House Pricing Model

Submitted by:

Vandana Jain

Internship 10

ACKNOWLEDGMENT

The internship opportunity I had with FlipRobo was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this project period.

I would like to thank our SME for suggesting this project and for his whole hearted cooperation and constant encouragement throughout the project.

INTRODUCTION

- Business Problem Framing

  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?

- Review of Literature

  Every single organization in today’s real estate business is operating fruitfully to achieve a competitive edge over alternative competitors. There is a need to simplify the process for a normal human being while providing the best results. This paper proposes a system that predicts house prices using a regression machine learning algorithm. In case you're going to sell a house, you have to recognize what sticker price to put on it. What's more, a PC calculation can give you a precise gauge!. This regression model is built not only for predicting the price of the house which is ready for sale but also for houses that are under construction. Regression is a machine learning apparatus that encourages you to make expectations by taking in – from the current measurable information – the connections between your target parameter and a lot of different independent parameters. As per this definition, a house's cost relies upon parameters, for example, the number of rooms, living region, area, and so forth. On the off chance that we apply counterfeit figuring out how to these parameters, we can compute house valuations in a given land region. The target feature in this proposed model is the price of the real estate property and the independent features are: no. of bedrooms, no. of bathrooms, carpet area, built-up area, the floor, age of the property. Other than those of the mentioned features, which are generally required for predicting the house prices. The whole implementation is done using the python programming language. For the construction of the predictive model, a Decision tree regressor is used from the "Scikit-learn" machine learning library.

- Motivation for the Problem Undertaken

  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.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

Machine Learning is defined by Tom Mitchell in his book as "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E". Supervised learning is when the output is known for the corresponding inputs, and is also provided for the machine to learn.

- Data Sources and their formats

The data is provided to us from our client database. It is hereby given to us for model the price of houses with the available independent variables. It is given in the csv file format.

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

LotArea: Lot size in square feet

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
IR1Slightly irregular
IR2Moderately Irregular
IR3Irregular

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)

NoSewr Electricity, 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
NridgHt	Northridge 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

YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)

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
BrkFace	Brick Face
CBlock	Cinder Block
CemntBd	Cement Board
HdBoard	Hard Board
ImStucc	Imitation 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
BrkFace	Brick Face
CBlock	Cinder Block
CemntBd	Cement Board
HdBoard	Hard Board
ImStucc	Imitation 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

BrkCmn Brick Common
BrkFace Brick 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 Pre-processing Done

  Data pre-processing is the process of cleaning our data set. There might be missing values or outliers in the dataset. These can be handled by data cleaning. If there are many missing values in a variable we will drop those values or substitute it with the average value.

  - Data Exploration

  Data Exploration is the key to getting insights from data. Practitioners say a good data exploration strategy can solve even complicated problems in a few hours. A good data exploration strategy comprises the following:

1. **Univariate Analysis** - It is used to visualize one variable in one plot. Examples: histogram, density plot, etc.
2. **Bivariate Analysis** - It is used to visualize two variables (x and y axis) in one plot. Examples: bar chart, line chart, area chart, etc.
3. **Multivariate Analysis** - As the name suggests, it is used to visualize more than two variables at once. Examples: stacked bar chart, dodged bar chart, etc.

4. **Cross Tables** -They are used to compare the behavior of two categorical variables (used in pivot tables as well).

- Data Inputs- Logic- Output Relationships

  Some factors which I can think of that directly influence house prices are the following:

  - Area of House
  - How old is the house
  - Location of the house
  - How close/far is the market
  - Connectivity of house location with transport
  - How many floors does the house have
  - What material is used in the construction
  - Water /Electricity availability
  - Play area / parks for kids (if any)
  - If terrace is available
  - If car parking is available
  - If security is available

- Hardware and Software Requirements and Tools Used

  Hardware : Since the computational aspect of the project is of importance to PANDA, it is important to know the hardware that was used in the evaluation process. The training and evaluation of the neural network model has been done on a Windows 10 computer using a quad-core CPU at i3.

  Software : Anaconda 3 , Windows 10 , Microsoft office.

  Tools used : Python , Machine learning libraries.

Model/s Development and Evaluation

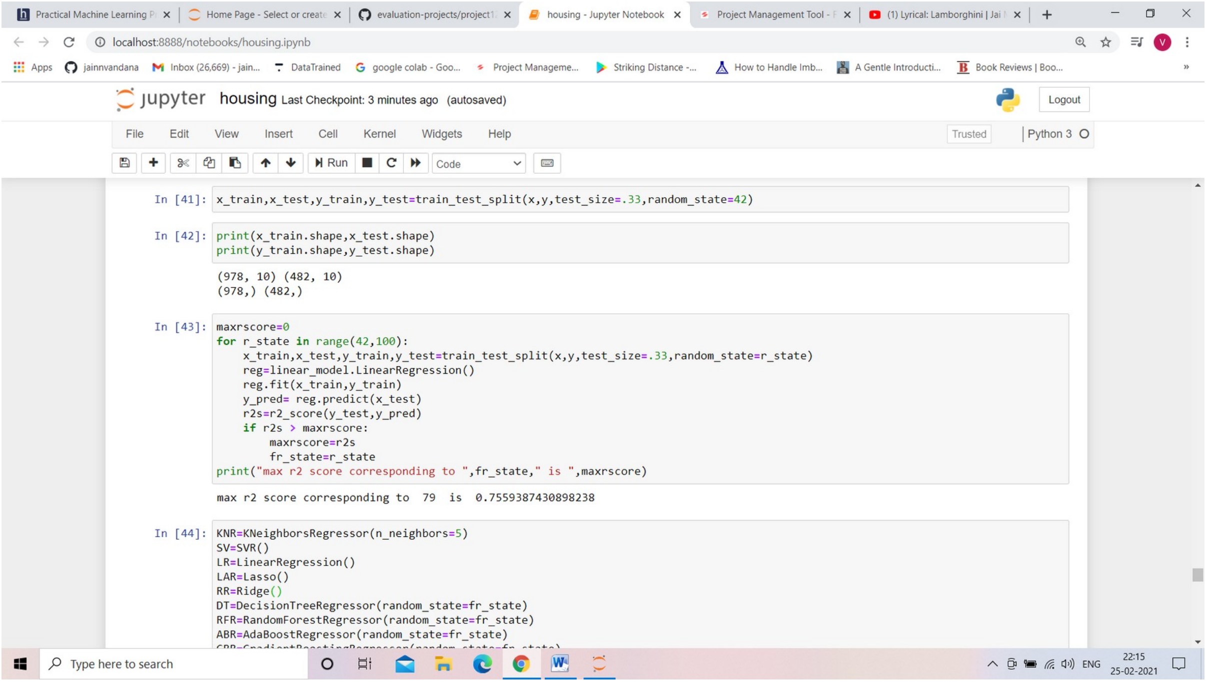
- Identification of possible problem-solving approaches (methods)
    - K Neighbors Regressor: The K-Nearest Neighbors algorithm uses the entire data set as the training set, rather than splitting the data set into a training set and test set.When an outcome is required for a new data instance, the KNN algorithm goes through the entire data set to find the k-nearest instances to the new instance, or the k number of instances most similar to the new record, and then outputs the mean of the outcomes (for a regression problem) or the mode (most frequent class) for a classification problem. The value of k is user-specified.
    - Support Vector Regression : Support Vector Machine (SVM) is another most powerful algorithm with strong theoretical foundations based on the Vapnik-Chervonenkis theory, as defined by Oracle docs. This supervised machine learning algorithm has strong regularization and can be leveraged both for classification or regression challenges. They are characterized by usage of kernels, the sparseness of the solution and the capacity control gained by acting on the margin, or on number of support vectors, etc. The capacity of the system is controlled by parameters that do not depend on the dimensionality of feature space. Since the SVM algorithm operates natively on numeric attributes, it uses a z-score normalization on numeric attributes.

- Linear Regression : linear regression is a statistical method that enables users to summarise and study relationships between two continuous (quantitative) variables. Linear regression is a linear model wherein a model that assumes a linear relationship between the input variables (x) and the single output variable (y). Here the y can be calculated from a linear combination of the input variables (x). When there is a single input variable (x), the method is called a simple linear regression. When there are multiple input variables, the procedure is referred as multiple linear regression.
- Lasso Regression : LASSO stands for Least Absolute Selection Shrinkage Operator wherein shrinkage is defined as a constraint on parameters. The goal of lasso regression is to obtain the subset of predictors that minimize prediction error for a quantitative response variable. The algorithm operates by imposing a constraint on the model parameters that causes regression coefficients for some variables to shrink toward a zero.
- Ridge Regression : Ridge Regression is a technique used when the data suffers from multicollinearity (independent variables are highly correlated). In multicollinearity, even though the least squares estimates (OLS) are unbiased, their variances are large which deviates the observed value far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.
- Decision Tree Regressor : Decision tree methods construct a model of decisions made based on actual values of attributes in the data.Decisions fork in tree structures until a prediction decision is made for a given record. Decision trees are trained on data for classification and regression problems. Decision trees are often fast and accurate and a big favorite in machine learning.
- Random Forest Regressor : Unlike a decision tree, where each node is split on the best feature that minimizes error, in Random Forests, we choose a random selection of

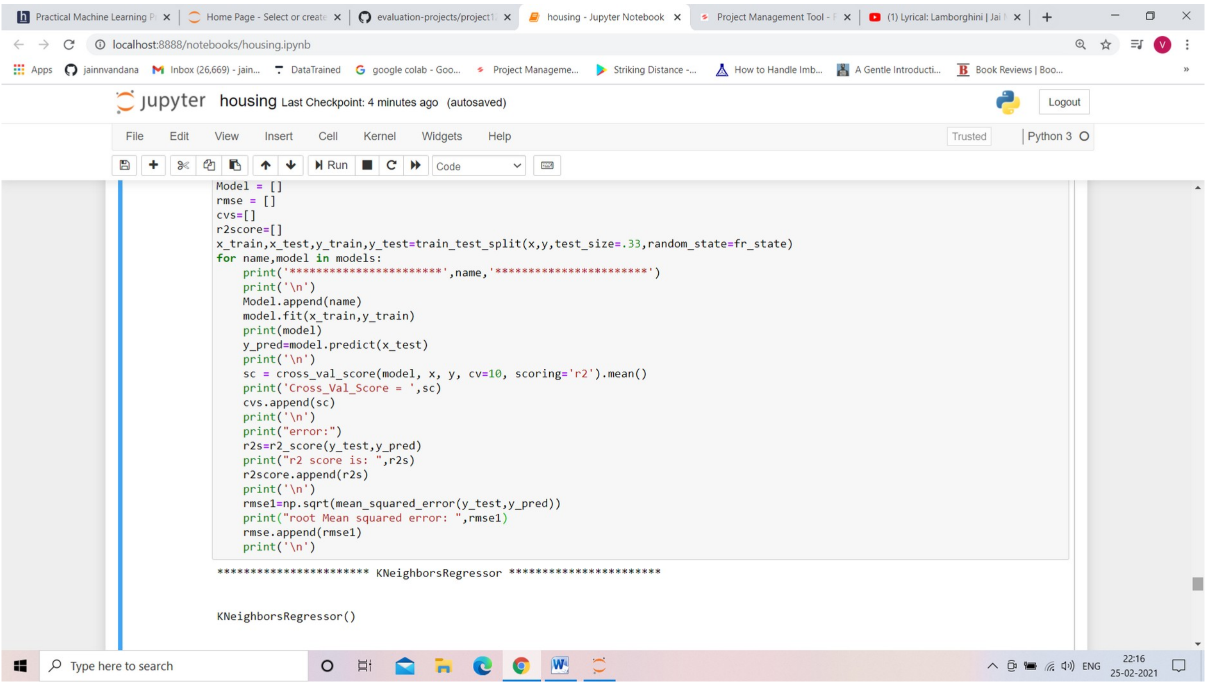
features for constructing the best split. The reason for randomness is: even with bagging, when decision trees choose the best feature to split on, they end up with similar structure and correlated predictions. But bagging after splitting on a random subset of features means less correlation among predictions from subtrees.

- AdaBoost Regressor : Adaboost stands for Adaptive Boosting. Bagging is a parallel ensemble because each model is built independently. On the other hand, boosting is a sequential ensemble where each model is built based on correcting the misclassifications of the previous model.
- Gradient Boosting Regressor : Gradient boosting Regression calculates the difference between the current prediction and the known correct target value.
  This difference is called residual. After that Gradient boosting Regression trains a weak model that maps features to that residual. This residual predicted by a weak model is added to the existing model input and thus this process nudges the model towards the correct target. Repeating this step again and again improves the overall model prediction.

- Run and Evaluate selected models



```
In [41]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.33,random_state=42)

In [42]: print(x_train.shape,x_test.shape)
         print(y_train.shape,y_test.shape)

(978, 10) (482, 10)
(978,) (482,)

In [43]: maxrscore=0
         for r_state in range(42,100):
             x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.33,random_state=r_state)
             reg=linear_model.LinearRegression()
             reg.fit(x_train,y_train)
             y_pred= reg.predict(x_test)
             r2s=r2_score(y_test,y_pred)
             if r2s > maxrscore:
                 maxrscore=r2s
                 fr_state=r_state
         print("max r2 score corresponding to ",fr_state," is ",maxrscore)

max r2 score corresponding to  79  is  0.7559387430898238

In [44]: KNR=KNeighborsRegressor(n_neighbors=5)
         SV=SVR()
         LR=LinearRegression()
         LAR=Lasso()
         RR=Ridge()
         DT=DecisionTreeRegressor(random_state=fr_state)
         RFR=RandomForestRegressor(random_state=fr_state)
         ABR=AdaBoostRegressor(random_state=fr_state)
```

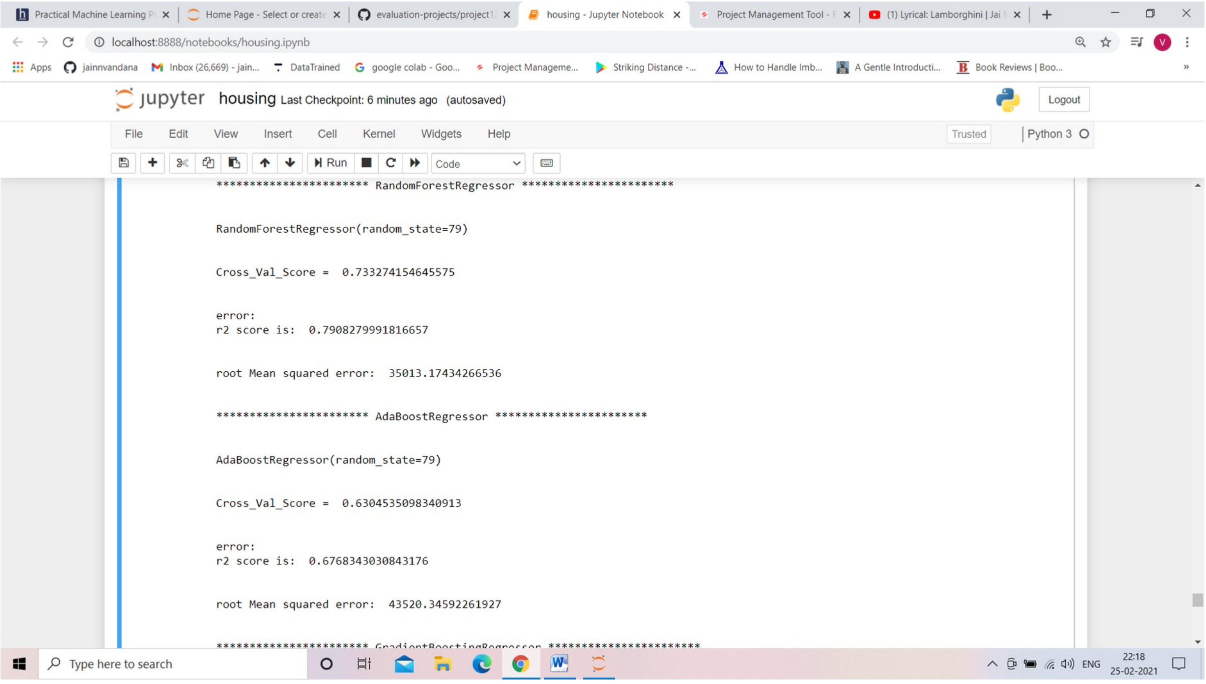


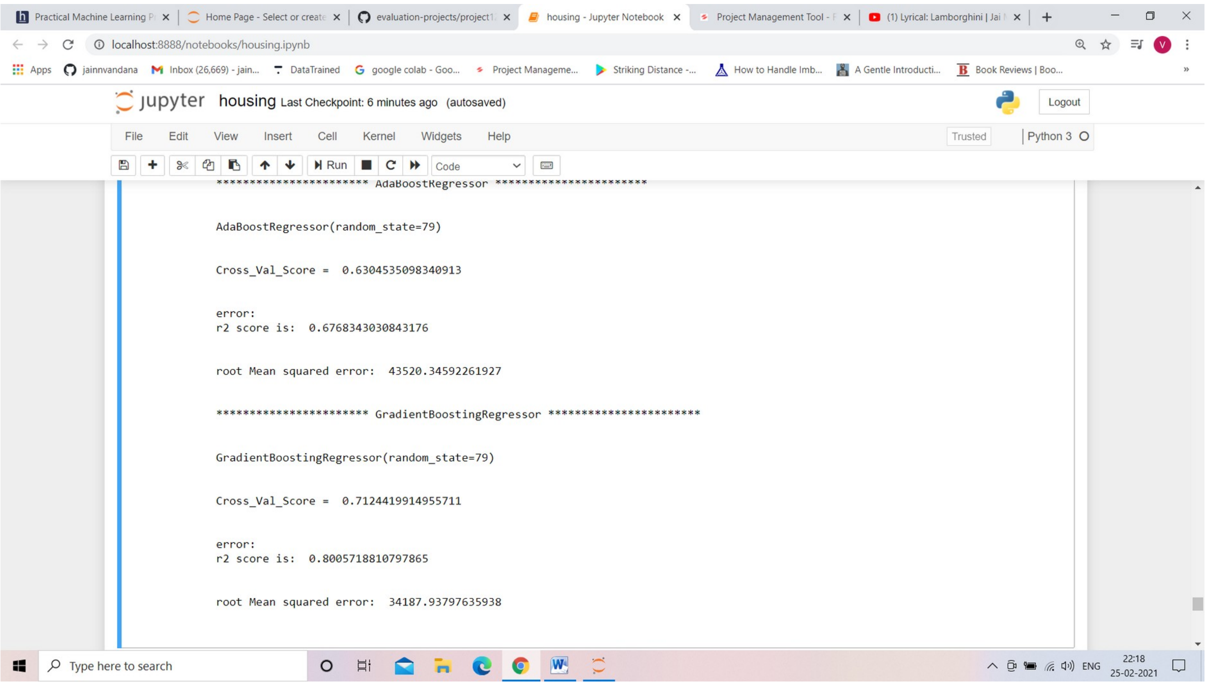
```
Model = []
rmse = []
cvs=[]
r2score=[]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.33,random_state=fr_state)
for name,model in models:
    print('***********************',name,'***********************')
    print('\n')
    Model.append(name)
    model.fit(x_train,y_train)
    print(model)
    y_pred=model.predict(x_test)
    print('\n')
    sc = cross_val_score(model, x, y, cv=10, scoring='r2').mean()
    print('Cross_Val_Score = ',sc)
    cvs.append(sc)
    print('\n')
    print("error:")
    r2s=r2_score(y_test,y_pred)
    print("r2 score is: ",r2s)
    r2score.append(r2s)
    print('\n')
    rmse1=np.sqrt(mean_squared_error(y_test,y_pred))
    print("root Mean squared error: ",rmse1)
    rmse.append(rmse1)
    print('\n')

*********************** KNeighborsRegressor ***********************


KNeighborsRegressor()
```

```
    print('\n')
********************** KNeighborsRegressor **********************


KNeighborsRegressor()


Cross_Val_Score =  0.654936971489361


error:
r2 score is:  0.6757848974429239


root Mean squared error:  43590.94976602254


********************** SVR **********************


SVR()


Cross_Val_Score =  -0.053366655816703856


error:
r2 score is:  -0.0232049623704331 24


root Mean squared error:  77439.23068187149
```

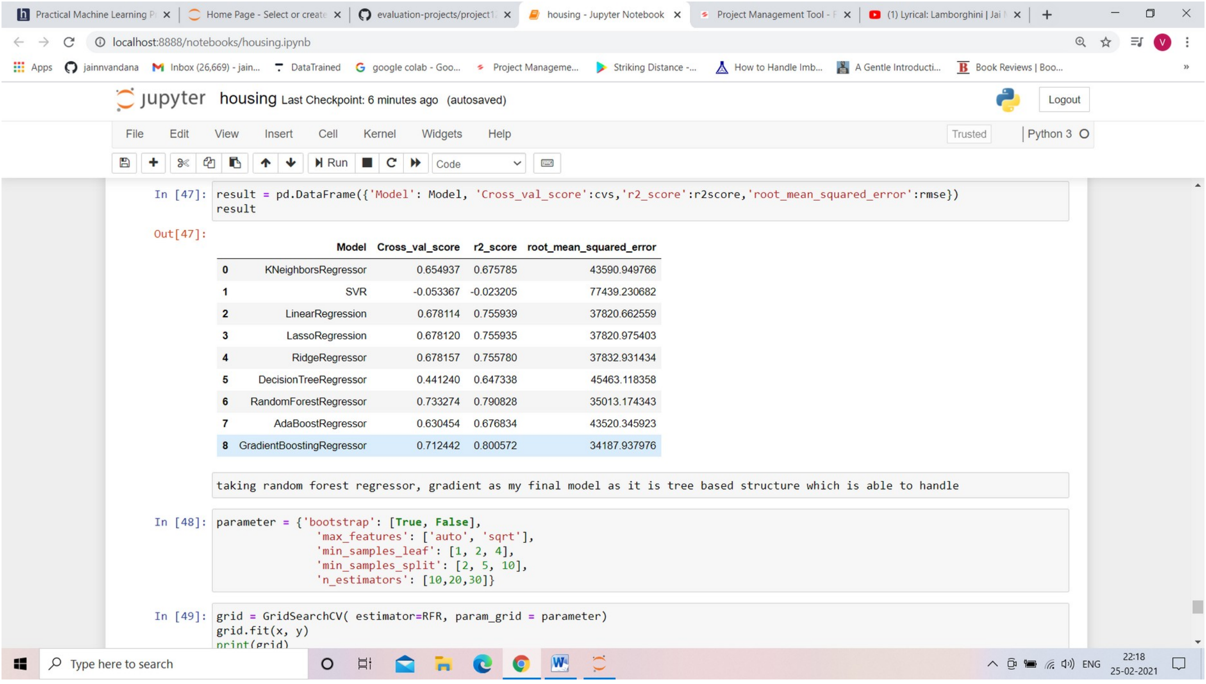
```
********************** LinearRegression **********************


LinearRegression()


Cross_Val_Score =  0.6781142803909679


error:
r2 score is:  0.7559387430898238


root Mean squared error:  37820.66255949312


********************** LassoRegression **********************


Lasso()


Cross_Val_Score =  0.6781195622647582


error:
r2 score is:  0.7559347054335854


root Mean squared error:  37820.97540349892
```

```
********************** RidgeRegressor **********************


Ridge()


Cross_Val_Score =  0.6781572749657295


error:
r2 score is:  0.755780372384263


root Mean squared error:  37832.93143357123


********************** DecisionTreeRegressor **********************


DecisionTreeRegressor(random_state=79)


Cross_Val_Score =  0.44124047354785195


error:
r2 score is:  0.6473377124725819


root Mean squared error:  45463.11835798277
```





- Key Metrics for success in solving problem under consideration



jupyter housing Last Checkpoint: 6 minutes ago (autosaved)

In [47]:
```
result = pd.DataFrame({'Model': Model, 'Cross_val_score':cvs,'r2_score':r2score,'root_mean_squared_error':rmse})
result
```

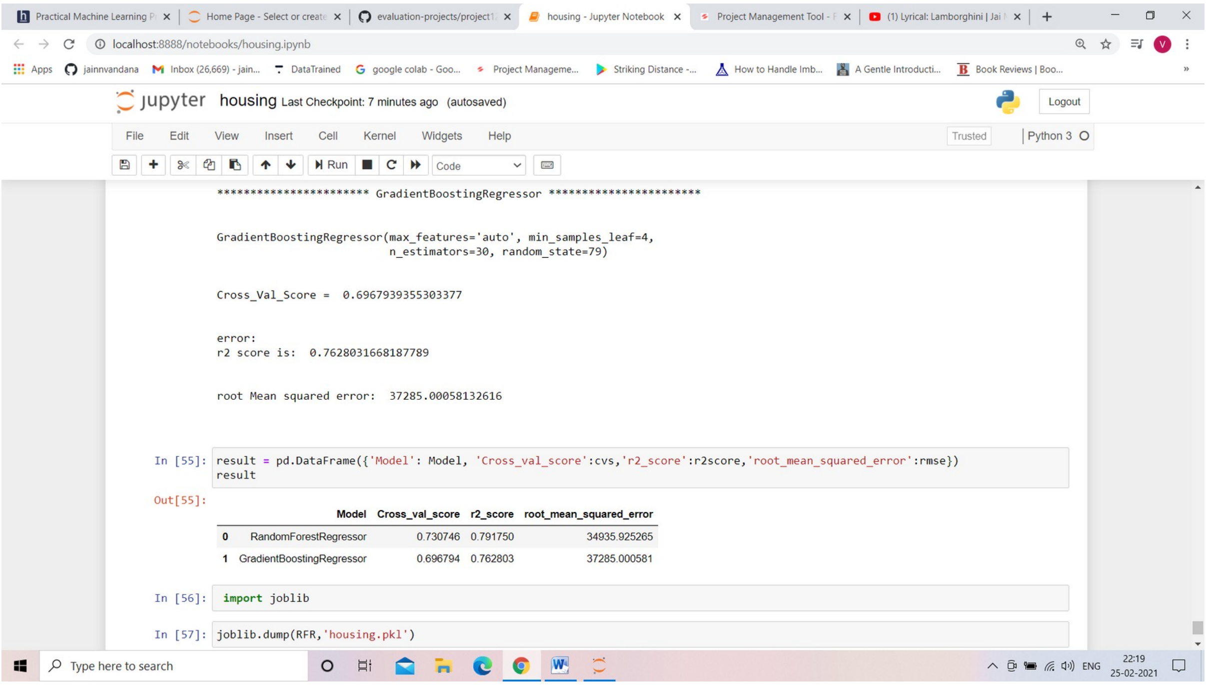
Out[47]:

	Model	Cross_val_score	r2_score	root_mean_squared_error
0	KNeighborsRegressor	0.654937	0.675785	43590.949766
1	SVR	-0.053367	-0.023205	77439.230682
2	LinearRegression	0.678114	0.755939	37820.662559
3	LassoRegression	0.678120	0.755935	37820.975403
4	RidgeRegressor	0.678157	0.755780	37832.931434
5	DecisionTreeRegressor	0.441240	0.647338	45463.118358
6	RandomForestRegressor	0.733274	0.790828	35013.174343
7	AdaBoostRegressor	0.630454	0.676834	43520.345923
8	GradientBoostingRegressor	0.712442	0.800572	34187.937976

taking random forest regressor, gradient as my final model as it is tree based structure which is able to handle

In [48]:
```
parameter = {'bootstrap': [True, False],
             'max_features': ['auto', 'sqrt'],
             'min_samples_leaf': [1, 2, 4],
             'min_samples_split': [2, 5, 10],
             'n_estimators': [10,20,30]}
```

In [49]:
```
grid = GridSearchCV( estimator=RFR, param_grid = parameter)
grid.fit(x, y)
print(grid)
```



jupyter housing Last Checkpoint: 7 minutes ago (autosaved)

```
********************** GradientBoostingRegressor **********************


GradientBoostingRegressor(max_features='auto', min_samples_leaf=4,
                          n_estimators=30, random_state=79)


Cross_Val_Score =  0.6967939355303377


error:
r2 score is:  0.7628031668187789


root Mean squared error:  37285.00058132616
```

In [55]:
```
result = pd.DataFrame({'Model': Model, 'Cross_val_score':cvs,'r2_score':r2score,'root_mean_squared_error':rmse})
result
```

Out[55]:

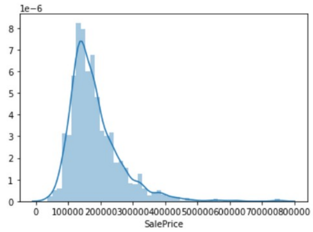
	Model	Cross_val_score	r2_score	root_mean_squared_error
0	RandomForestRegressor	0.730746	0.791750	34935.925265
1	GradientBoostingRegressor	0.696794	0.762803	37285.000581

In [56]:
```
import joblib
```

In [57]:
```
joblib.dump(RFR,'housing.pkl')
```

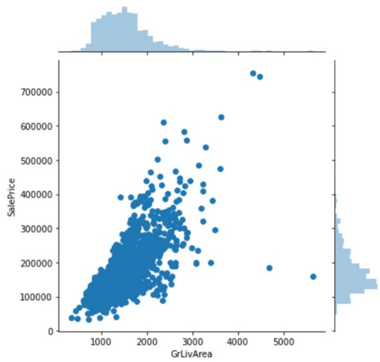
- Visualizations

In [16]:
```
#univarite analysis
sns.distplot(df['SalePrice'])
plt.show()
```



In [17]:
```
sns.countplot(df['YearBuilt'])
plt.show()
```

In [23]:
```
sns.jointplot(x=df['GrLivArea'], y=df['SalePrice'])
plt.show()
```

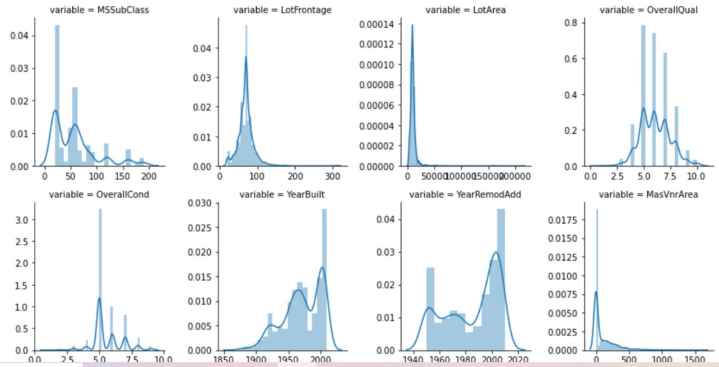


In [24]:
```
sns.barplot(x=df['LotArea'], y=df['SalePrice'])
```

jupyter housing Last Checkpoint: 2 minutes ago (autosaved)

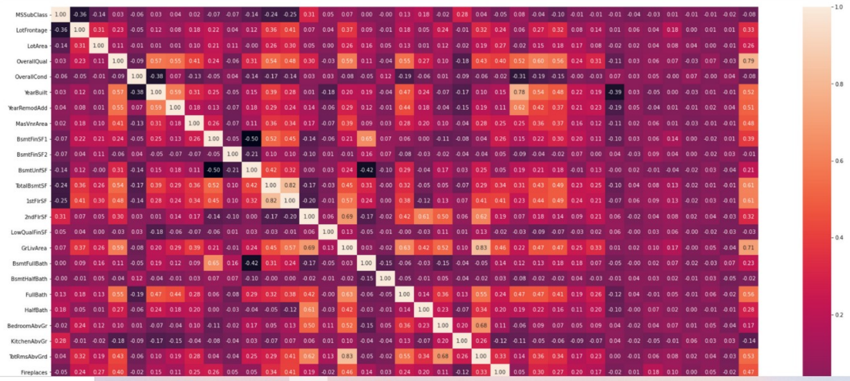
In [28]:
```python
#create numeric plots
nd = pd.melt(df, value_vars = nf)
n1 = sns.FacetGrid (nd, col='variable', col_wrap=4, sharex=False, sharey = False)
n1 = n1.map(sns.distplot, 'value')
n1
```

Out[28]: <seaborn.axisgrid.FacetGrid at 0x1ad42ed9190>



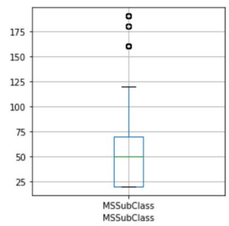
jupyter housing Last Checkpoint: 2 minutes ago (autosaved)

In [30]:
```
plt.figure(figsize=(30,20))
sns.heatmap(df.corr(), annot=True, fmt='.2f')
plt.show()
```

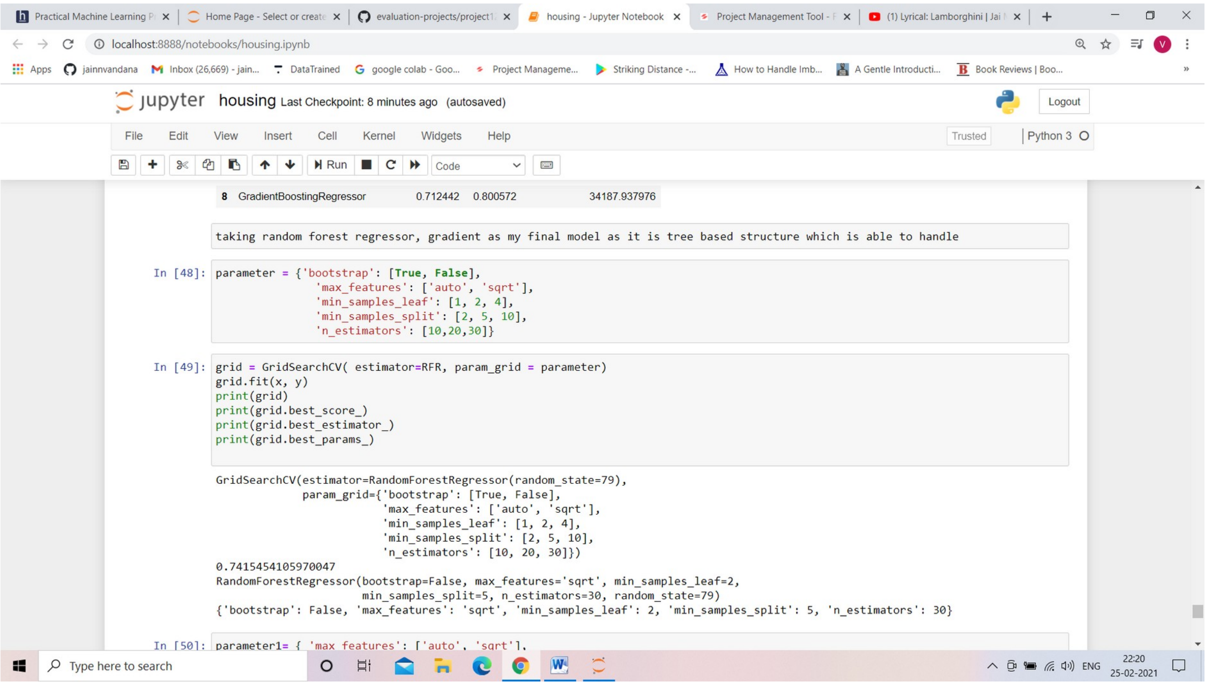


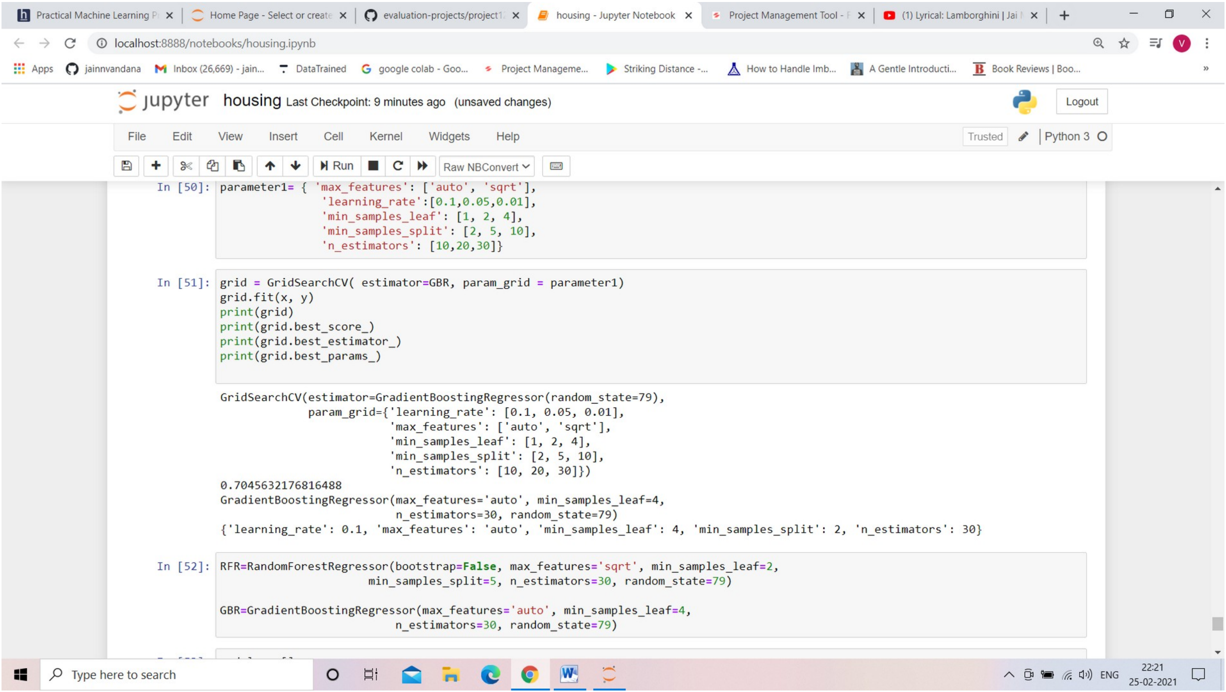
jupyter housing Last Checkpoint: 3 minutes ago (autosaved)

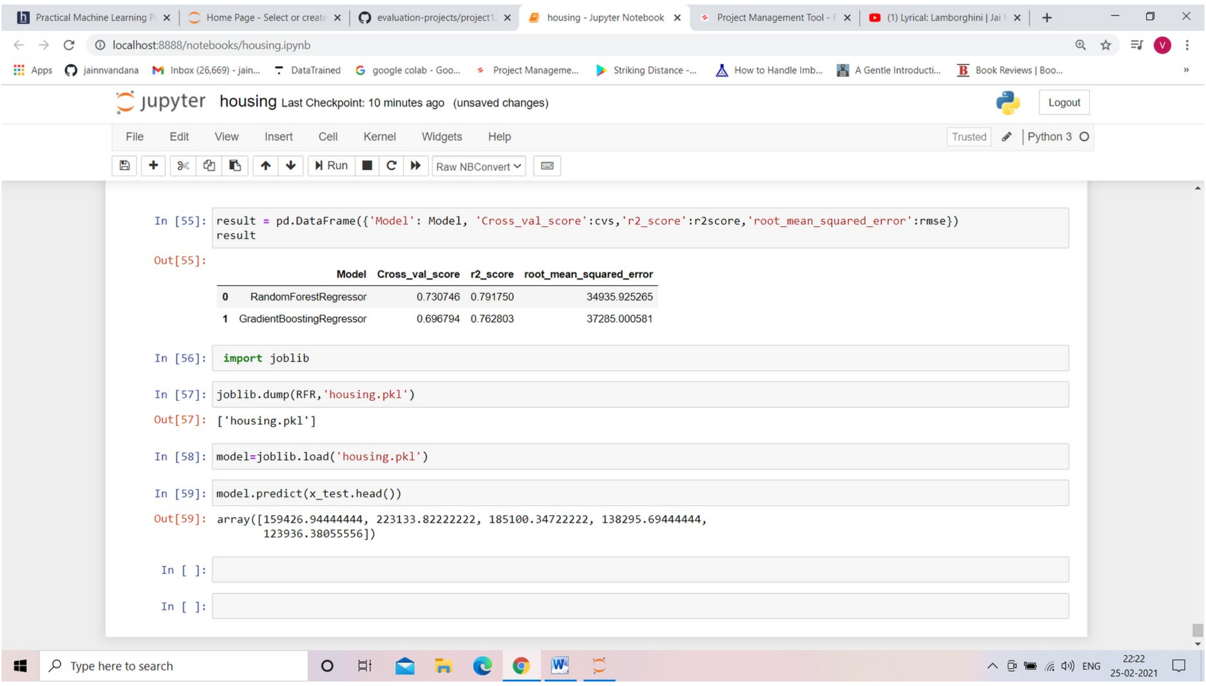
In [33]:
```
for i in df.columns:
        plt.figure(figsize = (4,4))
        df.boxplot(column=[i])
        plt.xlabel(i)
```



- Interpretation of the Results
-







CONCLUSION

Here , we used PCA because it is use to reduce the no of independent features. It is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of “summary indices” that can be more easily visualized and analyzed. And using standard scaller with it provides the best dataset for modelling.

We can see that all the regression models performed but Random forest Regressor and Gradient boosting Regressor performed well amongst them.

After hypertunning these two regressor we can analysis that best performer is Random Forest Regressor .