

# covid19\_LR

September 29, 2020

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.stats import norm
from scipy import stats
from datetime import timedelta
from datetime import datetime
import plotly.express as px
import plotly.graph_objs as go
%matplotlib inline
import seaborn as sns
import plotly.offline as py
import plotly.express as ex
import os
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import KFold
from sklearn import metrics
from sklearn.metrics import r2_score
```

```
[2]: data = pd.read_csv("new_data/covid_19_india1.csv")
data.tail(15)
```

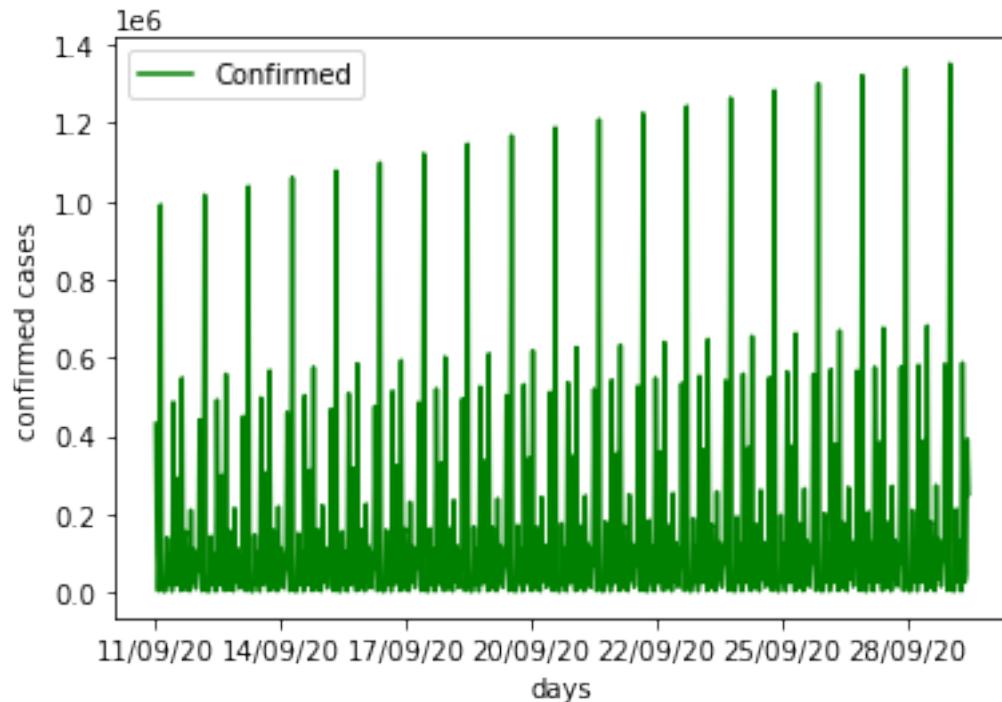
```
[2]:
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	\
6791	6792	29/09/20	8:00 AM	Manipur	-	
6792	6793	29/09/20	8:00 AM	Meghalaya	-	
6793	6794	29/09/20	8:00 AM	Mizoram	-	
6794	6795	29/09/20	8:00 AM	Nagaland	-	
6795	6796	29/09/20	8:00 AM	Odisha	-	
6796	6797	29/09/20	8:00 AM	Puducherry	-	

6797	6798	29/09/20	8:00 AM	Punjab	-
6798	6799	29/09/20	8:00 AM	Rajasthan	-
6799	6800	29/09/20	8:00 AM	Sikkim	-
6800	6801	29/09/20	8:00 AM	Tamil Nadu	-
6801	6802	29/09/20	8:00 AM	Telangana	-
6802	6803	29/09/20	8:00 AM	Tripura	-
6803	6804	29/09/20	8:00 AM	Uttarakhand	-
6804	6805	29/09/20	8:00 AM	Uttar Pradesh	-
6805	6806	29/09/20	8:00 AM	West Bengal	-

	ConfirmedForeignNational	Cured	Deaths	Confirmed
6791	-	7982	64	10477
6792	-	3868	46	5362
6793	-	1459	0	1958
6794	-	4938	17	5957
6795	-	177585	813	212609
6796	-	21156	515	26685
6797	-	90345	3284	111375
6798	-	109472	1456	130971
6799	-	2164	34	2896
6800	-	530708	9383	586397
6801	-	158690	1116	189283
6802	-	19203	276	25353
6803	-	36856	580	47502
6804	-	331270	5652	390875
6805	-	219844	4837	250580

```
[3]: data.tail(650).plot(kind='line',x='Date', y='Confirmed',color='green')
plt.xlabel('days')
plt.ylabel('confirmed cases')
plt.show()
```

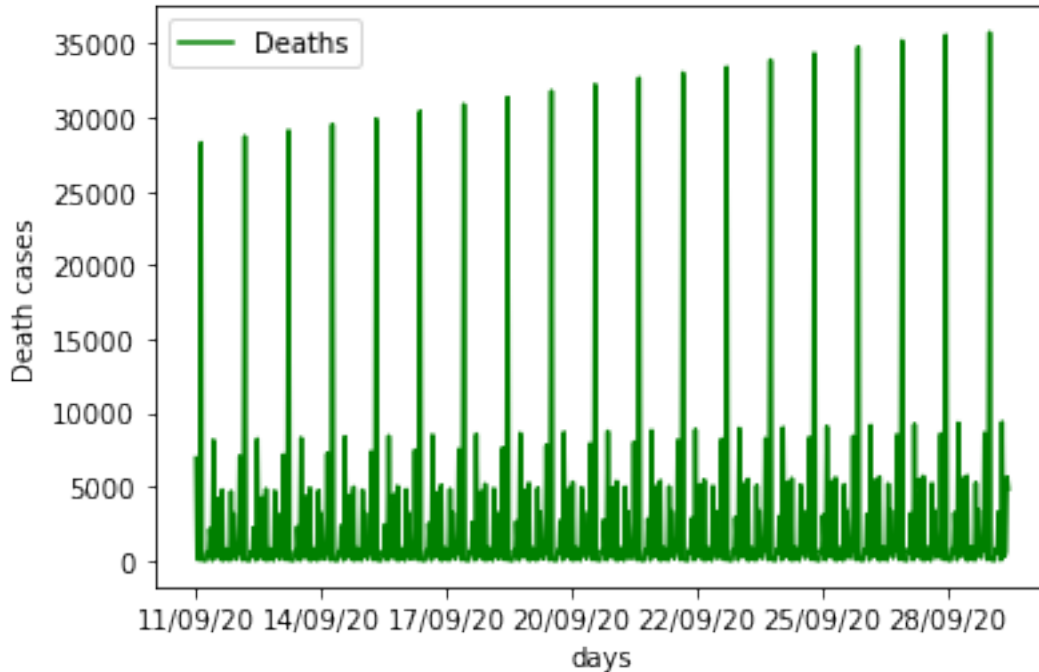


```
[4]: import plotly.graph_objs as go
fig = go.Figure()
fig.add_trace(go.Scatter(x=data['Date'], y=data['Cured'],
                        mode='lines+markers',marker_color='blue'))
fig.update_layout(title_text = 'Trend of Cured case of India')
fig.update_layout(plot_bgcolor='rgb(275, 270, 273)',width=600, height=600)
fig.show()
```

```
[5]: import plotly.graph_objs as go
fig = go.Figure()
fig.add_trace(go.Scatter(x=data['Date'], y=data['Confirmed'],
                        mode='lines+markers',marker_color='blue'))
fig.update_layout(title_text = 'Trend of Confirmed case of India')
fig.update_layout(plot_bgcolor='rgb(275, 270, 273)',width=600, height=600)
fig.show()
```

```
[6]: import plotly.graph_objs as go
fig = go.Figure()
fig.add_trace(go.Scatter(x=data['Date'], y=data['Deaths'],
                        mode='lines+markers',marker_color='blue'))
fig.update_layout(title_text = 'Trend of death cases of India')
fig.update_layout(plot_bgcolor='rgb(275, 270, 273)',width=600, height=600)
fig.show()
```

```
[7]: data.tail(650).plot(kind='line',x='Date', y='Deaths',color='green')
plt.xlabel('days')
plt.ylabel('Death cases')
plt.show()
```



```
[8]: data['Active']= data['Confirmed']- (data['Cured'] + data['Deaths'])
date_wise_data = data[['State/
↳UnionTerritory','Date','Confirmed','Deaths','Cured','Active']]
#date_wise_data['Date'] = date_wise_data['Date'].apply(pd.to_datetime,↳
↳dayfirst=True)
date_wise_data = date_wise_data.groupby(["Date"]).sum().reset_index()
date_wise_data.to_csv('date_wise_data.csv')
date_wise_data.head()
```

```
[8]:
```

	Date	Confirmed	Deaths	Cured	Active
0	01/02/20	2	0	0	2
1	01/03/20	3	0	0	3
2	01/04/20	1834	41	144	1649
3	01/05/20	34972	1152	9065	24755
4	01/06/20	190535	5394	91819	93322

```
[9]: import plotly.express as px
import plotly.express as ex
temp = date_wise_data.copy()
```

```
fig = px.bar(temp,x="Confirmed", y="Date",
             title='Date Wise Confirmed Cases',
             orientation='h',height=1700,text='Confirmed')
#fig.update_layout(plot_bgcolor='rgb(275, 270, 273)')
fig.show()
```

[ ]:

[ ]:

```
[10]: data['Date2'] = pd.to_datetime(data['Date'])
today = data[data['Date2'] == max(data['Date2'])]
today.to_csv('Today.csv')
today.tail()
```

```
[10]:
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	\
6206	6207	12/09/20	8:00 AM	Telengana	-	
6207	6208	12/09/20	8:00 AM	Tripura	-	
6208	6209	12/09/20	8:00 AM	Uttarakhand	-	
6209	6210	12/09/20	8:00 AM	Uttar Pradesh	-	
6210	6211	12/09/20	8:00 AM	West Bengal	-	

	ConfirmedForeignNational	Cured	Deaths	Confirmed	Active	Date2
6206	-	121925	950	154880	32005	2020-12-09
6207	-	10734	182	18281	7365	2020-12-09
6208	-	19428	388	29221	9405	2020-12-09
6209	-	227442	4282	299045	67321	2020-12-09
6210	-	169043	3828	196332	23461	2020-12-09

[ ]:

[ ]:

```
[11]: state_wise_data = today[['State/
UnionTerritory',"Confirmed","Deaths","Cured","Active"]]
#date_wise_data['Date'] = date_wise_data['Date'].apply(pd.to_datetime,
dayfirst=True)
state_wise_data = today.sort_values('Confirmed', ascending=False)
state_wise_data.to_csv('state_wise_data.csv')
state_wise_data.tail(100)
```

```
[11]:
```

	Sno	Date	Time	State/UnionTerritory	\
6195	6196	12/09/20	8:00 AM	Maharashtra	
6177	6178	12/09/20	8:00 AM	Andhra Pradesh	
6205	6206	12/09/20	8:00 AM	Tamil Nadu	
6191	6192	12/09/20	8:00 AM	Karnataka	
6209	6210	12/09/20	8:00 AM	Uttar Pradesh	

6184	6185	12/09/20	8:00 AM	Delhi
6210	6211	12/09/20	8:00 AM	West Bengal
6180	6181	12/09/20	8:00 AM	Bihar
6206	6207	12/09/20	8:00 AM	Telengana
6200	6201	12/09/20	8:00 AM	Odisha
6179	6180	12/09/20	8:00 AM	Assam
6186	6187	12/09/20	8:00 AM	Gujarat
6192	6193	12/09/20	8:00 AM	Kerala
6203	6204	12/09/20	8:00 AM	Rajasthan
6187	6188	12/09/20	8:00 AM	Haryana
6194	6195	12/09/20	8:00 AM	Madhya Pradesh
6202	6203	12/09/20	8:00 AM	Punjab
6190	6191	12/09/20	8:00 AM	Jharkhand
6182	6183	12/09/20	8:00 AM	Chhattisgarh
6189	6190	12/09/20	8:00 AM	Jammu and Kashmir
6208	6209	12/09/20	8:00 AM	Uttarakhand
6185	6186	12/09/20	8:00 AM	Goa
6201	6202	12/09/20	8:00 AM	Puducherry
6207	6208	12/09/20	8:00 AM	Tripura
6188	6189	12/09/20	8:00 AM	Himachal Pradesh
6196	6197	12/09/20	8:00 AM	Manipur
6181	6182	12/09/20	8:00 AM	Chandigarh
6178	6179	12/09/20	8:00 AM	Arunachal Pradesh
6199	6200	12/09/20	8:00 AM	Nagaland
6176	6177	12/09/20	8:00 AM	Andaman and Nicobar Islands
6197	6198	12/09/20	8:00 AM	Meghalaya
6193	6194	12/09/20	8:00 AM	Ladakh
6183	6184	12/09/20	8:00 AM	Dadra and Nagar Haveli and Daman and Diu
6204	6205	12/09/20	8:00 AM	Sikkim
6198	6199	12/09/20	8:00 AM	Mizoram

	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths \
6195	-	-	715023	28724
6177	-	-	446716	4779
6205	-	-	435422	8231
6191	-	-	334999	7067
6209	-	-	227442	4282
6184	-	-	178154	4687
6210	-	-	169043	3828
6180	-	-	139458	797
6206	-	-	121925	950
6200	-	-	112062	605
6179	-	-	108329	430
6186	-	-	91343	3180
6192	-	-	73900	410
6203	-	-	81970	1207
6187	-	-	68525	932

6194	-	-	62936	1691
6202	-	-	53308	2212
6190	-	-	43328	532
6182	-	-	27123	519
6189	-	-	34689	854
6208	-	-	19428	388
6185	-	-	18065	276
6201	-	-	13783	365
6207	-	-	10734	182
6188	-	-	5839	71
6196	-	-	6002	44
6181	-	-	4600	86
6178	-	-	4126	10
6199	-	-	3802	10
6176	-	-	3157	51
6197	-	-	1889	24
6193	-	-	2387	38
6183	-	-	2413	2
6204	-	-	1486	8
6198	-	-	790	0

	Confirmed	Active	Date2
6195	1015681	271934	2020-12-09
6177	547686	96191	2020-12-09
6205	491571	47918	2020-12-09
6191	440411	98345	2020-12-09
6209	299045	67321	2020-12-09
6184	209748	26907	2020-12-09
6210	196332	23461	2020-12-09
6180	155445	15190	2020-12-09
6206	154880	32005	2020-12-09
6200	143117	30450	2020-12-09
6179	138339	29580	2020-12-09
6186	110809	16286	2020-12-09
6192	102254	27944	2020-12-09
6203	99036	15859	2020-12-09
6187	88332	18875	2020-12-09
6194	83619	18992	2020-12-09
6202	74616	19096	2020-12-09
6190	59040	15180	2020-12-09
6182	58643	31001	2020-12-09
6189	50712	15169	2020-12-09
6208	29221	9405	2020-12-09
6185	23445	5104	2020-12-09
6201	19026	4878	2020-12-09
6207	18281	7365	2020-12-09
6188	8784	2874	2020-12-09

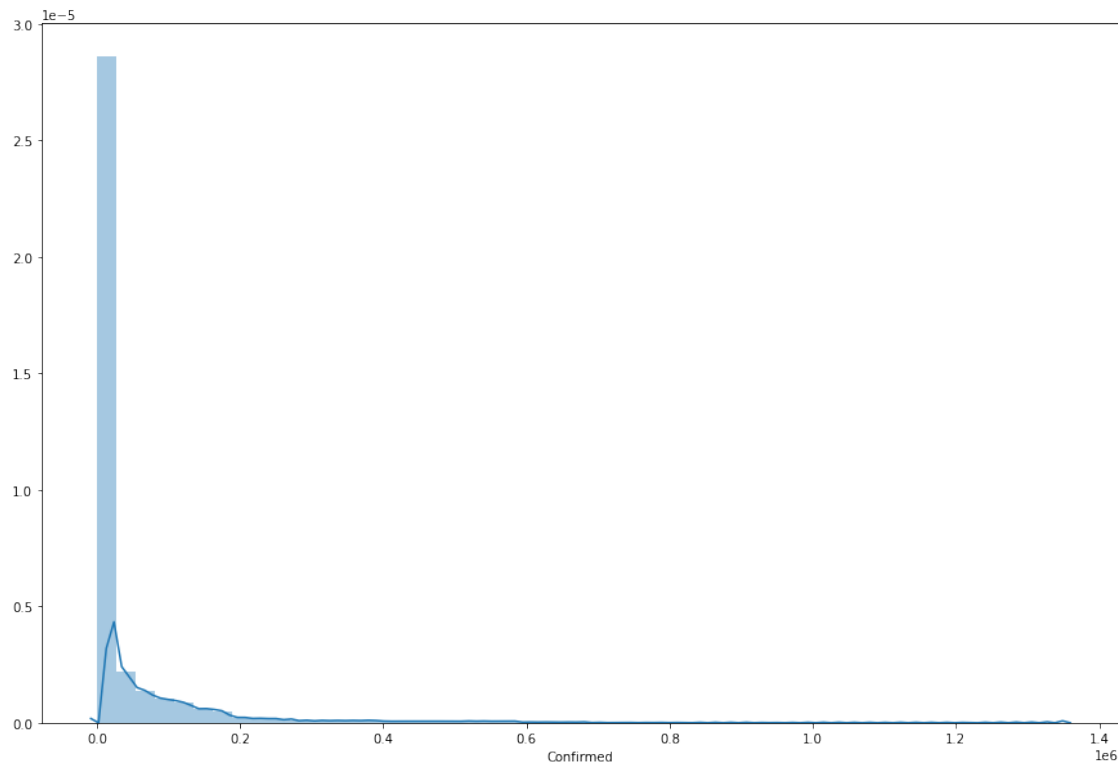
6196	7579	1533	2020-12-09
6181	7292	2606	2020-12-09
6178	5825	1689	2020-12-09
6199	4946	1134	2020-12-09
6176	3494	286	2020-12-09
6197	3447	1534	2020-12-09
6193	3228	803	2020-12-09
6183	2695	280	2020-12-09
6204	2026	532	2020-12-09
6198	1379	589	2020-12-09

```
[12]: temp = state_wise_data.sort_values('Confirmed', ascending=True)
fig = px.bar(temp,x="Confirmed", y="State/UnionTerritory",
             title='State Wise Confirmed Cases',
             orientation='h',height=1000,text='Confirmed')
#fig.update_layout(plot_bgcolor='rgb(275, 270, 273)')
fig.show()
```

```
[13]: active= today['Active'].sum()
cured = today['Cured'].sum()
death= today['Deaths'].sum()
fig = go.Figure(data=[go.Pie(labels=['Active Cases','Cured','Death'],
                               values= [active,cured,death],hole =.3)])
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=20,
                  marker=dict(colors=['#263fa3', '#2fcc41','#cc3c2f'],
                                line=dict(color='FFFFFF', width=2)))
fig.update_layout(title_text='Current Situation in
                    India',plot_bgcolor='rgb(275, 270, 273)')
fig.show()
```

```
[14]: plt.figure(figsize=(15,10))
plt.tight_layout()
sns.distplot(data['Confirmed'])
plt.show()
```





```
[16]: from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
data['Date1']=lb.fit_transform(data['Date'])
data.head()
x=data[['Date1','Deaths','Cured']]
y=data['Confirmed'].values.reshape(-1,1)
data = data.replace(np.inf, np.nan).replace(-np.inf, np.nan).dropna()

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
regression=LinearRegression()
regression.fit(x_train,y_train)
print('Intercept',regression.intercept_)
print('Coefficient',regression.coef_)
```

```
Intercept [2760.29313675]
Coefficient [[-4.07962157  7.32179572  1.0904547  ]]
```

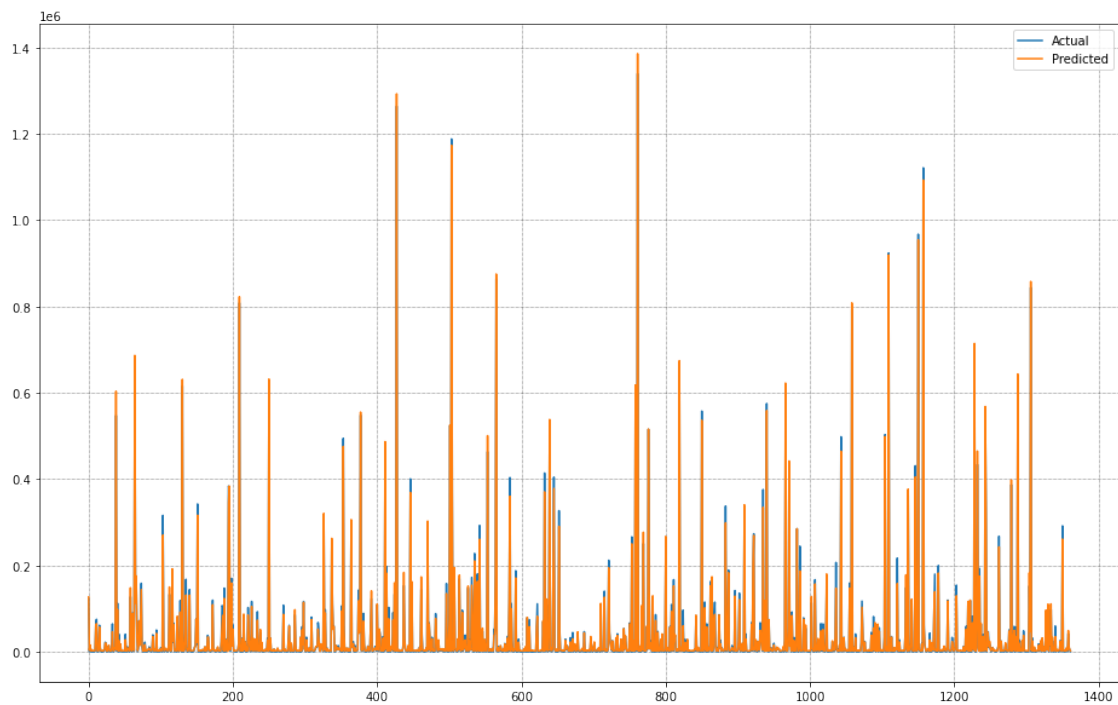
```
[17]: y_pred=regression.predict(x_test)
df=pd.DataFrame({'Actual':y_test.flatten(),'Predicted':y_pred.flatten()})
print(df.tail(50))
```

	Actual	Predicted
1312	7	2037.391862

1313	1	1842.378284
1314	732	2745.023532
1315	1374	2993.611543
1316	1407	2815.681835
1317	18281	15410.236658
1318	2533	3087.782659
1319	9328	9701.448986
1320	24392	25822.868608
1321	415	2348.176806
1322	31071	32080.814403
1323	1008	2426.168748
1324	1046	3345.968456
1325	10988	11499.663881
1326	1	2168.748009
1327	84311	96393.643609
1328	5451	6825.436655
1329	5579	5184.253186
1330	103464	110285.246858
1331	2	1933.220413
1332	32362	31795.586631
1333	13542	15894.657522
1334	97629	110936.126938
1335	45	1908.742684
1336	25601	33236.298429
1337	1145	3508.401684
1338	32	2266.658927
1339	1597	3507.306316
1340	59652	34976.107304
1341	109	1977.287993
1342	2657	3457.143842
1343	3816	5509.492113
1344	437	2504.868734
1345	101	2765.052155
1346	3580	5280.774253
1347	26094	24160.001437
1348	2688	3503.019608
1349	361	2101.791500
1350	291826	261152.822676
1351	1387	3467.149888
1352	152	2391.228483
1353	310	2055.532109
1354	2783	5039.665184
1355	794	3094.890905
1356	8729	9966.948479
1357	230	2771.671552
1358	41820	49121.163560
1359	10298	11050.265405
1360	3920	5506.500216

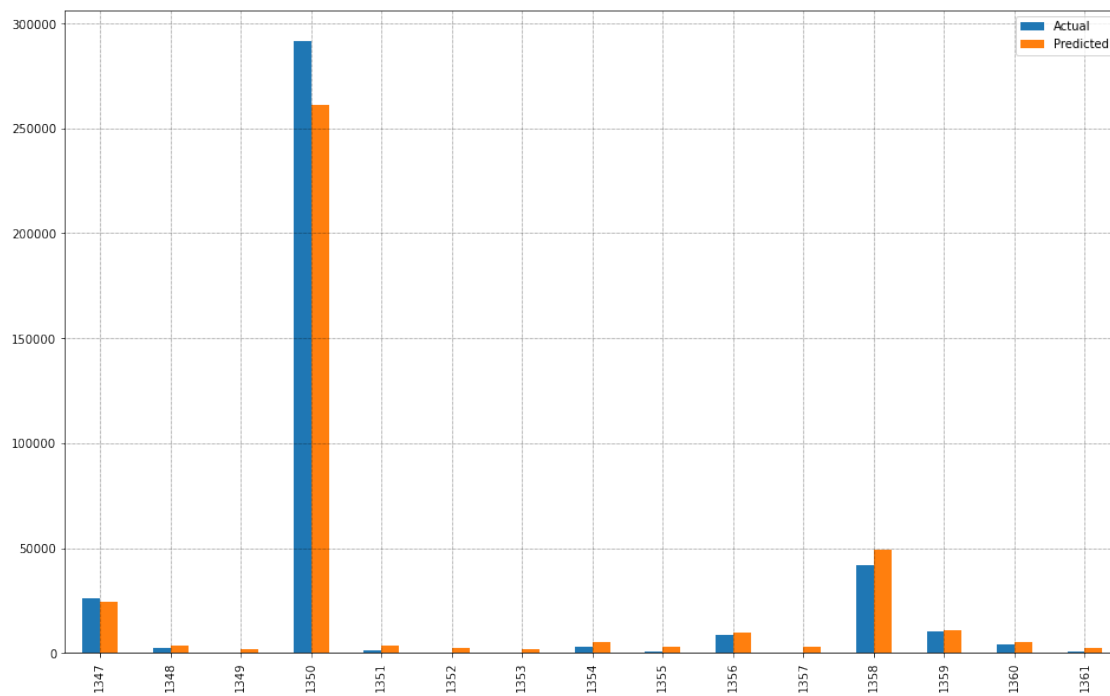
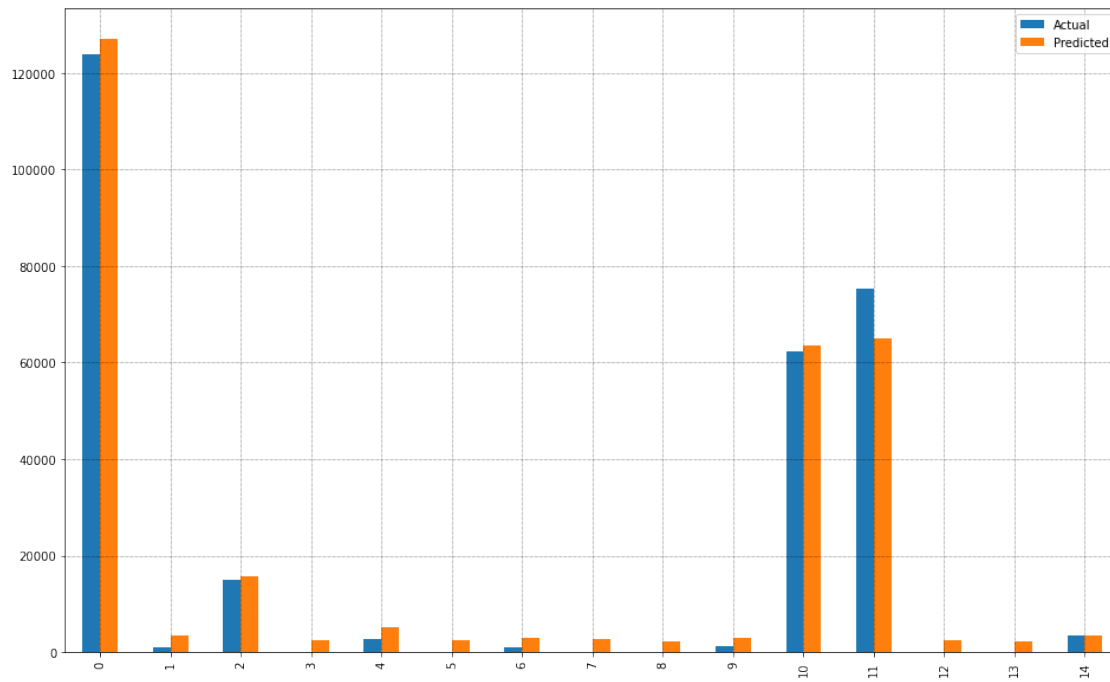
1361      839      2514.093390

```
[18]: df1=df
df1.plot(kind='line',figsize=(16,10))
plt.grid(which='major',linestyle='-',linewidth='0.5', color='green')
plt.grid(which='major',linestyle=':',linewidth='0.5', color='black')
plt.show()
plt.savefig('c1.png')
```



<Figure size 432x288 with 0 Axes>

```
[19]: df1=df.head(15)
df1.plot(kind='bar',figsize=(16,10))
plt.grid(which='major',linestyle='-',linewidth='0.5', color='green')
plt.grid(which='major',linestyle=':',linewidth='0.5', color='black')
plt.show()
df1=df.tail(15)
df1.plot(kind='bar',figsize=(16,10))
plt.grid(which='major',linestyle='-',linewidth='0.5', color='green')
plt.grid(which='major',linestyle=':',linewidth='0.5', color='black')
plt.show()
```



```
[20]: print('Mean Absolute Error:',metrics.mean_absolute_error(y_test,y_pred))
      print('Mean Squared Error:',metrics.mean_squared_error(y_test,y_pred))
```

```
print('Root Mean Squared Error:',np.sqrt(metrics.  
↪mean_squared_error(y_test,y_pred)))  
  
print('r2_score:',r2_score(y_test,y_pred))
```

Mean Absolute Error: 4377.691157379052  
Mean Squared Error: 72874832.17557167  
Root Mean Squared Error: 8536.675709875108  
r2\_score: 0.9952772025844584

```
[21]: ax1 = sns.distplot(y, hist=False, color="r", label="Actual Value")  
sns.distplot(y_pred, hist=False, color="b", label="Predicted Values" , ax=ax1)
```

[21]: <matplotlib.axes.\_subplots.AxesSubplot at 0x20444936e20>

