Module Name: Artificial Intelligence

University of Sunderland

Name: Shingidzano Lesetedi

Local Center: Botswana Accountancy College (Francistown)

Student Number: BH44GG

Title: Machine learning Model for House Price Prediction

Contents

[Introduction 3](#_Toc535867217)

[Problem Statement 3](#_Toc535867218)

[Literature Review 3](#_Toc535867219)

[Justification 3](#_Toc535867220)

[Goal 4](#_Toc535867221)

[Prototype Development 4](#_Toc535867222)

[Data Preparation 4](#_Toc535867223)

[Splitting of Variables into Different Categories 8](#_Toc535867224)

[Gathering Insight of Data 11](#_Toc535867225)

[Skewed Distribution 12](#_Toc535867226)

[Supply and Demand Analysis 14](#_Toc535867227)

[House Characteristics 16](#_Toc535867228)

[Relationship Between Garage and Price Category 18](#_Toc535867229)

[Log Transformations 20](#_Toc535867230)

[Skewedness and Kurtosis 22](#_Toc535867231)

[Skewedness: 22](#_Toc535867232)

[Kurtosis: 22](#_Toc535867233)

[Feature Engineering 23](#_Toc535867234)

[Correcting of Missing Values 24](#_Toc535867235)

[Correcting Numerical and Categorical Values 25](#_Toc535867236)

[Pipelines 27](#_Toc535867237)

[Predictive Models 28](#_Toc535867238)

[Residual Plot 28](#_Toc535867239)

[Random Forest Regressor 30](#_Toc535867240)

[Gradient Boosting Regressor 33](#_Toc535867241)

[Parameter Tuning 34](#_Toc535867242)

[Stacking Regressor 35](#_Toc535867243)

[Making Data Persistent 37](#_Toc535867244)

[Deployment of the Model 37](#_Toc535867245)

[Development Reflection 37](#_Toc535867246)

[Evaluation Report 37](#_Toc535867247)

[Alternative Predictive Models 38](#_Toc535867248)

[Use of Keras and Tensorflow 38](#_Toc535867249)

[Conclusion 38](#_Toc535867250)

# Introduction

Predictive analysis is a data science practice which involves analysis, manipulation and visualization of data. The basic concepts of this where covered in lectures and tutorials through the use of basic regression models. This has laid the foundation for this project which will be house price prediction. In particular the linear regression model covered in the E-portfolio is what drove interest into this research area. House prices have several factors that impact the value of the asset and therefore impact the price that it is given. These factors range from tangible factors such as the general size (area), and the quality of the house which may entail the design and materials used to build it. Other factors are more inherent such as economic performance. The task at hand which is coming up with a model which predicts house prices is a challenging one knowing there are both internal and external factors that are bound to affect the price. In order to make this task possible one would need to filter out the least impacting factors and use the features that carry the most weight to create the model.

# Problem Statement

## Literature Review

Before the financial crisis that occurred before 2007 it was common logical belief that house prices where to go up and many people invested in property with the belief that prices would go up and they would make a sturdy return on their investment. Unfortunately, this was not the case due to the subprime mortgage crisis which was caused by a lot of borrowing money and a terribly flawed financial model based on the assumption that house prices would only appreciate over time (PRITCHARD, 2018). Due to the fact that economic factors having a strong impact on the prices of houses it is important to understand what features of a house have that directly impact its value hence its price. This will allow house price prediction to be more practical rather than based on the popular assumption of house appreciation over time which has proven not to be reliable in the past.

## Justification

This experiment will be carried out using a dataset which contains housing data which will be extensively analyzed in order to establish the story behind data. After obtaining an in depth understanding of the data in the dataset and how it affects the price of houses. For the predictive model in this experiment we’ll be using regression analysis. Regression analysis is most suitable because it is specially designed to deal with predictive analytics and forecasting which is highly related to the experiment which is house price prediction. The experiment will use a residual plot to give us the prediction errors the model is making. Two additional regressor models will be used to build the model and the final model being the stacking regressor will combine the previously used regressor models into one.

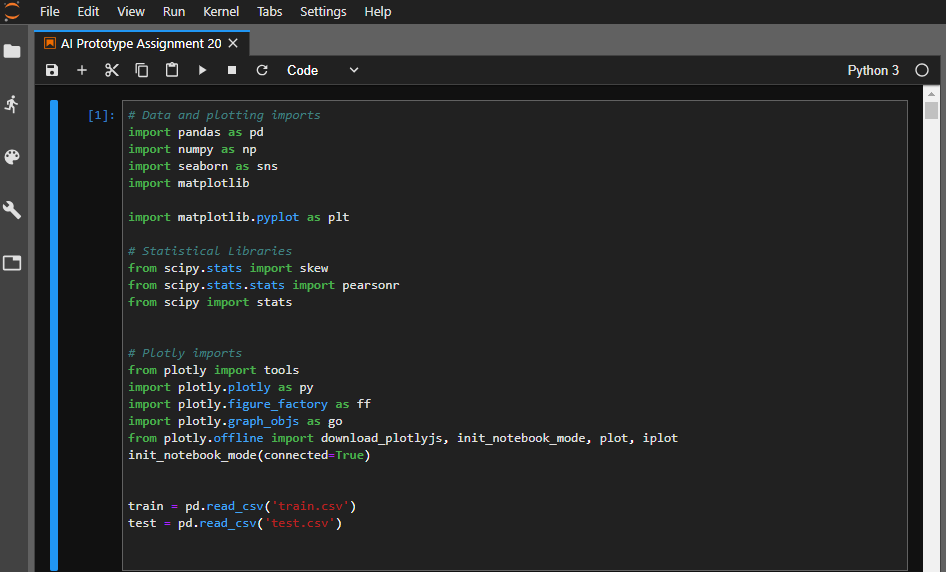
# Goal

The project goal is split in to two phases, namely:

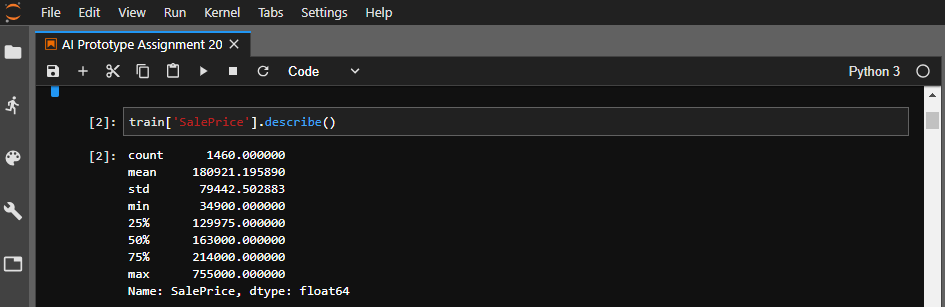
1. Exploratory Analysis of Data: This phase involves building a better understanding of the data that is involved in the experiment. Mainly, focusing on features that have the highest correlation to what is being predicted, the sale price.
2. Advanced Regression Modelling: This will be the implementation of a regression model which is able to predict sale price of a house.

# Prototype Development

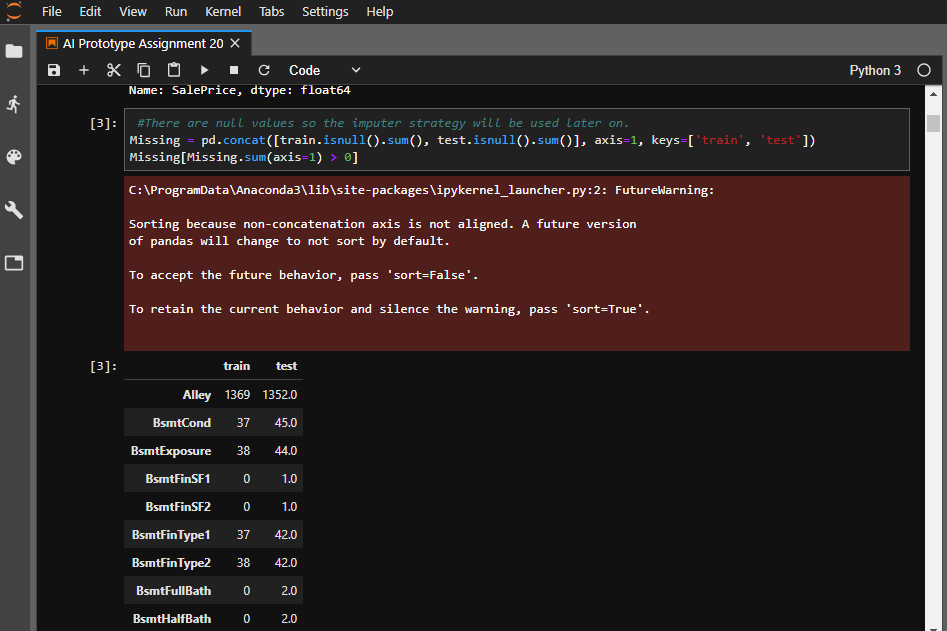
## Data Preparation

The first step is importing the datasets to be used in the experiment which is split into the train and test. The dataset chosen to perform this experiment is a csv file that contains a large bank of house information. The training set will be used to train the model and the test set will be used to test the trained model (Developers, 2018). Figure 1.0 below shows the primary library and dataset being imported.

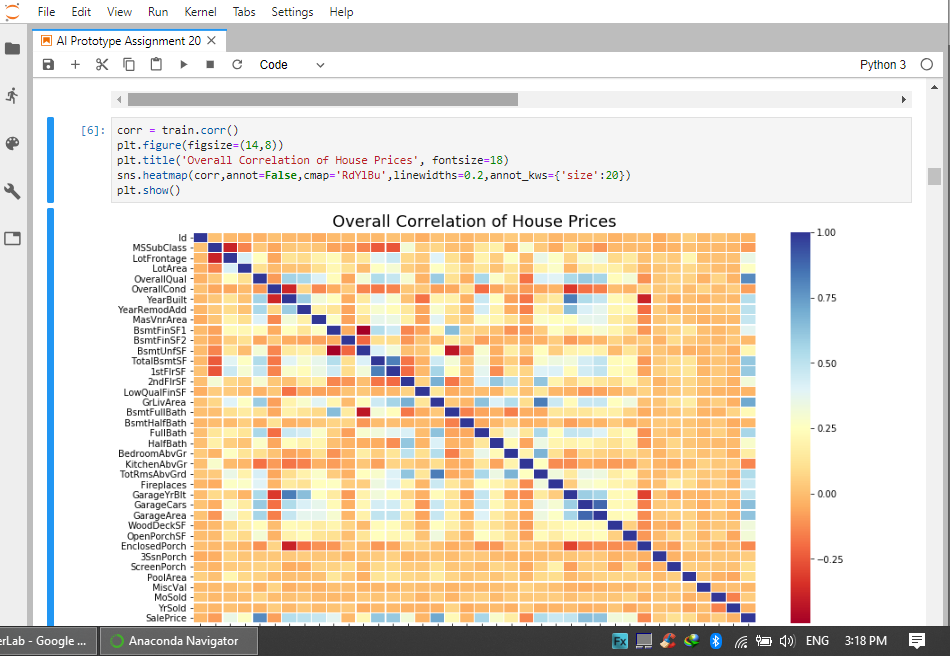
*Figure 1.0*

Figure 1.1 shows the sale price column data being described using the general analytics method.

*Figure 1.1*

**Figure 1.2 shows the missing values that were detected in the train and test datasets which will be address later. Columns with null values are to be replaced. It is important to know which columns have null values as they may affect our further experiments and accuracy at later stages.

*Figure 1.2*

Figure 1.3 shows the use of a heatmap to establish the correlation between house price and other variables in the train dataset. Values which are highly correlated to each other which are indicated by the dark blue color will be considered being dropped on either axis to prevent similar column data being fitted into the model. This excludes the sale price which is the value being predicted.

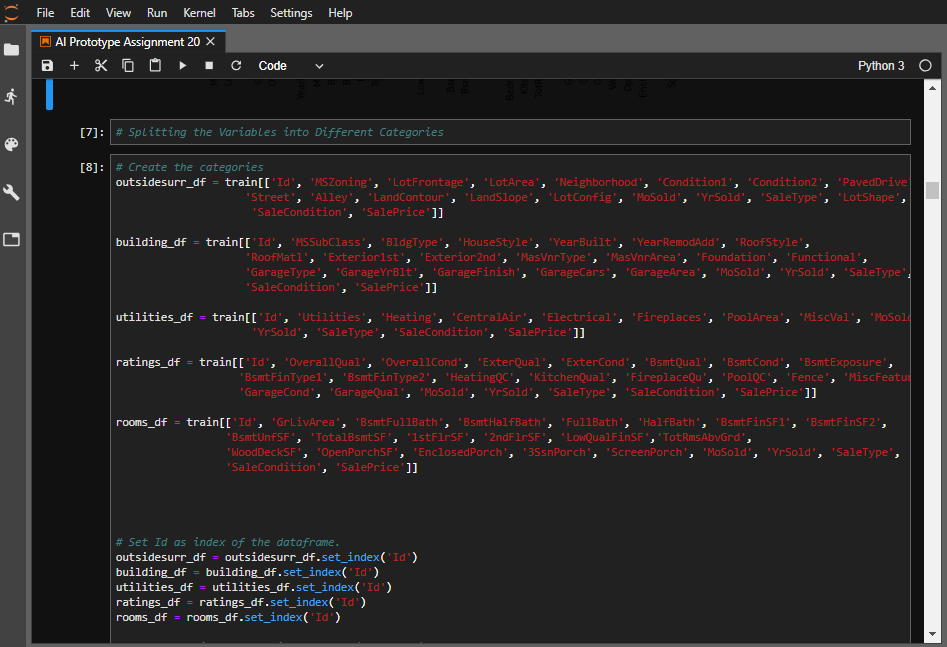
*Figure 1.3*

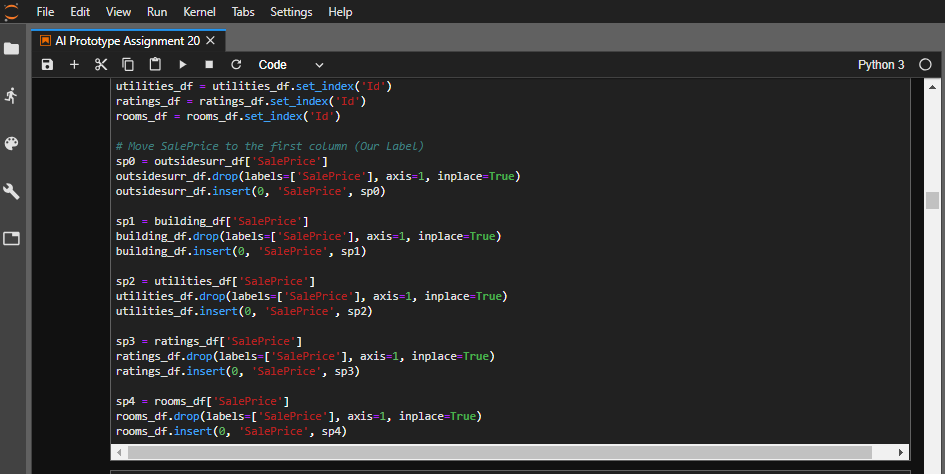
## Splitting of Variables into Different Categories

This step involves splitting data into different categories in order to split the analysis into segments.

This mainly involves:

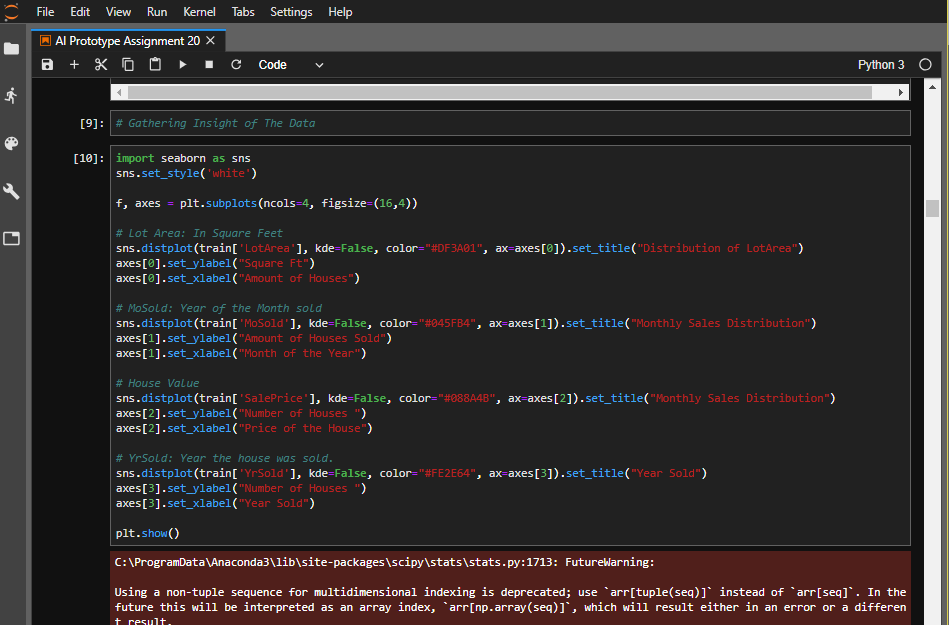
1. Separation of the data into different segments to make analysis easier.
2. The separate categories will include the sale price in order to see any notable patterns.
3. From there the linear regression model will be able to make accurate predictions of the sale price.
4. Each category will contain id, salesprice, MoSold, YearSold, SaleType and SaleCondition.

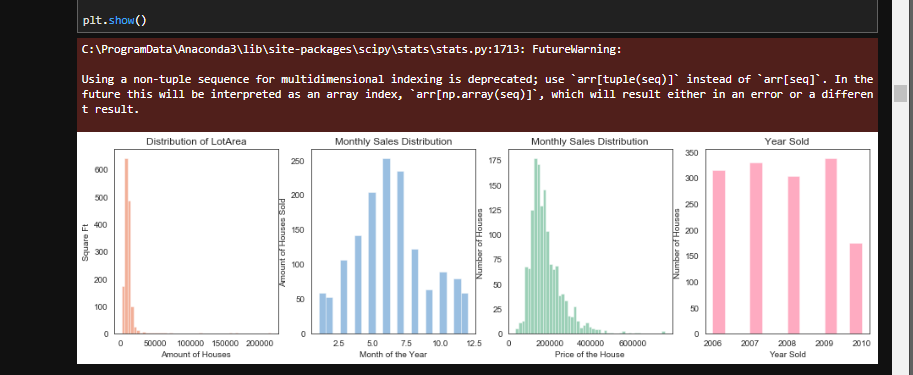
**This will allow for a more in-depth analysis of features used in the dataset after the data has been split up into segments that can be more carefully analyzed. Figure 1.5 and Figure 1.6 shows the splitting of data into five categories which are outsidesurr\_df, building\_df, utitilties\_df, ratings\_df and rooms\_df.

*Figure 1.5*

*Figure 1.6*

## Gathering Insight of Data

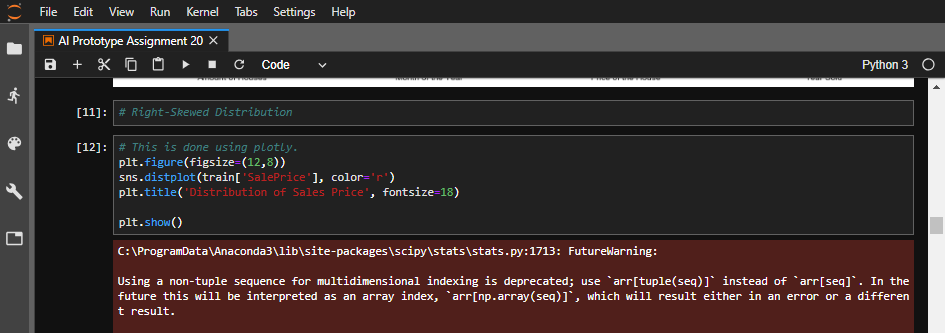
Figure 1.7 and 1.8 show the distribution plots of the LotArea, Monthly Sales (Month of the Year and Price of The House) and the Year Sold. From the graphs plotted two conclusions can be made:

*Figure 1.7*

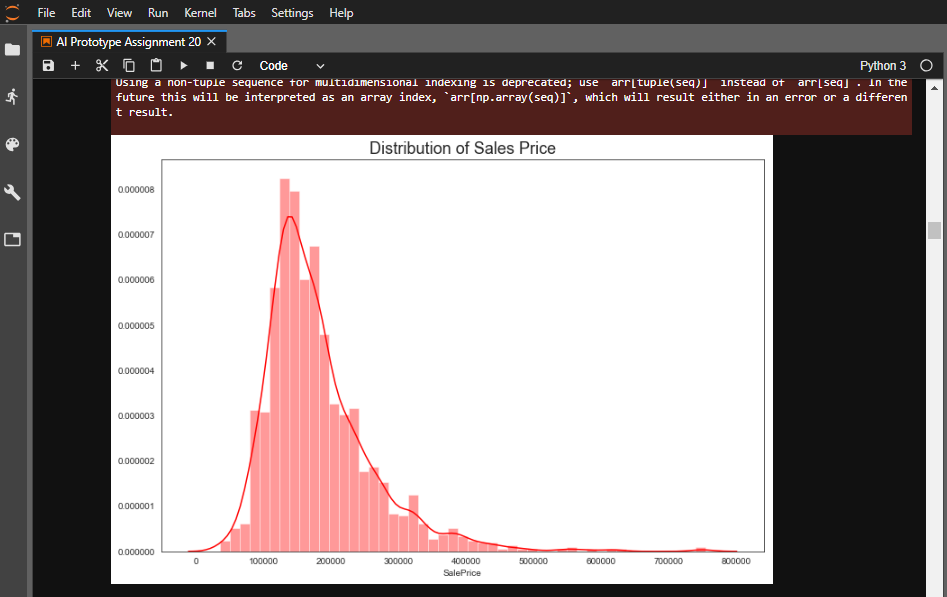
*Figure 1.8*

1. The distribution of the house prices is skewed to the right.
2. A drop in the number of houses sold is shown in the year 2010.

## Skewed Distribution

A skewed distribution occurs when one tail is longer than another in a distribution plot. A right-skewed distribution has a long right tail. Right-skewed distributions are also called positive-skew distributions. That’s because there is a long tail in the positive direction on the number line (Glen, 2018). This results in more outliers to the right of the distribution. Log transformations will be used at a later stage to transform the histogram into a normal distribution. Figures 1.9 and 2.0 show how the skewed distribution is plotted using the plotly.library.

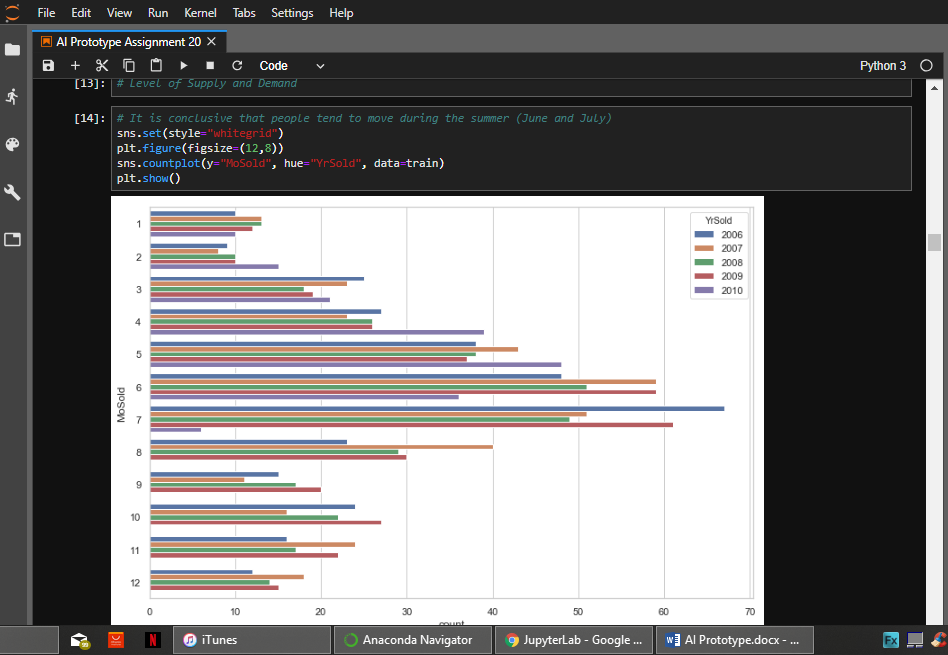
*Figure 1.9*



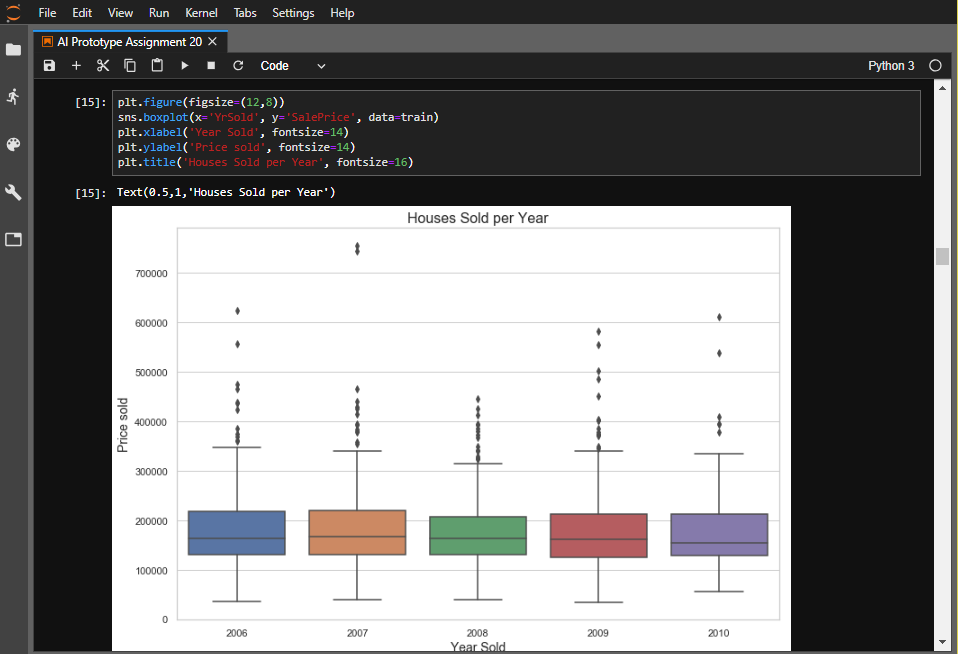
*Figure 2.0*

## Supply and Demand Analysis

As with commodities traded on the market, housing prices continually fluctuate, sometimes with drastic changes over a short period of time. Availability is a huge factor affecting prices within a set region, such as in a specific suburb of a metropolitan area (Kathy, 2018). House prices and sales are also affected by seasons of the year. Figure 2.1 shows a horizontal bar chart plot of the number of houses sold plotted against the month in which houses were sold. Figure 2.2 shows a boxplot of the price sold stacked against the year sold. From these plots the following can be concluded:

1. Most houses were sold in June and July.
2. The median price of houses was at its peak in the year 2007 and was lowest in 2010. This is possibly a result of economic recession discussed earlier.
3. Less houses were built and sold in 2010.

*Figure 2.1*

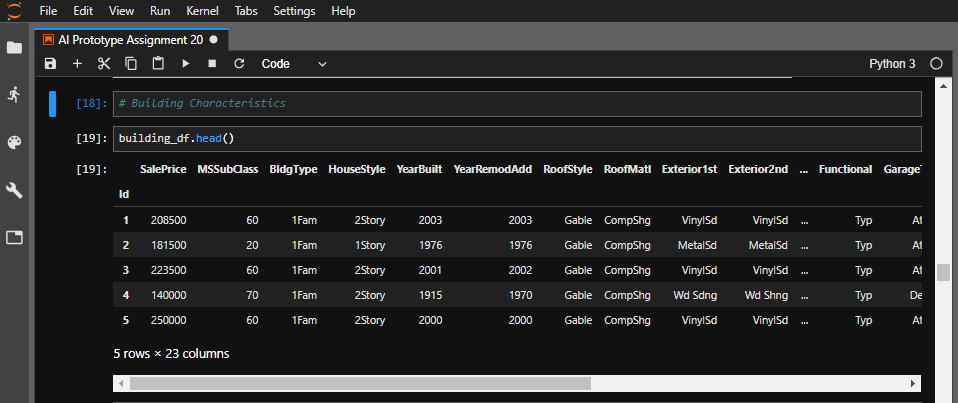
*Figure 2.2*

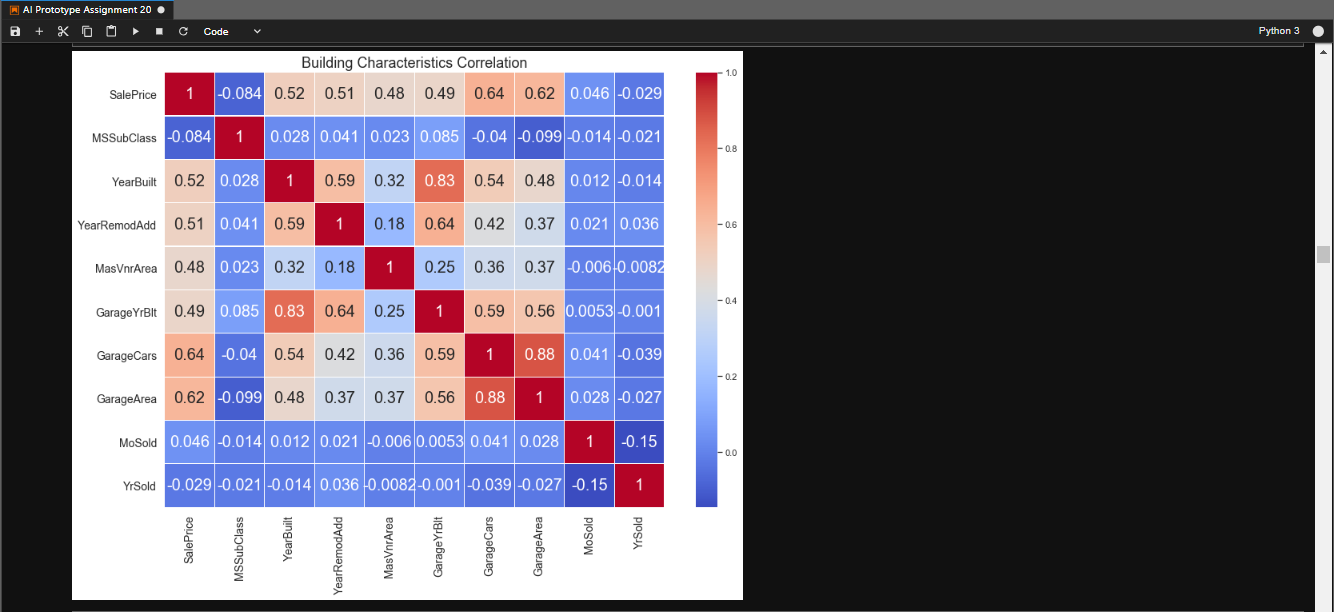
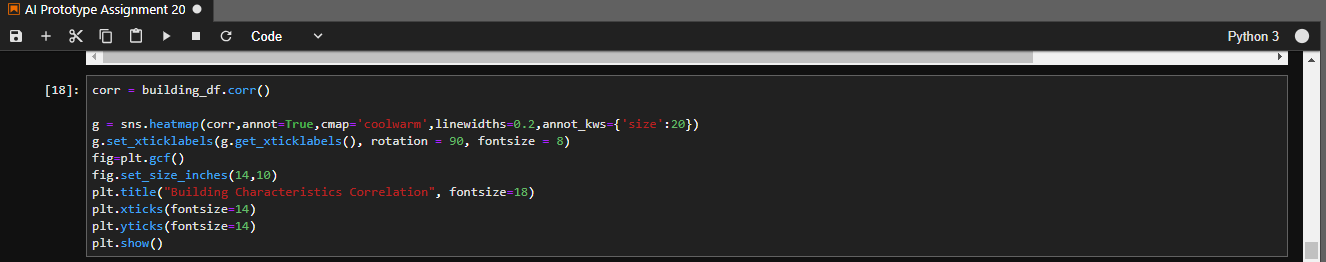
## House Characteristics

Figure 2.3 shows the columns that are characteristics of a house which will be used to establish correlation to the house price. This is the building\_df category which was split eariler Figure 2.4 is the correlation heatmap of the building\_df dataframe which is used to establish the correlation between different building characteristics and the house price.

From the heatmap above we can conclude that the following variables have high correlation with the Sale price:

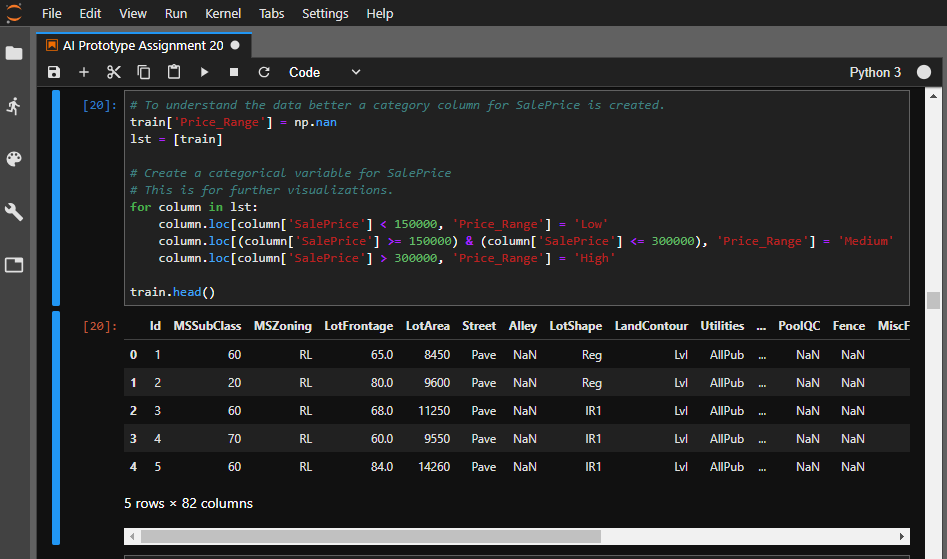
1. YearBuilt - the date the building was constructed
2. MasVnArea – masonry area
3. YearRemodAdd- the last time the building was remodeled
4. GarageYrBlt - year that the garage was built
5. GarageCars – the garage car capacity
6. GarageArea – the size of the garage



*Figure 2.3*

*Figure 2.4*

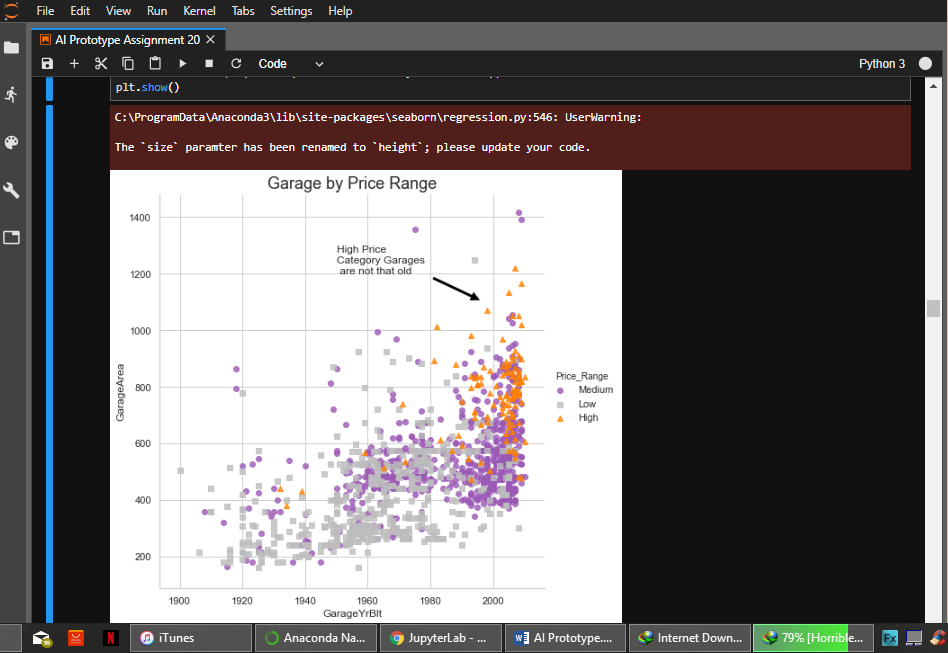
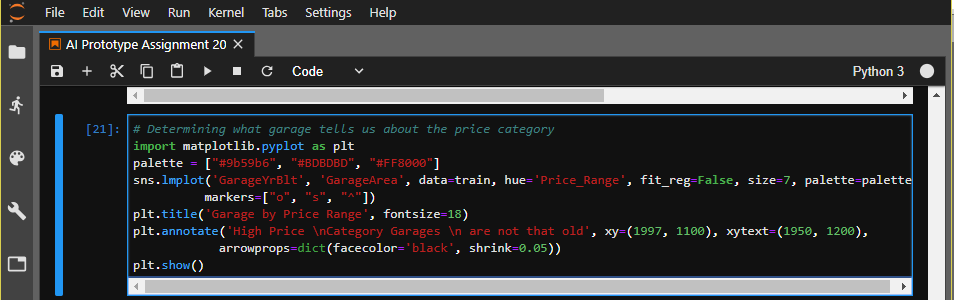
Figure 2.5 shows how better insight for house price was made possible by creating a category column which separated houses into Low, Medium and High cost houses using range bound criteria as follows:

1. Houses costing less than 150000 are low cost houses.
2. Houses costing greater than 150000 are medium cost houses.
3. Houses costing greater than 300000 are high cost houses.

*Figure 2.5*

## Relationship Between Garage and Price Category

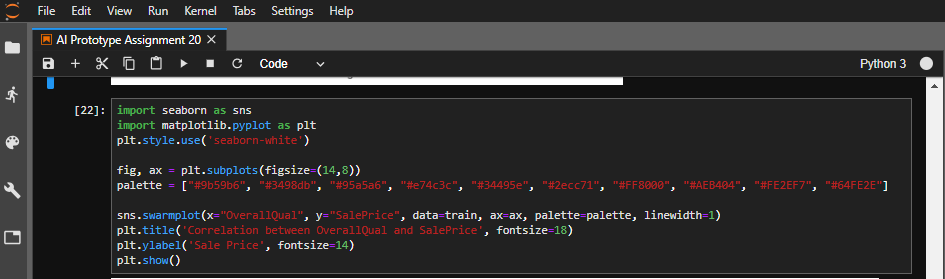
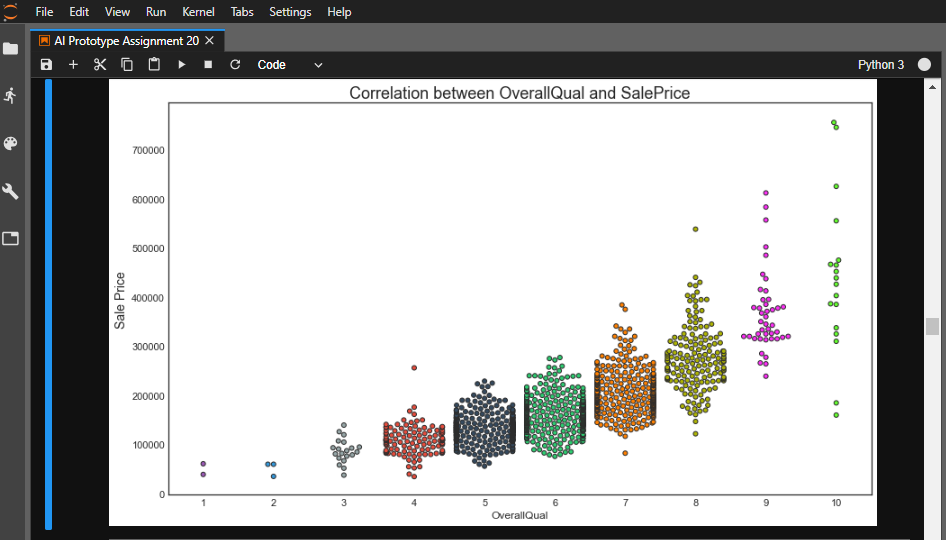
Figure 2.6 shows how lmplot which is a combination of regplot() and FacetGrid (Waskom, 2018) was used to establish what the garage in a house can tell us about the price category it falls into. The plot in the figure above shows that garages of high price category are more concentrated in the GarageYrBlt of 2000, therefore we can conclude that high price category garages are not very old.



*Figure 2.6*

Figure 2.7 shows a swarm plot of Overall house quality stacked against Sale Price. The following conclusions can be drawn from this plot:

1. Houses in the Overall quality extremes have very low occurrences.
2. Houses of overall quality 10 have somewhat even sale price distribution.
3. Majority of houses fall in the quality categories 4 to 8.

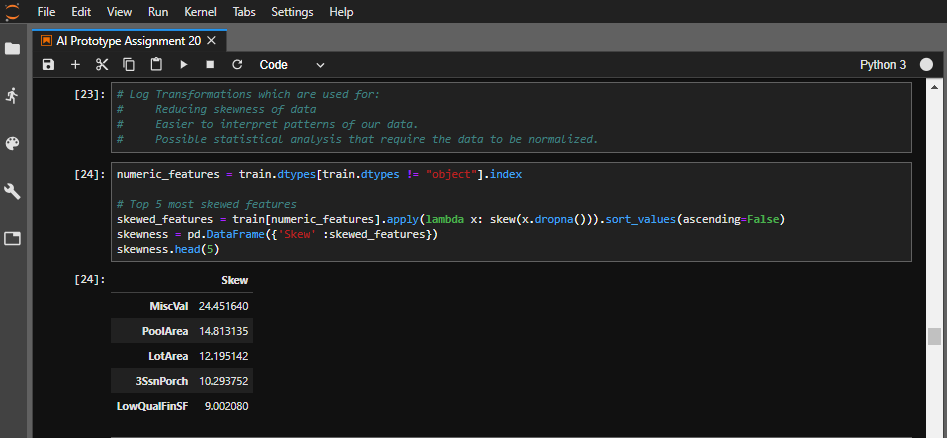
*Figure 2.7*

## Log Transformations

The log transformation can be used to make highly skewed distributions less skewed. This can be valuable both for making patterns in the data more interpretable and for helping to meet the assumptions of inferential statistics (Lane, 2018). The main purpose for log transformations in this experiment is to:

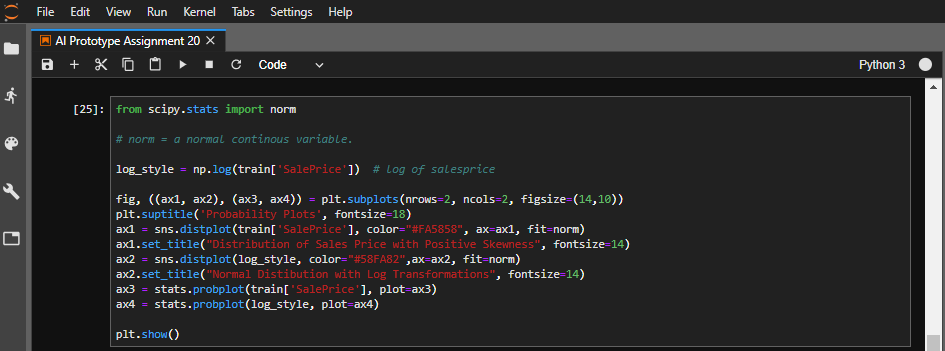
1. Make interpretation of patterns in this data easier.
2. Make statistical analysis that requires data to be normalized possible.

Figure 2.8 shows the values of the five most skewed features of the dataset in descending order.



*Figure 2.8*

Figure 2.9 shows the distribution of sales price before and after normalization with log transformation. The plot is no longer skewed to the left as it was before after normalization. The probability plot before normalization also shows the data plots bending up, while the probability plot on normalized data shows the data points approximately along a straight line showing that the distribution correctly represents the data.

*Figure 2.9*

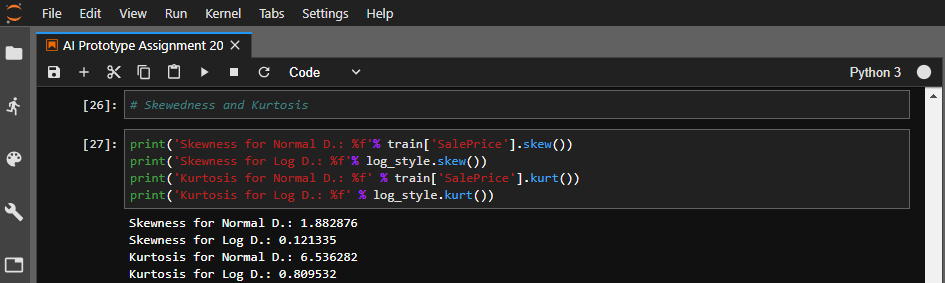
## Skewedness and Kurtosis

### Skewedness:

1. A skewedness of zero or a value close to zero indicates a symmetrical distribution
2. A negative value indicates skewedness to the left.
3. A positive value indicates skewedness to the right.

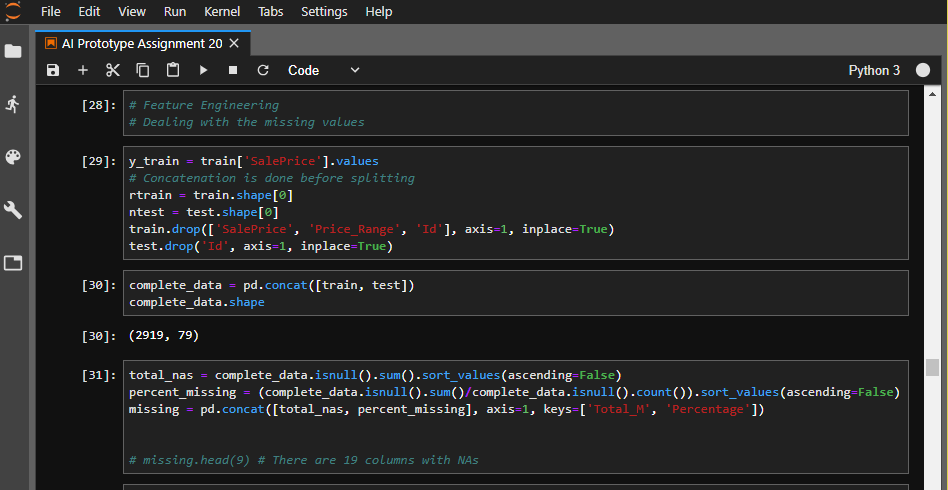
### Kurtosis:

This is a measure of how extreme observations in a dataset are. A larger kurtosis coefficient indicates a more peaked distribution around the mean (KENTON, 2018).

Greater kurtosis coefficient also means larger tails which increases extreme results (KENTON, 2018). Figure 3.0 shows the skewedness and kurtosis of the sale price in both the normal distribution and the log transformed distribution.

*Figure 3.0*

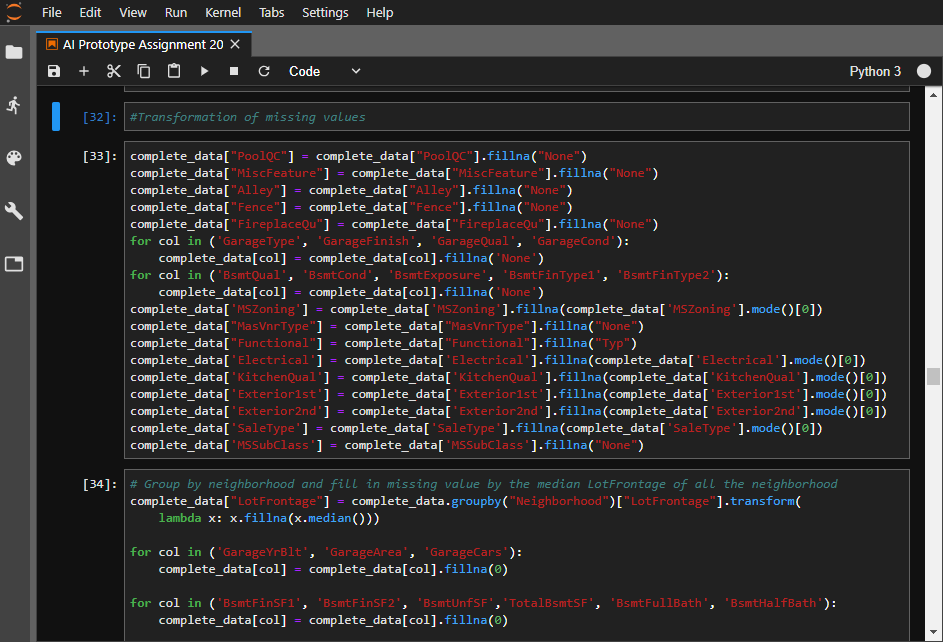
## Feature Engineering

This is the phase of the project where missing values are filled. The missing values can either be a result of some information not being captured or recorded but can also be a result of some data corruption. Presence of missing values in a dataset can result in errors when implementing some machine learning algorithms, hence this important phase which will be outlined.

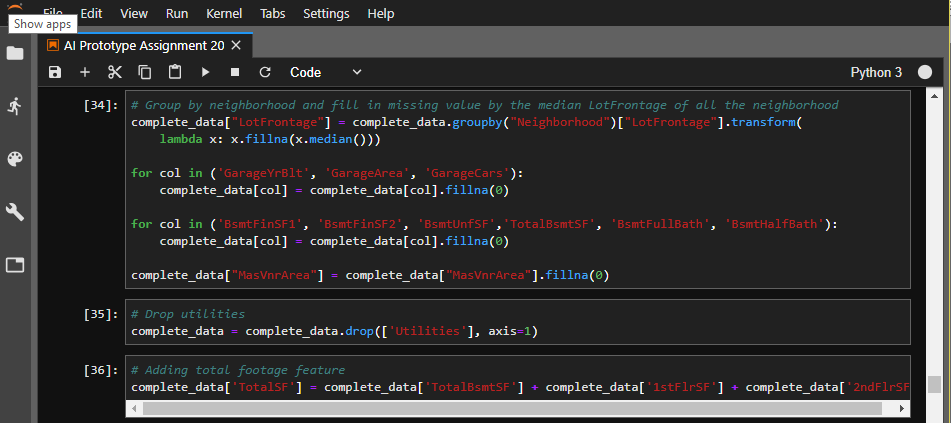
*Figure 3.1*

### Correcting of Missing Values

Figure 3.2 shows the first stage of feature engineering which involves filling columns with string values based on presence of a feature in a house.



*Figure 3.2*

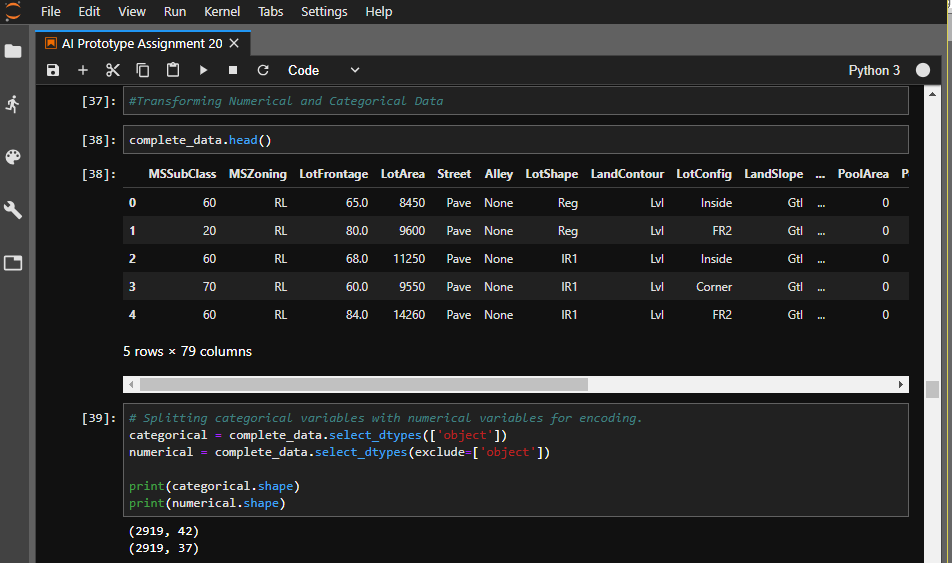


*Figure 3.3*

Figure 3.3 shows how the LotFrontage field missing values were addressed by grouping the values by their neighbors and filling the missing values using the median value. Other columns with NA values were filled with a zero value instead of being left with a blank field. The utilities column was dropped as it had insignificant correlation to Sale Price and is therefore irrelevant.

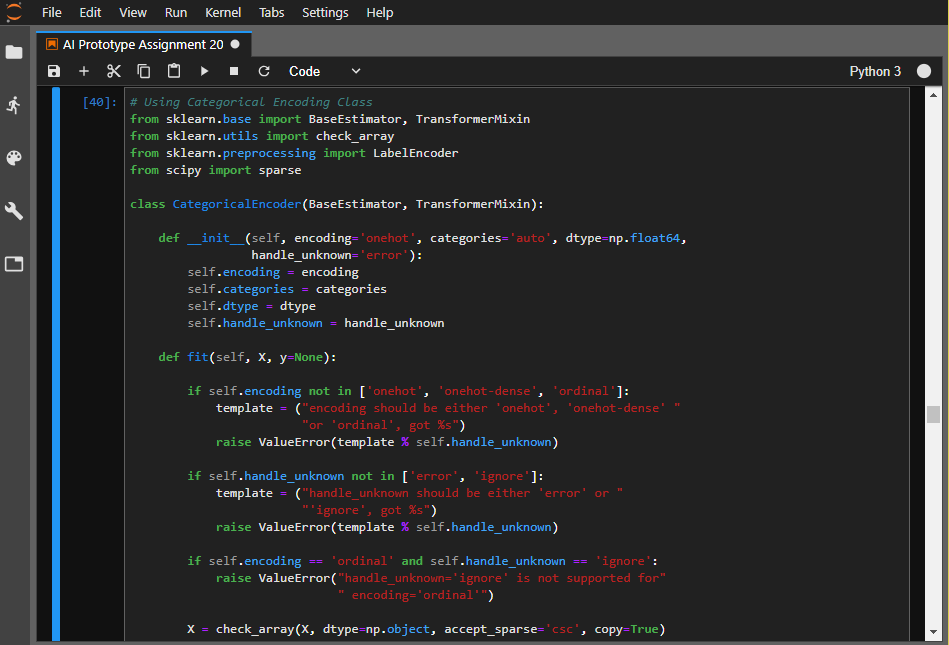
### Correcting Numerical and Categorical Values

This step will involve the following procedures:

1. Separation of features and labels from training dataset.
2. Separation of numerical and categorical variables in order for them to be ran in separate pipelines and scaled with their respective scalers.

*Figure 3.4*

Figure 3.4 shows how the variables of the dataset were split into categorial and numerical data as a preparatory step before encoding.

*Figure 3.5*

The categorical encoding class is used to encode features in such a way that assumption that two nearby values are more similar than two distant values is avoided. Figure 3.5 shows a code snippet of the categorical encoding class used with the OneHotEncoder.

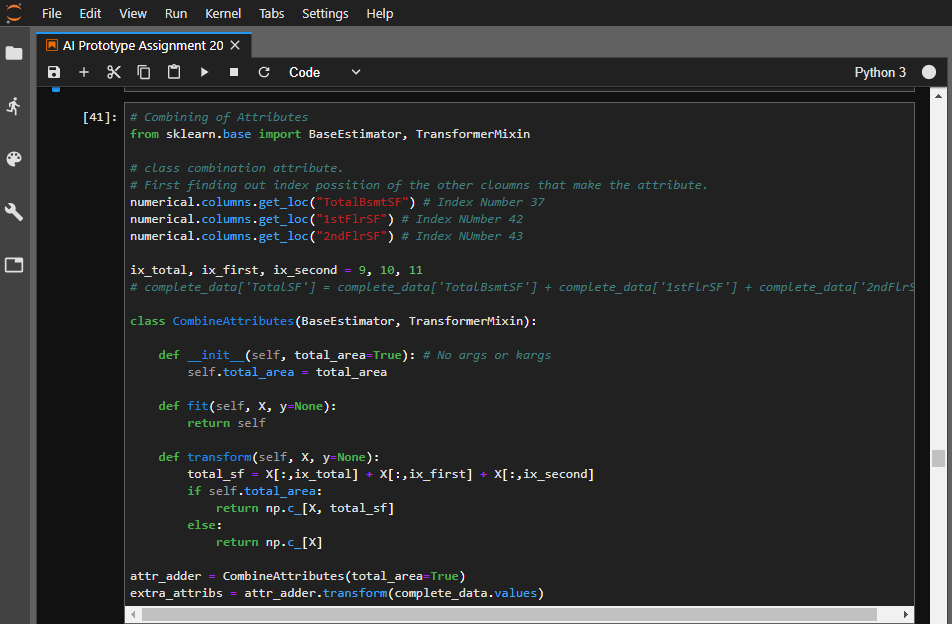
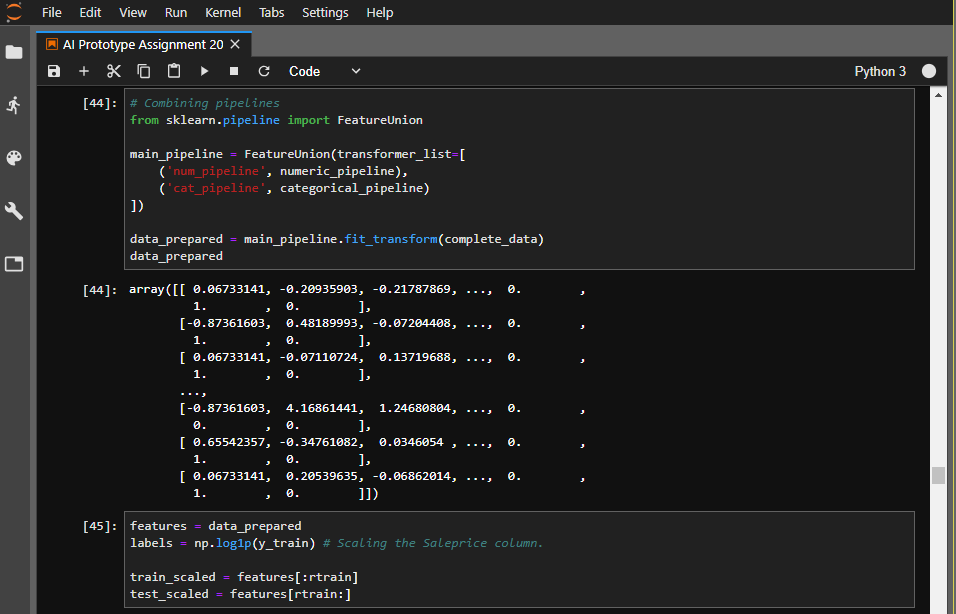
*Figure 3.6*

Figure 3.6 shows the combing attribute class which is used to create a total area variable which is a combination of TotalBsmtSF, 1stFlrSF and 2ndFlrSF. This value will be added to the pipeline and further scaled.

### Pipelines

*Figure 3.7*

Pipelines will be used to scale features of the numerical and categorical data. Figure 3.7 shows the creation of the two pipelines and the classes used to correct values in the two data categories.

*Figure 3.8*

Figure 3.8 shows the combining process of the two pipelines created to form the main pipeline. The main pipeline is then transformed using the final estimator. Prepared data is split into train and test scaled data before predictive models can be created.

# Predictive Models

## Residual Plot

According to (Stattrek, 2018) residual plot is a graph that shows the residuals on the vertical axis and the independent variable on the horizontal axis. If the points in a residual plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a non-linear model is more appropriate.

The formula for the residual plot is:

Residual = Observed value - Predicted value

The linear model is denoted by: γ where is the vector of fixed coefficients and γ is the vector of random effect coefficients.

Residual plots give actual prediction errors models make. For the residual plot in this experiment a simple linear regression model will be used along with the yellowbrick library for visualizations. The residual plot will be built based on validation set.

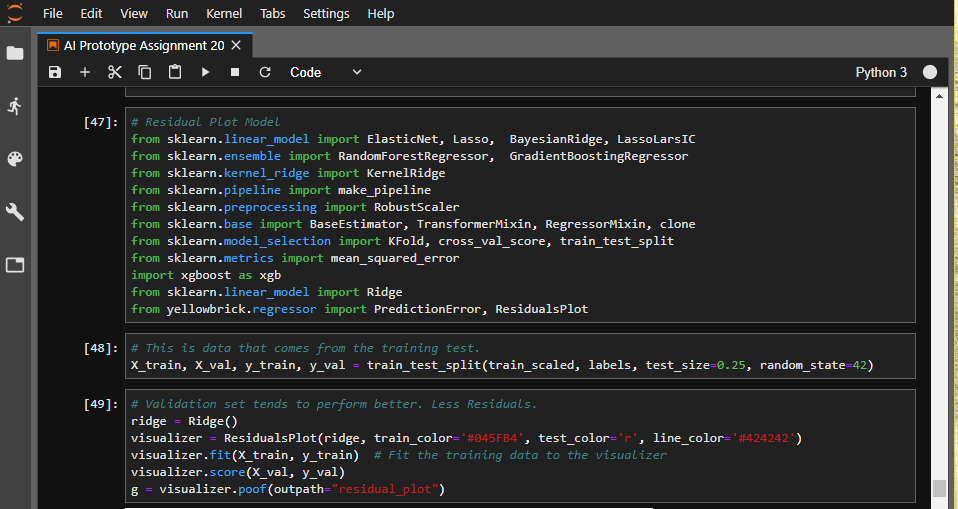
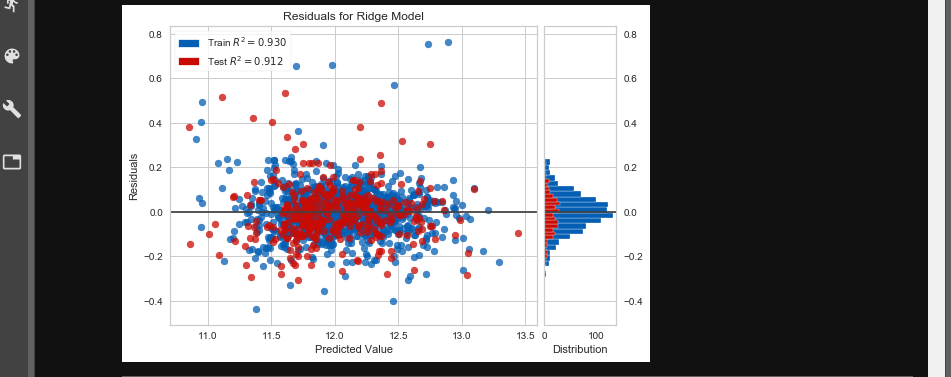
*Figure 3.9*

Figure 3.9 shows the residual plot of the train and test sets.

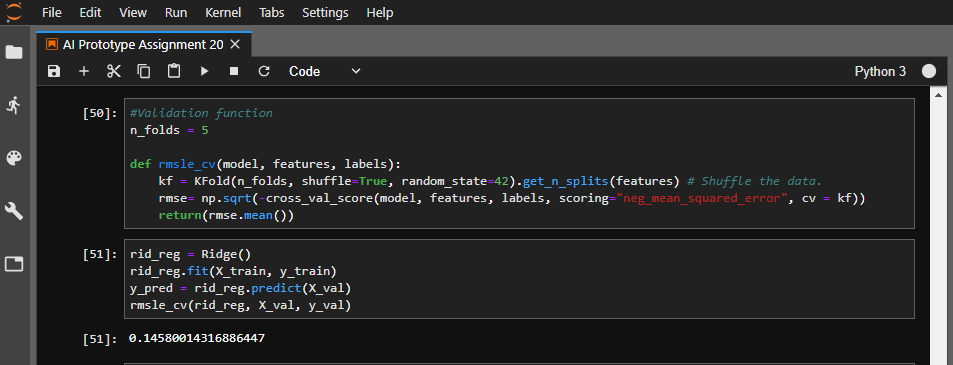
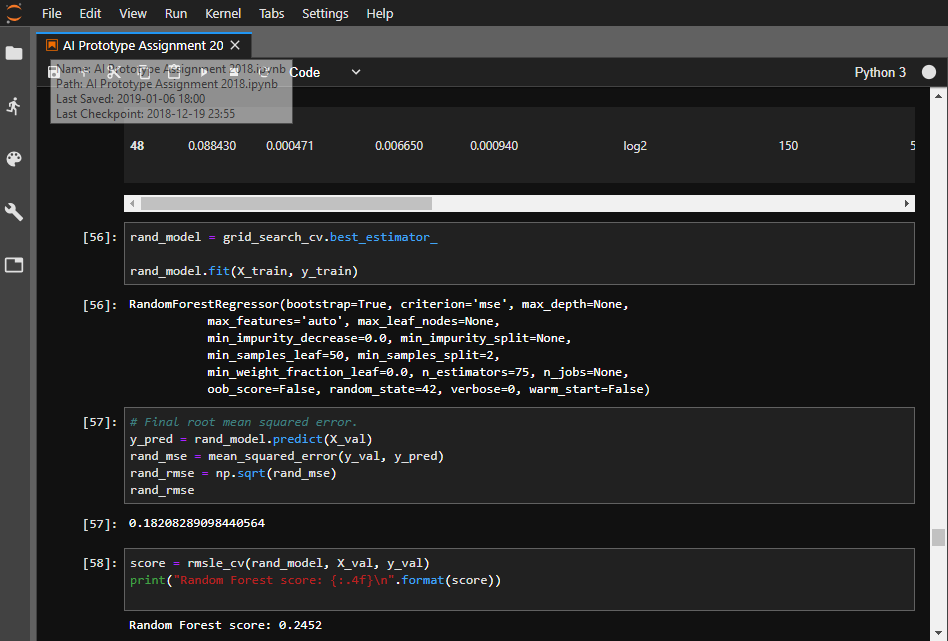
*Figure 4.0*

Figure 4.0 shows the validation function of the residual plot. The scoring criteria of the model is the mean square error which measures the accuracy of models in terms of error. Less is better, the best possible score being 0. The residual plot had a score of 0.146.

### Random Forest Regressor

The Random Forest is one of the most effective machine learning models for predictive analytics (Tutori, 2018). It gives the experiment randomness, instead of having to look through values for the best it picks its features in a completely random way and limit variance. This however results in a higher bias.



*Figure 4.1*

The final score for the random forest regressor is shown in figure 4.1. The scoring criteria is the root mean square error which is 0.2452 which according to me is to be expected from a random feature model.

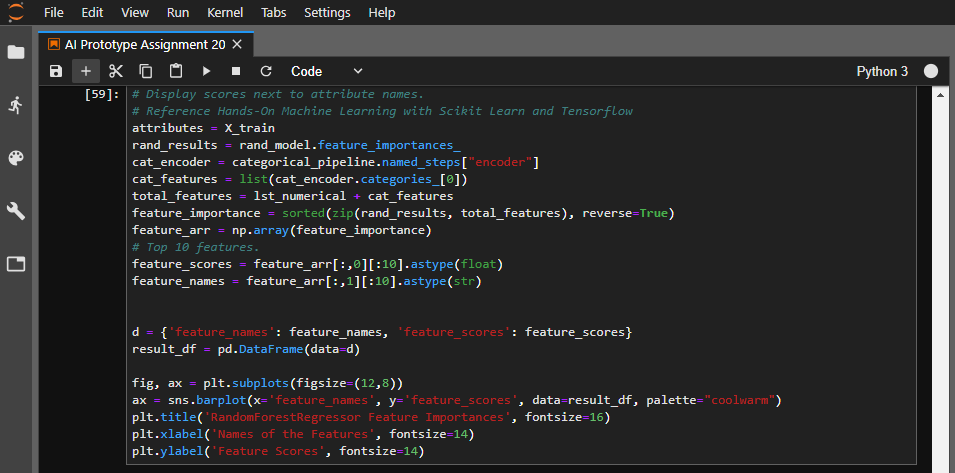
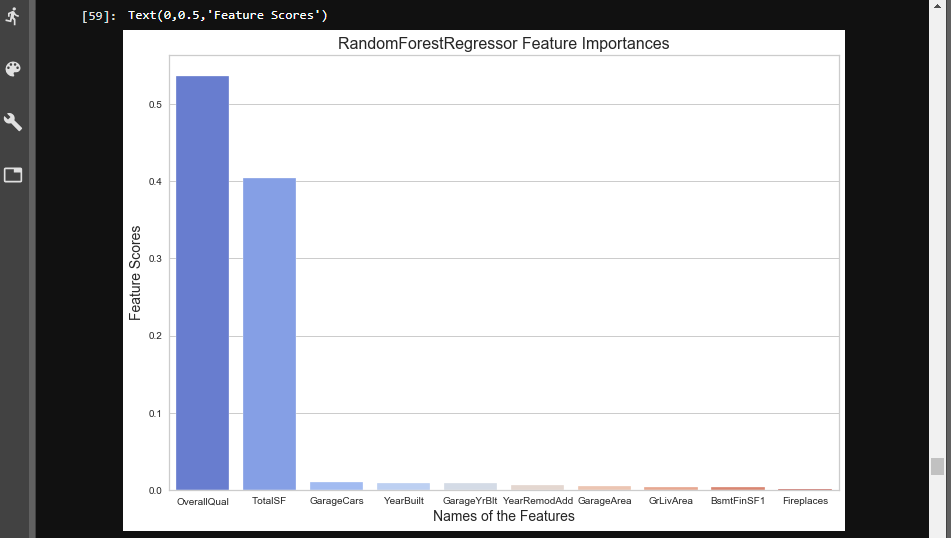
*Figure 4.2*

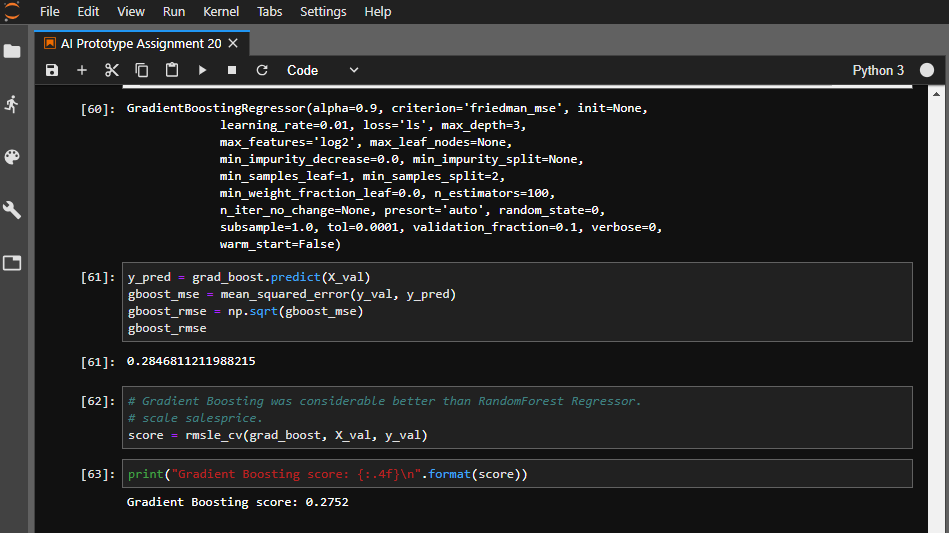
Figure 4.2 shows a further step which was taken to plot a bar chart which displays the features which the random forest regressor determined as most significant. These are the Overal Quality and the total surface area.

### Gradient Boosting Regressor

A gradient boosting regressor is an ensembled decision tree regressor model. At each step, a new tree is trained against the negative gradient of the loss function, which is analogous to (or identical to, in the case of least-squares error) the residual error (Group, 2018). It is suitable for building this model because it trains models over the residuals (prediction errors) which results in small variance and high accuracy.

Its objective is to define the loss function and minimize it. In this case the mean square error will be used as loss definition:

Where is the target value, is the prediction value and is the loss function.



*Figure 4.3*

Figure 4.3 shows the gradient boosting regressor’s default configuration and mean squared error score which is 0.2752.

### Parameter Tuning

*Figure 4.4*

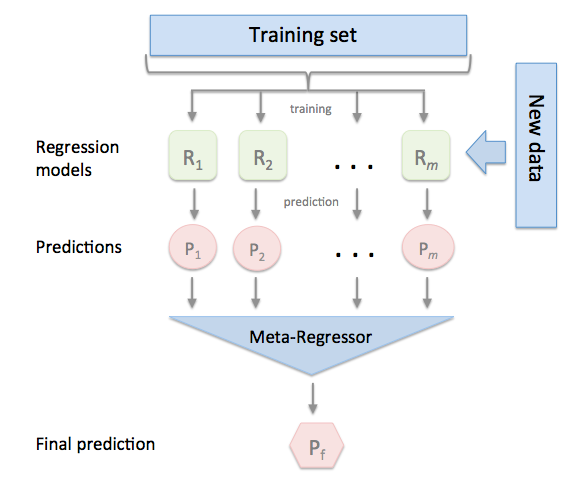
Figure 4.4 shows the gradient boosting regressor with tuned parameters with guidance from scikit learn documentation. Parameters altered are as follows:

1. Learning rate “0.5” from “0.1”
2. Loss function “huber” instead of “ls”
3. Max depth “2” from “3”
4. Min\_sample\_leaf “14” from “1”
5. Min\_sample\_split “10” from “2”
6. N\_estimators “3000” from “100”

Parameter tuning yielded an accuracy of 0.1403 which is 0.1349 better than the initial regressor.

### Stacking Regressor

A stacking regressor is an ensemble learning technique that combines multiple regression models via a meta-regressor. Each individual model is trained based on the complete training set then the meta-regressor is fitted based on the outputs (Breiman, 1996).



*Figure 4.5 (mlxtend, 2018)*

Figure 4.5 shows the flow of operation in a stacking regressor. The training set is fitted into each regression model selected to be used and the predictions are then loaded into the meta-regressor which is fitted based on outputs given from the individual regression models. This then results in the final output.

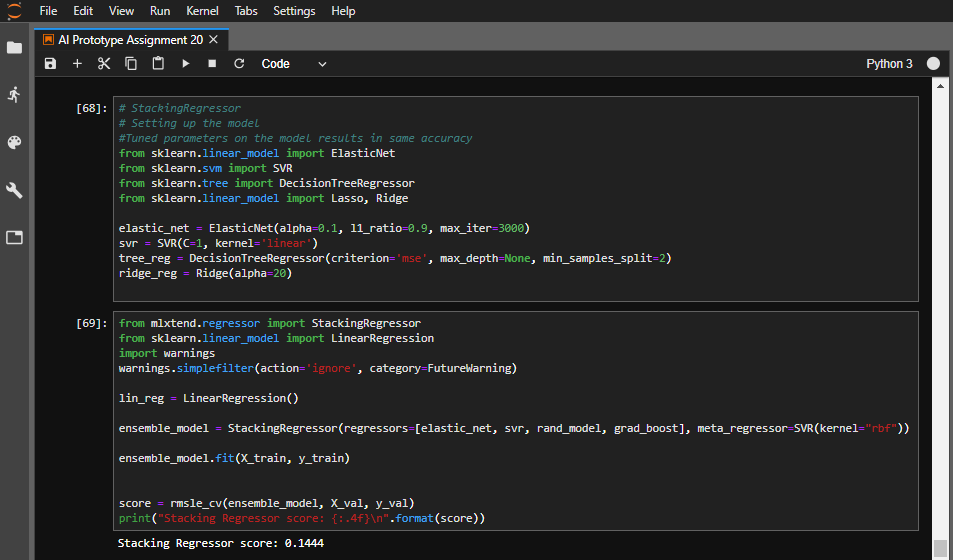
*Figure 4.6*

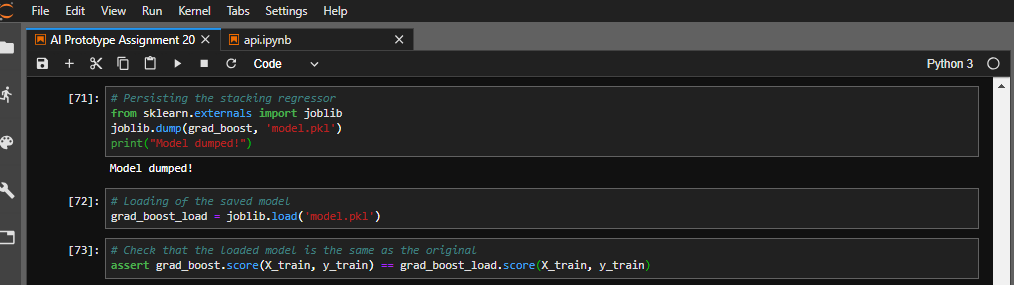
Figure 4.6 shows the stacking regressor model with the following models stacked:

1. Elastic\_net
2. Svr
3. Rand\_model
4. Gradient boost

These four models were loaded via the meta-regressor and resulted in a mean square logarithmic error of 0.1444.

Confusion Matrix could not be used to assess the models because classification models can not be used to access multiclass and continuous targets used in the models.

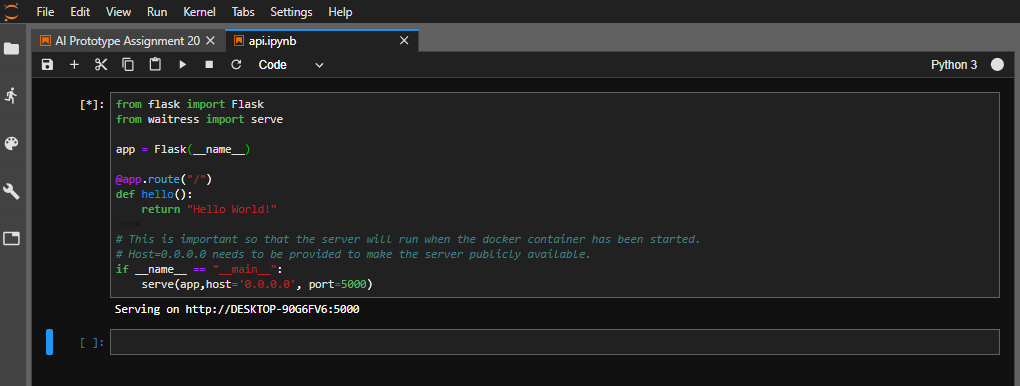
## Making Data Persistent

This is the process of saving data so that it can be reverted back if the need arises for the data to be used again. Staying true to the term persistence, the data must remain the same. The model persisted in this case is the Gradient boosting regressor which was the most accurate model with a mean square logarithmic error closest to zero.

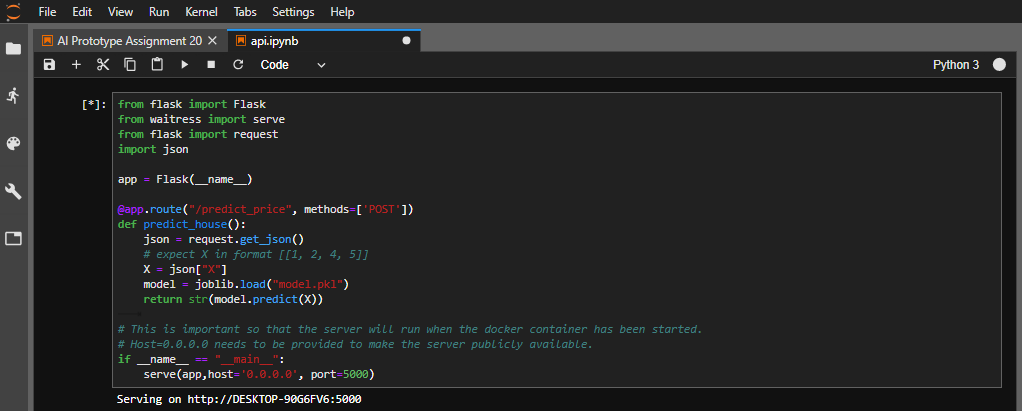
*Figure 4.7*

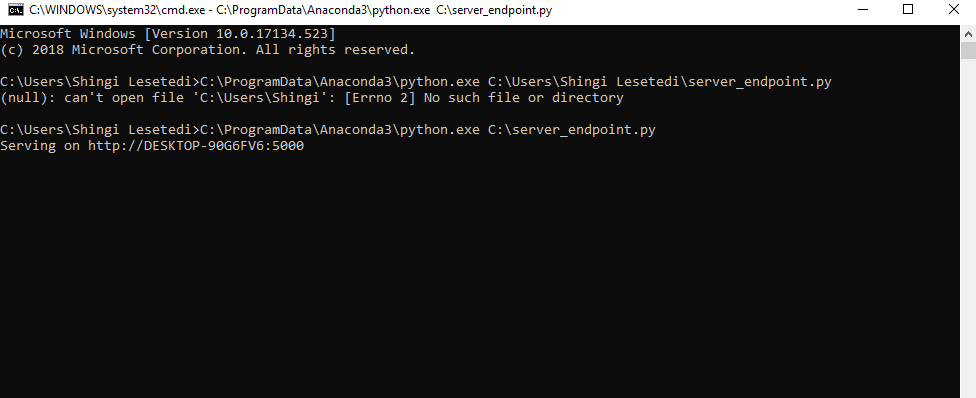
Figure 4.7 shows how the gradient boosting model being dumped via pickle and showing that the persisted model and the original model have the same output. This was done using the assert statement which did not throw an error.

## Deployment of the Model

This step involves wrapping the machine learning model into an API. This will involve developing an API using flask that will predict the house price using the persisted model above. A test sever was created to check the functionality of the local server. See figure 4.8 below.

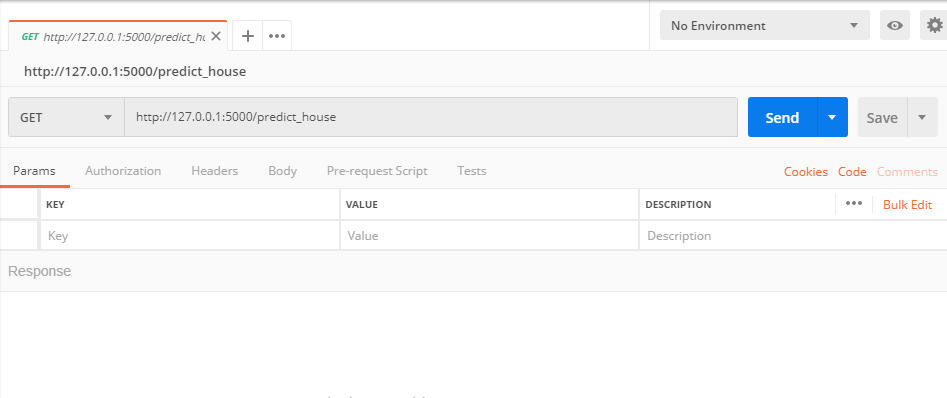
*Figure 4.8*

*Figure 4.9*

Figure 4.9 above shows creation of the endpoint using the pickled model.

*Figure 5.0*

Figure 5.0 shows the server endpoint being run as an executable python file. Problems are encountered when trying to run the server in jupyter notebook/jupyter lab.

*Figure 5.1*

After running the server, Postman was used to deploy the API. Postman is an API platform software that is used to run web applications and return results based on the input parameters given.

# Development Reflection

The development of the prototype was carried out using the phases of machine learning model development namely, data preparation, feature engineering, model development and deployment of the most accurate model. The code used to develop the entire project is commented to describe steps taken and the reasons behind taking the steps.

# Evaluation Report

The prototype developed for predicting house prices served its purpose of predicting the sale price of a house. The evaluation criteria used for all of the regression algorithms used was the mean square logarithmic error.

## Alternative Predictive Models

### Use of Keras and Tensorflow

The accuracy of the neural network is not as accurate as that of the ensemble and gradient boosting. An alternative model would be the implementation of a neural network with a regression model and showing the results of the neural networks using tensorboard.

# Conclusion

This prototype is not yet complete but demonstrates a comprehensive visualization of data in House Price dataset and utilizes various regressor algorithms which helped in predicting the price of house. Four algorithms were used to predict house prices and the most accurate of the models was the Gradient boosting regressor which was the most accurate with a root mean square logarithmic error of 0.1403. This model was then attempted to be deployed after being persisted to not much success. Future improvements of the project would be to include the neural network models which would be compared with regressor models implemented above. In addition further effort would be made to deploy the model successfully.

# References

Breiman, L., 1996. Stacked Regressions. *Machine Learning Issue 1*, pp. 49-64.

Developers, G., 2018. *Training and Test Sets: Splitting Data  |  Machine Learning Crash Course.* [Online]   
Available at: https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data  
[Accessed 5 December 2018].

Géron, A., 2017. *Hands-On Machine Learning with Scikit-Learn and TensorFlow.* s.l.:O'Reilly Media, Inc..

Glen, S., 2018. *Skewed Distribution: Definition, Examples - Statistics How To.* [Online]   
Available at: https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/skewed-distribution/#whatisskew  
[Accessed 6 November 2018].

Group, O. D., 2018. *Gradient Boosting Regressor | Open Data Group.* [Online]   
Available at: https://opendatagroup.github.io/Knowledge%20Center/Tutorials/Gradient%20Boosting%20Regressor/  
[Accessed 7 December 2018].

Jain, S., 2017. *A comprehensive beginners guide for Linear, Ridge and Lasso Regression.* [Online]   
Available at: https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/  
[Accessed 7 December 2018].

KENTON, W., 2018. *Mesokurtic.* [Online]   
Available at: https://www.investopedia.com/terms/m/mesokurtic.asp  
[Accessed 7 December 2018].

Lane, D. M., 2018. *Log Transformations.* [Online]   
Available at: http://onlinestatbook.com/2/transformations/log.html  
[Accessed 7 December 2018].

Marcelino, P., 2017. *Comprehensive data exploration with Python.* [Online]   
Available at: https://www.kaggle.com/pmarcelino/comprehensive-data-exploration-with-python  
[Accessed 7 December 2018].

mlxtend, 2018. *StackingRegressor - mlxtend.* [Online]   
Available at: http://rasbt.github.io/mlxtend/user\_guide/regressor/StackingRegressor/  
[Accessed 7 December 2018].

PRITCHARD, J., 2018. *What Caused the Mortgage Crisis?.* [Online]   
Available at: https://www.thebalance.com/mortgage-crisis-overview-315684  
[Accessed 5 December 2018].

Ribeiro, V. H. A., 2018. *Simple House Price Prediction Stacking.* [Online]   
Available at: https://www.kaggle.com/vhrique/simple-house-price-prediction-stacking  
[Accessed 6 December 2018].

Serigne, 2018. *Stacked Regressions to predict House Prices.* [Online]   
Available at: https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard  
[Accessed 6 December 2018].

Solutions, S., 2018. *Missing Values in Data - Statistics Solutions.* [Online]   
Available at: https://www.statisticssolutions.com/missing-values-in-data/  
[Accessed 7 December 2018].

Stattrek, 2018. *Residual Plot: Definition.* [Online]   
Available at: https://stattrek.com/statistics/dictionary.aspx?definition=residual%20plot  
[Accessed 7 December 2018].

Tutori, 2018. *Random Forest Regression | Turi Machine Learning Platform User Guide.* [Online]   
Available at: https://turi.com/learn/userguide/supervised-learning/random\_forest\_regression.html  
[Accessed 7 December 2018].

Waskom, M., 2018. *seaborn.lmplot — seaborn 0.9.0 documentation.* [Online]   
Available at: https://seaborn.pydata.org/generated/seaborn.lmplot.html  
[Accessed 7 December 2018].