Housing Price Prediction

Submitted by:

Khushboo Pandey

ACKNOWLEDGMENT

I would like to thank everyone who helped me during the making of the projects.

The help was provided by:

1. Data Trained faculty
2. Dr. Deepika Sharma

References include

1. Scikit-learn.org
2. Kaggle.com
3. Github.com
4. Stack-Overflow
5. Learning.datatrained.com

INTRODUCTION

- Business Problem Framing

Problem Statement:
Houses are one of the necessary need of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company.
A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file below.
The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. For this company wants to know:
• Which variables are important to predict the price of variable?
• How do these variables describe the price of the house?

Business Goal:
You are required to model the price of houses with the available independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can

accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market.
Technical Requirements:

- Data contains 1460 entries each having 81 variables.
- Data contains Null values. You need to treat them using the domain knowledge and your own understanding.
- Extensive EDA has to be performed to gain relationships of important variable and price.
- Data contains numerical as well as categorical variable. You need to handle them accordingly.
- You have to build Machine Learning models, apply regularization and determine the optimal values of Hyper Parameters.
- You need to find important features which affect the price positively or negatively.
- Two datasets are being provided to you (test.csv, train.csv). You will train on train.csv dataset and predict on test.csv file.

• Review of Literature

In this section, we look at five recent studies that are related to our topic and see how models were built and what results were achieved in these studies.

Stock Market Prediction Using Bayesian-Regularized Neural Networks

In a study done by Ticknor (2013), he used Bayesian regularized artificial neural network to predict the future operation of financial market. Specifically, he built a model to predict future stock prices. The input of the model is previous stock statistics in addition to some financial technical data. The output of the model is the next-day closing price of the corresponding stocks.

The model proposed in the study is built using Bayesian regularized neural network. The weights of this type of networks are given a probabilistic nature. This allows the network to penalize very complex models (with many hidden layers) in an automatic manner. This in turn will reduce the overfitting of the model.

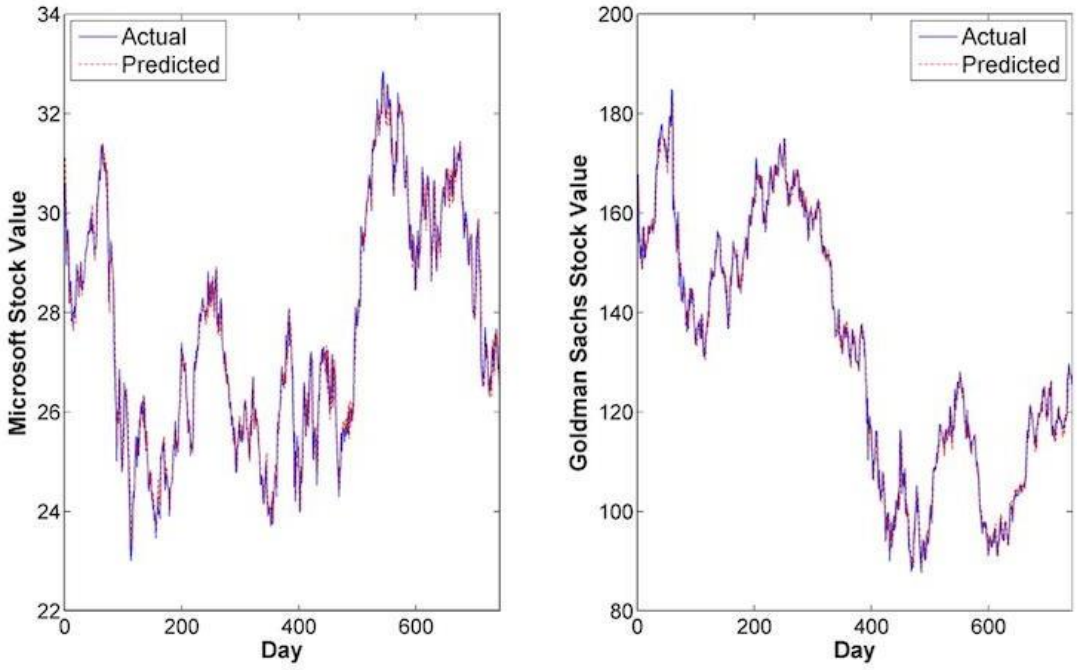
The model consists of a feedforward neural network which has three layers: an input layer, one hidden layer, and an output layer. The author chose the number of neurons in the hidden layer based on experimental methods. The input data of the model is normalized to be between -1 and 1, and this operation is reversed for the output so the predictedprice appears in the appropriate scale.

The data that was used in this study was obtained from Goldman Sachs Group (GS), Inc. and Microsoft Corp. (MSFT) . The data covers 734 trading days (4 January 2010 to 31 December 2012). Each instance of the data consisted of daily statistics: low price, high price, opening price, close price, and trading volume. To facilitate the training and testing of the model, this data was split into training data and test data with 80% and 20% of the original data, respectively. In addition to the daily-statistics variables in the data, six more variables were created to reflect financial indicators.

The performance of the model were evaluated using mean absolute percentage error (MAPE) performance metric. MAPE was calculated using this formula:

$$MAPE=\sum_{i=1}^{r}(abs(y_i-p_i)/y_i)r\times 100$$

where p_i is the predicted stock price on day i, y_i is the actual stock price on day i, and r is the number of trading days.
When applied on the test data, The model achieved a MAPE score of 1.0561 for MSFT part, and 1.3291 for GS part. Figure shows the actual values and predicted values for both GS and MSFT data.



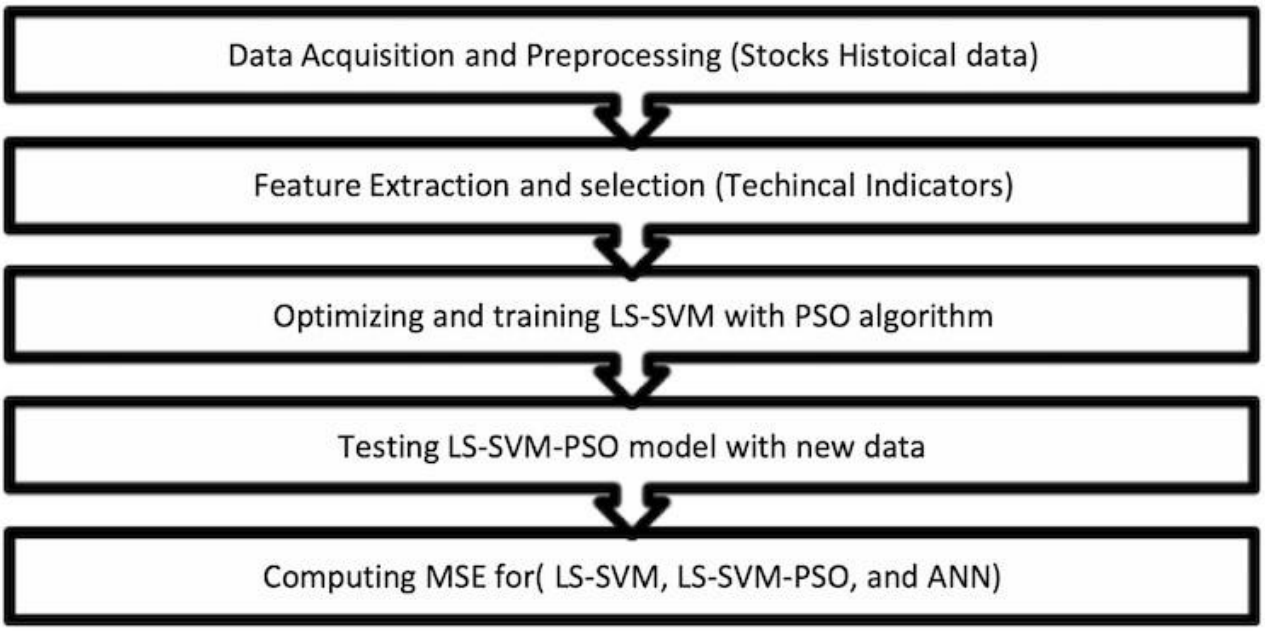
Stock Market Prediction Using A Machine Learning Model

In another study done by Hegazy, Soliman, and Salam (2014), a system was proposed to predict daily stock market prices. The system combines particle swarm optimization (PSO) and least square support vector machine (LS-SVM), where PSO was used to optimize LV-SVM.

The authors claim that in most cases, artificial neural networks (ANNs) are subject to the overfitting problem. They state that support vector machines algorithm (SVM) was developed as an alternative that doesn't suffer from overfitting. They attribute this advantage to SVMs being based on the solid foundations of VC-theory. They further elaborate that LS-SVM method was reformulation of traditional SVM method that uses a regularized least squares function with equality constraints to obtain a linear system that satisfies Karush-Kuhn-Tucker conditions for getting an optimal solution.

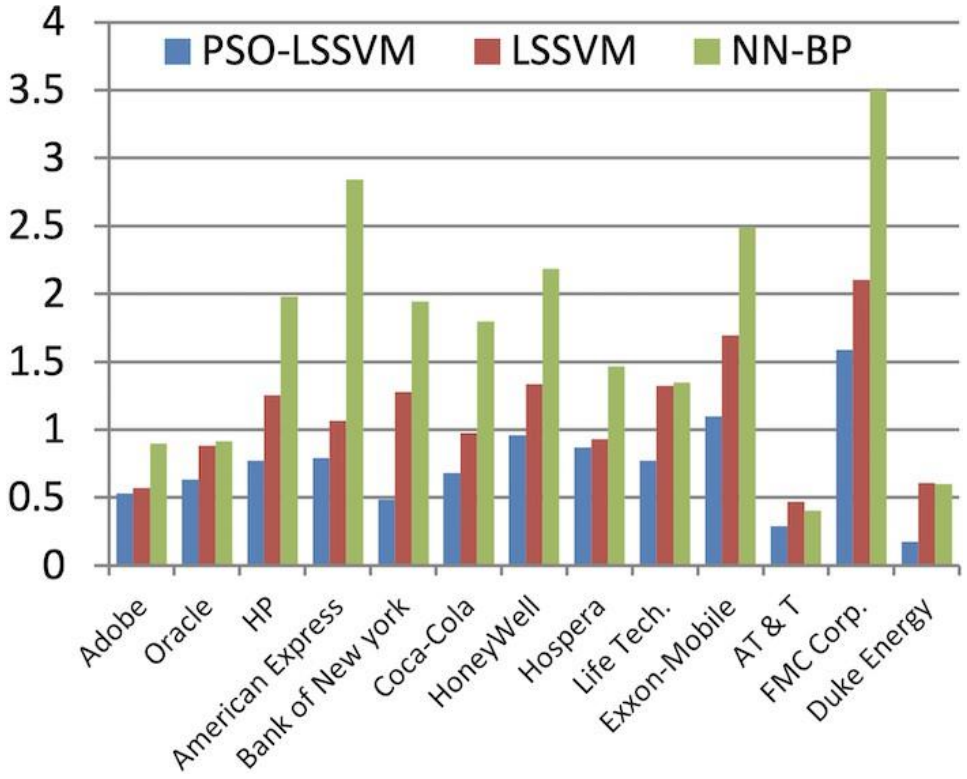
The authors describe PSO as a popular evolutionary optimization method that was inspired by organism social behavior like bird flocking. They used it to find the optimal parameters for LS-SVM. These parameters are the cost penalty CC, kernel parameter γγ, and insensitive loss function єє. The model proposed in the study was based on the analysis of historical data and technical financial indicators and using LS-SVM optimized by PSO to predict future daily stock prices. The model input was six vectors

representing the historical data and the technical financial indicators. The model output was the future price. The model used is represented in Figure .



Regarding the technical financial indicators, five were derived from the raw data: relative strength index (RSI), money flow index (MFI), exponential moving average (EMA), stochastic oscillator (SO), and moving average convergence/divergence (MACD). These indicators are known in the domain of stock market.

The model was trained and tested using datasets taken from https://finance.yahoo.com/. The datasets were from Jan 2009 to Jan 2012 and include stock data for many companies like Adobe and HP. All datasets were partitioned into a training set with 70% of the data and a test set with 30% of the data. Three models were trained and tested: LS-SVM-PSO model, LS-SVM model, and ANN model. The results obtained in the study showed that LS-SVM-PSO model had the best performance. Figure shows a comparison between the mean square error (MSE) of the three models for the stocks of many companies.



House Price Prediction Using Multilevel Model and Neural Networks

A different study was done by Feng and Jones (2015) to preduct house prices. Two models were built: a multilevel model (MLM) and an artificial neural network model (ANN). These two models were compared to each other and to a hedonic price model (HPM).

The multilevel model integrates the micro-level that specifies the relationships between houses within a given neighbourhood, and the macro-level equation which specifies the relationships between neighbouhoods. The hedonic price model is a model that estimates house prices using some attributes such as the number of bedrooms in the house, the size of the house, etc.

The data used in the study contains house prices in Greater Bristol area between 2001 and 2013. Secondary data was obtained from the Land Registry, the Population Census and Neighbourhood Statistics to be used in order to make the models suitable for national usage. The authors listed many reasons on why they chose the Greater Bristol area such as its diverse urban and rural blend and its different property types. Each record in the dataset contains data about a house in the area: it contains

the address, the unit postcode, property type, the duration (freehold or leasehold), the sale price, the date of the sale, and whether the house was newly-built when it was sold. In total, the dataset contains around 65,000 entries. To enable model training and testing, the dataset was divided into a training set that contains data about house sales from 2001 to 2012, and a test set that contains data about house sales in 2013.

The three models (MLM, ANN, and HPM) were tested using three senarios. In the first senario, locational and measured neighbourhood attributes were not included in the data. In the second senario, grid references of house location were included in the data. In the third senario, measured neighbourhood attributes were included in the data. The models were compared in goodness of fit where R2R2 was the metric, predictive accuracy where mean absolute error (MAE) and mean absolute percentage error (MAPE) were the metrics, and explanatory power. HPM and MLM models were fitted using MLwiN software, and ANN were fitted using IBM SPSS software. Figure shows the performance of each model regarding fit goodness and predictive accuracy. It shows that MLM model has better performance in general than other models.

COMPARISONS OF GOODNESS-OF-FIT

	R^2 (training set)	R^2 (test set)
HPM1	0.39	0.23
MLM1	0.75	0.75
ANN1	0.39	0.23
HPM2	0.43	0.3
MLM2	0.75	0.75
ANN2	0.41	0.26
HPM3	0.68	0.65
MLM3	0.75	0.74
ANN3	0.69	0.67

COMPARISON OF PREDICTIVE ACCURACY

Test set	MAE (lnP)	MAPE (lnP)	MAE (raw price)	MAPE (raw price)
HPM1	0.319	5.89%	80.4	30.9%
MLM1	0.178	3.29%	48.6	17.5%
ANN1	0.318	5.85%	80.1	30.0%
HPM2	0.304	5.61%	77.0	29.4%
MLM2	0.178	3.29%	48.6	17.5%
ANN2	0.313	5.76%	79.0	29.8%
HPM3	0.210	3.89%	25.3	20.7%
MLM3	0.178	3.30%	48.8	17.6%
ANN3	0.216	4.00%	55.7	20.9%

Composition of Models and Feature Engineering to Win Algorithmic Trading Challenge

A study done by de Abril and Sugiyama (2013) introduced the techniques and ideas used to win Algorithmic Trading Challenge, a competition held on Kaggle. The goal of the competition was to develop a model that can predict the short-term response of order-driven markets after a big liquidity shock. A liquidity shock happens when a trade or a sequence of trades causes an acute shortage of liquidity (cash for example).

The challenge data contains a training dataset and a test dataset. The training dataset has around 754,000 records of trade and quote observations for many securities of London Stock Exchange before and after a liquidity shock. A trade event happens when shares are sold or bought, whereas a quote event happens when the ask price or the best bid changes.

A separate model was built for bid and another for ask. Each one of these models consists of K random-forest sub-models. The models predict the price at a particular future time.

The authors spent much effort on feature engineering. They created more than 150 features. These features belong to four categories: price features, liquidity-book features, spread features (bid/ask spread), and rate features (arrival rate of orders/quotes). They applied a feature selection algorithm to obtain the optimal feature set (FbFb) for bid sub-models and the optimal feature set (FaFa) of all ask sub-models. The algorithm applied eliminates features in a backward manner in order to get a feature set with reasonable computing time and resources.
Three instances of the final model proposed in the study were trained on three datasets; each one of them consists of 50,000 samples sampled randomly from the training dataset. Then, the three models were applied to the test dataset. The predictions of the three models were then averaged to obtain the final prediction. The proposed method achieved a RMSE score of 0.77 approximately.

Using K-Nearest Neighbours for Stock Price Prediction

Alkhatib, Najadat, Hmeidi, and Shatnawi (2013) have done a study where they used the k-nearest neighbours (KNN) algorithm to predict stock prices. In this study, they expressed the stock prediction problem as a similarity-based classification, and they represented the historical stock data as well as test data by vectors.

The authors listed the steps of predicting the closing price of stock market using KNN as follows:

- The number of neaerest neighbours is chosen
- The distance between the new record and the training data is computed

- Training data is sorted according to the calculated distance
- Majority voting is applied to the classes of the k nearest neighbours to determine the predicted value of the new record

The data used in the study is stock data of five companies listed on the Jordanian stock exchange. The data range is from 4 June 2009 to 24 December 2009. Each of the five companies has around 200 records in the data. Each record has three variables: closing price, low price, and high price. The author stated that the closing price is the most important feature in determining the prediction value of a stock using KNN.

After applying KNN algorithm, the authors summarized the prediction performance evaluation using different metrics in a the table shown.

	kNN algorithm for K = 5		
Company	**Total squared RMS error**	**RMS error**	**Average error**
AIEI	0.263151	0.0378176	-5.43E-09
AFIN	0.2629177	0.0363482	-1.01E-08
APOT	22.74533	0.3372338	2.50E-08
IREL	1.2823397	0.1046908	4.27E-08
JOST	0.17963	0.0300444	1.508E-08

The authors used lift charts also to evaluate the performance of their model. Lift chart shows the improvement obtained by using the model compared to random estimation. As an example, the lift graph for AIEI company is shown in Figure. The area between the two lines in the graph is an indicator of the goodness of the model.

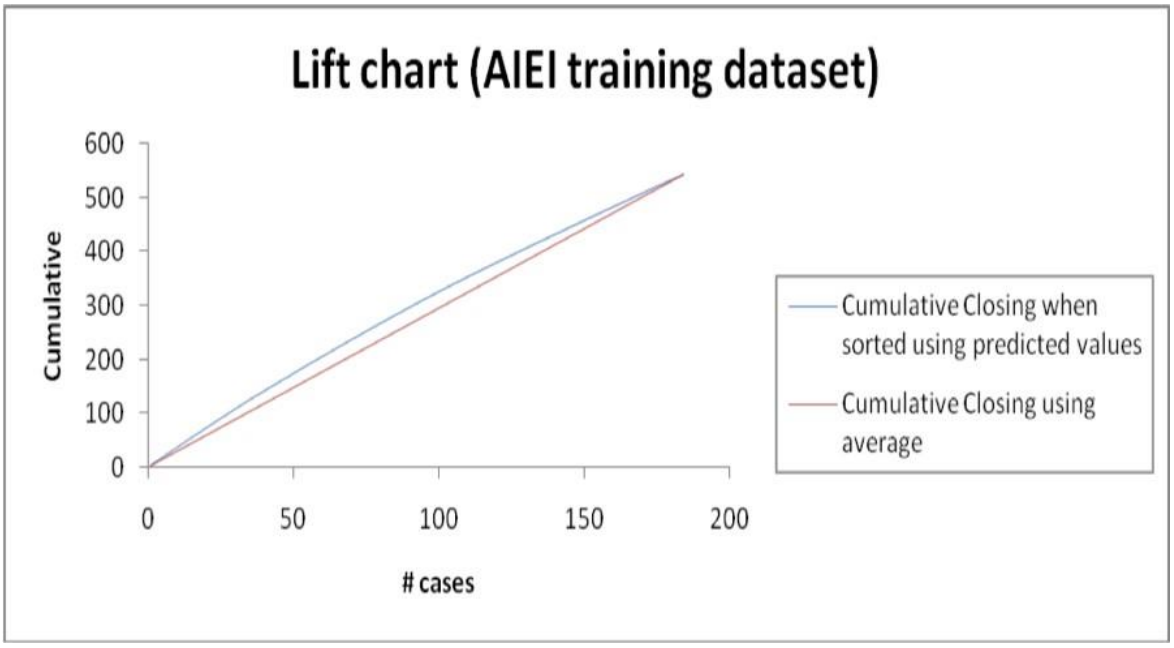
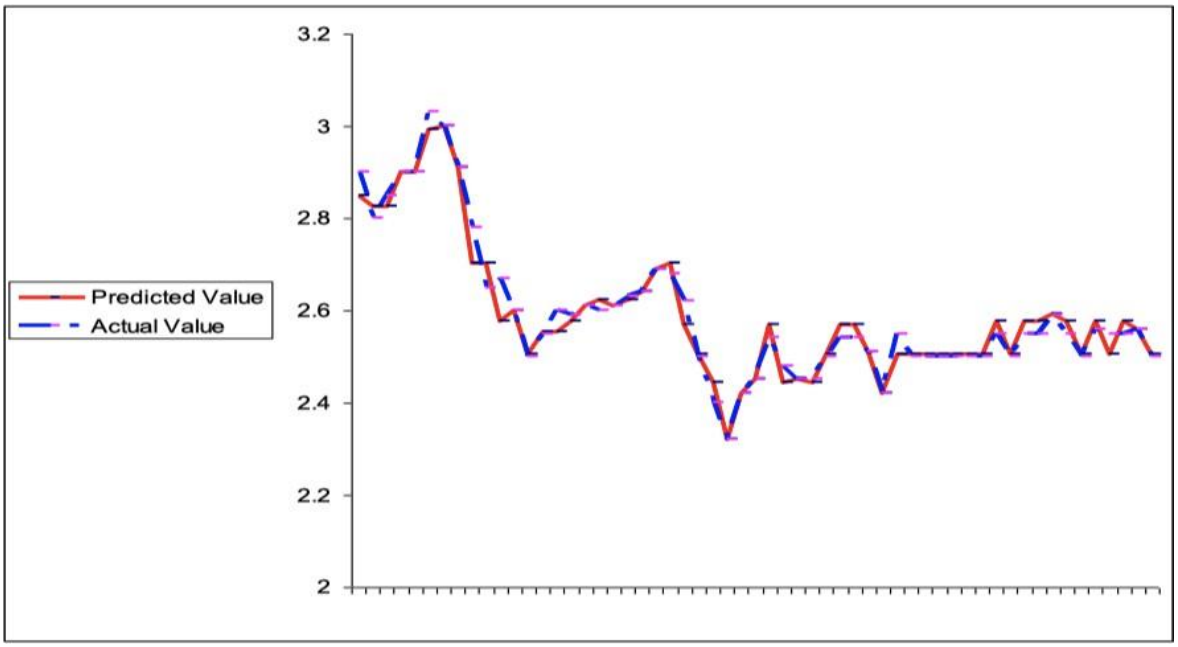


Figure shows the relationship between the actual price and predicted price for one year for the same company.



- Motivation for the Problem Undertaken

  It is very difficult to predict house prices as it keeps on changing with the upcoming scenarios.

For example-Covid 19 pandemic reduced the sales hence the price of a lot of stuff decreased. It almost seemed impossible for an outsider but as an aspiring data scientist I would love to take it up and predict the prices of the houses.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem
  1. Quality of Basement Kitchen affects the price of the houses.
  2. Maximum data contains MS subclass as 60.
  3. The train data contains 1168 rows and 81 columns and test contains 292 rows and 80 columns (Sale Price missing)
  4. The data contains continuous as well as descrete data.
  5. The data containg categorical data.
- Data Sources and their formats
- MSSubClass: Identifies the type of dwelling involved in the sale.

➢ 20 1-STORY 1946 & NEWER ALL STYLES
➢ 30 1-STORY 1945 & OLDER
➢ 40 1-STORY W/FINISHED ATTIC ALL AGES
➢ 45 1-1/2 STORY - UNFINISHED ALL AGES
➢ 50 1-1/2 STORY FINISHED ALL AGES
➢ 60 2-STORY 1946 & NEWER
➢ 70 2-STORY 1945 & OLDER
➢ 75 2-1/2 STORY ALL AGES
➢ 80 SPLIT OR MULTI-LEVEL
➢ 85 SPLIT FOYER
➢ 90 DUPLEX - ALL STYLES AND AGES
➢ 120 1-STORY PUD (Planned Unit Development) - 1946 & NEWER
➢ 150 1-1/2 STORY PUD - ALL AGES
➢ 160 2-STORY PUD - 1946 & NEWER
➢ 180 PUD - MULTILEVEL - INCL SPLIT LEV/FOYER
➢ 190 2 FAMILY CONVERSION - ALL STYLES AND AGES

- MSZoning: Identifies the general zoning classification of the sale.

  - A Agriculture
  - C Commercial
  - FV Floating Village Residential
  - I Industrial
  - RH Residential High Density
  - RL Residential Low Density
  - RP Residential Low Density Park
  - RM Residential Medium Density

- LotFrontage: Linear feet of street connected to property. Integer Format

- LotArea: Lot size in square feet. Integer format

- Street: Type of road access to property
  - Grvl Gravel
  - Pave Paved
- Alley: Type of alley access to property
  - Grvl Gravel
  - Pave Paved
  - NA No alley access

- LotShape: General shape of property
  - Reg Regular
  - IR1 Slightly irregular
  - IR2 Moderately Irregular
  - IR3 Irregular

- LandContour: Flatness of the property
  - Lvl Near Flat/Level
  - Bnk Banked - Quick and significant rise from street grade to building
  - HLS Hillside - Significant slope from side to side

- Low Depression

- Utilities: Type of utilities available
  - AllPub All public Utilities (E,G,W,& S)
  - NoSewrElectricity, Gas, and Water (Septic Tank)
  - NoSeWa Electricity and Gas Only
  - ELO Electricity only
- LotConfig: Lot configuration
  - Inside Inside lot
  - Corner Corner lot
  - CulDSac Cul-de-sac
  - FR2 Frontage on 2 sides of property
  - FR3 Frontage on 3 sides of property
- LandSlope: Slope of property
  - Gtl Gentle slope
  - Mod Moderate Slope
  - Sev Severe Slope
- Neighborhood: Physical locations within Ames city limits
  - Blmngtn Bloomington Heights
  - Blueste Bluestem
  - BrDale Briardale
  - BrkSide Brookside
  - ClearCr Clear Creek
  - CollgCr College Creek
  - Crawfor Crawford
  - Edwards Edwards
  - Gilbert Gilbert
  - IDOTRR Iowa DOT and Rail Road
  - MeadowV Meadow Village
  - Mitchel Mitchell
  - Names North Ames
  - NoRidge Northridge
  - NPkVill Northpark Villa
  - NridgHtNorthridge Heights
  - NWAmes Northwest Ames
  - OldTown Old Town

- SWISU South & West of Iowa State University
- Sawyer Sawyer
- SawyerW Sawyer West
- Somerst Somerset
- StoneBr Stone Brook
- Timber Timberland
- Veenker Veenker

- Condition1: Proximity to various conditions
  - Artery Adjacent to arterial street
  - Feedr Adjacent to feeder street
  - Norm Normal
  - RRNn Within 200' of North-South Railroad
  - RRAn Adjacent to North-South Railroad
  - PosN Near positive off-site feature--park, greenbelt, etc.
  - PosA Adjacent to postive off-site feature
  - RRNe Within 200' of East-West Railroad
  - RRAe Adjacent to East-West Railroad
- Condition2: Proximity to various conditions (if more than one is present)
  - Artery Adjacent to arterial street
  - Feedr Adjacent to feeder street
  - Norm Normal
  - RRNn Within 200' of North-South Railroad
  - RRAn Adjacent to North-South Railroad
  - PosN Near positive off-site feature--park, greenbelt, etc.
  - PosA Adjacent to postive off-site feature
  - RRNe Within 200' of East-West Railroad
  - RRAe Adjacent to East-West Railroad
- BldgType: Type of dwelling
  - 1Fam Single-family Detached
  - 2FmCon Two-family Conversion; originally built as one-family dwelling
  - Duplx Duplex
  - TwnhsE Townhouse End Unit
  - TwnhsI Townhouse Inside Unit

- HouseStyle: Style of dwelling
  - 1Story One story
  - 1.5Fin One and one-half story: 2nd level finished
  - 1.5Unf One and one-half story: 2nd level unfinished
  - 2Story Two story
  - 2.5Fin Two and one-half story: 2nd level finished
  - 2.5Unf Two and one-half story: 2nd level unfinished
  - SFoyer Split Foyer
  - SLvl Split Level
- OverallQual: Rates the overall material and finish of the house
  - 10 Very Excellent
  - 9 Excellent
  - 8 Very Good
  - 7 Good
  - 6 Above Average
  - 5 Average
  - 4 Below Average
  - 3 Fair
  - 2 Poor
  - 1 Very Poor
- OverallCond: Rates the overall condition of the house
  - 10 Very Excellent
  - 9 Excellent
  - 8 Very Good
  - 7 Good
  - 6 Above Average
  - 5 Average
  - 4 Below Average
  - 3 Fair
  - 2 Poor
  - 1 Very Poor
- YearBuilt: Original construction date(integer)
- YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)(integer)
- RoofStyle: Type of roof

- Flat Flat
- Gable Gable
- Gambrel Gabrel (Barn)
- Hip Hip
- Mansard Mansard
- Shed Shed

- RoofMatl: Roof material
  - ClyTile Clay or Tile
  - CompShg Standard (Composite) Shingle
  - Membran Membrane
  - Metal Metal
  - Roll Roll
  - Tar&Grv Gravel & Tar
  - WdShake Wood Shakes
  - WdShngl Wood Shingles
- Exterior1st: Exterior covering on house
  - AsbShng Asbestos Shingles
  - AsphShn Asphalt Shingles
  - BrkComm Brick Common
  - BrkFaceBrick Face
  - CBlock Cinder Block
  - CemntBd Cement Board
  - HdBoard Hard Board
  - ImStuccImitation Stucco
  - MetalSd Metal Siding
  - Other Other
  - Plywood Plywood
  - PreCast PreCast
  - Stone Stone
  - Stucco Stucco
  - VinylSd Vinyl Siding
  - Wd Sdng Wood Siding
  - WdShing Wood Shingles

- Exterior2nd: Exterior covering on house (if more than one material)

- AsbShng Asbestos Shingles
- AsphShn Asphalt Shingles
- BrkComm Brick Common
- BrkFaceBrick Face
- CBlock Cinder Block
- CemntBd Cement Board
- HdBoard Hard Board
- ImStuccImitation Stucco
- MetalSd Metal Siding
- Other Other
- Plywood Plywood
- PreCast PreCast
- Stone Stone
- Stucco Stucco
- VinylSd Vinyl Siding
- Wd Sdng Wood Siding
- WdShing Wood Shingles

- MasVnrType: Masonry veneer type
  - BrkCmnBrick Common
  - BrkFaceBrick Face
  - CBlock Cinder Block
  - None None
  - Stone Stone
- MasVnrArea: Masonry veneer area in square feet
- ExterQual: Evaluates the quality of the material on the exterior
  - Ex Excellent
  - Gd Good
  - TA Average/Typical
  - Fa Fair
  - Po Poor
- ExterCond: Evaluates the present condition of the material on the exterior
  - Ex Excellent
  - Gd Good
  - TA Average/Typical

- Fa Fair
- Po Poor

- Foundation: Type of foundation
  - BrkTil Brick & Tile
  - CBlock Cinder Block
  - PConc Poured Contrete
  - Slab Slab
  - Stone Stone
  - Wood Wood
- BsmtQual: Evaluates the height of the basement
  - Ex Excellent (100+ inches)
  - Gd Good (90-99 inches)
  - TA Typical (80-89 inches)
  - Fa Fair (70-79 inches)
  - Po Poor (<70 inches
  - NA No Basement
- BsmtCond: Evaluates the general condition of the basement
  - Ex Excellent
  - Gd Good
  - TA Typical - slight dampness allowed
  - Fa Fair - dampness or some cracking or settling
  - Po Poor - Severe cracking, settling, or wetness
  - NA No Basement
- BsmtExposure: Refers to walkout or garden level walls
  - Gd Good Exposure
  - Av Average Exposure (split levels or foyers typically score average or above)
  - Mn Mimimum Exposure
  - No No Exposure
  - NA No Basement
- BsmtFinType1: Rating of basement finished area
  - GLQ Good Living Quarters
  - ALQ Average Living Quarters
  - BLQ Below Average Living Quarters
  - Rec Average Rec Room

- LwQ Low Quality
- Unf Unfinshed
- NA No Basement

- BsmtFinSF1: Type 1 finished square feet
- BsmtFinType2: Rating of basement finished area (if multiple types)
  - GLQ Good Living Quarters
  - ALQ Average Living Quarters
  - BLQ Below Average Living Quarters
  - Rec Average Rec Room
  - LwQ Low Quality
  - Unf Unfinshed
  - NA No Basement
- BsmtFinSF2: Type 2 finished square feet
- BsmtUnfSF: Unfinished square feet of basement area
- TotalBsmtSF: Total square feet of basement area
- Heating: Type of heating
  - Floor Floor Furnace
  - GasA Gas forced warm air furnace
  - GasW Gas hot water or steam heat
  - Grav Gravity furnace
  - OthW Hot water or steam heat other than gas
  - Wall Wall furnace
- HeatingQC: Heating quality and condition
  - Ex Excellent
  - Gd Good
  - TA Average/Typical
  - Fa Fair
  - Po Poor
- CentralAir: Central air conditioning
  - N No
  - Y Yes
- Electrical: Electrical system
  - SBrkr Standard Circuit Breakers & Romex
  - FuseA Fuse Box over 60 AMP and all Romex wiring (Average)
  - FuseF 60 AMP Fuse Box and mostly Romex wiring (Fair)

- FuseP 60 AMP Fuse Box and mostly knob & tube wiring (poor)
- Mix Mixed

- 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
  - Ex Excellent
  - Gd Good
  - TA Typical/Average
  - Fa Fair
  - Po Poor
- TotRmsAbvGrd: Total rooms above grade (does not include bathrooms
- Functional: Home functionality (Assume typical unless deductions are warranted)
  - Typ Typical Functionality
  - Min1 Minor Deductions 1
  - Min2 Minor Deductions 2
  - Mod Moderate Deductions
  - Maj1 Major Deductions 1
  - Maj2 Major Deductions 2
  - Sev Severely Damaged
  - Sal Salvage only
- Fireplaces: Number of fireplaces
- FireplaceQu: Fireplace quality
  - Ex Excellent - Exceptional Masonry Fireplace

- Gd Good - Masonry Fireplace in main level
- TA Average - Prefabricated Fireplace in main living area or Masonry Fireplace in basement
- Fa Fair - Prefabricated Fireplace in basement
- Po Poor - Ben Franklin Stove
- NA No Fireplace

- GarageType: Garage location
  - 2Types More than one type of garage
  - Attchd Attached to home
  - Basment Basement Garage
  - BuiltIn Built-In (Garage part of house - typically has room above garage)
  - CarPort Car Port
  - Detchd Detached from home
  - NA No Garage
- GarageYrBlt: Year garage was built
- GarageFinish: Interior finish of the garage
  - Fin Finished
  - RFn Rough Finished
  - Unf Unfinished
  - NA No Garage
- GarageCars: Size of garage in car capacity
- GarageArea: Size of garage in square feet
- GarageQual: Garage quality
  - Ex Excellent
  - Gd Good
  - TA Typical/Average
  - Fa Fair
  - Po Poor
  - NA No Garage
- GarageCond: Garage condition
  - Ex Excellent
  - Gd Good
  - TA Typical/Average
  - Fa Fair

- Po Poor
- NA No Garage

- PavedDrive: Paved driveway
  - Y Paved
  - P Partial Pavement
  - N Dirt/Gravel
- 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
  - Ex Excellent
  - Gd Good
  - TA Average/Typical
  - Fa Fair
  - NA No Pool
- Fence: Fence quality
  - GdPrv Good Privacy
  - MnPrv Minimum Privacy
  - GdWo Good Wood
  - MnWw Minimum Wood/Wire
  - NA No Fence
- MiscFeature: Miscellaneous feature not covered in other categories
  - Elev Elevator
  - Gar2 2nd Garage (if not described in garage section)
  - Othr Other
  - Shed Shed (over 100 SF)
  - TenC Tennis Court
  - NA None
- MiscVal: $Value of miscellaneous feature
- MoSold: Month Sold (MM)
- YrSold: Year Sold (YYYY)

- SaleType: Type of sale
  - WD Warranty Deed - Conventional
  - CWD Warranty Deed - Cash
  - VWD Warranty Deed - VA Loan
  - New Home just constructed and sold
  - COD Court Officer Deed/Estate
  - Con Contract 15% Down payment regular terms
  - ConLw Contract Low Down payment and low interest
  - ConLI Contract Low Interest
  - ConLD Contract Low Down
  - Oth Other
- SaleCondition: Condition of sale
  - Normal Normal Sale
  - Abnorml Abnormal Sale - trade, foreclosure, short sale
  - AdjLand Adjoining Land Purchase
  - Alloca Allocation - two linked properties with separate deeds, typically condo with a garage unit
  - Family Sale between family members
  - Partial Home was not completed when last assessed (associated with New Homes)

- Data Preprocessing Done
  - OneHot Encoded all the data categorical data- We encode the categorical data in this step, to convert it to integer type, since the model does not work on 'string' data.
  - Regularised the data using standard Scaler.- The next step is to bring the data to a common scale, since there are certain columns with very small values and some columns with high values. This process is important as values on a similar scale allow the model to learn better. We use standard scaler for this process
  - Removed the skewness using power transform-yeo-Johnson- The Yeo–Johnson transformation allows also for zero and negative values in the dataset.

- Data Inputs- Logic- Output Relationships

- Since there were a lot of columns that had tremendous relationship with the Sale price used sklearn.decomposition.PCA to get the input for the desired output.

- Hardware and Software Requirements and Tools Used

  Listing down the hardware and software requirements along with the tools, libraries and packages used.

  1. Jupyter Notebook.- Used python to perform the machine Learning task
  2. Laptop with 8 GB RAM

Model/s Development and Evaluation

Different models I tried:

	Models	CVS	r2	diff
0	Linear regression	97.51	87.69	-9.82
1	Lasso	21.27	87.69	66.42
2	Ridge	70.97	87.14	16.17
3	Elastic Net	68.62	79.18	10.56
4	Decision Tree	55.56	97.44	41.88
5	KNN	54.13	100.00	45.87
6	Random Forest	99.99	72.66	-27.33
7	Extra tree Regressor	52.74	82.79	30.05
8	Ada Boost	68.14	82.21	14.07
9	XGB Regressor	76.36	99.99	23.63

#From the above analysis Linear Regressor has least difference between r2 and cvs

Using hyper parameter tuning on LinearRegressor further increased the accuracy.

Linear regression is an attractive model because the representation is so simple.
The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

The linear equation assigns one scale factor to each input value or column, called a coefficient and represented by the capital Greek letter Beta (B). One additional coefficient is also added, giving the line an additional degree of freedom (e.g. moving up and down on a two-dimensional plot) and is often called the intercept or the bias coefficient.

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

$$y = B0 + B1*x$$

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients (e.g. B0 and B1 in the above example).

It is common to talk about the complexity of a regression model like linear regression. This refers to the number of coefficients used in the model.

When a coefficient becomes zero, it effectively removes the influence of the input variable on the model and therefore from the prediction made from the model (0 * x = 0). This becomes relevant if you look at regularization methods that change the learning algorithm to reduce the complexity of regression models by putting pressure on the absolute size of the coefficients, driving some to zero.

In [78]:
```
from sklearn.linear_model import LinearRegression,Lasso,Ridge,ElasticNet
from sklearn.metrics import r2_score,mean_squared_error
from sklearn.model_selection import train_test_split
```

In [79]:
```
def fun(f,x,y):
    f.fit(x,y)
    pred=f.predict(x)
    print("MSE=",mean_squared_error(y,pred))
    print("r2 score=",r2_score(y,pred))
```

In [92]:
```
from sklearn.model_selection import cross_val_score,KFold
def cvs(m,x,y):
    cv1=KFold(n_splits=5,shuffle=True)
    score=cross_val_score(m,x,y,cv=cv1,scoring='r2')
    print("Cross val score",score)
    print(score.mean())
```

In [93]:
```
from sklearn.model_selection import GridSearchCV
def hypertuning(params,model,x,y):
    gd=GridSearchCV(model,params,cv=5)
    gd.fit(x,y)
    print(gd.best_params_)
```

In [94]:
```
lr=LinearRegression()
lr_params={'fit_intercept':[True,False],'normalize':[True,False],'copy_X':[True,False],'n_jobs':[None,1,2,3]}
hypertuning(lr_params,lr,X_train,y_train)
```

```
{'copy_X': True, 'fit_intercept': True, 'n_jobs': None, 'normalize': False}
```

In [98]:
```
lr=LinearRegression(copy_X=True,fit_intercept=True,n_jobs=None,normalize=False)
fun(lr,X_train,y_train)
cvs(lr,X_train,y_train)
```

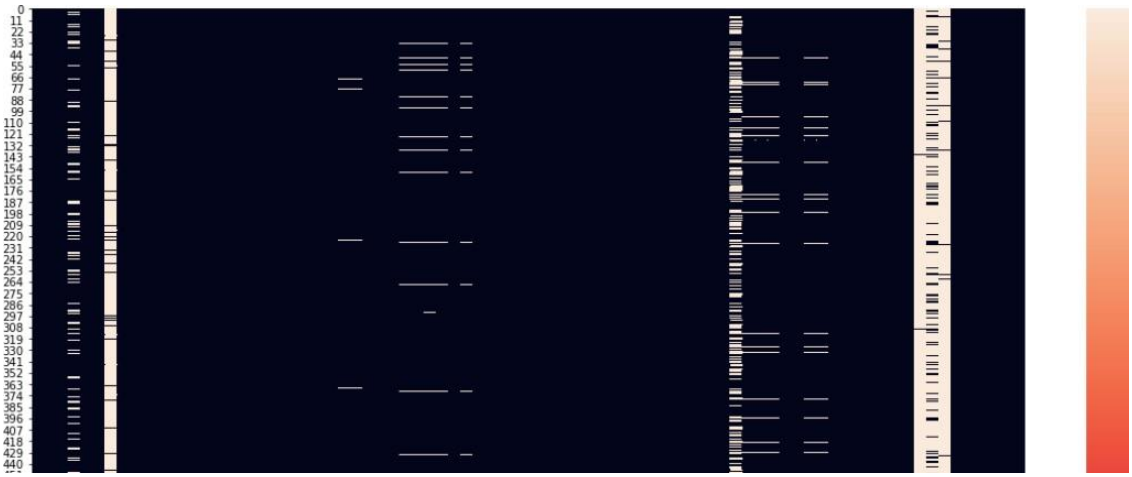
```
MSE= 769204450.7138395
r2 score= 0.8769732754407024
Cross val score [-3.03547066  0.62328189  0.07048686 -0.11140972 -2.42216748]
-0.9750558223074499
```

In [114]:
```
r2=[87.69]
CVS=[97.51]
model=['Linear regression']
```

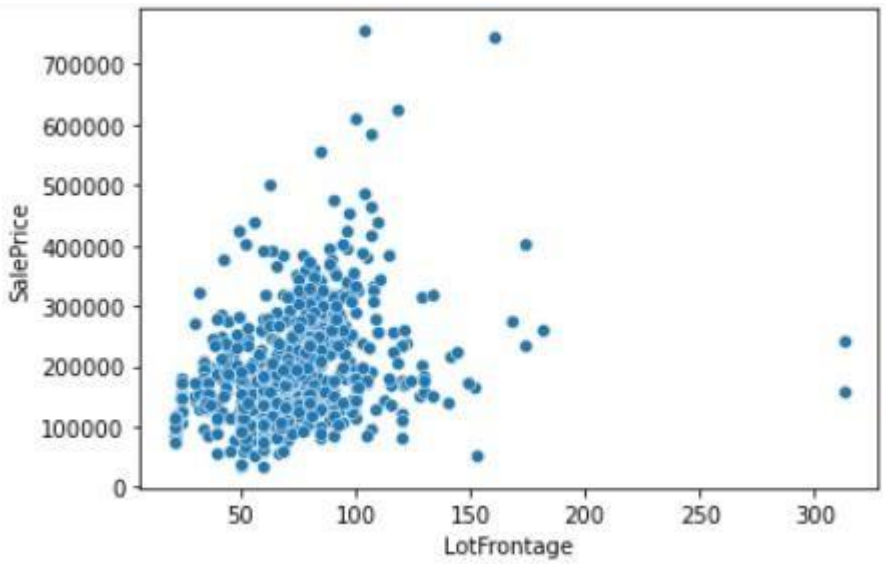
In [102]:
```
l=Lasso()
params_lasso={
    'alpha':[0.001,0.01,0.1,1],
    'fit_intercept':[True,False],
    'normalize':[True,False],
    'copy_X':[True,False]
}
```

Similarily did the same for all models shown in the table above

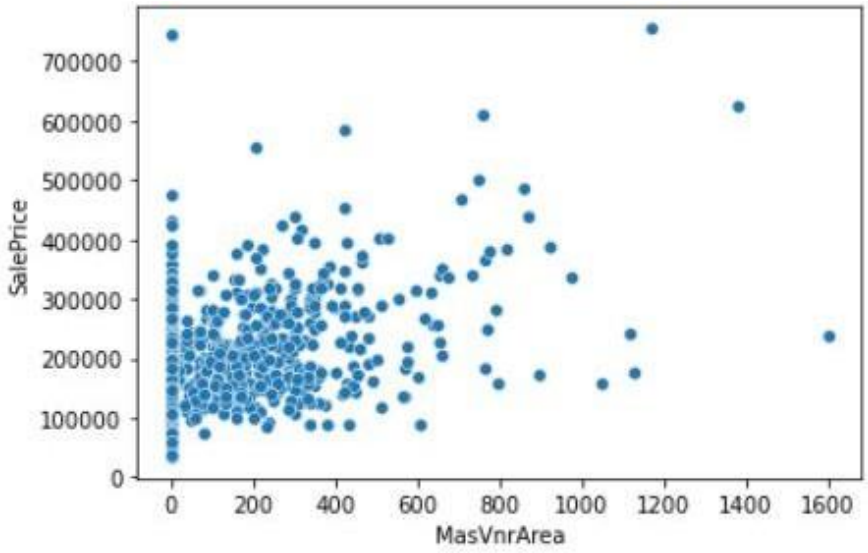
Visualizations



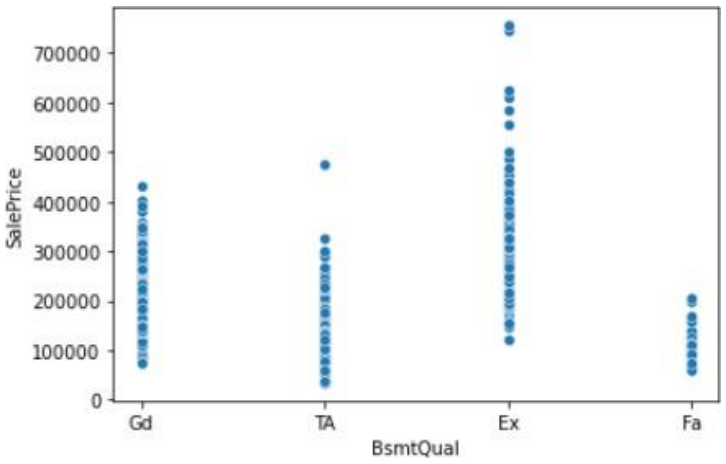
Data had a lot of null values.



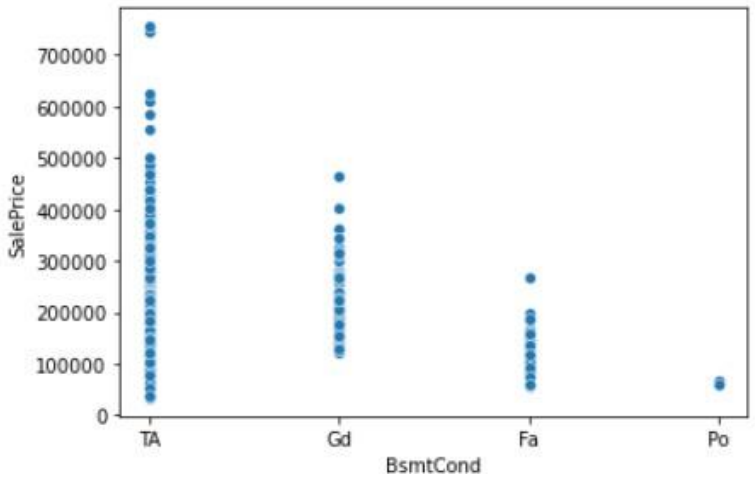
LotFrontage in related to Sales Price.



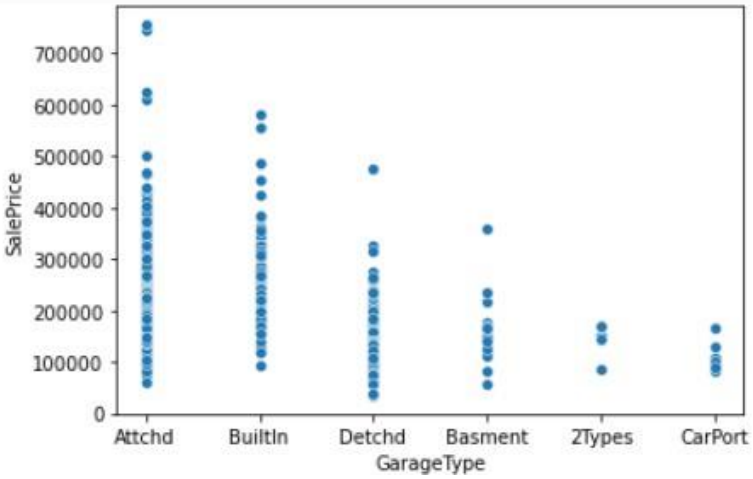
MasVnrArea also affects the price.



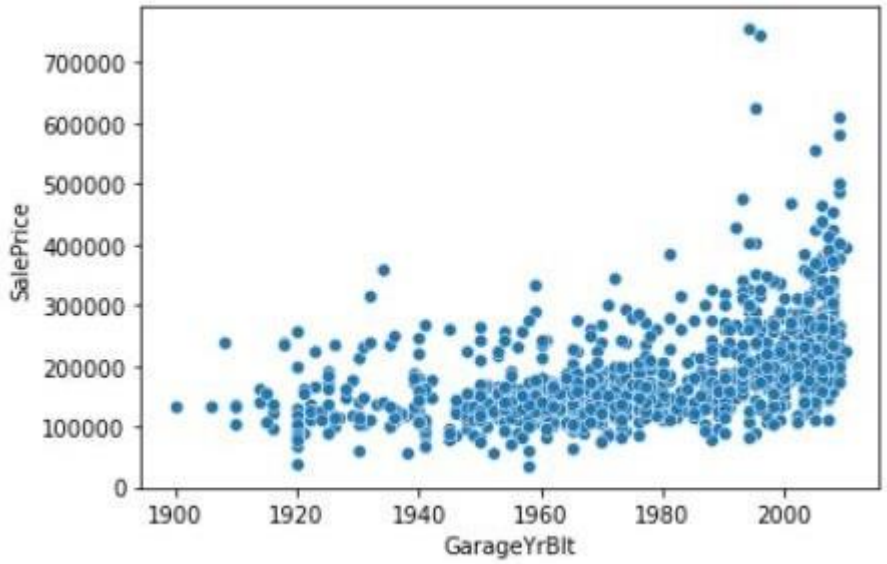
Basement Quality affects the price.



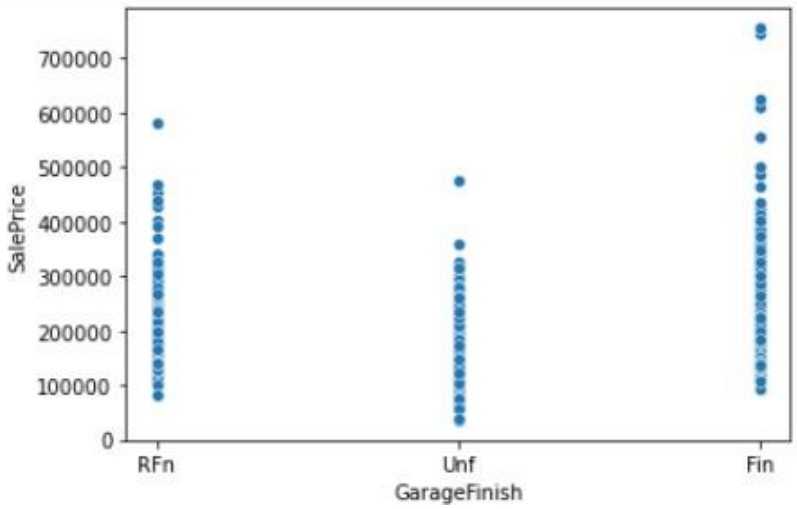
Basement Condition affects price.



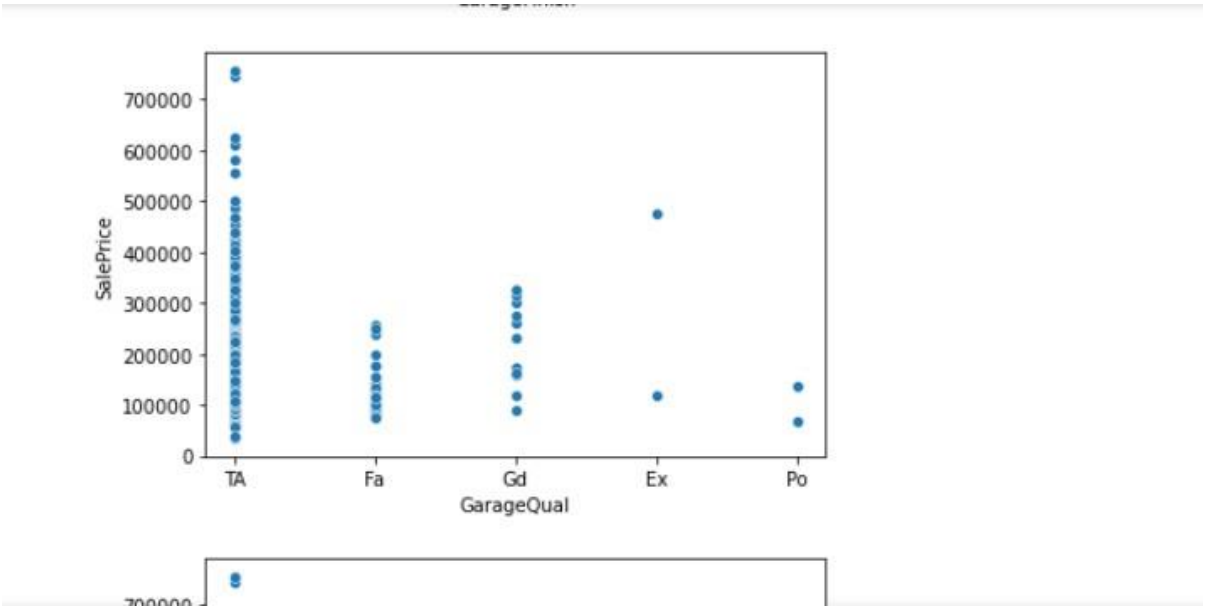
Garage type affects the price.



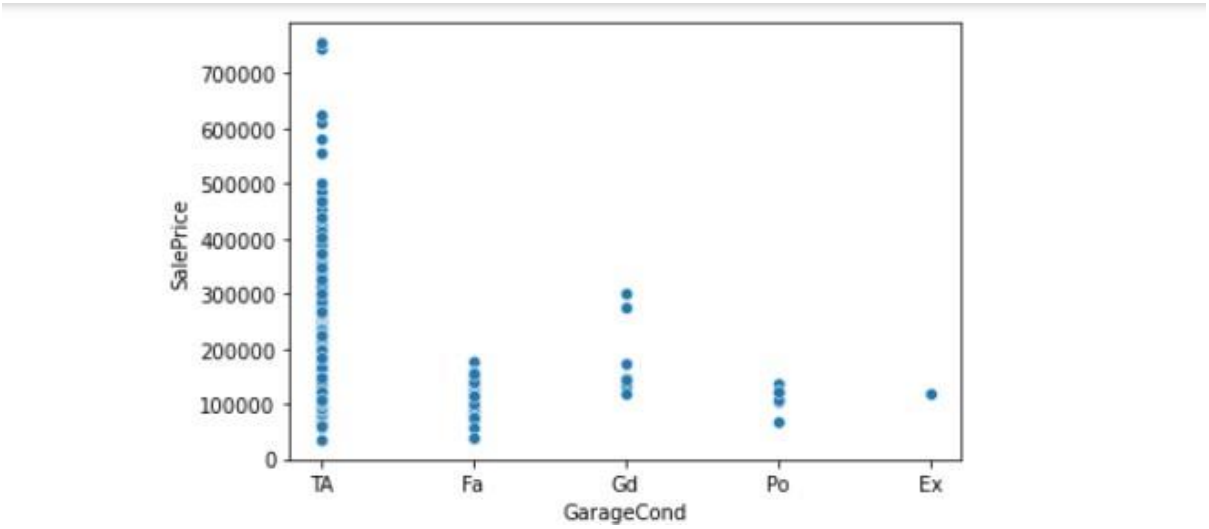
Newer the Garage, Higher the price.



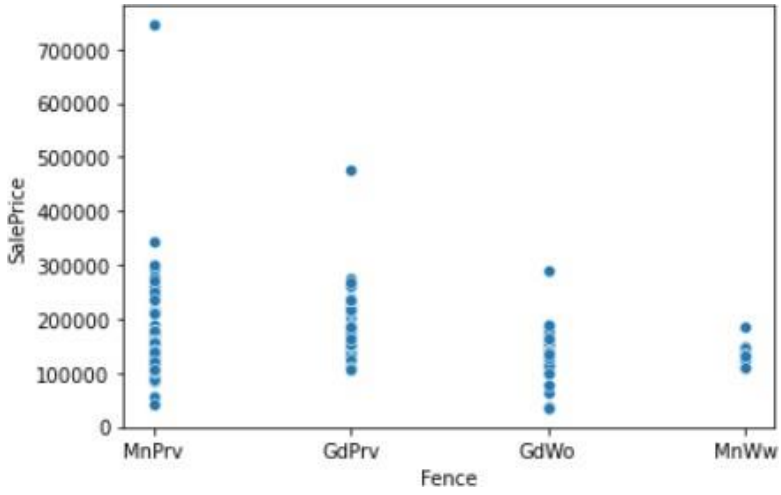
Whether the Garage is finished or not also affects the price.



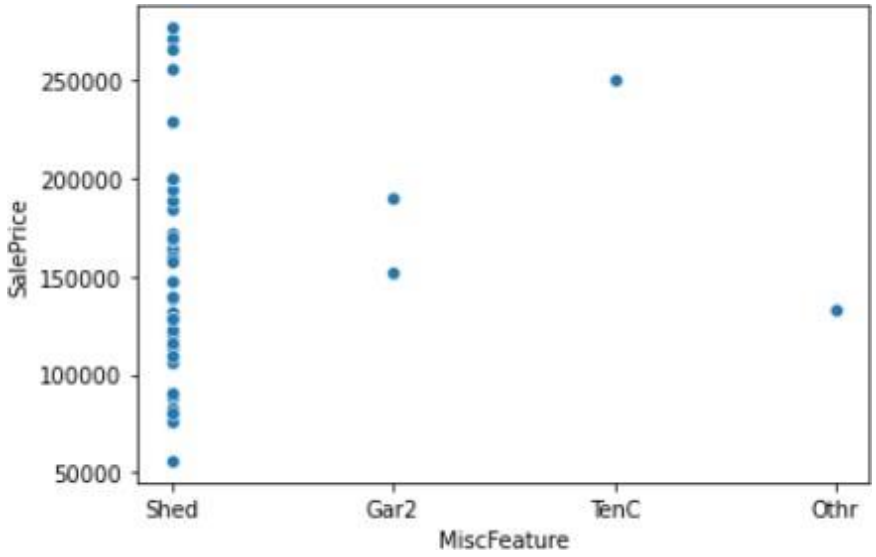
Garage Quality affects the price.



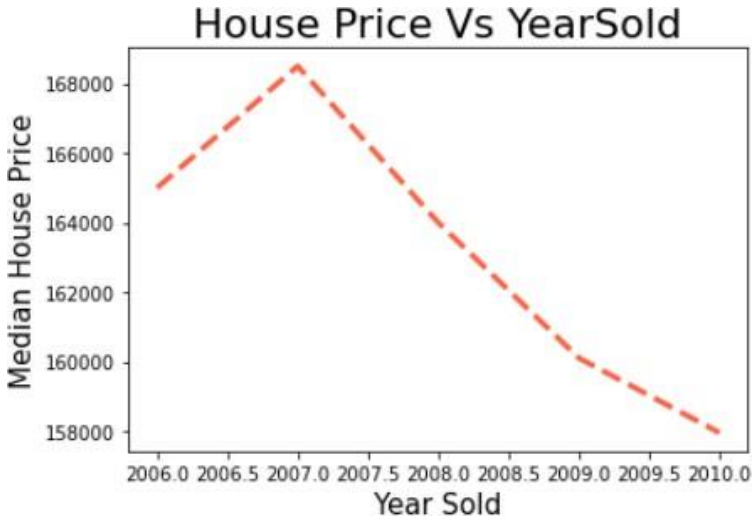
Garage Condition affects the price.



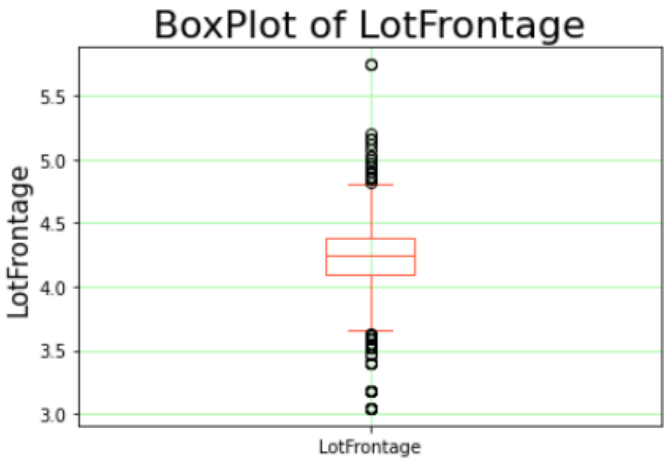
Fence affects the price.



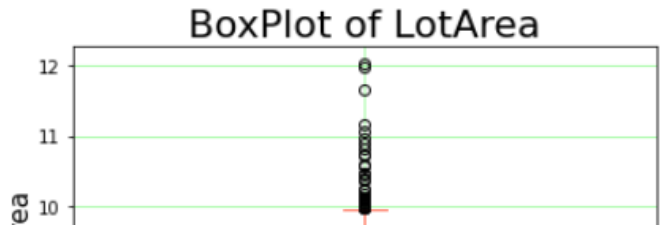
Misc features affect the price.



House Price was max if sold in 2007 and after that it started decreasing.




```
<Figure size 2160x864 with 0 Axes>
```



Data contains a lot of outliers.

CONCLUSION

- Key Findings and Conclusions of the Study
  The model that fits best is Linear Regression

```
In [94]: lr=LinearRegression()
         lr_params={'fit_intercept':[True,False],'normalize':[True,False],'copy_X':[True,False],'n_jobs':[None,1,2,3]}
         hypertuning(lr_params,lr,X_train,y_train)

         {'copy_X': True, 'fit_intercept': True, 'n_jobs': None, 'normalize': False}

In [98]: lr=LinearRegression(copy_X=True,fit_intercept=True,n_jobs=None,normalize=False)
         fun(lr,X_train,y_train)
         cvs(lr,X_train,y_train)

         MSE= 769204450.7138395
         r2 score= 0.8769732754407024
         Cross val score [-3.03547066  0.62328189  0.07048686 -0.11140972 -2.42216748]
         -0.9750558223074499
```

- Learning Outcomes of the Study in respect of Data Science
  - Since a lot of factors were related to the Sales Price it was very difficult to determine the best features hence PCA was used.
  - The data was highly spread and contained a lot of skewness which was removed by yeo-johnson which reduced the skewness in data.

- Used Linear Regression as it works best with a little bit of outliers in the dataset.

- Limitations of this work and Scope for Future Work
  Since a lot of data was not recorded we filled it using the best strategy so there could be some deviation with the actual data.
  The training dataset contained only 1168 records. The more the data the better the learning.
  Current recorded data to be provided for better results in future.