Capstone Project

For the capstone project I have chosen the Chennai housing sale dataset.

The sales price needs to be predicted from this dataset, SALES_PRICE is the dependent variable. This is a regression problem and it is supervised learning

Proprocessing

There are 3 fields [N_BEDROOM ,N_BATHROOM,QS_OVERALL]which are having null values and are handled using interpolate method

For other 6 [AREA,SALE_COND,PARK_FACIL,BUILDTYPE,UTILITY_AVAIL,STREET] columns the field values were not proper, so the column values are replaced to properly

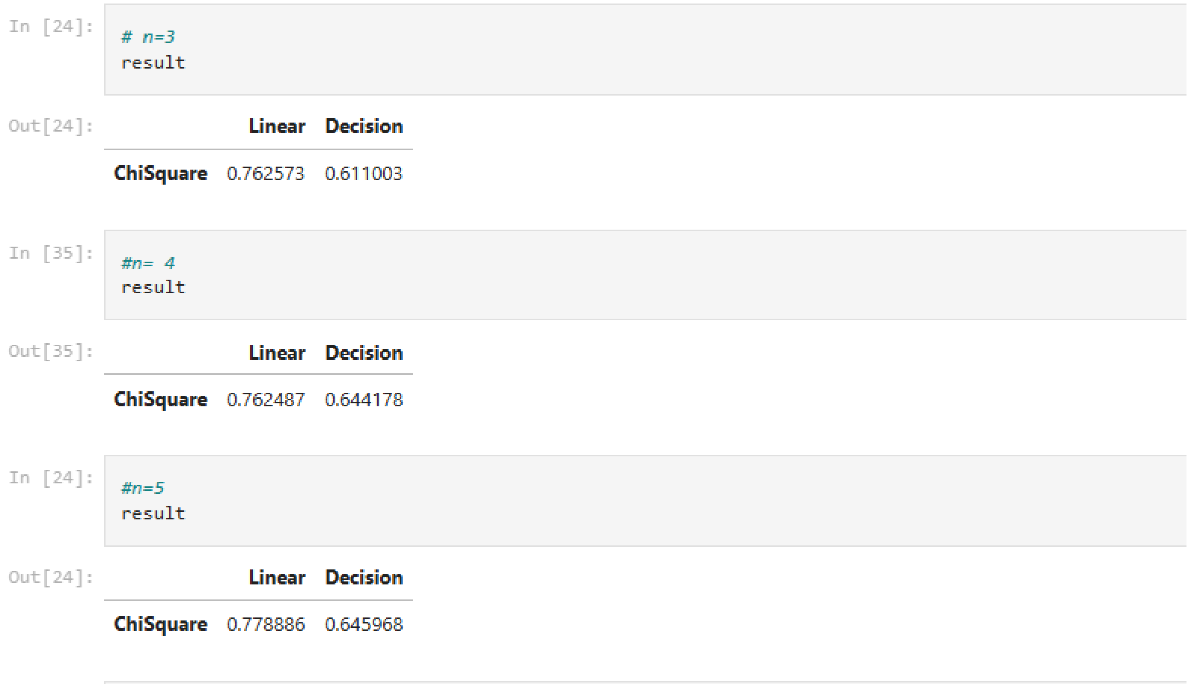
Univariate and Bivariate

Univariate and Bivariate analysis are done for this dataset

Feature Selection

The feature is selection is done using Select K and RFE algorithms

Select K



```
In [24]:  # n=3
          result
```

Out[24]:

	Linear	Decision
ChiSquare	0.762573	0.611003

```
In [35]:  #n= 4
          result
```

Out[35]:

	Linear	Decision
ChiSquare	0.762487	0.644178

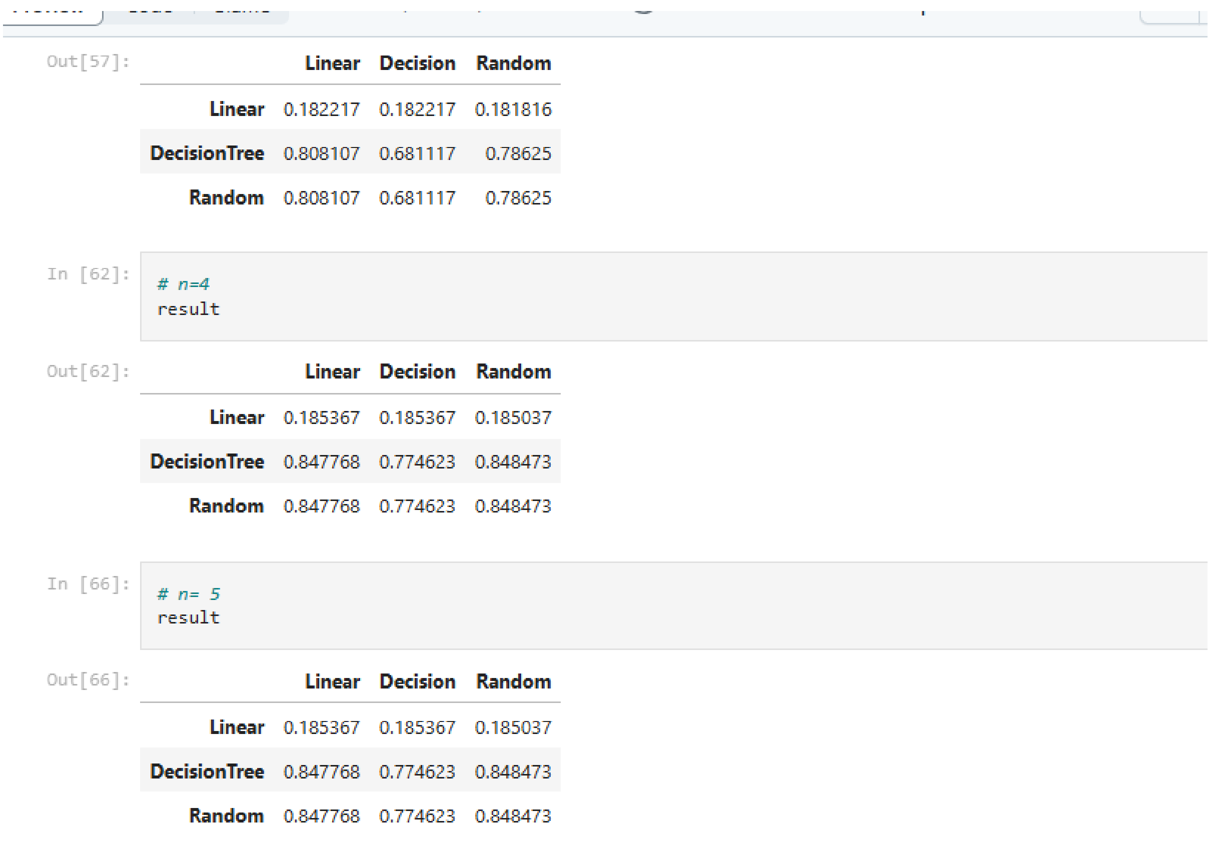
```
In [24]:  #n=5
          result
```

Out[24]:

	Linear	Decision
ChiSquare	0.778886	0.645968

The 5 best features using Select K are ['INT_SQFT', 'DIST_MAINROAD', 'REG_FEE', 'COMMIS', 'AREA_T Nagar']

RFE



Out[57]:

	Linear	Decision	Random
Linear	0.182217	0.182217	0.181816
DecisionTree	0.808107	0.681117	0.78625
Random	0.808107	0.681117	0.78625

In [62]:

```
# n=4
result
```

Out[62]:

	Linear	Decision	Random
Linear	0.185367	0.185367	0.185037
DecisionTree	0.847768	0.774623	0.848473
Random	0.847768	0.774623	0.848473

In [66]:

```
# n= 5
result
```

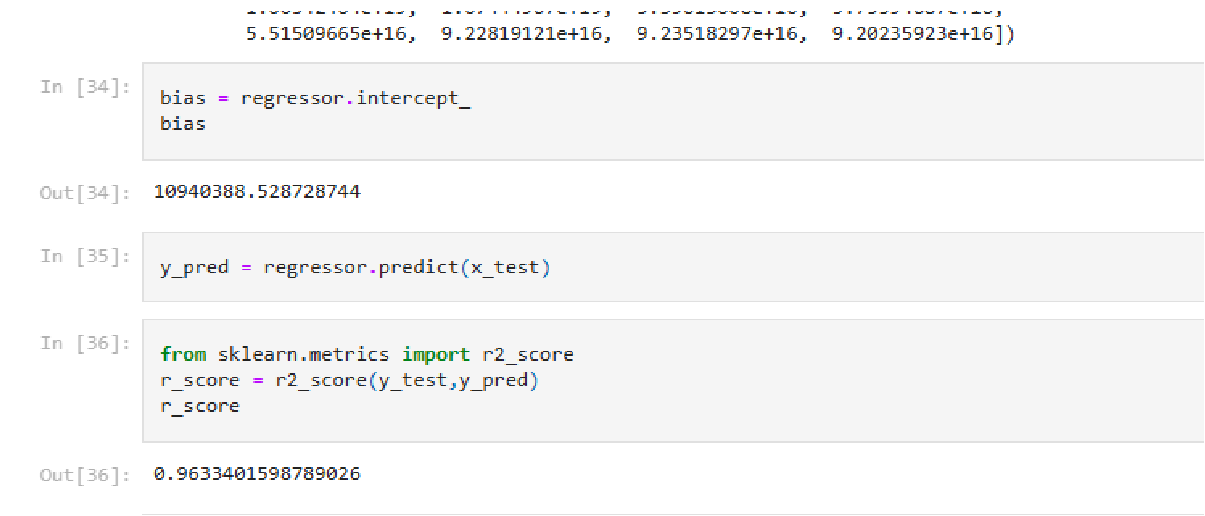
Out[66]:

	Linear	Decision	Random
Linear	0.185367	0.185367	0.185037
DecisionTree	0.847768	0.774623	0.848473
Random	0.847768	0.774623	0.848473

The random forest algorithm gives the best score

Model Creation

Linear Regression



```
       5.51509665e+16,  9.22819121e+16,  9.23518297e+16,  9.20235923e+16])
```

In [34]:

```
bias = regressor.intercept_
bias
```

Out[34]: 10940388.528728744

In [35]:

```
y_pred = regressor.predict(x_test)
```

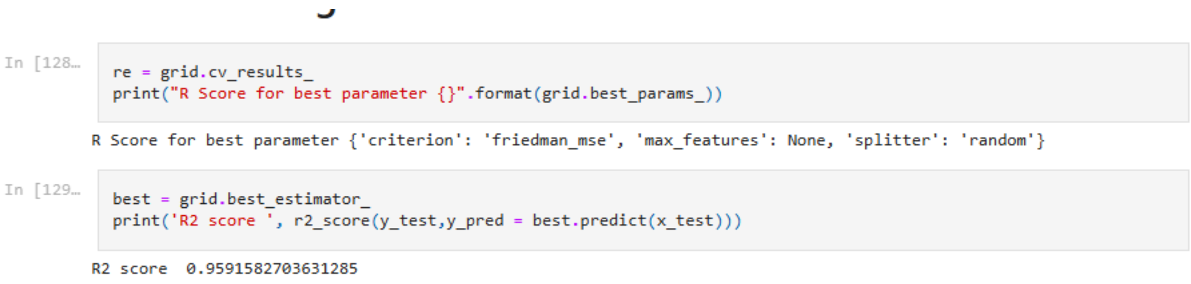
In [36]:

```
from sklearn.metrics import r2_score
r_score = r2_score(y_test,y_pred)
r_score
```

Out[36]: 0.9633401598789026

The R score using linear regression is **0.963**

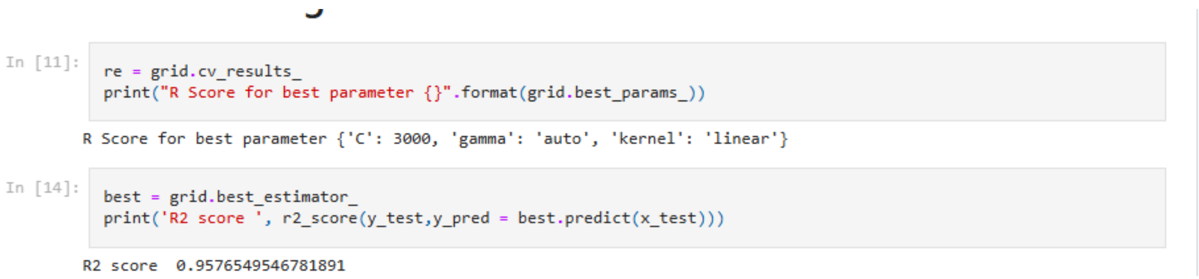
Decision Tree



```
In [128…  re = grid.cv_results_
          print("R Score for best parameter {}".format(grid.best_params_))

R Score for best parameter {'criterion': 'friedman_mse', 'max_features': None, 'splitter': 'random'}

In [129…  best = grid.best_estimator_
          print('R2 score ', r2_score(y_test,y_pred = best.predict(x_test)))

R2 score  0.9591582703631285
```

The R score using Decision tree is **0.959** and the best parameter **is {'criterion': 'friedman_mse', 'max_features': None, 'splitter': 'random'}**

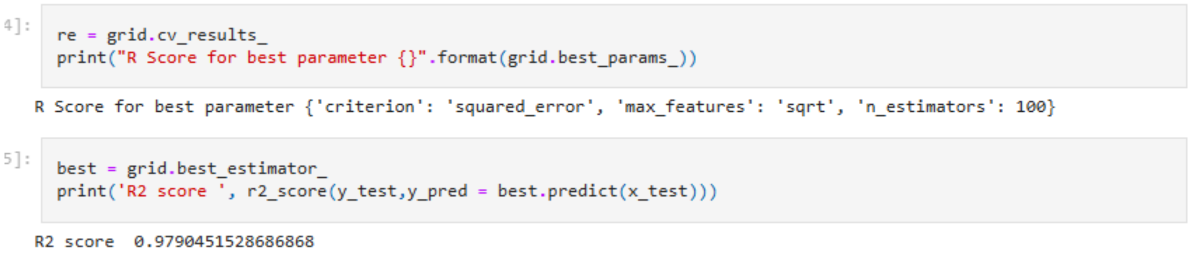
SVM



```
In [11]:  re = grid.cv_results_
          print("R Score for best parameter {}".format(grid.best_params_))

R Score for best parameter {'C': 3000, 'gamma': 'auto', 'kernel': 'linear'}

In [14]:  best = grid.best_estimator_
          print('R2 score ', r2_score(y_test,y_pred = best.predict(x_test)))

R2 score  0.9576549546781891
```

The R score using SVM is **0.957** and the best parameter **is *{'C': 3000, 'gamma': 'auto', 'kernel': 'linear'}***

Random forest



```
4]:  re = grid.cv_results_
     print("R Score for best parameter {}".format(grid.best_params_))

R Score for best parameter {'criterion': 'squared_error', 'max_features': 'sqrt', 'n_estimators': 100}

5]:  best = grid.best_estimator_
     print('R2 score ', r2_score(y_test,y_pred = best.predict(x_test)))

R2 score  0.9790451528686868
```

The R score using Random forest algorithm is **0.979** and the best parameter is **{'criterion': 'squared_error', 'max_features': 'sqrt', 'n_estimators': 100}**

Final Model

The best model is **Random forest** for the Chennai housing sale dataset as the R score is higher for this model