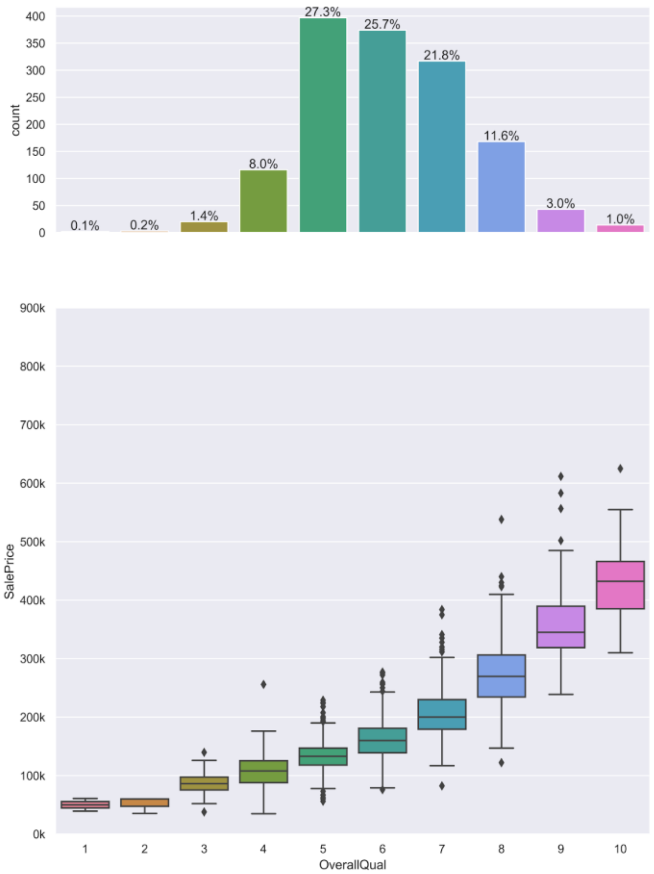


Figure 3. House overall quality. Top - distribution histogram. Bottom - box whisker plot.

Neighbourhood features are very interesting (fig. 4) and have a high impact on Sale price. It also shows how many house attributes are affected by the neighbourhood, for example, Overall quality and condition often is concentrated. In fig. 4 the quality and condition are represented with bubble distribution plot, where the size of the bubble indicates percentage.

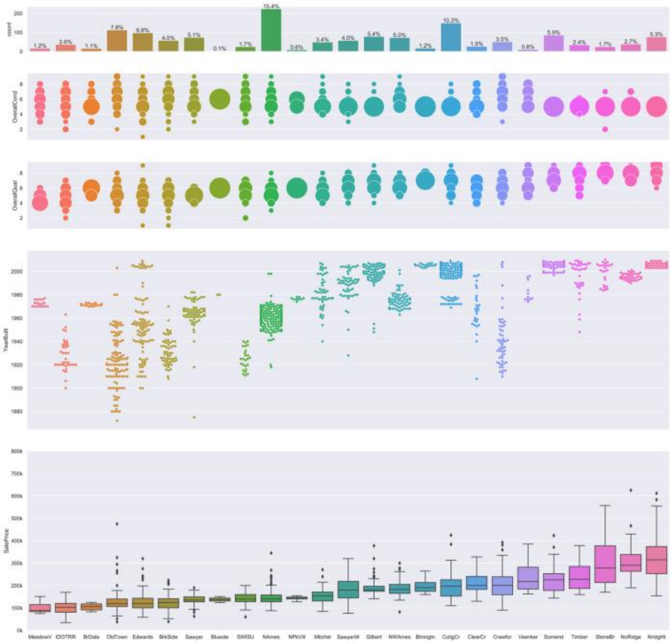


Figure 4. Neighbourhood visualizations. From top - distribution histogram, Overall condition, Overall quality, Year built, and Sale price. Review code for larger image.

For the data cleaning process, this is very important due to being able to impute missing values with not only median but also create new criteria based on exploration. The combination of different plotting techniques has been used, like swarm plot and bubble plot.

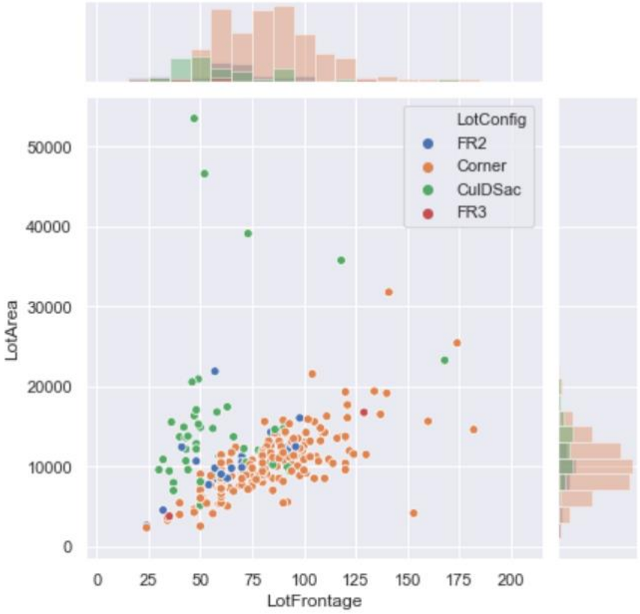


Figure 5. Scatter class plot of Lot Area and Lot Frontage and the Lot Shape class.

It is important to find categorical clusters of data. We used scatterplot (fig. 5) and colour to indicate category labels. In figure x we can see that the Lot configuration category is very likely to depend on Lot area size and Lot frontage.

There are many more visualisation plots available in the code document of this project.

B. *Correlations*

We used the Pandas function *corr()* on the data frame to convert all the numerical features into a correlation matrix to see the statistical dependencies between house price and other variables. We then use heat map visualisation from *Plotly* to display correlation coefficient (fig. 6).

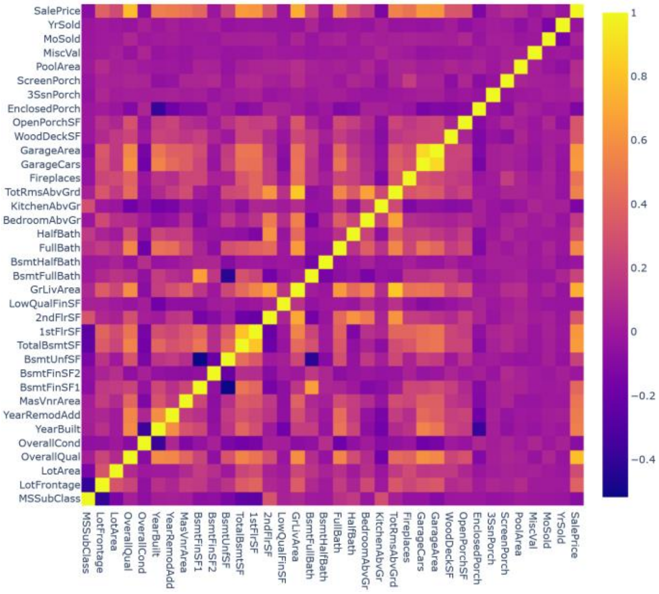


Figure 6. Correlation coefficient matrix of the whole data frame (review code for larger image.).

We then select the 10 most correlated features to explore and decide if it is useful for the analysis (fig. 7).

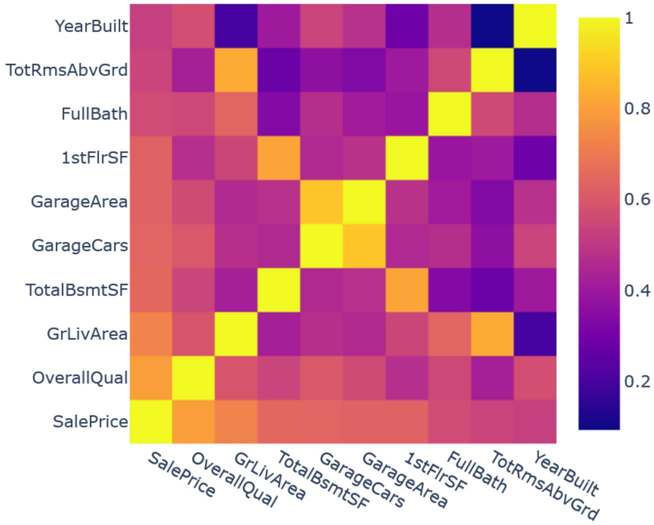


Figure 7. Correlation coefficient matrix of selected most correlated features.

The dependent variable is typically plotted along the y-axis but in *Plotly* the observation is made along the top of the matrix along the x-axis (fig. 8). A strong positive linear trend is observed for SalePrice as a function of TotalBsmtSF, GrLivArea, and 1stFlrSF.

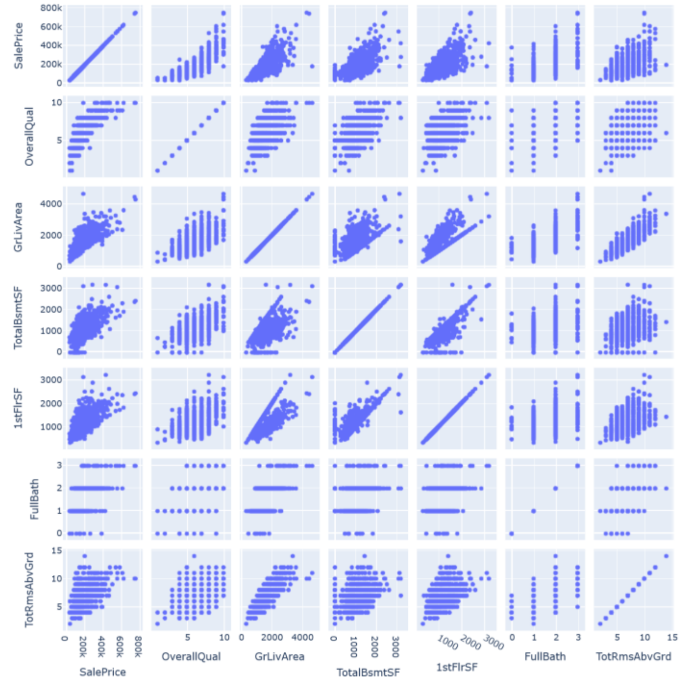


Figure 8. Scatter matrix of correlated features (review code for larger image.).

SalePrice tends to be higher if the house is newer, also the overall quality tends to be better for newer and more expensive houses with some minor exceptions. The number of rooms (not including basement) however has no particular relationship with other variables (fig. 9).

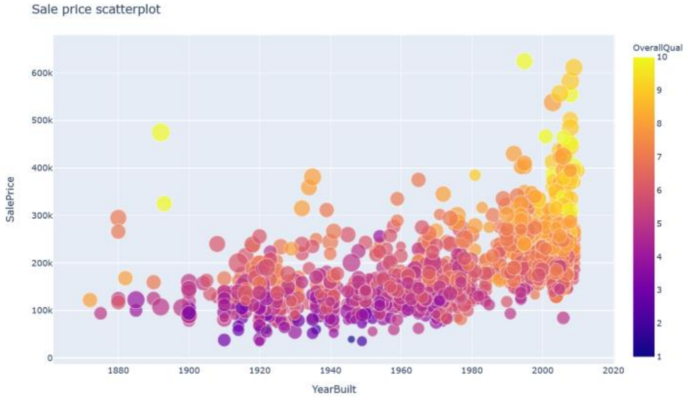


Figure 9. Sale Price plotted against Year Built, Overall Quality (colour) and Total Rooms Above Ground (bubble size). See Annex for higher resolution image.

C. *Summary*

From the analysis above together with further deeper analysis in the code, we conclude to organise the following groups of features for the pre-processing:

Numerical features:

SalePrice, LotArea, OverallQual, OverallCond, YearBuilt, YearRemodAdd, BsmtFinType1, TotalBsmtSF, BsmtUnfSF, CentralAir, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, FullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, TotRmsAbvGrd, Fireplaces, GarageCars, GarageArea, WoodDeckSF, *OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, MiscVal, MoSold, YrSold.*

Categorical features:

LandSlope, Neighborhood, MasVnrType(?), ExterQual, ExterCond, BsmtQual, BsmtCond, BsmtExposure, HeatingQC, Electrical, KitchenQual, Functional, FireplaceQu, GarageFinish, PavedDrive, SaleType, SaleCondition.

Dropped features due to several reasons, like poor distribution, duplicate or missing values or low descriptive power:

MSZoning, MSSubClass, Alley, Street, LotShape, LandContour, Utilities, LotConfig, Condition1, Condition2, BldgType, HouseStyle, RoofStyle, RoofMatl, Exterior1st, Exterior2nd, MasVnrArea, Foundation, BsmtFinType1, BsmtFinType2, Heating, Electrical, GarageYrBlt, GarageType, GarageQual, GarageCond, PoolQC, Fence, MiscFeature.

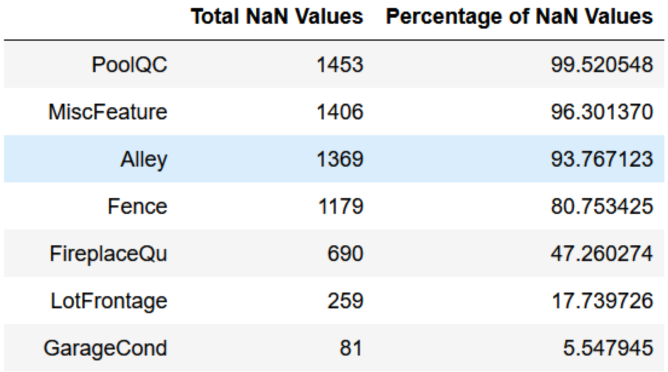
III. Methodology

Since the target prediction variable SalePrice is a continuous numerical value, the regression techniques would be the most appropriate for this project.

A. Cleaning

1) Missing values

To handle missing values, we need to investigate their distribution and nature (Sensor error, Unknown type, Corrupted file).



	Total NaN Values	Percentage of NaN Values
PoolQC	1453	99.520548
MiscFeature	1406	96.301370
Alley	1369	93.767123
Fence	1179	80.753425
FireplaceQu	690	47.260274
LotFrontage	259	17.739726
GarageCond	81	5.547945

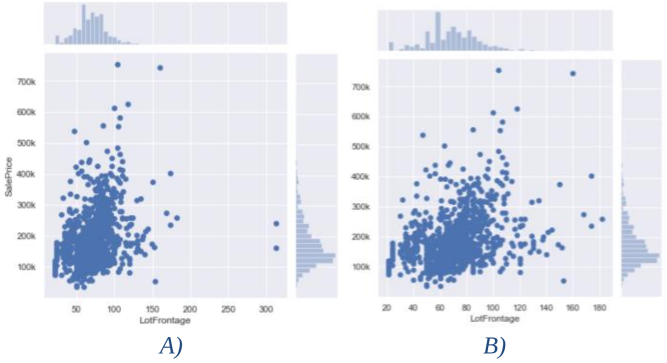
Figure 10. Top 7 missing values.

There were two different kinds of missing values in our dataset. In categorical features (such as *MiscFeature*), missing values were not errors but were used to indicate that a house does not belong to available categories in the feature so we have replaced them either by "No" or by "Other" (fig. 10).

In numeric features (such as *LotFrontage*), numbers were missing and we thought of several methods to fill them such as using the mean of the feature or inferring with nearest neighbours (data neighbours, not real). For *LotFrontage*, we used an intermediate method: we have filled missing values with the mean of the neighbourhood because building plot areas are most likely to be the same in a given neighbourhood.

2) Outliers

Searching for outliers is necessary in order to prepare data for certain models which are very sensitive to outliers. At the same time, it is very important to be careful when removing outliers since the test data might have them, resulting in inaccurate prediction. As noted by De Cock (2011)[2], there are some outliers present in the data. We will look at the scatter plot of SalePrice and *LotFrontage* (fig. 11). It can be observed that there are two records of low price but large *LotFrontage* which points to outliers and thus should be removed.



A) *B)*

Figure 11. SalePrice and GrLivArea scatter plots: A - with outliers, B - without outliers.

B. Feature engineering

Now that the dataset only contains data that we consider normal, we need to prepare it before feeding it into algorithms.

1) Categorical features: transformation and One Hot Encoding (OHE)

As seen before, most features of this dataset are categorical. This is an issue because the vast majority of models (especially for regression) can only be fed with numbers. Here are the two different methods we have used to deal with those features: 1) Transformation: a lot of categorical features were actually literal grades. For example, *ExterQual* categories were [Poor, Fair, Average, Good, Excellent]. We kept this feature and transformed its values from 0 (Poor) to 4 (Excellent). So those features became numeric then. 2) OHE: during exploration, we found that several features were very relevant for explaining sale price (such as Neighborhood). By using *scikit-learn OneHotEncoder()*, we have transformed each category of such feature into a new column, filled with 1 if a house belongs to this category else 0. OHE should be used carefully to prevent the curse of dimensionality so we can't apply it to all categorical features.

2) Numeric features: standardizing and Principal Component Analysis (PCA)

We standardize numeric features and plotted them to display the variance of PCA against a number of features in projections space (fig. 12).

[2] Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project

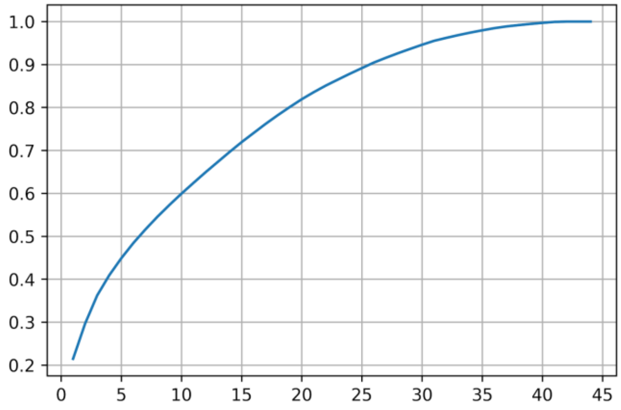


Figure 12. PCA line plot. X-axis – number of dimensions in principal component space; Y-axis – standardised numeric features.

An ideal PCA will show a sharp elbow. Our is pretty round but we have opted for a trade: keeping 90% of the variance and getting rid of 20 features. So numerical features were projected in the 25-principal components space.

C. *Modelisation*

1) *Data splitting*

Dataset ready for modelisation was separated in a train and a test set with an 80/20 slicing. For some models tuning, additional cross-validation datasets have been created from a train set.

2) *Prediction models*

Since we are trying to predict continuous numeric values, we will use regression models such as linear regression, Lasso, Elastic Net, Bayesian Ridge or Polynomial Regression. We chose to evaluate models performance with the Mean Squared Error between the target (SalePrice) and predictions.

IV. Results

For the prediction performance evaluation, the mean square error (MSE) has been utilised. The measured number should be as close to zero as possible. We apply equation

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \qquad (1)$$

by fitting the function with the test data to draw the regression line, generate residuals or the error number by subtracting real values and prediction, and then we square the residuals, sum them up and divide by the total number of observations.

Each model was tuned until we reached the lowest MSE on train dataset; then, the generalization of models was evaluated by computing the test score (MSE between predictions of X_test and y_test). Results can be observed in Table 1.

TABLE I
Regression model performance using mean square error (MSE)

Model	Best Train score	Best Test score
Linear	0.11261	0.09737
Lasso	0.11265	0.09621
Elastic Net	0.11262	0.09690
Kernel Ridge	0.03987	0.08150
Gradient Boosting	0.00054	0.10132
Polynomial Reg. (deg.=0.5)	0.13969	0.18460
xgBoost stacking	0.00000	0.08080

Must note that with the MSE, outliers also have a greater impact on the solution because errors are squared.

The best performing model based on MSE score is xgBoost stacking, 0.0808. Surprisingly, the second-best result was Kernel Ridge model with some parameter optimisation in place.

V. Discussion

The results of this project are satisfactory although there is plenty of room for improvement. Must note that initially the project subject and data were chosen based on having large and diverse enough data to demonstrate multiple techniques in exploration, pre-processing and modelling. There is a myriad of analysis and modelling done on this dataset which, we must admit, we did not know at the beginning of this project. Thus it was difficult to ignore many different solutions available online, nevertheless, we did our best using only the knowledge we have gained before and mostly during this module.

There was a difference in terms of competency levels between the members of the group in a variety of areas but with good collaboration and interpersonal skills, the project group reached a balanced workflow.

In terms of methods used for machine learning, we reached our goals by using regression techniques. Moreover, the regression is very popular and powerful and excels when working with numerical values, like house price. There are many more models available that could be very useful for this project, like Artificial Neuron Networks or different variations of model stacking approach. Also, hyperparameter optimization would improve the models we already have.

For future work, we are going to improve the polynomial and stacking model as they showed the most promise. We noticed that feature engineering has a huge impact on the results of our models. With the first iteration where the data was pre-processed using the simpler techniques (such as imputing missing values with median), the results were below average. After spending some more time in exploration and pre-processing stages we managed to improve the models greatly thus the more we spend on feature engineering, the higher accuracy will be produced by different models.

Project results will be posted in *Kaggle* competition and the aim is to be in the top 10%. We are still new to the field of data science but we must admit that this project provided a challenge and improved our knowledge greatly.

References

[1] D. De Cock, “Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project,” Journal of Statistics Education, vol. 19, no. 3, Nov. 2011.

[2] “Python | Mean Squared Error - GeeksforGeeks,” GeeksforGeeks, 28-Jun-2019. [Online]. Available: https://www.geeksforgeeks.org/python-mean-squared-error/. [Accessed: 01-Apr-2020].

[3] I. Goodfellow, Yoshua Bengio, and A. Courville, Deep Learning. Frechen: Mitp, 2018.

[4] Aurélien Géron, Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems. Sebastopol, Ca: O’reilly Media, 2017.

[5] D. Dietrich, B. Heller, B. Yang, and Emc Education Services, Data science & big data analytics : discovering, analyzing, visualizing and presenting data. Indianapolis, In: Wiley, 2015.

[6] M. Negnevitsky, Artificial intelligence : a guide to intelligent systems. Harlow, England: Addison Wesley, 2011.