Presented By,

1 : 18101A0016 Lav Dabade

2 : 19101A2006 Harshala Ardekar

3 : 19101A2007 Tanvi Gangar

## Zomato recipe rating Prediction using Linear Regression

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
df = pd.read_csv('/zomato.csv', encoding='latin-1')
```

```
df.head()
```

| | Restaurant ID | Restaurant Name | Country Code | City | Address | Locality | Locality Verbose | Lc |
|---|---|---|---|---|---|---|---|---|
| **0** | 6317637 | Le Petit Souffle | 162 | Makati City | Third Floor, Century City Mall, Kalayaan Avenu... | Century City Mall, Poblacion, Makati City | Century City Mall, Poblacion, Makati City, Mak... | 12 |
| **1** | 6304287 | Izakaya Kikufuji | 162 | Makati City | Little Tokyo, 2277 Chino Roces Avenue, Legaspi... | Little Tokyo, Legaspi Village, Makati City | Little Tokyo, Legaspi Village, Makati City, Ma... | 12 |
| **2** | 6300002 | Heat - Edsa Shangri-La | 162 | Mandaluyong City | Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal... | Edsa Shangri-La, Ortigas, Mandaluyong City | Edsa Shangri-La, Ortigas, Mandaluyong City, Ma... | 12 |
| | | | | | Third | | SM | |

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Restaurant ID         9551 non-null   int64  
 1   Restaurant Name       9551 non-null   object 
 2   Country Code          9551 non-null   int64  
 3   City                  9551 non-null   object 
 4   Address               9551 non-null   object 
 5   Locality              9551 non-null   object 
 6   Locality Verbose      9551 non-null   object 
 7   Longitude             9551 non-null   float64
 8   Latitude              9551 non-null   float64
 9   Cuisines              9542 non-null   object 
 10  Average Cost for two  9551 non-null   int64  
 11  Currency              9551 non-null   object 
 12  Has Table booking     9551 non-null   object 
 13  Has Online delivery   9551 non-null   object 
 14  Is delivering now     9551 non-null   object 
 15  Switch to order menu  9551 non-null   object 
 16  Price range           9551 non-null   int64  
 17  Aggregate rating      9551 non-null   float64
 18  Rating color          9551 non-null   object 
 19  Rating text           9551 non-null   object 
 20  Votes                 9551 non-null   int64  
dtypes: float64(3), int64(5), object(13)
memory usage: 1.5+ MB
```

```
df.describe()
```

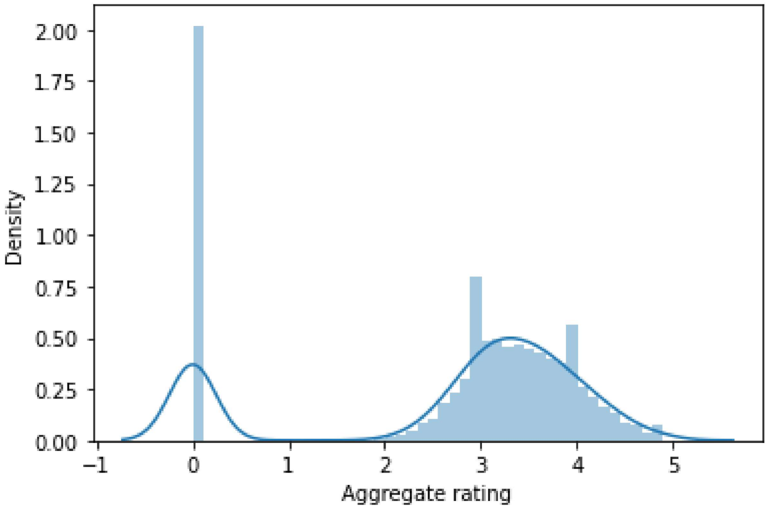
| | Restaurant ID | Country Code | Longitude | Latitude | Average Cost for two | Price range | |
|---|---|---|---|---|---|---|---|
| **count** | 9.551000e+03 | 9551.000000 | 9551.000000 | 9551.000000 | 9551.000000 | 9551.000000 | 95 |
| **mean** | 9.051128e+06 | 18.365616 | 64.126574 | 25.854381 | 1199.210763 | 1.804837 | |
| **std** | 8.791521e+06 | 56.750546 | 41.467058 | 11.007935 | 16121.183073 | 0.905609 | |
| **min** | 5.300000e+01 | 1.000000 | -157.948486 | -41.330428 | 0.000000 | 1.000000 | |
| **25%** | 3.019625e+05 | 1.000000 | 77.081343 | 28.478713 | 250.000000 | 1.000000 | |
| **50%** | 6.004089e+06 | 1.000000 | 77.191964 | 28.570469 | 400.000000 | 2.000000 | |
| **75%** | 1.835229e+07 | 1.000000 | 77.282006 | 28.642758 | 700.000000 | 2.000000 | |
| **max** | 1.850065e+07 | 216.000000 | 174.832089 | 55.976980 | 800000.000000 | 4.000000 | |

```
df.columns
```

```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
           'Votes'],
          dtype='object')
```

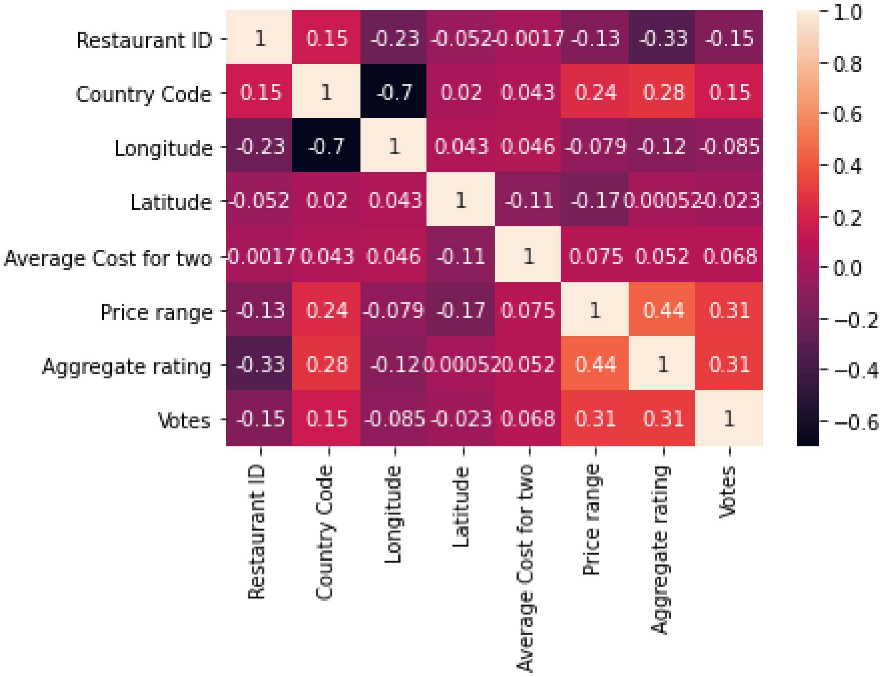
```
sns.distplot(df['Aggregate rating'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `di
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f2c85ad3f10>
```



```
sns.heatmap(df.corr(), annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2c85aad750>
```



```
x = df[['Restaurant ID', 'Country Code', 'Longitude', 'Latitude', 'Average Cost for two', 'Pr
y = df['Aggregate rating']
```

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.4, random_state=101)
x_train
```

| | Restaurant ID | Country Code | Longitude | Latitude | Average Cost for two | Price range | Aggregate rating | Votes |
|---|---|---|---|---|---|---|---|---|
| **6831** | 18416859 | 1 | 77.199017 | 28.560495 | 400 | 1 | 0.0 | 2 |
| **8587** | 18204489 | 1 | 77.340025 | 28.565491 | 700 | 2 | 3.8 | 113 |
| **257** | 17257684 | 216 | -93.637401 | 41.587219 | 10 | 1 | 4.4 | 659 |
| **669** | 2400027 | 1 | 81.835585 | 25.457687 | 600 | 3 | 3.4 | 83 |
| **2042** | 18462602 | 1 | 77.094712 | 28.481085 | 400 | 1 | 0.0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **599** | 201044 | 214 | 55.278525 | 25.198291 | 285 | 3 | 3.2 | 506 |
| **5695** | 18349914 | 1 | 76.975319 | 28.611235 | 200 | 1 | 0.0 | 1 |
| **8006** | 312756 | 1 | 77.213345 | 28.562265 | 400 | 1 | 2.7 | 8 |
| **1361** | 307935 | 1 | 77.092692 | 28.490744 | 350 | 1 | 3.1 | 71 |
| **1547** | 3855 | 1 | 77.102077 | 28.480604 | 2500 | 4 | 3.5 | 57 |

```
from sklearn.linear_model import LinearRegression
```

```
lm = LinearRegression()
```

```
lm.fit(x_train, y_train)
```

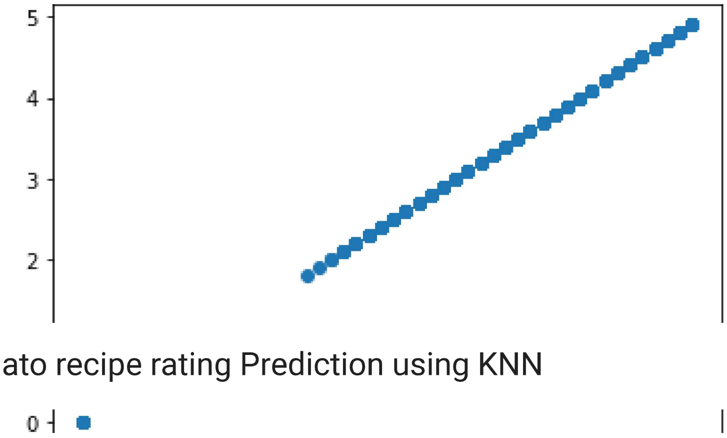
```
     LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
predictions = lm.predict(x_test)
predictions
```

```
     array([ 3.30000000e+00,  3.00000000e+00,  3.10000000e+00, ...,
             2.90000000e+00, -1.07182713e-14,  4.20000000e+00])
```

```
plt.scatter(y_test, predictions)
```

```
<matplotlib.collections.PathCollection at 0x7f2c76f589d0>
```



## Zomato recipe rating Prediction using KNN

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

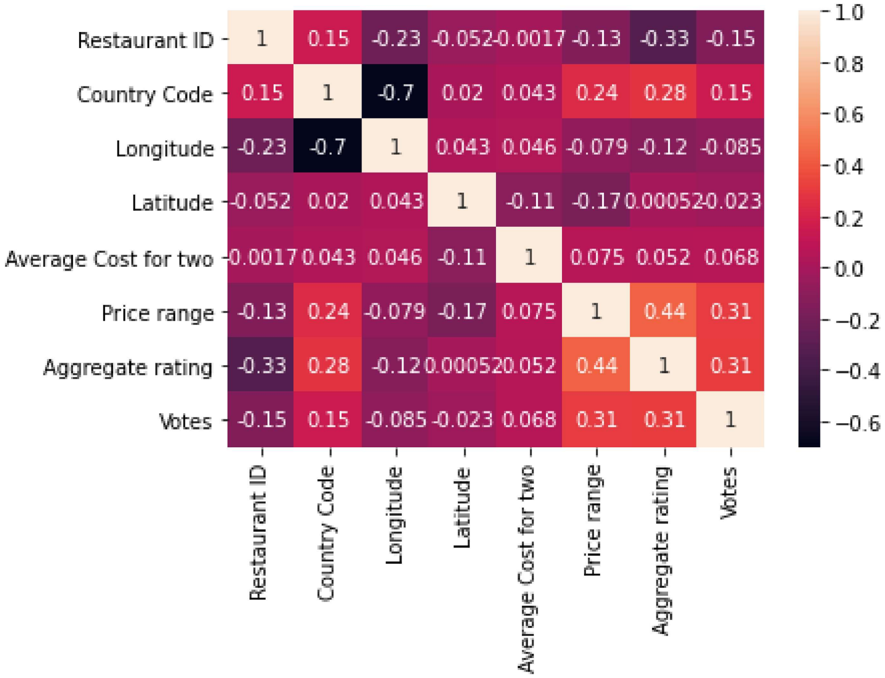
```
data = pd.read_csv('/zomato.csv', encoding='latin-1')
```

```
data.head()
```

| | Restaurant ID | Restaurant Name | Country Code | City | Address | Locality | Locality Verbose | Lc |
|---|---|---|---|---|---|---|---|---|

```
sns.heatmap(data.corr(), annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2c76f27a10>
```



```
x = data[['Restaurant ID', 'Country Code', 'Longitude', 'Latitude', 'Average Cost for two', '
y = data['Aggregate rating']
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.4, random_state=101)
```

```
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

```
classifier = KNeighborsRegressor(n_neighbors=11,p=2,metric='euclidean')
```

```
classifier.fit(x_train,y_train)
```

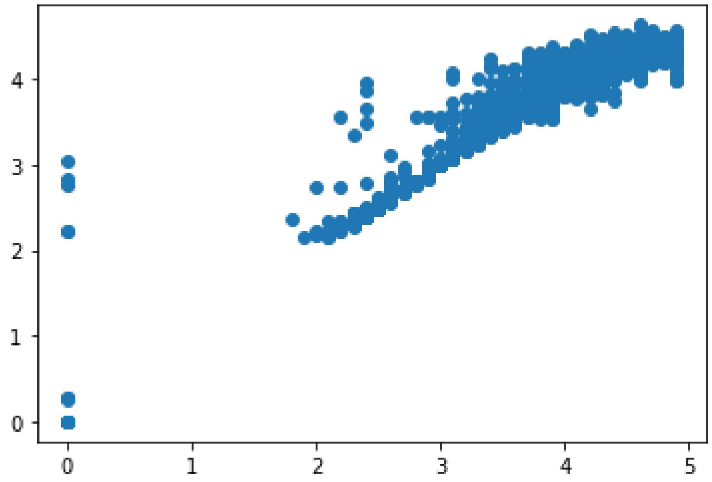
```
KNeighborsRegressor(algorithm='auto', leaf_size=30, metric='euclidean',
                    metric_params=None, n_jobs=None, n_neighbors=11, p=2,
                    weights='uniform')
```

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

```
array([3.3       , 3.        , 3.11818182, ..., 2.9       , 0.        ,
       4.03636364])
```

```
plt.scatter(y_test, y_pred)
```

```
<matplotlib.collections.PathCollection at 0x7f2c76f588d0>
```


4
3
2
1
0
0
1
2
3
4
5