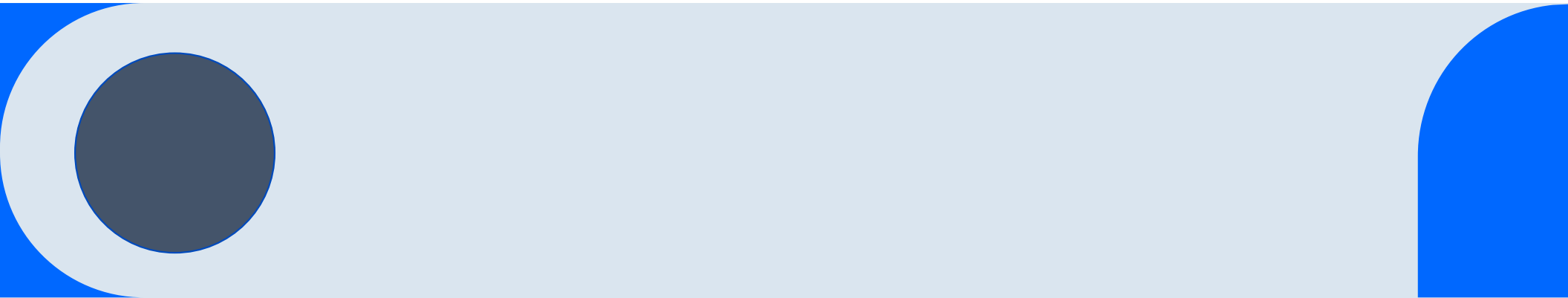




Surprise Housing – Advanced Regression assignment

Member Name - Pamela Roy

Batch – MLC51 EPGP ML&AI



Agenda

- Problem Statement
- Reading and Understanding the data
- Data preparation for Modelling
- Building , Training and Testing Models
- Prediction and evaluation on different models

Problem Statement

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 on at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia.

The company is looking at prospective properties to buy to enter the market.

The company wants to know:

- Which variables are significant in predicting the price of a house, and
- How well those variables describe the price of a house

- **Reading and Understanding the Data**

Data files provided – train.csv , data_description.txt



Reading and Understanding the Data

On analysing train.csv file below are the observations :

- The data file 'train.csv' is uploaded in G-drive & code is written in google.colab to read the file
- On checking the detailed file information we find :
 - Total number of rows - 1460
 - Total number of columns - 81
 - Data types of the columns : 35 columns int64 , 43 columns object , 3 columns float64
 - Columns having very few records maintained : 'Alley' , 'FireplaceQc' , 'PoolQc' , 'Fence' , 'MiscFeatures' . These columns were dropped as they won't add much value in model building process
 - Columns mentioned do not have all records maintained against them (few missing) and hence has been identified as possible candidates for data imputation .
COLs : MasVnrType , MasVnrArea , BsmtQual, BsmtCond , BsmtExposure , BsmtFinType1 , BsmtFinType2 , Electrical , GarageType , GarageYrBlt , GarageFinish , GarageQual , GarageCond

- **Data preparation for Modelling**



Data preparation for Modelling

- Check for duplicate data is done . NO duplicates were found
- Below columns were dropped because of reasons provided below , reducing the no.columns to 64

Dropping ID as it do not add any information

Only 91 rows have data for Alley - hence dropping

Only 770 rows have data for FireplaceQu - hence dropping

Only 7 records avialbel for PoolQC - hence dropping

Only 54 records available for MiscFeature - hence dropping

Only 281 records available Fence - Hence Dropping

MiscVal col has 1408 records with val 0 out of 1460 recs - hence dropping

PoolArea col has 1453 records as 0 - hence dropping

ScreenPorch has 1344 records as 0 - hence dropping

3SsnPorch has 1463 records as 0 - hence dropping

EnclosedPorch has 1200+ records with val 0 - hence dropping

BsmtFinSF2 has almost 1300 records with val 0 - Hence dropping

LowQualFinSF has almost 1300 recs with val 0 - hence dropping

BsmtHalfBath has 1200+ records woth val 0 - hence dropping

Street have 1400+ records with value 'Pave' - hence dropping

Utilities Column has all rows with value AllPub - Hence Dropping

CentralAir colm has around 80 rows with value 'N' - Hence Dropping

- Data imputing was done for the below columns :

Replace NA values with 0 for continuous variables LotFrortage , MasVnrArea , GarageYrBlt

Data preparation for Modelling

Data preparation continued... :

- Non Binary Categorical variables were converted to original values for below columns in order to help in model building :

-MSSubClass , MSZoning , OverallQual , OverallCond , MoSold , LandSlope , Functional

- Checking the statistical information for the 25 continuous variables below is what we get :
This shows the value / ranges for the values are very big or small . Later scaling to be performed in order to represent all data in 0& 1 range .

	LotFrontage	LotArea	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtUnfSF	TotalBsmtSF	1stFlrSF	2ndFlrSF	...	Kitchen/
count	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	...	1460.00
mean	57.623288	10516.828082	1971.267808	1984.865753	103.117123	443.639726	567.240411	1057.429452	1162.626712	346.992466	...	1.00
std	34.664304	9981.264932	30.202904	20.645407	180.731373	456.098091	441.866955	438.705324	386.587738	436.528436	...	0.20
min	0.000000	1300.000000	1872.000000	1950.000000	0.000000	0.000000	0.000000	0.000000	334.000000	0.000000	...	0.00
25%	42.000000	7553.500000	1954.000000	1967.000000	0.000000	0.000000	223.000000	795.750000	882.000000	0.000000	...	1.00
50%	63.000000	9478.500000	1973.000000	1994.000000	0.000000	383.500000	477.500000	991.500000	1087.000000	0.000000	...	1.00
75%	79.000000	11601.500000	2000.000000	2004.000000	164.250000	712.250000	808.000000	1298.250000	1391.250000	728.000000	...	1.00
max	313.000000	215245.000000	2010.000000	2010.000000	1600.000000	5644.000000	2336.000000	6110.000000	4692.000000	2065.000000	...	3.00

Data preparation for Modelling

Data preparation continued... :

- For all categorical variables dummies were added in order to turn the values into 0 & 1 (Binary)
Like : for MSSubClass , MonthSold , MSZoning , LotShape , LandContour , LotConfig etc

- After adding the dummy variables the total number of columns increased to 258
- The data types for all the variables now got changed to continuous variables with data type either float64 , int64 or uint8 .

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Columns: 258 entries, LotFrontage to Wall
dtypes: float64(3), int64(22), uint8(233)
memory usage: 617.5 KB
```

- The data set is then split into TRAIN & TEST sets in the ration of 70:30 .
 - 70 % of the data provided will be used for training the model

Records in training set – 1021

- 30% of the data provided will be used for evaluation & test of the model

Records in testing set – 439

- Rescaling of the continuous variables done on order to change all data values in a range of 0 & 1

	LotFrontage	LotArea	YearBuilt	YearRemodAdd	HasVnrArea	BsmtFinSF1	BsmtUnfSF	TotalBsmtSF	1stFlrSF	2ndFlrSF
count	1021.000000	1021.000000	1021.000000	1021.000000	1021.000000	1021.000000	1021.000000	1021.000000	1021.000000	1021.000000
mean	0.184049	0.042162	0.719719	0.583072	0.065306	0.079406	0.241329	0.173813	0.184401	0.000000
std	0.109960	0.048221	0.219718	0.343416	0.117088	0.082409	0.192097	0.075139	0.092106	0.000000
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.137380	0.027937	0.594203	0.283333	0.000000	0.000000	0.092466	0.129787	0.116667	0.000000
50%	0.191693	0.037555	0.731884	0.733333	0.000000	0.069454	0.197774	0.162684	0.165278	0.000000
75%	0.249201	0.048943	0.927536	0.900000	0.098750	0.126152	0.345034	0.215057	0.243056	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

- **Building and Training
and Testing the Model**



Building & Training the Model

LM Summary

LINEAR REGRESSION MODEL (With RFE)

- Using Linear Regression model the X_train and y_train dataset is fit
- Using RFE** - Recursive feature Elimination method we bring down the number of columns from 258 to **35 columns**
- The RFE model ranks the variables from 1 to 35
- The **selected predictor variables** are :
 - 'LotArea', 'YearBuilt', 'BsmtFinSF1', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GrLivArea', 'BedroomAbvGr', 'KitchenAbvGr', 'GarageArea', 'SevereSlope', 'PosA', 'PosN', 'RR Ae', 'Excellent', 'Very Excellent', 'Very Good', 'Excellent', 'Fair', 'Shed', 'CompShg', 'Membran', 'Metal', 'Roll', 'Tar&Grv', 'WdShake', 'WdShngl', 'CBlock', 'Stone', 'Wood', 'Po', 'Severely Damaged', 'Con', 'Partial', 'OthW', 'Stone', 'Po', 'Mix', 'Fa', 'Gd', 'Po', 'TA', 'Fa', 'Gd', 'Po', 'TA', 'Partial', 'GasA'
- Add a constant to the model using add_constant method from StatsModels library
- Then we run the Linear regression using the OLS method from same library on the TRAIN dataset
- Next we check the summary :
 - The R2 value is .918**
 - Coefficients of the 35 variables are given
 - For all the 35 variables the P value are 0 or very near to 0

OLS Regression Results						
=====						
Dep. Variable:	SalePrice	R-squared:	0.918			
Model:	OLS	Adj. R-squared:	0.914			
Method:	Least Squares	F-statistic:	232.7			
Date:	Wed, 02 Aug 2023	Prob (F-statistic):	0.00			
Time:	05:34:12	Log-Likelihood:	2085.5			
No. Observations:	1021	AIC:	-4075.			
Df Residuals:	973	BIC:	-3838.			
Df Model:	47					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

const	-1.1687	0.038	-30.539	0.000	-1.244	-1.094
LotArea	0.2418	0.029	8.195	0.000	0.184	0.300
YearBuilt	0.0850	0.006	13.535	0.000	0.073	0.097
BsmtFinSF1	0.1802	0.016	11.387	0.000	0.149	0.211
TotalBsmtSF	0.1842	0.028	6.584	0.000	0.129	0.239
1stFlrSF	0.4822	0.096	5.014	0.000	0.293	0.671
2ndFlrSF	0.2285	0.045	5.034	0.000	0.139	0.318
GrLivArea	-0.0297	0.113	-0.263	0.793	-0.251	0.192
BedroomAbvGr	-0.0525	0.013	-3.955	0.000	-0.078	-0.026
KitchenAbvGr	-0.1009	0.010	-10.018	0.000	-0.121	-0.081
GarageArea	0.0358	0.009	3.853	0.000	0.018	0.054
SevereSlope	-0.0700	0.017	-4.053	0.000	-0.104	-0.036

Testing the Model

LINEAR REGRESSION MODEL .. continued

- Using Linear Regression model the X_test and y_test dataset is fit
- Using RFE - Recursive feature Elimination method we bring down the number of columns from 258 to 35 columns
- The RFE model ranks the variables from 1 to 35
- The **selected predictor variables** are :
'LotArea', 'BsmtFinSF1', '1stFlrSF', 'GrLivArea',
'2 FAMILY CONVERSION - ALL STYLES AND AGES', '2-1/2 STORY ALL AGES',
'2-STORY 1946 & NEWER', 'DUPLEX - ALL STYLES AND AGES', 'SPLIT FOYER',
'FR3', 'SevereSlope', 'NridgHt', 'StoneBr', 'PosA', 'Duplex', 'Excellent', 'Poor',
'Mansard', 'Tar&Grv', 'WdShake', 'WdShngl', 'Other', 'Stone', 'Po', 'Mix', 'Fa', 'Gd',
'Po', 'TA', 'Fa', 'Gd', 'Po', 'TA', 'Partial', 'GasA'
- Add a constant to the model using add_constant method from StatsModels library
- Then we run the Linear regression using the OLS method from same library on the TRAIN dataset
- Next we check the summary :
 - The R2 value is .903
 - Coefficients of the 35 variables are given
 - For all the 35 variables the P value are 0 or very near to 0

LM Summary

OLS Regression Results

Dep. Variable:	SalePrice	R-squared:	0.903
Model:	OLS	Adj. R-squared:	0.889
Method:	Least Squares	F-statistic:	66.10
Date:	Wed, 02 Aug 2023	Prob (F-statistic):	8.73e-163
Time:	05:34:14	Log-Likelihood:	752.56
No. Observations:	439	AIC:	-1395.
Df Residuals:	384	BIC:	-1170.
Df Model:	54		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	0.1058	0.060	1.755	0.080	-0.013	0.224
LotArea	0.3105	0.057	5.406	0.000	0.198	0.423
BsmtFinSF1	0.0803	0.014	5.685	0.000	0.053	0.108
1stFlrSF	0.1273	0.026	4.951	0.000	0.077	0.178
GrLivArea	0.2862	0.027	10.471	0.000	0.232	0.340
2 FAMILY CONVERSION - ALL STYLES AND AGES	-0.0250	0.023	-1.111	0.267	-0.069	0.019
2-1/2 STORY ALL AGES	-0.0594	0.023	-2.547	0.011	-0.105	-0.014
2-STORY 1946 & NEWER	0.0337	0.009	3.745	0.000	0.016	0.051
DUPLEX - ALL STYLES AND AGES	-0.0155	0.008	-1.970	0.050	-0.031	-2.98e-05
SPLIT FOYER	0.0066	0.022	0.298	0.766	-0.037	0.050
FR3	-0.1015	0.052	-1.957	0.051	-0.204	0.000
SevereSlope	-0.2023	0.039	-5.155	0.000	-0.279	-0.125

Building , Training and testing the Model

Linear regression Model (Without RFE)

- Running Simple linear Regression for both train & test sets using all the 257 predictor variables we get the below metric scores – which indicates **OverFitting**

R2 Score on Train Set	0.9552118053411991
R2 Score on Test Set	0.9590772069514153
RSS on Train Set	0.5512200041630403
RSS on Test Set	0.351268485850443
MSE Score on Train Set	0.0005398824722458768
MSE Score on Test Set	0.0008001560042151322
RMSE on Train Set	0.02323537114499953
RMSE on Test Set	0.028287028903989406

Building , Training and testing the Model

RIDGE regression Model

(Performed on the 35 variables chosen by RFE)

- **20 hyperparameters** used for tuning in RIDGE regression
Lambda (alpha in Python code) :
0.0001 , 0.005 , 0.001, 0.05 , 0.01, 0.1, 0.2, 0.3 , 0.4 , 0.5, 0.6 , 0.7 , 0.8 , 0.9 , 1.0,10.0, 50, 100, 500, 1000
- **Cross validation** done using GridSearchCV from sklearn library with **5 folds**
- 20 x 5 = 100 fits were done on the model
- Best **Lambda value** returned by method best_params_ = **0.05**
- With optimum Lambda value of .05 when the model is run on both Train & Test sets the R2 scores for the are as shown in right

Metrics using RIDGE - Summary

R2 Score on Train Set

0.917572876742081

R2 score on Test Set

0.9028261696702273

Building , Training and testing the Model

LASSO regression Model

(Performed on the 35 variables chosen by RFE)

- **20 hyperparameters** used for tuning in LASSO regression
Lambda (alpha in Python code) :
0.0001 , 0.005 , 0.001, 0.05 , 0.01, 0.1, 0.2, 0.3 , 0.4 , 0.5, 0.6 , 0.7 , 0.8 , 0.9 , 1.0,10.0, 50, 100, 500, 1000
- **Cross validation** done using GridSearchCV from sklearn library with **5 folds**
- 20 x 5 = 100 fits were done on the model
- Best **Lambda value** returned by method best_params_ = **0.0001**
- With optimum Lambda value of 0.0001 when the model is run on both Train & Test sets the R2 scores for the are as shown in right

Metrics using Lasso - Summary

R2 Score on Train Set

0.8432992239239697

R2 score on Test Set

0.8937517201666982

**Prediction and evaluation
on test data**



Prediction and evaluation on test data

- Comparing the 3 models built we get the below metrics .

	Metric	Linear Regression	Ridge Regression	Lasso Regression
0	<u>R2 Score(Train)</u>	0.955212	0.908791	0.843299
1	<u>R2 Score(Test)</u>	0.959077	0.902188	0.893752
2	RSS(Train)	0.551220	1.122535	1.928557
3	RSS(Test)	0.351268	0.839590	0.912002
4	MSE(Train)	0.000540	0.001099	0.001889
5	MSE(Test)	0.000800	0.001913	0.002077
6	RMSE(Train)	0.023235	0.033158	0.043461
7	RMSE(Test)	0.028287	0.043732	0.045579

- The **Linear regression** model gives R2 values on train & test set as 95.5% & 95.9%
 - This is an indication of overfitting
- On Performing **Ridge Regression** the R2 values are reduced to 90.87 %& 90.21% for train & test set respectively , but are still doing pretty well for both Train & test sets
 - **Lambda value used is 0.05**
- On performing Lasso Regression the R2 values are further reduced to 84.3 & 89.3% for train & test set respectively
 - **Lambda value used is 0.0001**
- Comparing the 3 models built we get the below metrics .
 - **Ridge model** appears to be the best fit .



Thank you

Pamela Roy

Email : bhattacharya.pam@gmail.com

