**MMA867 – Predictive Modelling**

**Assignment One – Individual**

**Kaggle Name:** Vanessa Afolabi

**Total Number of Teams on Leaderboard:** 4740

**Last Position on Leaderboard:** 1676

**Github:** <https://github.com/VanessaAfolabi/MMA867---Predictive-Modelling>



1. The three competitions I identified are stated below.
   1. **Predict Future Sales**

**Kaggle’s Description is as follows:**

*“In this competition you will work with a challenging time-series dataset consisting of daily sales data, kindly provided by one of the largest Russian software firms -*[*1C Company*](http://1c.ru/eng/title.htm)*.*

*We are asking you to predict total sales for every product and store in the next month. By solving this competition, you will be able to apply and enhance your data science skills.”*

* 1. **House Prices: Advanced Regression Techniques**

**Kaggle’s Description is as follows:**

*“Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.*

*With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home.”*

* 1. **Walmart Recruiting – Store Sales Forecasting**

**Kaggle’s Description is as follows:**

*“In this recruiting competition, job-seekers are provided with historical sales data for 45 Walmart stores located in different regions. Each store contains many departments, and participants must project the sales for each department in each store. To add to the challenge, selected holiday markdown events are included in the dataset. These markdowns are known to affect sales, but it is challenging to predict which departments are affected and the extent of the impact.”*

**Explanation of my choice.**

The choice I made was Predicting House Prices using Advanced Regression Techniques. It is more of a straightforward regression problem and less of a Time Series problem like the other two options. I also found it to be rather interesting because I enjoy anything related to Real Estate. Also, looking at all aspects of residential homes and how it relates to Sale Prices is a stimulating topic to me.

1. Explanation of the model

**Data Preparation**

Both the train and test csvs were read as pandas dataframes. The ID column was saved and then dropped from both datasets because ID will not be used in the regression model. The ID column of the test set will be used in the end when preparing the final prediction set to be submitted to Kaggle.

**Outlier Removal**

I plotted SalePrice against GrLivArea and noticed that there were two outliers where the Greater Living Area was extremely high but the Sale Price was low. This is not a normal case in real life thus these two cases were treated as outliers and removed.

**Exploration and Log Transformation of SalePrice**

The SalePrice variable in the training set is being predicted and it is important to understand its behavior. A statistical summary of SalePrice was generated showing values such as maximum, minimum and mean. In addition, both a histogram and QQ Plot were generated in order to determine if SalePrice is normally distributed. It turns out that SalePrice is skewed and not normally distributed. To fix this problem, a log transformation was applied to SalePrice. This immediately made SalePrice normally distributed.

**Correlations, Heatmaps and Scatterplots**

I plotted a Correlation Matrix, HeatMap and Scatterplots to observe the linear relationship between the variables. I observed strong correlations between the following variables, among others.

* BsmtFullBath & BsmtUnfSF
* GarageYrBlt & EnclosedPorch
* GarageYrBlt & OverallCond
* YearBuilt & EnclosedPorch
* LotFrontage & MSSubClass

**Imputation of Null values**

To deal with the null values in the data, first the training set was concatenated to the test set to create a master file. All imputation of null values was performed on this master file. The first step was generating a list of the variables containing null values. The data description file from Kaggle was used to guide the imputation process for each of the variables containing null values. For some variables, the null values were imputed with the default null value as stated in the data description file. Others were imputed with the maximum value, while others were imputed with the mean values or zeroes. Each variable was imputed separately with accompanying explanations provided in the python notebook. Each categorical variable was also converted to type string to ensure levels in the data values.

**Total Square Footage**

A new variable called TotalSF was created by summing TotalBsmtSF, 1stFlrSF and 2ndFlrSF. When buying a home, buyers are interested in knowing the Total Square Footage. By adding this variable to the dataset this will enhance the Regression model.

**From Integer to Categorical**

Through discovery, I found out that the following variables are of type integer when they should be treated as being categorical in the regression model. To fix this problem each of there variables were converted to type string.

* MSSubClass
* OverallQual
* OverallCond
* YrSold
* MoSold
* GarageYrBlt
* YearBuilt
* YearRemodAdd

**Box-Cox Transformations for Skewness**

All numerical variables with a skewness value above 0.75 were transformed using Box-Cox Transformations. This was also done as an outlier reduction strategy.

**Label Encoding**

All categorical variables were encoded with values between 0 and n\_classes-1. As the scikit learn website states, Label Encoding can be used to normalize labels and to transform non-numerical labels to numerical labels. Instead of having text or string labels these values are transformed to numerical labels.

**Dummy Variables**

Dummy variables were created for each level of each categorical variable. This is a great feature engineering techniques that enhanced the Regression Modelling process.

**Model Building**

After all the imputation, feature engineering and data preparation the master file was separated into a train and test set ensuring that the test set had 1459 rows. Many Regression models were built. The following is an output of each Regression model and its corresponding **Root Mean Squared Error (RSME).**

* Linear Regression RSME: 0.12434643448220742
* Ridge Regression RSME: 0.12402213279933252
* Lasso Regression RSME: 0.12361439787506197
* LassoCV Regression RSME: 0.18872479548952584
* ElasticNet Regression RSME: 0.12355496287204451
* BayesianRidge Regression RSME: 0.12485109397482888
* LassoLarsIC Regression RSME: 0.13120025041849642
* Random Forest Regressor RSME: 0.14449664091720715
* KNeighbors Regressor RSME: 0.23778660277251343
* DecisionTree Regressor RSME: 0.19613435009634456
* Support Vector Regressor RSME: 0.39496324569859115
* KernelRidge Regression RSME: 0.2869253478149764

**Model Revision**

Many techniques were utilized to refine the model. The following techniques were used.

* Regression models by themselves with no other techniques in the Pipeline.
  + Linear Regression
  + Ridge Regression
  + Lasso Regression
  + LassoCV Regression
  + ElasticNet Regression
  + BayesianRidge Regression
  + LassoLarsIC Regression
  + Random Forest Regressor
  + KNeighbors Regressor
  + DecisionTree Regressor
  + Support Vector Regressor
  + KernelRidge Regression
* Regression models with Polynomial Features of degree 2 in the Pipeline
  + Polynomial Ridge Regression
  + Polynomial ElasticNet Regression
* Bagging Regressor using Ridge Regression
* Regression models with SelectKBest in the Pipeline.
  + SelectKBest ElasticNet Regression

**Prediction Quality**

A couple submissions were made.

**Appendix**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**