**Verified Data Prediction**

No. of Kiranas Shops Prediction Report

**Nandan Kumar**

**PROBLEM STATEMENT**

We have the merged csv file of some of the verified regions of India. Our motive is to build a model on the verified data set, so that it can be used to predict that data points of the non- verified merged data set.

This project includes Outlier detection, null detection, missing values handling and build the model for the targeted data points. And also validation and visualization of the results.

**DATA DESCRIPTION**

The data is of the dimension (722\*40) including object, float64 and int64 data types columns, including our target variable – “Number of Kirana Shops”, having 0 missing value in the dataset.

Missing values of All the column:-

Region 0

Area 0

Population 0

Density per square kilometer 0

Senior citizen % 0

No of cases 12

Educational institutes 1

Hospitals 0

Supermarts/malls 89

Number of Households 0

Number of Notified Households 0

No. of household with water connection inside the house 0

Total number of water connections 292

No of households having private washroom 0

Density of retail store 0

Per capita income 11

Gender Ratio 0

Domestic Violence % 0

Internet Penetration % 0

Police to Population Ratio per 1000 12

Number of Kirana Shops 0

Number of Pharmacies 3

Number of Liquor Stores 1

Estimates of Alcoholism % in Population 0

% Unemployed(55%) 0

Age Group Children % 0

Literacy Rate 1

Slum percentage 0

Agriculture and Food 39

Manufacturing 39

Mining 39

Electricity Gas Water 39

Construction 39

Trade Hotels Restaurants 39

Transport Storage Communication 39

Financing Real Estate Business Services 39

Community Social Public Admin 39

Rural 0

Semi Urban 0

Urban 0

No. of Kiranas Shops Statistical values---

· Count : 722

· mean :919.498968

· std :3425.778813

· min : 0.000000

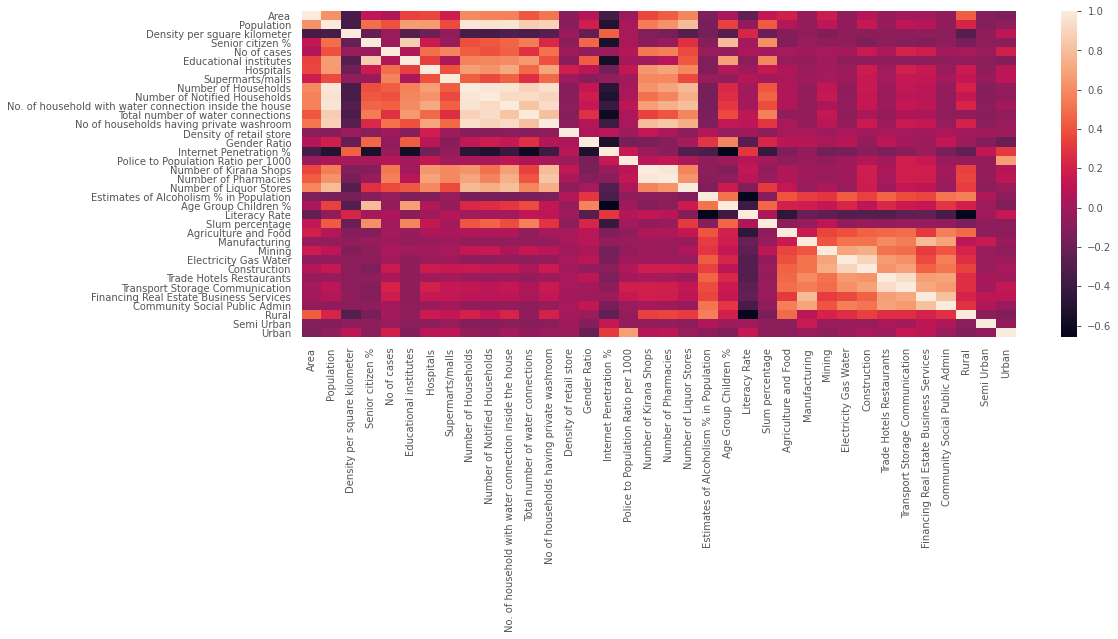
· max :43135.000000

**METHODOLOGY**

**Feature Extraction**

To find the columns or variables that shows correlation with the target variable, I plotted the heatmap, using the Pearson Correlation among all the columns of the data.

I have selected the relevant features or columns, depending on the correlation value of the particular column with the target data point, that is different for all and the range lies between -0.6 and 1.0



Only those columns will be extracted which shows the correlation value greater than 0.5, with respect to the target data point i.e. “Number of Kirana Shops”.

We get the following columns, that shows strong positive correlation with the target data point :

Number of Kirana Shops 1.000000

Number of Pharmacies 0.976323

No of households having private washroom 0.797194

No. of household with water connection inside the house 0.673672

Hospitals 0.649215

Number of Households 0.633946

Supermarts/malls 0.589862

Number of Liquor Stores 0.574851

Population 0.547821

No of cases 0.530176

Number of Notified Households 0.502139

**Data Pre-processing**

After doing the feature extraction, we have got a new data with dimension (722\*10),instead of this take the all columns except “Number of Kirana Shops” . now ,in this also we have to drop some columns name as "Region", "Per capita income", "Domestic Violence %","% Unemployed(55%)", because dtype of this columns is object type

Now, , we have got a new data with dimension (722\*36), including the 35 independent variables and 1 dependent or target variable

Now, we move to the handling of missing values, in the data. We have got 0 missing value in “Number of Kirana Shops” and 292 missing value in ‘Total number of water connections’. I filled all the missing values with 0.

Then we divided the data in two : x (independent variables) and y (dependent variable). And splitted it into train set (data set to train the model on) and test set (data set to test the built model), with the test size of 30%. Thus, we get :

x\_train.shape – (505, 35)

x\_test.shape – (217, 35)

y\_train.shape – (505)

y\_test.shape – (217)

**Building the model**

As , this is a regression problem, I applied few machine learning regression algorithms, on the top of this data and evaluate their performance on the basis of r^2 value i.e. the “coefficient of determination” and also mean squared error.

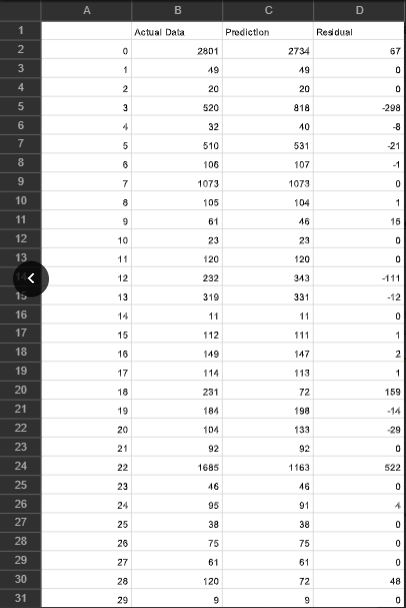
We get the results as follows :

|  |  |
| --- | --- |
| Algorithm or Models | R^2 Score |
| 1. Linear regression | 91.23994554523007 |
| 1. Lasso | 91.23410794553413 |
| 1. KNeighborsRegressor | 82.21029554648494 |
| 1. DecisionTreeRegressor | 90.78708466840347 |
| 1. RandomForestRegressor | 95.86345520802412 |
| 1. GradientBoostingRegressor | 94.21028734663209 |
| 1. XGBRegressor | 94.57876747996035 |

**RESULTS**

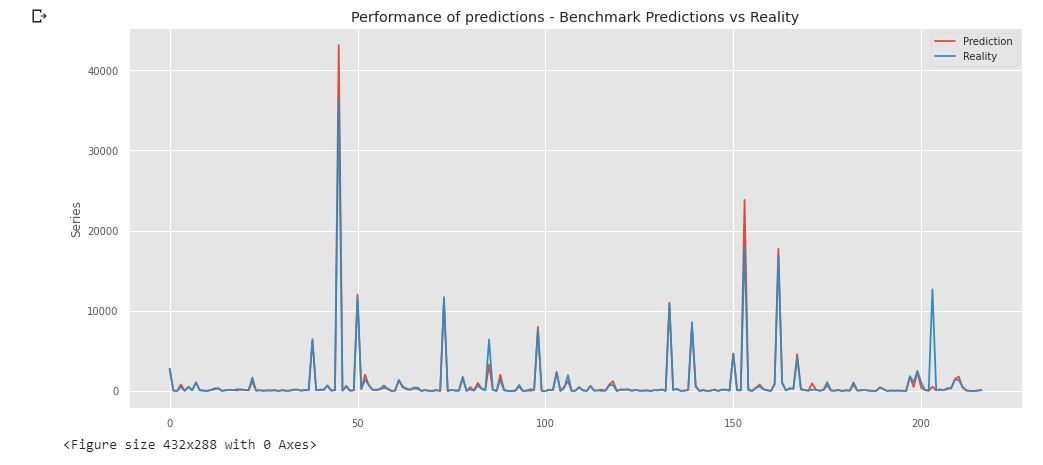
By looking at the r^2 value of the algorithms, we get the best result with random forest regressor , so I used it to predict the values of target value, using the x\_test data points. And later comparing it with the actual values of the target variable i.e. y\_test.

Then I created a dataframe having the actual values and predicted values of the target data point and also calculated their residual value (actual values - predicted values). And we got results as following :

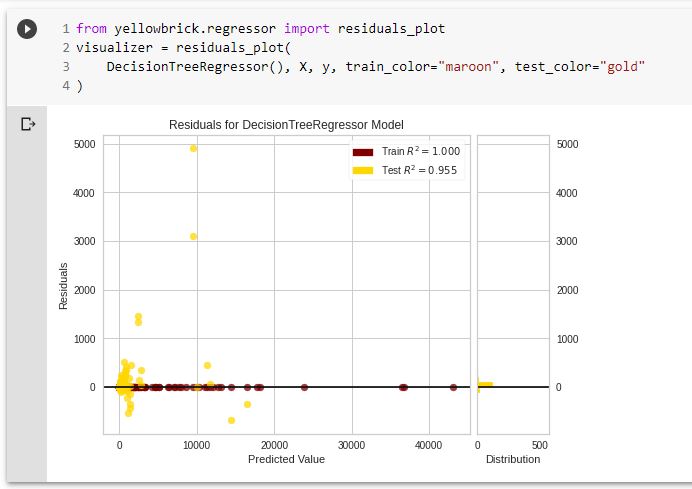


**Graphical Plots :-**

* **GGPLOT between the Actual and predicted values**



* Residual plot:-



* Prediction Error :-

