**“A MACHINE LEARNING APPROCH TO PREDICT PRICES OF HOUSES.”**

**Module 03 – Final Report**

### (AIDI 1003)

Compiled by :

|  |  |
| --- | --- |
| Mohamad Althaf | 100757321 |
| Vinod Vijayaraghavan | 100772404 |
| Shivam Rana | 100766734 |
| TarunJot singh | 100766653 |

Date : 07.12.2019

Course Facilitator: Amit Maraj

**EXECUTIVE SUMMARY**

It is true that many factors have to be considered before making a purchase decision. Especially when it comes to buying a land or a building, the stress of decision-making is way too high due to economical, feasibility and aesthetic factors. In this project, we analyze the problem in real-estate deals and propose a machine learning solution that can help buyers and sellers to predict the prices of house prices.

**INTRODUCTION**

Since the beginning of civilization, we humans needed a shelter to survive. With the industrial revolution and breakthrough in new technologies, the needs of humans have increased as well. Let it be shopping, ordering a dish in a restaurant, or buying a car, we have to make a lot of difficult decisions in our daily lives. But, having a safe shelter was always been as the primary need. Furthermore, as stated in the executive summary, stress of decision-making is way too high when it comes to real estate deals.

When analysing two decades of median house price values from different regions of United States, it is found that the unit prices are hugely varies from cities to cities apart from other features of houses, and other economic factors. So, we found it is quite complicated to find a formula to come up with a reliable price in a way both the sellers and buyers are happy in finishing the deals.

Hence our group decided to create a machine learning model that can predict house prices using historical data of house prices from available public databases.

**RATIONAL STATEMENT**

Canada is a country with highest record of immigration and there is a huge demand to be addressed when it comes to buying lands and buildings. Again, the price of any such fixed properties is very different from province to province and cities to cities due to various reasons. The initial proposal of our project was to implement a predictive machine learning model that can suggest a very close approximate price of any such properties for the requirements of prospective dealers in Canada. This will hugely benefit, both the real-estate sector and the individual buyers in Canada to spend money wisely.

But unfortunately, we were not been able to find a public dataset to use it for whole Canada. With the permission of our course facilitator, we were allowed to modify the scope by selecting a slightly different dataset that include most features available in the original proposal.

In this project, more than 70 independent features were initially considered when it comes to finalizing the price of a land or a building. Some of the important factors includes:

* Province where the property is located.
* City where the property is located.
* Whether it has a garage or not.
* The total area of the property in square feet.
* Number of stairs in the house
* Number of bedrooms
* Number of bathrooms

With the help of our deployed application, the individuals with zero experience in real estate deals can easily finalize their housing plans without any stress. Sellers can also confidently value their lands and buildings easily.

**PROBLEM**

Buying a land or a building is one of the rare things we do in our lifetime. Some of us spend our whole savings on it. Hence it is a very crucial and difficult decision to make. Many people buy houses and live the rest of their lives in them, so the outcomes of real estate decisions have a long-term impact. The following are some of the critical problems associated with real-estate deals in Canada:

1. Buying a house is obviously very challenging, especially for first time buyers who have no or little experience in making real-estate deals. Without proper knowledge or right information in hand, they may end up with a bad deal.
2. Price of any real estate may vary from province to province and cities to cities and also countries to countries due to many factors such as household income and the development of the area. Since there is no system to validate the price of a property, there is a huge possibility, buyers spending too much money for properties that are not worthy.
3. Sellers do have a problem on setting a fair sales price for their property. There are possibilities for a seller setting an over price and wait for the deal to come forever, or a seller setting an under price and lose a lot of money.

**DATA REQUIREMENT**

We used the following features in our training data to predict the sales price of houses. Further we assumed we can collect these data of houses when deployed.

|  |  |
| --- | --- |
| Feature Names | Description |
| MSSubClass | Identifies the type of dwelling involved in the sale. |
| MSZoning | Identifies the general zoning classification of the sale. |
| LotFrontage | Linear feet of street connected to property |
| LotArea | Lot size in square feet |
| Street | Type of road access to property |
| Alley | Type of alley access to property |
| LotShape | General shape of property |
| LandContour | Flatness of the property |
| Utilities | Type of utilities available |
| LotConfig | Lot configuration |
| LandSlope | Slope of property |
| Neighborhood | Physical locations within Ames city limits |
| Condition1 | Proximity to various conditions |
| Condition2 | Proximity to various conditions (if more than one is present) |
| BldgType | Type of dwelling |
| HouseStyle | Style of dwelling |
| OverallQual | Rates the overall material and finish of the house |
| OverallCond | Rates the overall condition of the house |
| YearBuilt | Original construction date |
| YearRemodAdd | Remodel date (same as construction date if no remodeling or additions) |
| RoofStyle | Type of roof |
| RoofMatl | Roof material |
| Exterior1st | Exterior covering on house |
| Exterior2nd | Exterior covering on house (if more than one material) |
| MasVnrType | Masonry veneer type |
| MasVnrArea | Masonry veneer area in square feet |
| ExterQual | Evaluates the quality of the material on the exterior |
| ExterCond | Evaluates the present condition of the material on the exterior |
| Foundation | Type of foundation |
| BsmtQual | Evaluates the height of the basement |
| BsmtCond | Evaluates the general condition of the basement |
| BsmtExposure | Refers to walkout or garden level walls |
| BsmtFinType1 | Rating of basement finished area |
| BsmtFinSF1 | Type 1 finished square feet |
| BsmtFinType2 | Rating of basement finished area (if multiple types) |
| BsmtFinSF2 | Type 2 finished square feet |
| BsmtUnfSF | Unfinished square feet of basement area |
| TotalBsmtSF | Total square feet of basement area |
| Heating | Type of heating |
| HeatingQC | Heating quality and condition |
| CentralAir | Central air conditioning |
| Electrical | Electrical system |
| 1stFlrSF | First Floor square feet |
| 2ndFlrSF | Second floor square feet |
| LowQualFinSF | Low quality finished square feet (all floors) |
| GrLivArea | Above grade (ground) living area square feet |
| BsmtFullBath | Basement full bathrooms |
| BsmtHalfBath | Basement half bathrooms |
| FullBath | Full bathrooms above grade |
| HalfBath | Half baths above grade |
| Bedroom | Bedrooms above grade (does NOT include basement bedrooms) |
| Kitchen | Kitchens above grade |
| KitchenQual | Kitchen quality |
| TotRmsAbvGrd | Total rooms above grade (does not include bathrooms) |
| Functional | Home functionality (Assume typical unless deductions are warranted) |
| Fireplaces | Number of fireplaces |
| FireplaceQu | Fireplace quality |
| GarageType | Garage location |
| GarageYrBlt | Year garage was built |
| GarageFinish | Interior finish of the garage |
| GarageCars | Size of garage in car capacity |
| GarageArea | Size of garage in square feet |
| GarageQual | Garage quality |
| GarageCond | Garage condition |
| PavedDrive | Paved driveway |
| WoodDeckSF | Wood deck area in square feet |
| OpenPorchSF | Open porch area in square feet |
| EnclosedPorch | Enclosed porch area in square feet |
| 3SsnPorch | Three season porch area in square feet |
| ScreenPorch | Screen porch area in square feet |
| PoolArea | Pool area in square feet |
| PoolQC | Pool quality |
| Fence | Fence quality |
| MiscFeature | Miscellaneous feature not covered in other categories |
| MiscVal | $Value of miscellaneous feature |
| MoSold | Month Sold (MM) |
| YrSold | Year Sold (YYYY) |
| SaleType | Type of sale |
| SaleCondition | Condition of sale |

**Requirements to meet the solution**

* The selected features (independent variables) should have enough correlation with the price (dependent variable) of the property.
* The selected features should be correctly encoded to an appropriate form where it can be used to create a model. For example, province, cities and types of the property should be encoded by one hot encoder.
* Missing data should be correctly imputed with appropriate mechanics.

**Requirements, the solution does not need to meet**

* Dataset does not need to have all the features that affect the price.
* Dataset does not need to be scaled.
* Model doesn’t have a restriction on selecting the number of features.

**Constraints & Limitations**

* Difficulties in creating dataset by merging from number of data sources.
* Concerns regarding the reliability of the data sources.
* Concerns in reliability of model’s life time.

**DATA**

We used multiple datasets to create our model, but the one found in Kaggle with the name ‘Ames Housing Dataset’ gave us promising results.

1. **Ames Housing Dataset**:

This dataset describes the sale of individual residential property in Ames, Iowa from 2006 to 2010. The data set contains 2930 observations and a large number of explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous) involved in assessing house prices.

**Source:** <https://www.kaggle.com/c/house-prices-advanced-regression-techniques>

1. **House sale prices for King County, US:**

This dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015.

**Source**: <https://www.kaggle.com/harlfoxem/housesalesprediction>

1. **Boston House Prices:**

A Dataset derived from information collected by the U.S. Census Service concerning housing in the area of Boston Mass. This dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass.

**Source:** <https://www.kaggle.com/vikrishnan/boston-house-prices>

1. Sydney and Melbourne House prices:

This dataset contains house sale prices in Melbourne and Sydney. It includes homes sold between May 2009 and May 2011.

**Source:** <https://www.kaggle.com/anthonypino/melbourne-housing-market>

**MODEL BUILDING APPROCH**

There is an obvious demand in finding real estate prices for prospective buyers and sellers due to the huge immigration rate in Canada. Budget to purchase, the feasibility of accessing outdoor services (supermarket, public transit, hospital, etc.), aesthetic preference (with or without garden) and many more factors are depended on the purchase decision of a buyer. The crucial decision for seller is setting the right price. By implementing our proposed solution, an easy and informed purchase decisions can be made for buyers as well as sellers and can set a fair price without any hassles.

Data Collection

Data Pre-processing

Creating a regression model

Test the model

Data Collection

Validation Failed

Validation Success

We reached the good predictive model using the following steps:

**Step 01 – Select the right machine learning algorithm or combination of algorithms:**

It is vital to select the appropriate machine learning algorithm to create the right predictive model. Since this system is associated with several independent variables to predict the price of the real estate, there can be a possible situation to use combination of machine learning algorithms.

We chose five algorithms that could produce a better result,

1. Multiple Linear Regression
2. Ridge Regression
3. Lasso Regression
4. Support Vector Regression
5. Random Forest Regression

**Step 02 – Data Collection:**

Without the data, machine learning is become a void algorithm. Therefore, creating a data set is equally important to selecting the right algorithm. This was the hardest part of in our initial phase.

As stated in the Data section, we initially selected 4 datasets which is described and finalized with Ames Housing Data at the end.

**Step 03 – Data Preprocessing:**

Initially the collected raw data is mostly noisy and incomplete. Hence it is very important to preprocess the collected data before passing to the algorithm. Preprocessing is simply a cleaning step that includes managing the missing data, data wrangling, data encoding and normalization. Finally splitting the dataset into training dataset and testing dataset.

**For Missing Values:**

We deleted some of the complete data point which are missing and make no sense. But many of them are imputed with mean of that feature.

**For Categorical Data:**

Since our dataset include categorical data, they were encoded by label encoder to make a numerical relationship with dependent variables.

**Feature Scaling:**

Since the dataset we have are in different units representing different parameters, they need to be scaled to fit the data accurately. So, we did scale the data normalization.

**Data Splitting:**

To validate the regression model, we split the data in to test set and training set. As per the background readings, training: test data split is recommended in 8:2 ratio for the most efficient model.

**Step 04 – Exploratory data analysis:**

In this step, we analysed the training data for its co-relation, skewness of its features and feature outliers. Using correlation metrics, high correlating features between independent variables were analyzed and one of such variable was removed above a threshold to address multi co-linearity issue.

Skewness of the feature distribution was the adjusted by log transform. Further, outliers were removed, the numerical one by referring to pair plots and categorical values by box plot.

**Step 05 – Create a model with training data set:**

At this step, we had clean and scaled data, and a clearly selected algorithm ready to build the model. Each of our individual members selected an algorithm and evaluated its performances.

**Step 06 – Test the model with the testing data set:**

We then tested with remaining 20% of data we derived from the split and evaluated the model performance.

**Technical Requirements:**

01. Python 3.0 (or later) Environment: We use Anaconda Python with Jupiter Notebook Installed.

02. Libraries:

- NumPy: A library to perform mathematical computations  
 - Pandas: A Library to load and preprocess the dataset  
 - sklearn: Machine Learning library with different model fitting functionalities.  
 - pickle: This library allows the system to save the model for deployment and future use.  
 - Seaborn: A visualization library  
 - Matplotlib: A visualization library

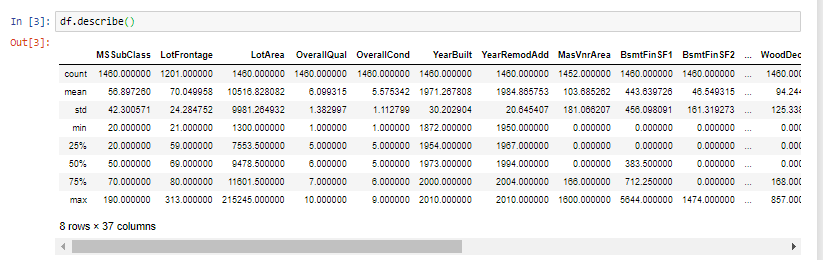
03. System Requirement: PC with following minimum requirements  
 - 4GB RAM  
 - Core i7 Processor with at least 4 cores  
 - 100 GB Hard Disk  
 - Windows OS Installed

**PROJECT WORK BREAKDOWN**

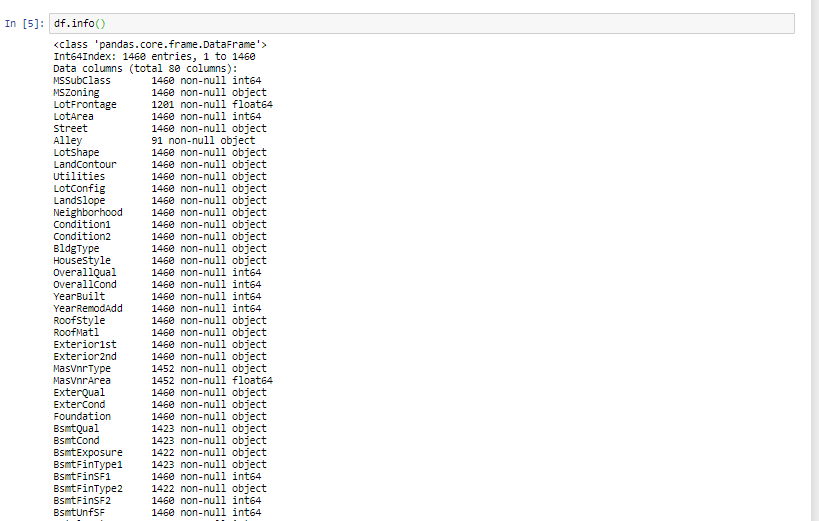
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tasks | Althaf | Shivam | Vinod | Tarun |
| Background Study |  |  |  |  |
| Collect information |  |  |  |  |
| Analyze Dataset |  |  |  |  |
| Clean Dataset |  |  |  |  |
| Develop Data Preprocessing Pipeline |  |  |  |  |
| Fit Linear Regression Model |  |  |  |  |
| Fit Ridge Regression Model |  |  |  |  |
| Fit Lasso Regression Model |  |  |  |  |
| Fit SVR Model |  |  |  |  |
| Fit Random Forest Model |  |  |  |  |
| Documentation |  |  |  |  |

**EXPLORATORY DATA ANALYSIS (EDA)**

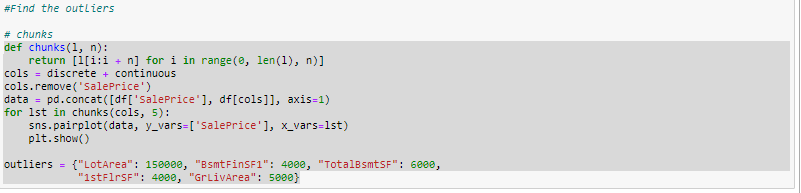
1. Describe function – It Give the statistical information.

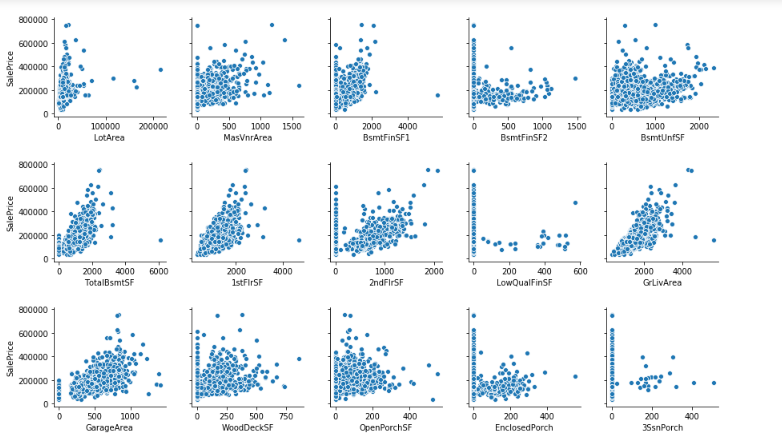


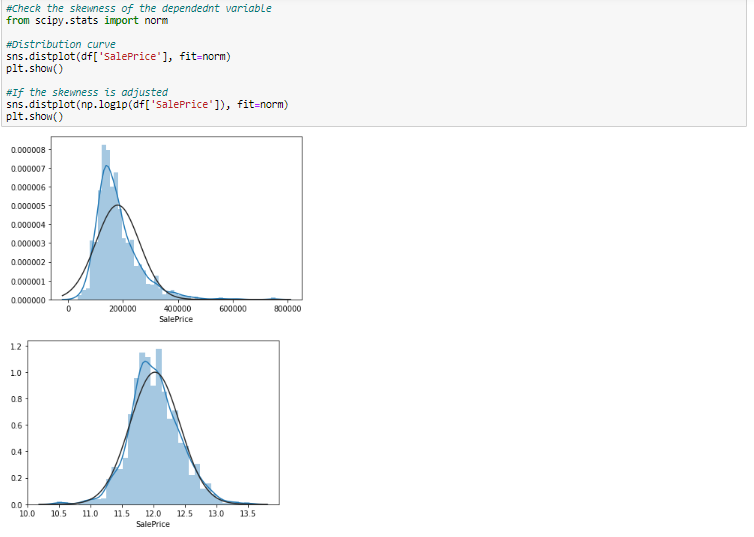
1. Info function – It give the general information regarding the type of data.



1. PairPlots – We Used PairPlots to identifiy the Outliers in our dataset.

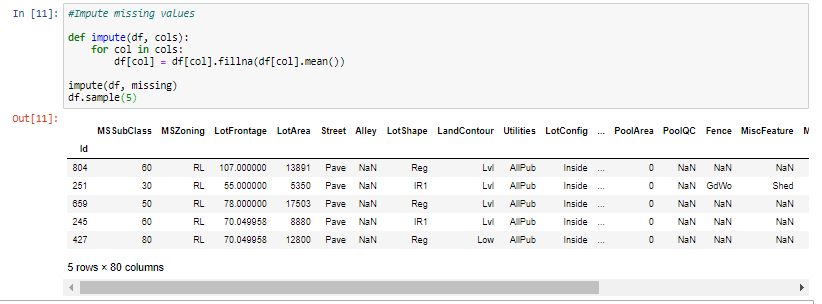




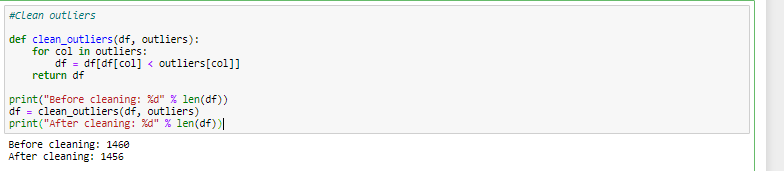
1. Distribution Plot –Distribution plot to check the skewness in data. 
2. Heat Map – To find the Correlation between features and target.

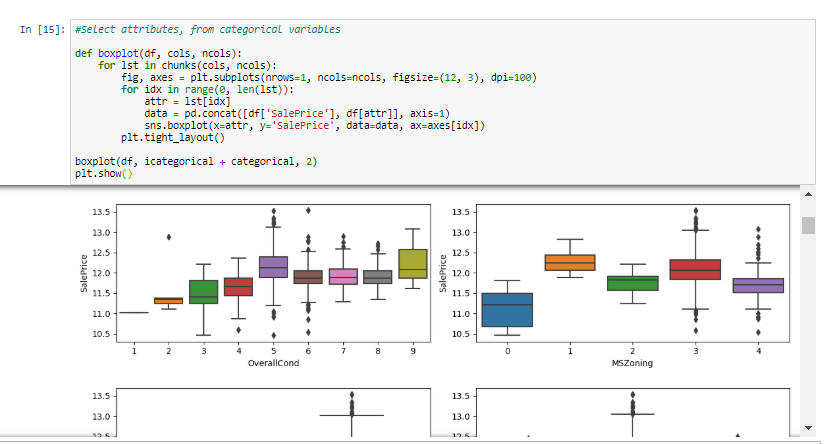
**PRE-PROCESSING PIPELINE.**

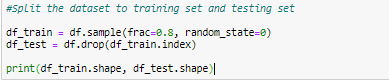
1. Mean – Imputed the Missing value with mean of the corresponding feature.



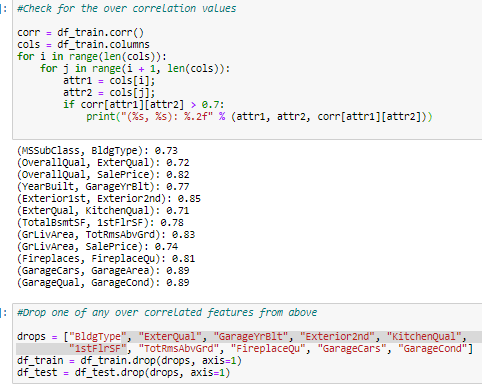
1. Clean Outliers – Remove the features that have outliers as found using pairplot in EDA section.



1. Box Plot – To identify Outliers in categorical variables. 
2. Splitting Data – Separating training and testing datasets.



1. Over Correlated features – Removed all the over Correlated Attributes.



**ALGORITHM EVALUATION**

We have tested 5 different algorithms on our dataset and all got good r2 score. Below are the algorithms that we have tried.

* Linear regression
* Ridge regression
* Lasso Regression
* SVR (support vector regression)
* Random forest regression

#### Top performing algorithms:

Among the above algorithms Linear and Ridge regression performed the best, and SVR performed the worst. Below are the performance metrics of above algorithms.

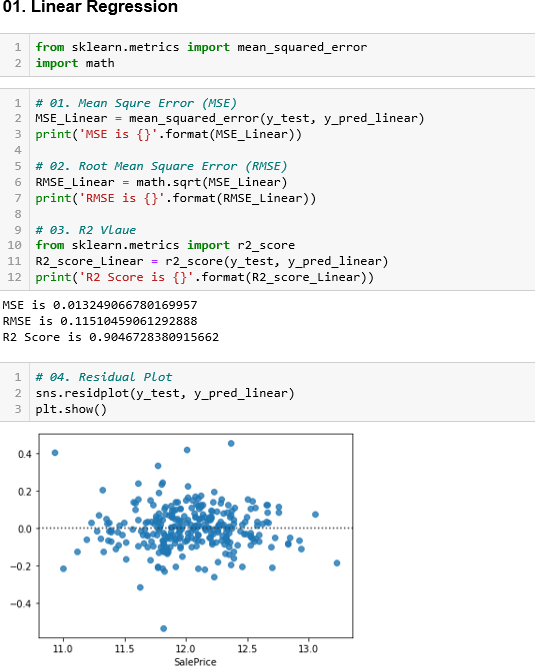
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Linear** | **Ridge** | **Lasso** | **SVR** | **Random forest** |
| **R2 Score** | 0. 9046728380 | 0. 9046952005 | 0. 7104762926 | 0. 6671602919 | 0. 8800400090 |
| **MSE** | 0. 01246678 | 0.01324595873 | 0. 0402395168 | 0. 0462598007 | 0. 0166726659 |
| **RMSE** | 0. 1141049061 | 0. 1150910888 | 0. 2005978983 | 0. 21508091683 | 0. 1291226780 |

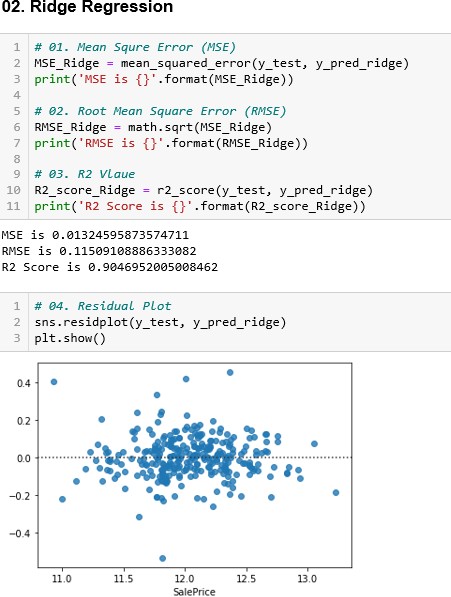
1. Linear Regression
   * + - * Pros
2. Linear Regression performs well when the dataset is linearly separable. We can use it to find the nature of the relationship among the variables
3. Linear Regression is easier to implement, interpret and very efficient to train.
4. Linear Regression is prone to over-fitting but it can be easily avoided using some dimensionality reduction techniques, regularization (L1 and L2) techniques and cross-validation.

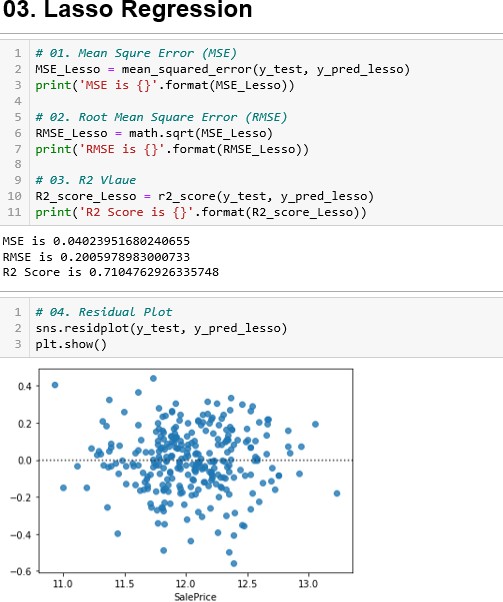
* + - * + Cons

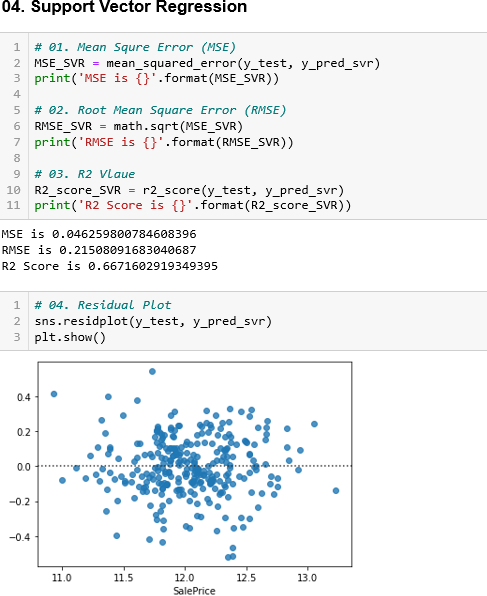
1. Main limitation of Linear Regression is the assumption of linearity between the dependent variable and the independent variables. In the real world, the data is rarely linearly separable. It assumes that there is a straight-line relationship between the dependent and independent variables which is incorrect many times.
2. Prone to noise and overfitting: If the number of observations are lesser than the number of features, Linear Regression should not be used, otherwise it may lead to overfit because is starts considering noise in this scenario while building the model.
3. Prone to outliers: Linear regression is very sensitive to outliers (anomalies). So, outliers should be analyzed and removed before applying Linear Regression to the dataset.
4. Ridge Regression
   * + - * Pros
5. Prevents Overfitting: A high-dimensional dataset having too many features can sometimes lead to overfitting.
6. Simplicity: An over-complex model having too many features can be hard to interpret especially when features are correlated with each other.
7. Computational Efficiency: A model trained on a lower dimensional dataset is computationally efficient (execution of algorithm requires less computational time).
   * + - * Cons
8. Regularization leads to dimensionality reduction, which means the machine learning model is built using a lower dimensional dataset. This generally leads to a high bias error.
9. If regularization is performed before training the model, a perfect balance between bias-variance trade-off must be used.
10. Lasso Regression
    * + - * Pros
11. As any regularization method, it can avoid overfitting. It can be applied even when number of features is larger than number of data.
12. It can do feature selection.
13. It is fast in terms of inference and fitting.
    * + - * Cons
14. The model selected by lasso is not stable. For example, on different bootstrapped data, the feature selected can be very different.
15. When there are highly correlated features, lasso may randomly select one of them of part of them. The result depends on the implementation. To improve, people introduced elastic net.
16. SVR (support vector regression)
    * + - * Pros
17. They’re accurate in high dimensional spaces.
18. they use a subset of training points in the decision function (called support vectors), so it’s also memory efficient.
19. Can work on Non-Linear data.
    * + - * Cons
20. The algorithm is prone for over-fitting, if the number of features is much greater than the number of samples.
21. Also, SVMs do not directly provide probability estimates, which are desirable in most classification problems.
22. SVMs are not very efficient computationally, if your dataset is very big, such as when you have more than one thousand rows.
23. Random forest regression
    * + - * Pros
24. It has an effective method for estimating missing data and maintains accuracy when large proportion of the data are missing.
25. It has methods for balancing errors in data sets where classes are imbalanced.
    * + - * Cons
26. It surely does a good job at classification but not as for regression problem as it does not give precise continuous nature prediction. In case of regression, it doesn’t predict beyond the range in the training data, and that they may over fit data sets that are particularly noisy.
27. Random forest can feel like a black box approach for a statistical model we have very little control on what the model does. You can at best try different parameters and random seeds.

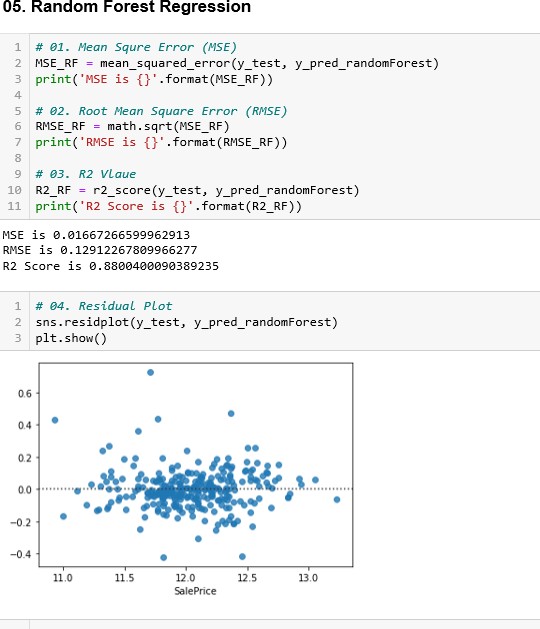
**Screenshots of the Code**











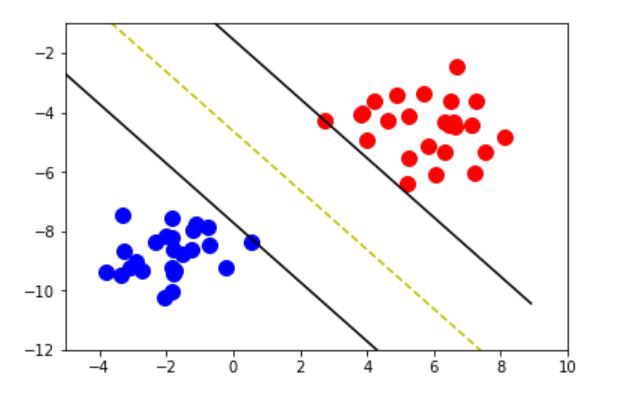
**CANDIDATE ALGORITHM SELECTION AND RATIONALE.**

Here are the algorithms we chose and the logic behind why we chose it

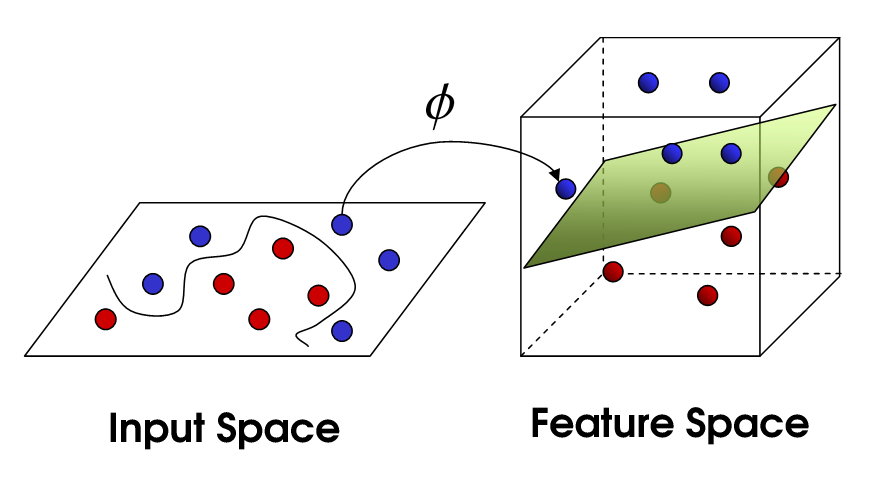
**Tarun**:

**SVM**

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side



SVM is very good at handling non-linear data. A function called Kernel function is used which represents the data into higher Dimension, so that algorithm can fit a straight line as shown in figure.



**Shiva:**

**Linear Regression**

Linear regression is a linear analysis technique for the relationship between the dependent variable and one or more independent variables.

Y = a\_0 + a\_1 \* x

The linear regression algorithm’s motive is to find the best a\_0 and a\_1 value. Regression techniques differ mostly based on the number of independent variables and the type of relationship between the independent and dependent variables.

The capacity to assess the relative effect of the criteria function of one or more predictor variables. Its ability to identify outliers, or anomalies. Linear regression is perfect when the relationship between the covariate and the response parameter is linear. This is useful because the emphasis moves from statistical modelling to data analysis and pre-processing. It’s great to learn to play with data without having to worry about the model’s intricate details.

After implemented this algorithm, we got 0.78 R2 score.

**Althaf**:

**Ridge Regression**

Since the dataset we use consists of 52 independent features, which could be easily fallen to over fitting problem due to its higher number of dimensions. Ridge regression, which contain the regularization factor will prevent this problem, and that is why it is been selected as a preferred model. Further it is simple to implement and computationally less complex when creating the model.

As result stated, the created model was very effective as RMSE, MSE are very low. Further R2 score was in 0.90 which shows the model is very closer to the actual results.

**Vinod**:

**Lasso Regression**

There are many advantages in using LASSO method, first of all it can provide a very good prediction accuracy, because shrinking and removing the coefficients can reduce variance without a substantial increase of the bias, this is especially useful when you have a small number of observation and a large number of features. Hence the reason why I chose Lasso over other techniques.

In terms of the tuning parameter λ we know that bias increases and variance decreases when λ increases, indeed a trade-off between bias and variance has to be found. Moreover, the LASSO helps to increase the model interpretability by eliminating irrelevant variables that are not associated with the response variable, this way also overfitting is reduced. This is the point where we are more interested in because in this paper the focus is on the feature selection task.

The LASSO method puts a constraint on the sum of the absolute values of the model parameters, the sum has to be less than a fixed value (upper bound). In order to do so the method apply a shrinking (regularization) process where it penalizes the coefficients of the regression variables shrinking some of them to zero. During features selection process the variables that still have a non-zero coefficient after the shrinking process are selected to be part of the model. The goal of this process is to minimize the prediction error.

The r2 score for Lasso regression in our model is 0.71. Which was not as good as other techniques like Linear regression and ridge regression.

**Team**:

**Random forest**   
The random forest model is a type of additive model that combines decisions from a sequence of base models to make predictions.  
 g(x)=f 0(x)+f1(x)+f2(x)+…  
where the final model g(x) is the sum of simple base models functions ‘ f ’. Here, each base classifier is a simple decision tree. The Random Forest is one of the most powerful models of machine learning for predictive analysis.

As we have categorised and numerical data in our dataset. The random forest method is very good at handling tabular information with numerical features, or categorical features. Unlike linear models, nonlinear interaction between the features and the target can be captured by random forests. An important note is that tree-based models are not intended to work with very sparse characteristics. When dealing with sparse input data, either we can pre-process sparse features to produce numerical statistics, or we can turn to a linear model that is better suited to such scenarios. We have chosen the n\_estimator hyperparameters which is actually refer to the number of trees in the forest.

We implemented this algorithm and we got 0.88 R2 score which is actually better than linear regression and SVM algorithms.

**FINAL INFERENCE**

First, we use Label encoder Then we apply (lambda x: x.cat.codes) to transform the categorical feature into the numerical feature.

|  |  |  |
| --- | --- | --- |
| Algorithm | Prior R2\_score accuracy | New R2\_score Accuracy |
| Linear regression | 0.89 | 0.90 |
| Ridge regression | 0.82 | 0.88 |
| Lasso Regression | 0.76 | 0.72 |
| Support Vector Regression | 0.64 | 0.62 |
| Random Forest Regression | 0.83 | 0.85 |

After dropping some columns like “BldgType”, "ExterQual", "GarageYrBlt", "Exterior2nd". It showed minor impact on the accuracy.

|  |  |  |
| --- | --- | --- |
| Algorithm | Prior R2\_score accuracy | New R2\_score Accuracy |
| Linear regression | 0.90 | 0.90 |
| Ridge regression | 0.88 | 0.90 |
| Lasso Regression | 0.72 | 0.71 |
| Support Vector Regression | 0.62 | 0.66 |
| Random Forest Regression | 0.85 | 0.88 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Linear** | **Ridge** | **Lasso** | **SVR** | **Random forest** |
| **R2 Score** | 0. 9046728380 | 0. 9046952005 | 0. 7104762926 | 0. 6671602919 | 0. 8800400090 |
| **MSE** | 0. 013246678 | 0.01324595873 | 0. 0402395168 | 0. 0462598007 | 0. 0166726659 |
| **RMSE** | 0. 1141049061 | 0. 1150910888 | 0. 2005978983 | 0. 21508091683 | 0. 1291226780 |

**APPENDICES**

We are happy with the results we got. We know some algorithms performed better than others. We understood that certain algorithms will perform better in certain scenarios. We want to dig deep into machine learning algorithms and learn more.

We as a team worked very well in this project. Everyone contributed their best. Capstone term helped us in many ways more than a mere technical project. It helped us understand each other better, helped us with our communication and team management skills. Moreover, it made us good friends.

We are thankful to Amit who is a great mentor. Amit helped us when we needed support and he always had a lenient attitude towards us. We as a team greatly appreciate your support. We thank you for making us learn some brand new out of the box machine learning stuffs.

We look forward to similar new experiences. We as a team thank every one in the Ai batch. We wish you all the best.

**Thank you!**

****