FLIP ROBO

HOUSING PROJECT

Submitted by:

Praveen Kumar Singh

ACKNOWLEDGMENT

I would like to thank Shubham Yadav, who's the guide for the internship phase at FLIP ROBO, and the Dr. Deepika Sharma from DataTrained who conducted the training in the institute.

References:

https://stackoverflow.com/

https://medium.com/analytics-vidhya/

https://www.youtube.com/user/krishnaik06/

INTRODUCTION

Description

Houses are one of the necessary needs 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.

Dataset

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.

Conceptual Background

Knowledge of real-life real-estate companies would help to better understand the different variables given in the dataset. It will in turn help us to leverage the correlation of the variables.

Review of literature

I did research https://www.magicbricks.com/ to find the prospect factors that would influence the prices of houses in an area. Even though the filters available in the commercial website it limited, we can definitely get an insight on how different factors affect the pricing. The important ones I noticed was the square ft area, and the neighbourhood. There might also be differences that must be taken into account when checking houses in the US and in India, but the differences are evident in the filters.

From the business perspective, the info is limited apart from the filters the website provides. There are no other detailed insights given by any competitor platforms.

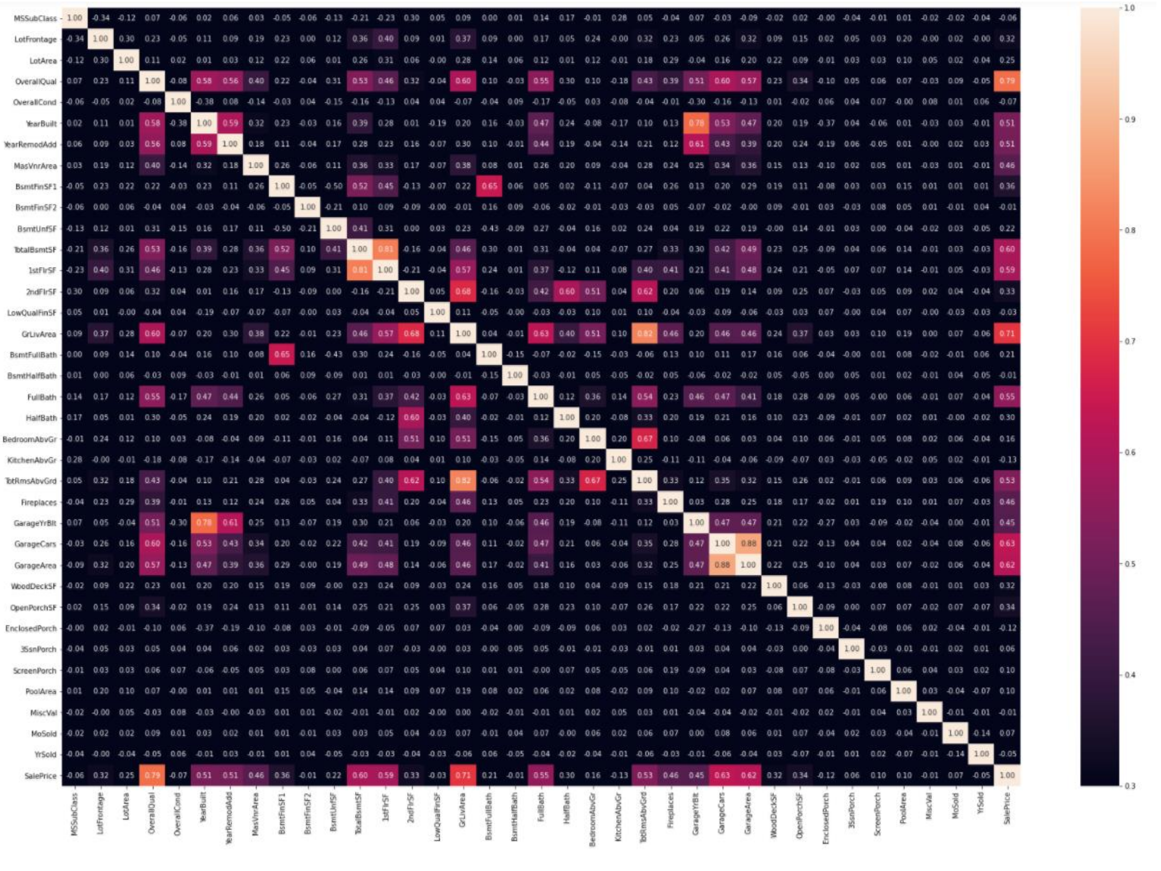
Motive

The motive behind this project is to create a ML model that predicts with the given conditions the price of a house up for sale.

ANALYTICAL PROBLEM FRAMING

Getting Started

- Reviewing the dataset, I was able to find a couple of columns which doesn’t contribute any significate data first hand. Those were removed in the initial stage of the pre-processing. Those columns were ['Id', 'Alley', 'FireplaceQu', 'PoolQC', 'Fence','MiscFeature'].
- Started by separating the columns based on the datatype as to easily do visualization on them and extrapolate from the visual data.
- I was able to find that the SalePrice had direct relation with ('YearBuilt', 'YearRemodAdd') columns. This was evident from the lineplots.
- Further reviewed the relationship between the output variable SalePrice compared with the different object, numerical, categorical and non-categorical values.
- Pie charts were plotted for different categorical columns to understand the distribution of variables in different columns. The extrapolations made were noted down, and the distribution of the prices were plotted to find the average price.
- Plotted count-plots to visualize the categorical columns.
- Checked the outliers in the float dtype columns and choose to go with median to replace the null values in them. For the object dtype columns, I choose mode.
- Once the null values had been handled, I plotted the summary and correlation plots to filter out the columns which contribute less to the model building process. PFB the correlation plot below.



- Since this would be a regression model, there wasn’t a need to balance the dataset. With regards to the correlation plot, the columns listed as following were removed. ['MSSubClass', 'LotArea', 'OverallCond', 'BsmtFinSF2', 'LowQualFinSF', 'BsmtHalfBath', 'KitchenAbvGr', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', 'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'SaleType', 'SaleCondition']
- I checked for outliers in the numerical columns ['BsmtFinSF1', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'GarageArea', 'SalePrice', 'LotFrontage', 'GarageYrBlt'], post which I checked for ZScore and removed the outliers which accounted for 5.47% of the total dataset.
- I plotted distplots to check the skewness and use the PowerTransformer (Yeo-Johnson method) to remove the skewness from the columns listed.
- I encoded the list of objects dtype columns using LabelEncoder, and scaled the dataset using the StandardScaler. With a bit of confusion on which comes first, I choose to encode the object variables before scaling the dataset.

Hardware and Software Requirements and Tools Used

- Intel i7 – 10705H
- 8 GB DDR4
- 1TB SSD
- Anaconda – Jupyter Notebook
- Chrome

MODEL(S) DEVELOPMENT AND EVALUATION

Possible Problem-Solving Approach

The output variable was continuous in nature, so the problem had to be modelled using the Regression method.

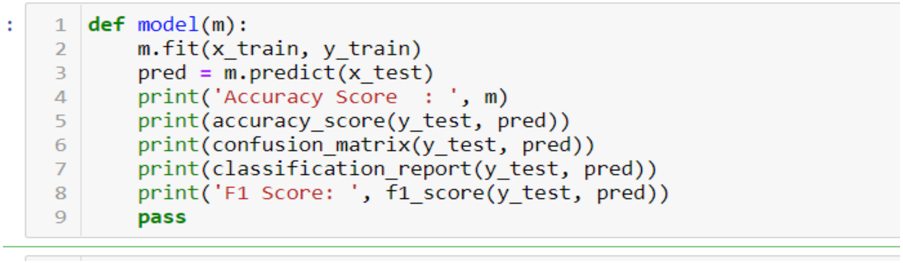
Testing of Identified Approaches (Algorithms)

Models:

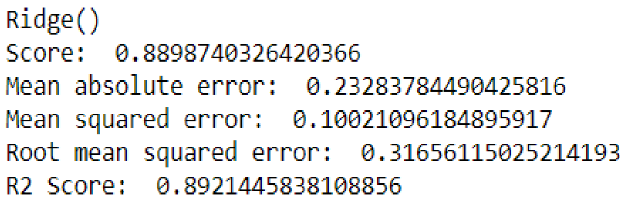
- Linear Regression
- K-Neighbors Regressor
- Decision Tree Regressor
- SVR
- Random Forest Regressor
- Lasso
- Ridge
- Elastic Net

Run and Evaluate selected models

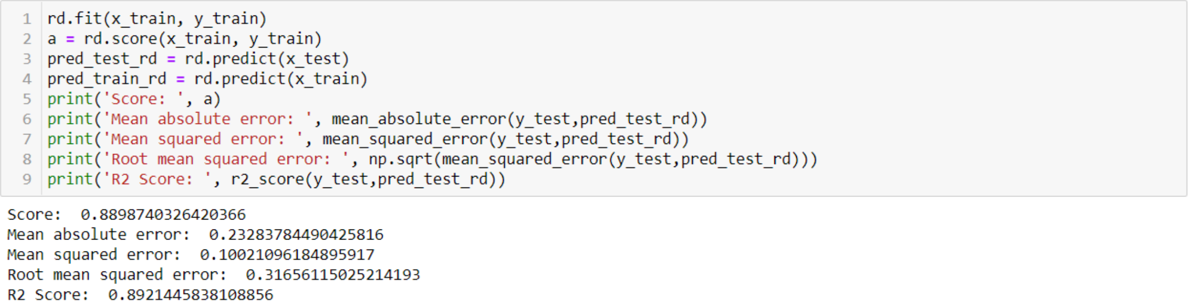
After running all the models, Ridge turned out to be the best model.



```python
def model(m):
    m.fit(x_train, y_train)
    pred = m.predict(x_test)
    print('Accuracy Score  : ', m)
    print(accuracy_score(y_test, pred))
    print(confusion_matrix(y_test, pred))
    print(classification_report(y_test, pred))
    print('F1 Score: ', f1_score(y_test, pred))
    pass
```



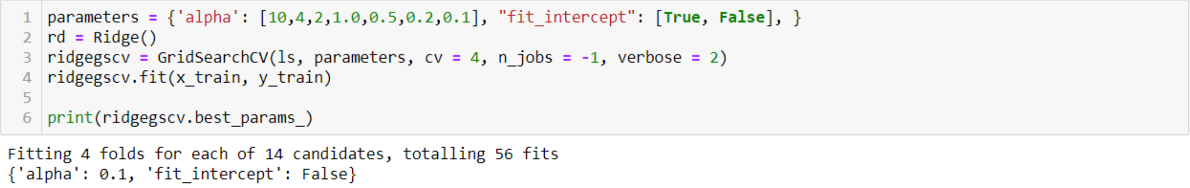
```
Ridge()
Score:  0.8898740326420366
Mean absolute error:  0.23283784490425816
Mean squared error:  0.10021096184895917
Root mean squared error:  0.31656115025214193
R2 Score:  0.8921445838108856
```



```
rd.fit(x_train, y_train)
a = rd.score(x_train, y_train)
pred_test_rd = rd.predict(x_test)
pred_train_rd = rd.predict(x_train)
print('Score: ', a)
print('Mean absolute error: ', mean_absolute_error(y_test,pred_test_rd))
print('Mean squared error: ', mean_squared_error(y_test,pred_test_rd))
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test,pred_test_rd)))
print('R2 Score: ', r2_score(y_test,pred_test_rd))
```

```
Score:  0.8898740326420366
Mean absolute error:  0.23283784490425816
Mean squared error:  0.10021096184895917
Root mean squared error:  0.31656115025214193
R2 Score:  0.8921445838108856
```

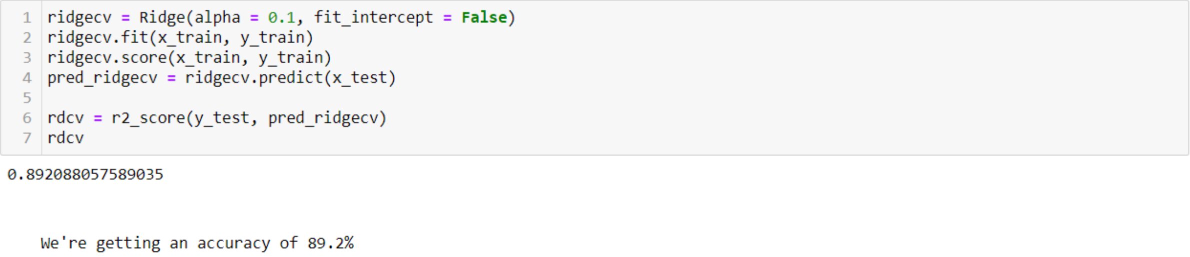
Running the Cross validation with the parameters set as shown below, and re-running the model with the best parameters, we get a 89.2% accuracy.



```
parameters = {'alpha': [10,4,2,1.0,0.5,0.2,0.1], "fit_intercept": [True, False], }
rd = Ridge()
ridgegscv = GridSearchCV(ls, parameters, cv = 4, n_jobs = -1, verbose = 2)
ridgegscv.fit(x_train, y_train)

print(ridgegscv.best_params_)
```

```
Fitting 4 folds for each of 14 candidates, totalling 56 fits
{'alpha': 0.1, 'fit_intercept': False}
```



```
ridgecv = Ridge(alpha = 0.1, fit_intercept = False)
ridgecv.fit(x_train, y_train)
ridgecv.score(x_train, y_train)
pred_ridgecv = ridgecv.predict(x_test)

rdcv = r2_score(y_test, pred_ridgecv)
rdcv
```

```
0.892088057589035
```

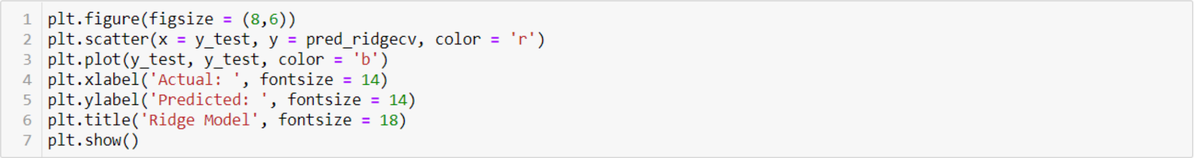
```
We're getting an accuracy of 89.2%
```

Key Metrics for success in solving problem under consideration

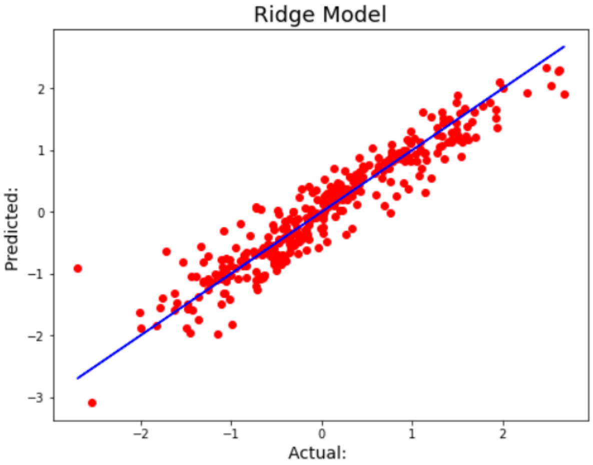
- alpha = 0.1
- fit_intercept = False

Visualizations

The plot below shows the code and the fitting of the model. We can conclude that the model is not over/under fit, and works as it says it does.



```python
plt.figure(figsize = (8,6))
plt.scatter(x = y_test, y = pred_ridgecv, color = 'r')
plt.plot(y_test, y_test, color = 'b')
plt.xlabel('Actual: ', fontsize = 14)
plt.ylabel('Predicted: ', fontsize = 14)
plt.title('Ridge Model', fontsize = 18)
plt.show()
```



Interpretation of the Results

We’re getting a model with about 89.2% accuracy post Cross Validation and Grid Search CV with all listed parameters.

CONCLUSION

Key findings and Conclusion

Both Lasso and Ridge models gave a good R2 Score in the preliminary run, but on further inspection, Ridge turned out to be the best working model. On further running with the Cross validation, we were able to find that the difference between R2 Score and CV score were less for all the models, however, I went with ridge taking into consideration the low difference in values for all the models and the comparatively higher score for the Ridge Model.

Learning Outcomes of the Study in respect of Data Science

I have created a model which gives me a good accuracy in terms of predicting the prospect price of a house with the parameters listed.

Limitations of this work and Scope for Future Work

I feel that the libraries which we run are limited in a way that they don't have an option for parallel processing.

Some libraries like TensorFlow-GPU, H2O4GPU (Scikit replacement) etc utilizes the GPU which has more cores and help run the models a lot faster while not compromising on the accuracy.

If the common Classification models that we use come up with a replacement that utilizes the GPU, we'll be more comfortable running those models on personal laptops.