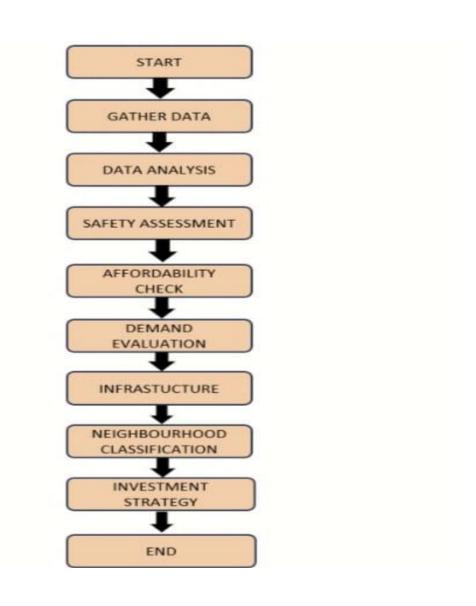# Neighbourhood Classification for Real Estate Investment.

Flow Chart

Architecture Diagram

Program Code

509	0.77763	0.0	12.70		0	0.561	6.222	34.0	2.5430	3	329	23.0	343.00	76.00	67.0
510	0.65432	0.0	12.80		0	0.561	6.760	67.0	2.9870	3	345	23.0	321.00	45.00	24.0

511 rows × 14 columns

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 511 entries, 0 to 510
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   CRIM     511 non-null    float64
 1   ZN       511 non-null    float64
 2   INDUS    511 non-null    float64
 3   CHAS     511 non-null    int64
 4   NOX      511 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      511 non-null    float64
 7   DIS      511 non-null    float64
 8   RAD      511 non-null    int64
 9   TAX      511 non-null    int64
 10  PTRATIO  511 non-null    float64
 11  B        511 non-null    float64
 12  LSTAT    511 non-null    float64
 13  MEDV     511 non-null    float64
dtypes: float64(11), int64(3)
memory usage: 56.0 KB
```

```python
df.describe()
```

```
[5] df.describe()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
count	511.000000	511.000000	511.000000	511.000000	511.000000	506.000000	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000
mean	3.584139	11.252446	11.151096	0.068493	0.554757	6.287589	68.616243	3.783876	9.485323	407.440313	18.500000	356.600900	12.879550	22.682192
std	8.564433	23.234838	6.828175	0.252838	0.115310	0.703802	28.099130	2.098631	8.688469	167.903532	2.200348	90.882679	7.797416	9.484262
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	1.730000	5.000000
25%	0.082325	0.000000	5.190000	0.000000	0.449000	5.885500	45.050000	2.100350	4.000000	279.500000	17.400000	374.710000	7.065000	17.050000
50%	0.261690	0.000000	9.690000	0.000000	0.538000	6.209000	77.300000	3.152300	5.000000	330.000000	19.100000	391.340000	11.450000	21.200000
75%	3.621175	12.500000	18.100000	0.000000	0.624000	6.629750	94.050000	5.118000	24.000000	666.000000	20.200000	396.210000	17.105000	25.000000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	23.000000	396.900000	76.000000	67.000000

```
[8] df['RM']=df['RM'].fillna(df['RM'].mean())
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 511 entries, 0 to 510
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype
---  ------   --------------   -----
 0   CRIM     511 non-null     float64
 1   ZN       511 non-null     float64
 2   INDUS    511 non-null     float64
 3   CHAS     511 non-null     int64
```

completed at 12:56 PM

```
print("Total Number of Missing Value",df.isna().sum())
```

```
Total Number of Missing Value CRIM        0
ZN           0
INDUS        0
CHAS         0
NOX          0
RM           0
AGE          0
DIS          0
RAD          0
TAX          0
PTRATIO      0
B            0
LSTAT        0
MEDV         0
dtype: int64
```

```
y=df['MEDV'].copy()
x=df.drop('MEDV',axis=1)
```

```
[13] x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.7)
```

```
[14] scaler=StandardScaler()
     scaler.fit(x_train)
     x_train=pd.DataFrame(scaler.transform(x_train),columns=x_train.columns)
     x_test=pd.DataFrame(scaler.transform(x_test),columns=x_test.columns)
```

```
[15] print(x_train.shape)
     print(x_test.shape)
```

```
[15]  print(x_train.shape)
      print(x_test.shape)
      print(y_train.shape)
      print(y_test.shape)

      (357, 13)
      (154, 13)
      (357,)
      (154,)
```

```
      model=RandomForestRegressor()
      model.fit(x_train,y_train)
      print(model.score(x_test,y_test))
```

```
      0.8573633024153
```

```
[ ]
```