CT4103 Semester 2

Benjamin butcher – s4005578

Melbourne housing

Analysis on Melbourne housing Data using 3 different learning models and correlation graphs

Scenario:

To report on the similarities and condition for pricing in Melbourne housing

Data:

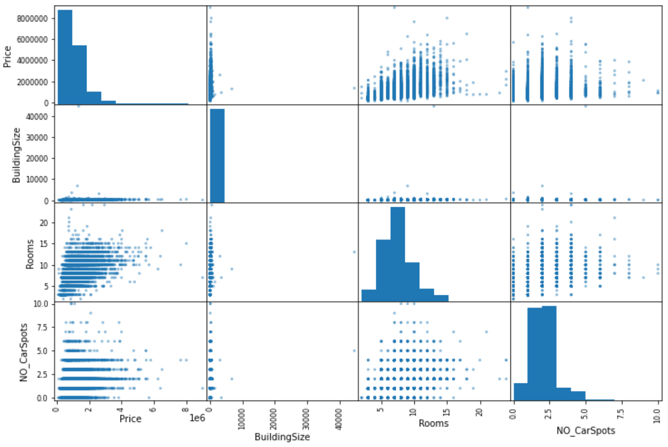
We took this data from a Kaggle dataset called Melbourne housing data which has total of 21 columns which are

- Rooms: Number of rooms
- Price: Price in dollars
- Method: S - property sold; SP - property sold prior; PI - property passed in; PN - sold prior not disclosed; SN - sold not disclosed; NB - no bid; VB - vendor bid; W - withdrawn prior to auction; SA - sold after auction; SS - sold after auction price not disclosed. N/A - price or highest bid not available.
- Type: br - bedroom(s); h - house,cottage,villa, semi,terrace; u - unit, duplex; t - townhouse; dev site - development site; o res - other residential.
- SellerG: Real Estate Agent
- Date: Date sold
- Distance: Distance from CBD
- Regionname: General Region (West, North West, North, North east …etc)
- Propertycount: Number of properties that exist in the suburb.
- Bedroom2 : Scraped # of Bedrooms (from different source)
- Bathroom: Number of Bathrooms
- Car: Number of carspots
- Landsize: Land Size
- BuildingArea: Building Size
- CouncilArea: Governing council for the area

First thing we see from the data is that rooms is individual to bedrooms and bathrooms so I took the liberty and put them all together so rooms becomes the total number of rooms in the building.

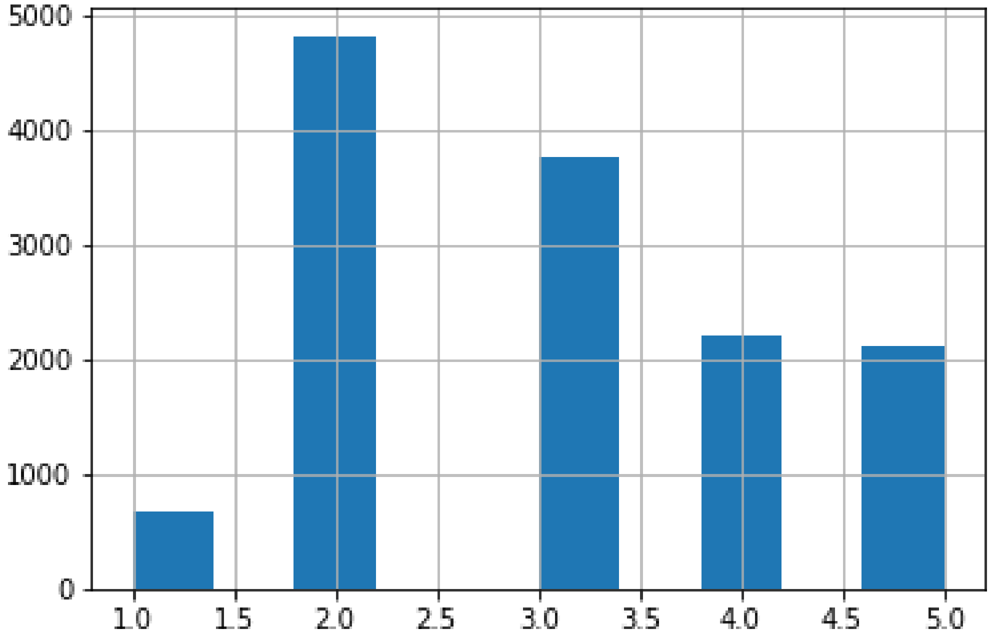
Secondly we see what is correlating and what isn't so we can find like terms and put them together we also want to rename some of our data to make it clearer to what it is

This is a Scatter Graph of the price, building size, rooms and car spots all put together so we can see if any of the have a correlations. First in this scatter graph we can see that there is a slight positive correlation between the building size and price, car spots and price and a tad more normal positive correlations between bedrooms and price.

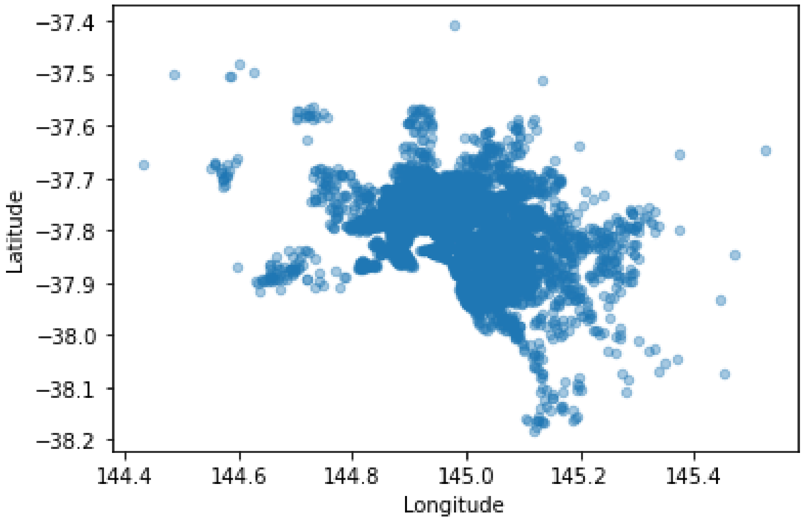


We can also see corrolations between rooms and landsize, parking and landsize and finally bedrooms per landsize. Because of this we can then combine them to create like terms which we can use for our data.

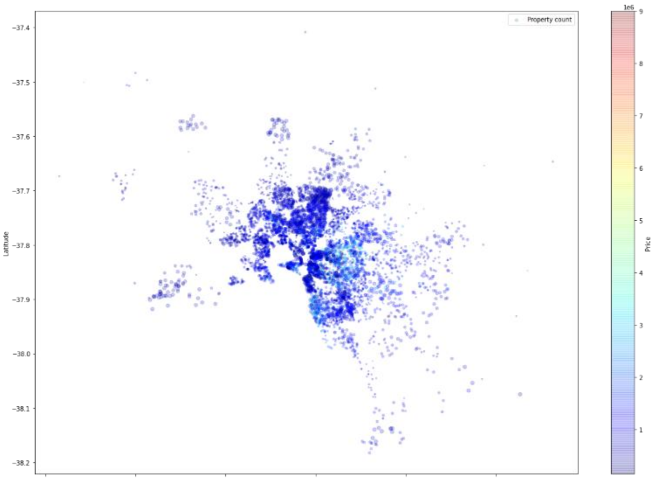
Next we can see from this histogram of the price categories that the majority of properties are in a price range of 0.8 to 1.2mill and that the mean of the prices are around 1.07 million



We can expand apon this using a scatter graph of the longitude and latitude to see the layout of properties on the Melbourne map



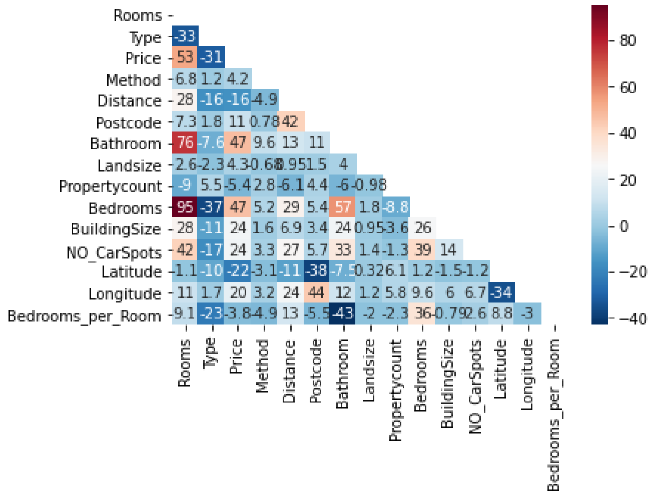
And using further techniques we can see the pricey areas using another scatter graph



Sunbury
M31
M79
Melton
M8
Yarra Valley
Lilydale
M3
Melbourne
Werribee
M1
Port Phillip
Dandenong
M1
Geelong
Frankston
M11
Portsea
French Island
Mornington Peninsula
Phillip Island
km 10 20 30
mi 10 20
Bass Strait

Which closely relates to the most expensive properties being in the centre of Melbourne and around the coast. We can also see from this that the properties near the airports are cheaper most likely due to noise pollution.

Another method of finding correlations of data is using a heat map which I have done here



| | Rooms | Type | Price | Method | Distance | Postcode | Bathroom | Landsize | Propertycount | Bedrooms | BuildingSize | NO_CarSpots | Latitude | Longitude | Bedrooms_per_Room |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rooms | | | | | | | | | | | | | | | |
| Type | -33 | | | | | | | | | | | | | | |
| Price | 53 | -31 | | | | | | | | | | | | | |
| Method | 6.8 | 1.2 | 4.2 | | | | | | | | | | | | |
| Distance | 28 | -16 | -16 | -4.9 | | | | | | | | | | | |
| Postcode | 7.3 | 1.8 | 11 | 0.78 | 42 | | | | | | | | | | |
| Bathroom | 76 | -7.6 | 47 | 9.6 | 13 | 11 | | | | | | | | | |
| Landsize | 2.6 | -2.3 | 4.3 | -0.68 | 0.95 | 1.5 | 4 | | | | | | | | |
| Propertycount | -9 | 5.5 | -5.4 | 2.8 | -6.1 | 4.4 | -6 | -0.98 | | | | | | | |
| Bedrooms | 95 | -37 | 47 | 5.2 | 29 | 5.4 | 57 | 1.8 | -8.8 | | | | | | |
| BuildingSize | 28 | -11 | 24 | 1.6 | 6.9 | 3.4 | 24 | 0.95 | -3.6 | 26 | | | | | |
| NO_CarSpots | 42 | -17 | 24 | 3.3 | 27 | 5.7 | 33 | 1.4 | -1.3 | 39 | 14 | | | | |
| Latitude | 1.1 | -10 | -22 | -3.1 | -11 | -38 | -7.5 | 0.32 | 6.1 | 1.2 | -1.5 | -1.2 | | | |
| Longitude | 11 | 1.7 | 20 | 3.2 | 24 | 44 | 12 | 1.2 | 5.8 | 9.6 | 6 | 6.7 | -34 | | |
| Bedrooms_per_Room | 9.1 | -23 | -3.8 | -4.9 | 13 | -5.5 | -43 | -2 | -2.3 | 36 | -0.79 | 2.6 | 8.8 | -3 | |

From this we can see that the more bathrooms we have the more bedrooms we have due to more citizens living there. We can also see that the lower the year build the lower the price due to the building not being as stable and needing constant rebuilding due to its withering state especially the extremely olden era of building that use outdated building methods.

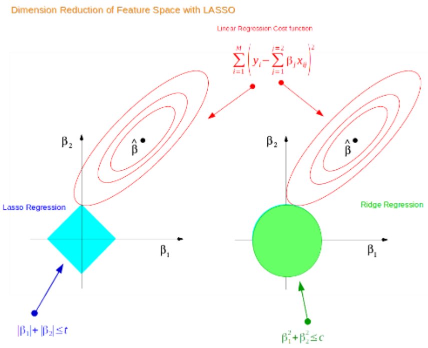
We can also see that the price goes up the more rooms, bedrooms and bathrooms there are most likely due to more building space and land size to fill that space especially in the main parts of Melbourne as in cities space is very limited.

When putting this through an learning model it was hard to decide what model I should use for this data so I decided to use 3 different types.

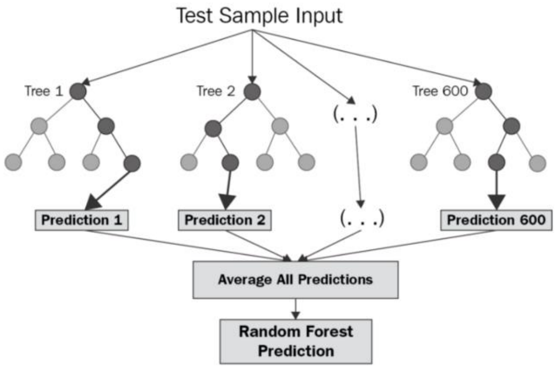
Ridge, lasso and random forest each with their own benefits and downfalls.

In ridge regression we add a penalty term and a coefficient to control that term. In case of zero we add a constraint to the coefficient meaning it can never be absolute zero this is good as it causes the coefficient to tend towards zero but never reach it which leads to a low variance and low bias . problem is these decreases the complexity of the model but doesn't reduce the number of variables since it never has a coefficient of zero but rather minimizes it.

In Lasso regression we add a penalty term to the cost function this term is an absolute of the coefficients and the value of coefficients increases from zero this means we decrease the value of coefficients in order to reduce loss and the difference between this and ridge is that is coefficient can become absolute zero this also means that lasso can struggle with different types of data which we hopefully sorted in our dataset by limiting it to obj, float, int and categories.

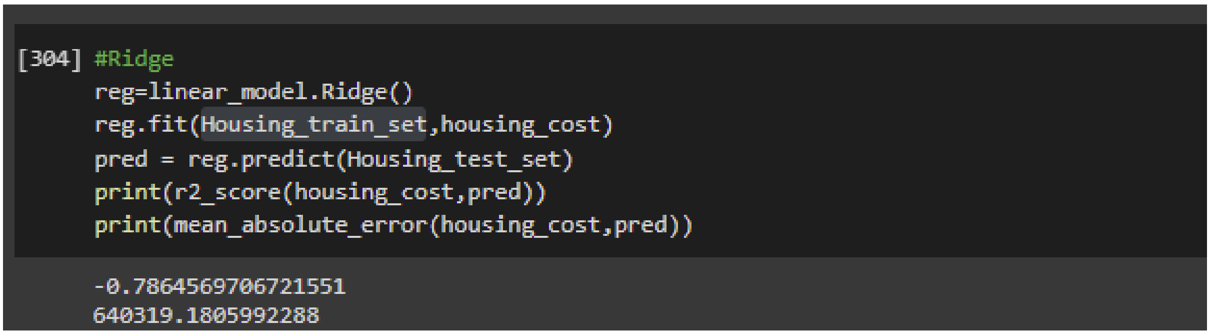

Dimension Reduction of Feature Space with LASSO
Linear Regression Cost function
$\sum_{i=1}^{M}\left(y_i-\sum_{j=1}^{j=2}\beta_j x_{ij}\right)^2$
$\beta_2$
$\hat{\beta}$
$\beta_1$
Lasso Regression
$|\beta_1|+|\beta_2|\leq t$
$\beta_2$
$\hat{\beta}$
$\beta_1$
Ridge Regression
$\beta_1^2+\beta_2^2\leq c$

In random forest we use decision tress where it takes multiple and different decision trees and makes them choose each one. Each tree then need to predict an expected price and base on that the decision criteria is picked after this we calculate the average of all predictions and find a great estimate from averaging them all together. We benefit from this as random forest is extremely accurate and is very good when predicting future cost and prices.



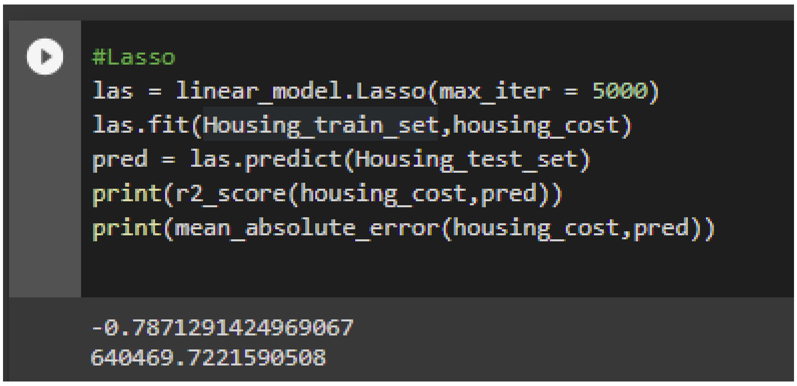
We then use stratified sampling to separate our data into 2 different bits of data a test set and a training set. by divinding a population of out data into smaller groups, known as a strata, and then taking a random number of each row of data from the strata into our final bit of data.

By running this code we get an r2 value and a MSE value where r2 is the proportional variance in the dependent variable that shows how well the data fits the regression model and the MSE is the square of the difference between the actual and estimated values.



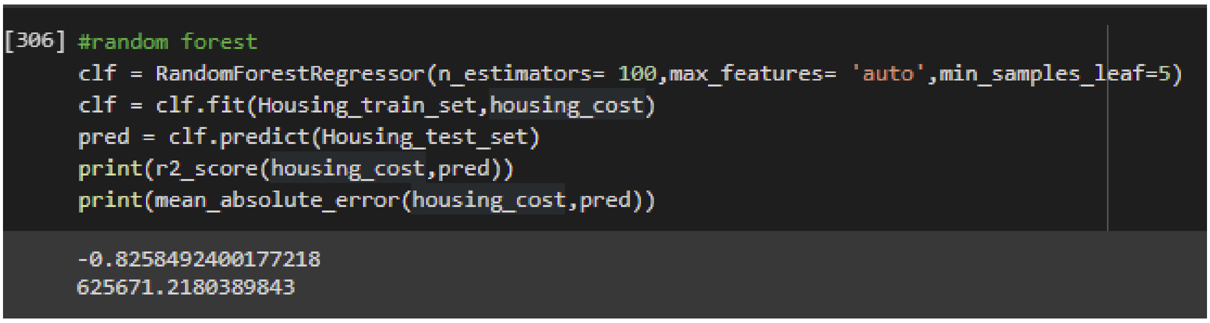
```
[304] #Ridge
      reg=linear_model.Ridge()
      reg.fit(Housing_train_set,housing_cost)
      pred = reg.predict(Housing_test_set)
      print(r2_score(housing_cost,pred))
      print(mean_absolute_error(housing_cost,pred))

      -0.7864569706721551
      640319.1805992288
```



```
#Lasso
las = linear_model.Lasso(max_iter = 5000)
las.fit(Housing_train_set,housing_cost)
pred = las.predict(Housing_test_set)
print(r2_score(housing_cost,pred))
print(mean_absolute_error(housing_cost,pred))

-0.7871291424969067
640469.7221590508
```



```
[306] #random forest
      clf = RandomForestRegressor(n_estimators= 100,max_features= 'auto',min_samples_leaf=5)
      clf = clf.fit(Housing_train_set,housing_cost)
      pred = clf.predict(Housing_test_set)
      print(r2_score(housing_cost,pred))
      print(mean_absolute_error(housing_cost,pred))

      -0.8258492400177218
      625671.2180389843
```

From this we see all of my R2 values are negative so there is no real correlation meaning that all of the data input into the models is rubbish most likely due to the data being extremely skewed the opposite way and with all the MSE being extremely high means that  the difference between the values is extremely high.

This was due to some problems with the data being mostly having wrong bits of data in it and because my pipelines didn't work I had to try and do it manually which may of skewed the data drastically.

In conclusion I had trouble with the dataset but we can see from the previous graphs that if there is a house in a good spot with a large amount of land and a decent amount of rooms and parking space it will be very expensive and vice versa if there is a property in a rather dense area with not a lot around it with a low size and room count it will be fairly cheap.