In [ ]:

```
from google.colab import files
uploaded = files.upload()
```

Choose File No file selected

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving crop_production.csv to crop_production.csv
```

In [ ]:

```
import os
import json
import numpy as np
import pandas as pd
from pandas.io.json import json_normalize
import matplotlib.pyplot as plt
import seaborn as sns
color = sns.color_palette()

%matplotlib inline

from plotly import tools
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go

from sklearn import model_selection, preprocessing, metrics

pd.options.mode.chained_assignment = None
pd.options.display.max_columns = 999
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning:
pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

In [ ]:

```
import pandas as pd
train_df=pd.read_csv('crop_production.csv')
```

In [ ]:

```
train_df.dtypes
```

Out[ ]:

```
State_Name        object
District_Name     object
Crop_Year          int64
Season            object
Crop              object
Area             float64
Production       float64
dtype: object
```

In [ ]:

```
train_df.head()
```

Out[ ]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
	Andaman and Nicobar						

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0

In [ ]:

```
train_df.describe()
```

Out[ ]:

	Crop_Year	Area	Production
count	246091.000000	2.460910e+05	2.423610e+05
mean	2005.643018	1.200282e+04	5.825034e+05
std	4.952164	5.052340e+04	1.706581e+07
min	1997.000000	4.000000e-02	0.000000e+00
25%	2002.000000	8.000000e+01	8.800000e+01
50%	2006.000000	5.820000e+02	7.290000e+02
75%	2010.000000	4.392000e+03	7.023000e+03
max	2015.000000	8.580100e+06	1.250800e+09

In [ ]:

```
# Null value in database
train_df.isnull().any()
```

Out[ ]:

```
State_Name        False
District_Name     False
Crop_Year         False
Season            False
Crop              False
Area              False
Production        False
dtype: bool
```

In [ ]:

```
train_df['Production'].fillna("0",inplace=True)
train_df.Production.isnull().any()
```

Out[ ]:

```
False
```

In [ ]:

```
#Unique value in database
for col in train_df.columns:
  print(train_df[col].unique())
  print('='*100)
```

```
['Andaman and Nicobar Islands' 'Andhra Pradesh' 'Arunachal Pradesh'
 'Assam' 'Bihar' 'Chandigarh' 'Chhattisgarh' 'Dadra and Nagar Haveli'
 'Goa' 'Gujarat' 'Haryana' 'Himachal Pradesh' 'Jammu and Kashmir '
 'Jharkhand' 'Karnataka' 'Kerala' 'Madhya Pradesh' 'Maharashtra' 'Manipur'
 'Meghalaya' 'Mizoram' 'Nagaland' 'Odisha' 'Puducherry' 'Punjab'
 'Rajasthan' 'Sikkim' 'Tamil Nadu' 'Telangana ' 'Tripura' 'Uttar Pradesh'
```

```
 'Uttarakhand' 'West Bengal']
==========================================================================================================

['NICOBARS' 'NORTH AND MIDDLE ANDAMAN' 'SOUTH ANDAMANS' 'ANANTAPUR'
 'CHITTOOR' 'EAST GODAVARI' 'GUNTUR' 'KADAPA' 'KRISHNA' 'KURNOOL'
 'PRAKASAM' 'SPSR NELLORE' 'SRIKAKULAM' 'VISAKHAPATANAM' 'VIZIANAGARAM'
 'WEST GODAVARI' 'ANJAW' 'CHANGLANG' 'DIBANG VALLEY' 'EAST KAMENG'
 'EAST SIANG' 'KURUNG KUMEY' 'LOHIT' 'LONGDING' 'LOWER DIBANG VALLEY'
 'LOWER SUBANSIRI' 'NAMSAI' 'PAPUM PARE' 'TAWANG' 'TIRAP' 'UPPER SIANG'
 'UPPER SUBANSIRI' 'WEST KAMENG' 'WEST SIANG' 'BAKSA' 'BARPETA'
 'BONGAIGAON' 'CACHAR' 'CHIRANG' 'DARRANG' 'DHEMAJI' 'DHUBRI' 'DIBRUGARH'
 'DIMA HASAO' 'GOALPARA' 'GOLAGHAT' 'HAILAKANDI' 'JORHAT' 'KAMRUP'
 'KAMRUP METRO' 'KARBI ANGLONG' 'KARIMGANJ' 'KOKRAJHAR' 'LAKHIMPUR'
 'MARIGAON' 'NAGAON' 'NALBARI' 'SIVASAGAR' 'SONITPUR' 'TINSUKIA'
 'UDALGURI' 'ARARIA' 'ARWAL' 'AURANGABAD' 'BANKA' 'BEGUSARAI' 'BHAGALPUR'
 'BHOJPUR' 'BUXAR' 'DARBHANGA' 'GAYA' 'GOPALGANJ' 'JAMUI' 'JEHANABAD'
 'KAIMUR (BHABUA)' 'KATIHAR' 'KHAGARIA' 'KISHANGANJ' 'LAKHISARAI'
 'MADHEPURA' 'MADHUBANI' 'MUNGER' 'MUZAFFARPUR' 'NALANDA' 'NAWADA'
 'PASHCHIM CHAMPARAN' 'PATNA' 'PURBI CHAMPARAN' 'PURNIA' 'ROHTAS'
 'SAHARSA' 'SAMASTIPUR' 'SARAN' 'SHEIKHPURA' 'SHEOHAR' 'SITAMARHI' 'SIWAN'
 'SUPAUL' 'VAISHALI' 'CHANDIGARH' 'BALOD' 'BALODA BAZAR' 'BALRAMPUR'
 'BASTAR' 'BEMETARA' 'BIJAPUR' 'BILASPUR' 'DANTEWADA' 'DHAMTARI' 'DURG'
 'GARIYABAND' 'JANJGIR-CHAMPA' 'JASHPUR' 'KABIRDHAM' 'KANKER' 'KONDAGAON'
 'KORBA' 'KOREA' 'MAHASAMUND' 'MUNGELI' 'NARAYANPUR' 'RAIGARH' 'RAIPUR'
 'RAJNANDGAON' 'SUKMA' 'SURAJPUR' 'SURGUJA' 'DADRA AND NAGAR HAVELI'
 'NORTH GOA' 'SOUTH GOA' 'AHMADABAD' 'AMRELI' 'ANAND' 'BANAS KANTHA'
 'BHARUCH' 'BHAVNAGAR' 'DANG' 'DOHAD' 'GANDHINAGAR' 'JAMNAGAR' 'JUNAGADH'
 'KACHCHH' 'KHEDA' 'MAHESANA' 'NARMADA' 'NAVSARI' 'PANCH MAHALS' 'PATAN'
 'PORBANDAR' 'RAJKOT' 'SABAR KANTHA' 'SURAT' 'SURENDRANAGAR' 'TAPI'
 'VADODARA' 'VALSAD' 'AMBALA' 'BHIWANI' 'FARIDABAD' 'FATEHABAD' 'GURGAON'
 'HISAR' 'JHAJJAR' 'JIND' 'KAITHAL' 'KARNAL' 'KURUKSHETRA' 'MAHENDRAGARH'
 'MEWAT' 'PALWAL' 'PANCHKULA' 'PANIPAT' 'REWARI' 'ROHTAK' 'SIRSA'
 'SONIPAT' 'YAMUNANAGAR' 'CHAMBA' 'HAMIRPUR' 'KANGRA' 'KINNAUR' 'KULLU'
 'LAHUL AND SPITI' 'MANDI' 'SHIMLA' 'SIRMAUR' 'SOLAN' 'UNA' 'ANANTNAG'
 'BADGAM' 'BANDIPORA' 'BARAMULLA' 'DODA' 'GANDERBAL' 'JAMMU' 'KARGIL'
 'KATHUA' 'KISHTWAR' 'KULGAM' 'KUPWARA' 'LEH LADAKH' 'POONCH' 'PULWAMA'
 'RAJAURI' 'RAMBAN' 'REASI' 'SAMBA' 'SHOPIAN' 'SRINAGAR' 'UDHAMPUR'
 'BOKARO' 'CHATRA' 'DEOGHAR' 'DHANBAD' 'DUMKA' 'EAST SINGHBUM' 'GARHWA'
 'GIRIDIH' 'GODDA' 'GUMLA' 'HAZARIBAGH' 'JAMTARA' 'KHUNTI' 'KODERMA'
 'LATEHAR' 'LOHARDAGA' 'PAKUR' 'PALAMU' 'RAMGARH' 'RANCHI' 'SAHEBGANJ'
 'SARAIKELA KHARSAWAN' 'SIMDEGA' 'WEST SINGHBHUM' 'BAGALKOT'
 'BANGALORE RURAL' 'BELGAUM' 'BELLARY' 'BENGALURU URBAN' 'BIDAR'
 'CHAMARAJANAGAR' 'CHIKBALLAPUR' 'CHIKMAGALUR' 'CHITRADURGA'
 'DAKSHIN KANNAD' 'DAVANGERE' 'DHARWAD' 'GADAG' 'GULBARGA' 'HASSAN'
 'HAVERI' 'KODAGU' 'KOLAR' 'KOPPAL' 'MANDYA' 'MYSORE' 'RAICHUR'
 'RAMANAGARA' 'SHIMOGA' 'TUMKUR' 'UDUPI' 'UTTAR KANNAD' 'YADGIR'
 'ALAPPUZHA' 'ERNAKULAM' 'IDUKKI' 'KANNUR' 'KASARAGOD' 'KOLLAM' 'KOTTAYAM'
 'KOZHIKODE' 'MALAPPURAM' 'PALAKKAD' 'PATHANAMTHITTA' 'THIRUVANANTHAPURAM'
 'THRISSUR' 'WAYANAD' 'AGAR MALWA' 'ALIRAJPUR' 'ANUPPUR' 'ASHOKNAGAR'
 'BALAGHAT' 'BARWANI' 'BETUL' 'BHIND' 'BHOPAL' 'BURHANPUR' 'CHHATARPUR'
 'CHHINDWARA' 'DAMOH' 'DATIA' 'DEWAS' 'DHAR' 'DINDORI' 'GUNA' 'GWALIOR'
 'HARDA' 'HOSHANGABAD' 'INDORE' 'JABALPUR' 'JHABUA' 'KATNI' 'KHANDWA'
 'KHARGONE' 'MANDLA' 'MANDSAUR' 'MORENA' 'NARSINGHPUR' 'NEEMUCH' 'PANNA'
 'RAISEN' 'RAJGARH' 'RATLAM' 'REWA' 'SAGAR' 'SATNA' 'SEHORE' 'SEONI'
 'SHAHDOL' 'SHAJAPUR' 'SHEOPUR' 'SHIVPURI' 'SIDHI' 'SINGRAULI' 'TIKAMGARH'
 'UJJAIN' 'UMARIA' 'VIDISHA' 'AHMEDNAGAR' 'AKOLA' 'AMRAVATI' 'BEED'
 'BHANDARA' 'BULDHANA' 'CHANDRAPUR' 'DHULE' 'GADCHIROLI' 'GONDIA'
 'HINGOLI' 'JALGAON' 'JALNA' 'KOLHAPUR' 'LATUR' 'MUMBAI' 'NAGPUR' 'NANDED'
 'NANDURBAR' 'NASHIK' 'OSMANABAD' 'PALGHAR' 'PARBHANI' 'PUNE' 'RAIGAD'
 'RATNAGIRI' 'SANGLI' 'SATARA' 'SINDHUDURG' 'SOLAPUR' 'THANE' 'WARDHA'
 'WASHIM' 'YAVATMAL' 'BISHNUPUR' 'CHANDEL' 'CHURACHANDPUR' 'IMPHAL EAST'
 'IMPHAL WEST' 'SENAPATI' 'TAMENGLONG' 'THOUBAL' 'UKHRUL'
 'EAST GARO HILLS' 'EAST JAINTIA HILLS' 'EAST KHASI HILLS'
 'NORTH GARO HILLS' 'RI BHOI' 'SOUTH GARO HILLS' 'SOUTH WEST GARO HILLS'
 'SOUTH WEST KHASI HILLS' 'WEST GARO HILLS' 'WEST JAINTIA HILLS'
 'WEST KHASI HILLS' 'AIZAWL' 'CHAMPHAI' 'KOLASIB' 'LAWNGTLAI' 'LUNGLEI'
 'MAMIT' 'SAIHA' 'SERCHHIP' 'DIMAPUR' 'KIPHIRE' 'KOHIMA' 'LONGLENG'
 'MOKOKCHUNG' 'MON' 'PEREN' 'PHEK' 'TUENSANG' 'WOKHA' 'ZUNHEBOTO' 'ANUGUL'
 'BALANGIR' 'BALESHWAR' 'BARGARH' 'BHADRAK' 'BOUDH' 'CUTTACK' 'DEOGARH'
 'DHENKANAL' 'GAJAPATI' 'GANJAM' 'JAGATSINGHAPUR' 'JAJAPUR' 'JHARSUGUDA'
 'KALAHANDI' 'KANDHAMAL' 'KENDRAPARA' 'KENDUJHAR' 'KHORDHA' 'KORAPUT'
 'MALKANGIRI' 'MAYURBHANJ' 'NABARANGPUR' 'NAYAGARH' 'NUAPADA' 'PURI'
 'RAYAGADA' 'SAMBALPUR' 'SONEPUR' 'SUNDARGARH' 'KARAIKAL' 'MAHE'
 'PONDICHERRY' 'YANAM' 'AMRITSAR' 'BARNALA' 'BATHINDA' 'FARIDKOT'
 'FATEHGARH SAHIB' 'FAZILKA' 'FIROZEPUR' 'GURDASPUR' 'HOSHIARPUR'
 'JALANDHAR' 'KAPURTHALA' 'LUDHIANA' 'MANSA' 'MOGA' 'MUKTSAR' 'NAWANSHAHR'
 'PATHANKOT' 'PATIALA' 'RUPNAGAR' 'S.A.S NAGAR' 'SANGRUR' 'TARN TARAN'
```

```
 'AJMER' 'ALWAR' 'BANSWARA' 'BARAN' 'BARMER' 'BHARATPUR' 'BHILWARA'
 'BIKANER' 'BUNDI' 'CHITTORGARH' 'CHURU' 'DAUSA' 'DHOLPUR' 'DUNGARPUR'
 'GANGANAGAR' 'HANUMANGARH' 'JAIPUR' 'JAISALMER' 'JALORE' 'JHALAWAR'
 'JHUNJHUNU' 'JODHPUR' 'KARAULI' 'KOTA' 'NAGAUR' 'PALI' 'PRATAPGARH'
 'RAJSAMAND' 'SAWAI MADHOPUR' 'SIKAR' 'SIROHI' 'TONK' 'UDAIPUR'
 'EAST DISTRICT' 'NORTH DISTRICT' 'SOUTH DISTRICT' 'WEST DISTRICT'
 'ARIYALUR' 'COIMBATORE' 'CUDDALORE' 'DHARMAPURI' 'DINDIGUL' 'ERODE'
 'KANCHIPURAM' 'KANNIYAKUMARI' 'KARUR' 'KRISHNAGIRI' 'MADURAI'
 'NAGAPATTINAM' 'NAMAKKAL' 'PERAMBALUR' 'PUDUKKOTTAI' 'RAMANATHAPURAM'
 'SALEM' 'SIVAGANGA' 'THANJAVUR' 'THE NILGIRIS' 'THENI' 'THIRUVALLUR'
 'THIRUVARUR' 'TIRUCHIRAPPALLI' 'TIRUNELVELI' 'TIRUPPUR' 'TIRUVANNAMALAI'
 'TUTICORIN' 'VELLORE' 'VILLUPURAM' 'VIRUDHUNAGAR' 'ADILABAD' 'HYDERABAD'
 'KARIMNAGAR' 'KHAMMAM' 'MAHBUBNAGAR' 'MEDAK' 'NALGONDA' 'NIZAMABAD'
 'RANGAREDDI' 'WARANGAL' 'DHALAI' 'GOMATI' 'KHOWAI' 'NORTH TRIPURA'
 'SEPAHIJALA' 'SOUTH TRIPURA' 'UNAKOTI' 'WEST TRIPURA' 'AGRA' 'ALIGARH'
 'ALLAHABAD' 'AMBEDKAR NAGAR' 'AMETHI' 'AMROHA' 'AURAIYA' 'AZAMGARH'
 'BAGHPAT' 'BAHRAICH' 'BALLIA' 'BANDA' 'BARABANKI' 'BAREILLY' 'BASTI'
 'BIJNOR' 'BUDAUN' 'BULANDSHAHR' 'CHANDAULI' 'CHITRAKOOT' 'DEORIA' 'ETAH'
 'ETAWAH' 'FAIZABAD' 'FARRUKHABAD' 'FATEHPUR' 'FIROZABAD'
 'GAUTAM BUDDHA NAGAR' 'GHAZIABAD' 'GHAZIPUR' 'GONDA' 'GORAKHPUR' 'HAPUR'
 'HARDOI' 'HATHRAS' 'JALAUN' 'JAUNPUR' 'JHANSI' 'KANNAUJ' 'KANPUR DEHAT'
 'KANPUR NAGAR' 'KASGANJ' 'KAUSHAMBI' 'KHERI' 'KUSHI NAGAR' 'LALITPUR'
 'LUCKNOW' 'MAHARAJGANJ' 'MAHOBA' 'MAINPURI' 'MATHURA' 'MAU' 'MEERUT'
 'MIRZAPUR' 'MORADABAD' 'MUZAFFARNAGAR' 'PILIBHIT' 'RAE BARELI' 'RAMPUR'
 'SAHARANPUR' 'SAMBHAL' 'SANT KABEER NAGAR' 'SANT RAVIDAS NAGAR'
 'SHAHJAHANPUR' 'SHAMLI' 'SHRAVASTI' 'SIDDHARTH NAGAR' 'SITAPUR'
 'SONBHADRA' 'SULTANPUR' 'UNNAO' 'VARANASI' 'ALMORA' 'BAGESHWAR' 'CHAMOLI'
 'CHAMPAWAT' 'DEHRADUN' 'HARIDWAR' 'NAINITAL' 'PAURI GARHWAL'
 'PITHORAGARH' 'RUDRA PRAYAG' 'TEHRI GARHWAL' 'UDAM SINGH NAGAR'
 'UTTAR KASHI' '24 PARAGANAS NORTH' '24 PARAGANAS SOUTH' 'BANKURA'
 'BARDHAMAN' 'BIRBHUM' 'COOCHBEHAR' 'DARJEELING' 'DINAJPUR DAKSHIN'
 'DINAJPUR UTTAR' 'HOOGHLY' 'HOWRAH' 'JALPAIGURI' 'MALDAH'
 'MEDINIPUR EAST' 'MEDINIPUR WEST' 'MURSHIDABAD' 'NADIA' 'PURULIA']
==========================================================================================================

[2000 2001 2002 2003 2004 2005 2006 2010 1997 1998 1999 2007 2008 2009
 2011 2012 2013 2014 2015]
==========================================================================================================

['Kharif     ' 'Whole Year ' 'Autumn     ' 'Rabi       ' 'Summer     '
 'Winter     ']
==========================================================================================================

['Arecanut' 'Other Kharif pulses' 'Rice' 'Banana' 'Cashewnut' 'Coconut '
 'Dry ginger' 'Sugarcane' 'Sweet potato' 'Tapioca' 'Black pepper'
 'Dry chillies' 'other oilseeds' 'Turmeric' 'Maize' 'Moong(Green Gram)'
 'Urad' 'Arhar/Tur' 'Groundnut' 'Sunflower' 'Bajra' 'Castor seed'
 'Cotton(lint)' 'Horse-gram' 'Jowar' 'Korra' 'Ragi' 'Tobacco' 'Gram'
 'Wheat' 'Masoor' 'Sesamum' 'Linseed' 'Safflower' 'Onion'
 'other misc. pulses' 'Samai' 'Small millets' 'Coriander' 'Potato'
 'Other  Rabi pulses' 'Soyabean' 'Beans & Mutter(Vegetable)' 'Bhindi'
 'Brinjal' 'Citrus Fruit' 'Cucumber' 'Grapes' 'Mango' 'Orange'
 'other fibres' 'Other Fresh Fruits' 'Other Vegetables' 'Papaya'
 'Pome Fruit' 'Tomato' 'Rapeseed &Mustard' 'Mesta' 'Cowpea(Lobia)' 'Lemon'
 'Pome Granet' 'Sapota' 'Cabbage' 'Peas  (vegetable)' 'Niger seed'
 'Bottle Gourd' 'Sannhamp' 'Varagu' 'Garlic' 'Ginger' 'Oilseeds total'
 'Pulses total' 'Jute' 'Peas & beans (Pulses)' 'Blackgram' 'Paddy'
 'Pineapple' 'Barley' 'Khesari' 'Guar seed' 'Moth'
 'Other Cereals & Millets' 'Cond-spcs other' 'Turnip' 'Carrot' 'Redish'
 'Arcanut (Processed)' 'Atcanut (Raw)' 'Cashewnut Processed'
 'Cashewnut Raw' 'Cardamom' 'Rubber' 'Bitter Gourd' 'Drum Stick'
 'Jack Fruit' 'Snak Guard' 'Pump Kin' 'Tea' 'Coffee' 'Cauliflower'
 'Other Citrus Fruit' 'Water Melon' 'Total foodgrain' 'Kapas' 'Colocosia'
 'Lentil' 'Bean' 'Jobster' 'Perilla' 'Rajmash Kholar' 'Ricebean (nagadal)'
 'Ash Gourd' 'Beet Root' 'Lab-Lab' 'Ribed Guard' 'Yam' 'Apple' 'Peach'
 'Pear' 'Plums' 'Litchi' 'Ber' 'Other Dry Fruit' 'Jute & mesta']
==========================================================================================================

[1.25400e+03 2.00000e+00 1.02000e+02 ... 3.02274e+05 1.14930e+04
 2.79151e+05]
==========================================================================================================

[2000.0 1.0 321.0 ... 729553.0 730136.0 597899.0]
==========================================================================================================
```

In [ ]:

```
#Replace the value
train_df['Production'].value_counts()
train_df['Production']=train_df['Production'].replace('=','0')
```

In [ ]:

```
#Missing values in dataset
for col in train_df.columns:
    nullrow=train_df[col].isnull().sum()
    notrow=train_df[col].notnull().sum()
    percentage=(nullrow*100)/(nullrow+notrow)
    print("Column is ",col,percentage,"% Missing Values")
```

```
Column is  State_Name 0.0 % Missing Values
Column is  District_Name 0.0 % Missing Values
Column is  Crop_Year 0.0 % Missing Values
Column is  Season 0.0 % Missing Values
Column is  Crop 0.0 % Missing Values
Column is  Area 0.0 % Missing Values
Column is  Production 1.5156994770227274 % Missing Values
```

In [ ]:

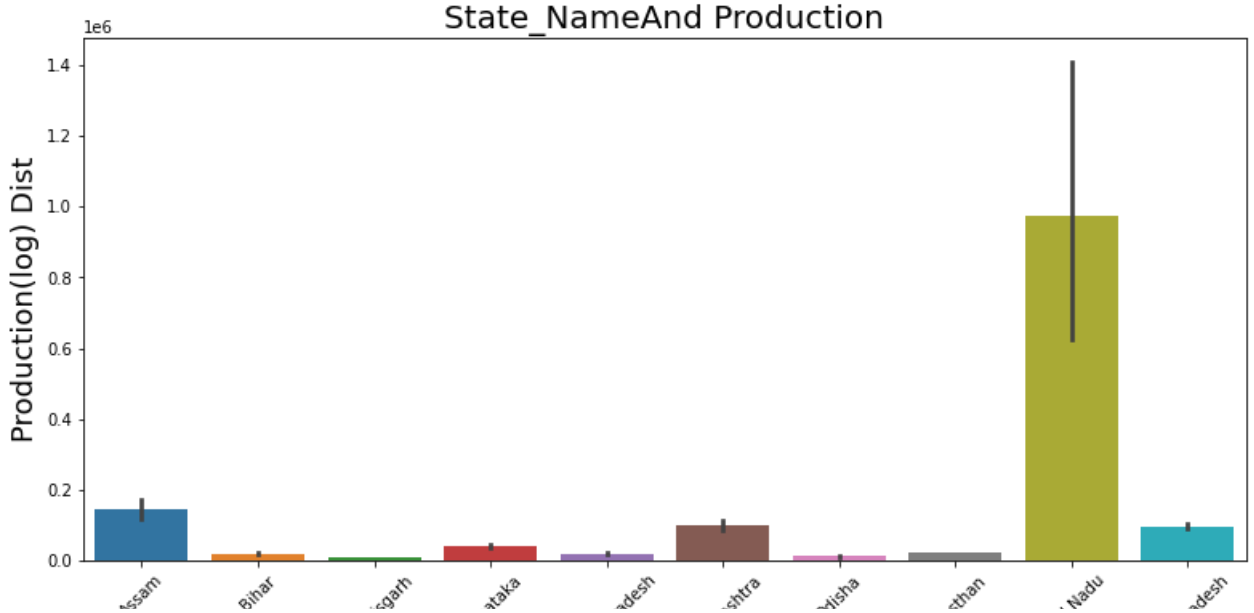
```
col=train_df.columns[:6]
col
```

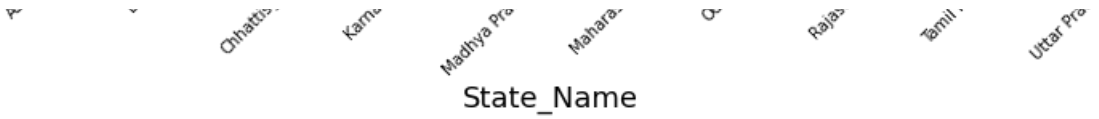
Out[ ]:

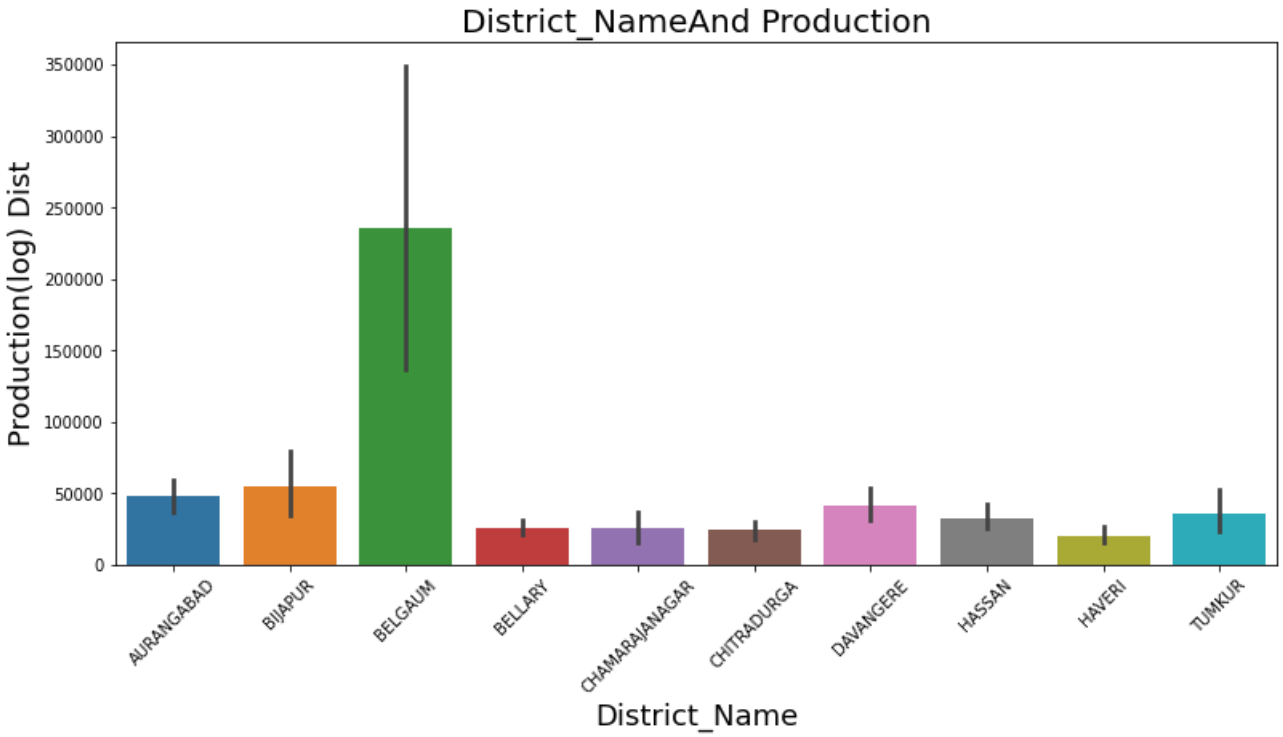
```
Index(['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area'], dtype='object')
```

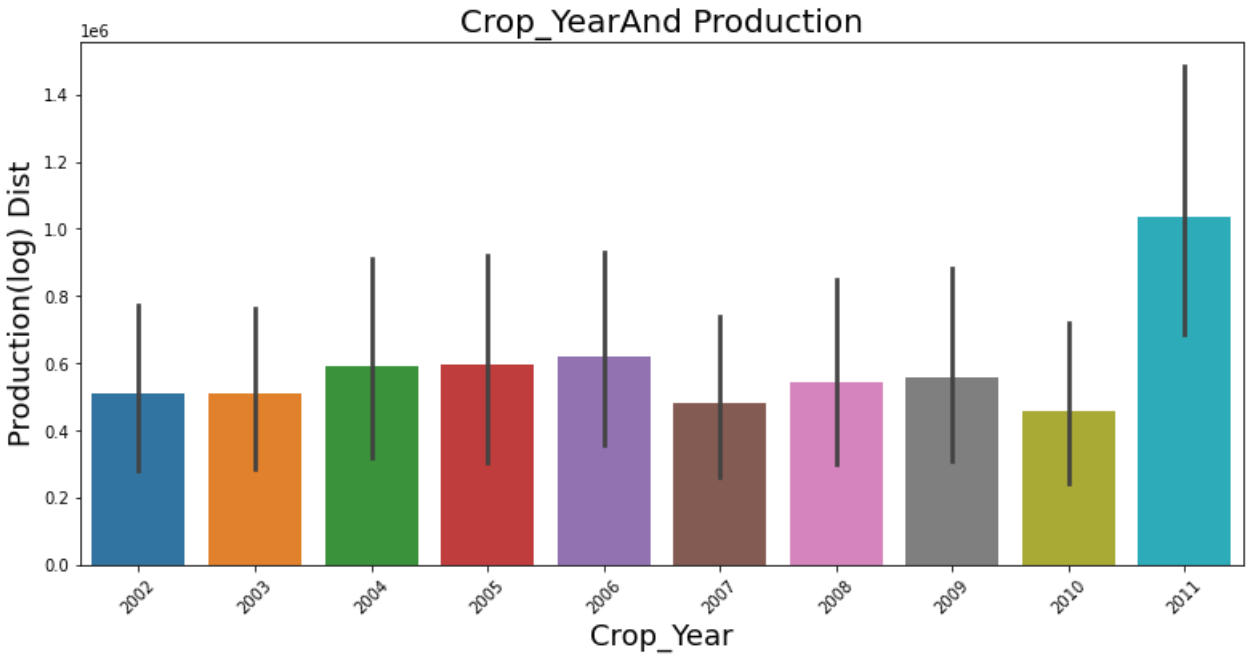
In [ ]:

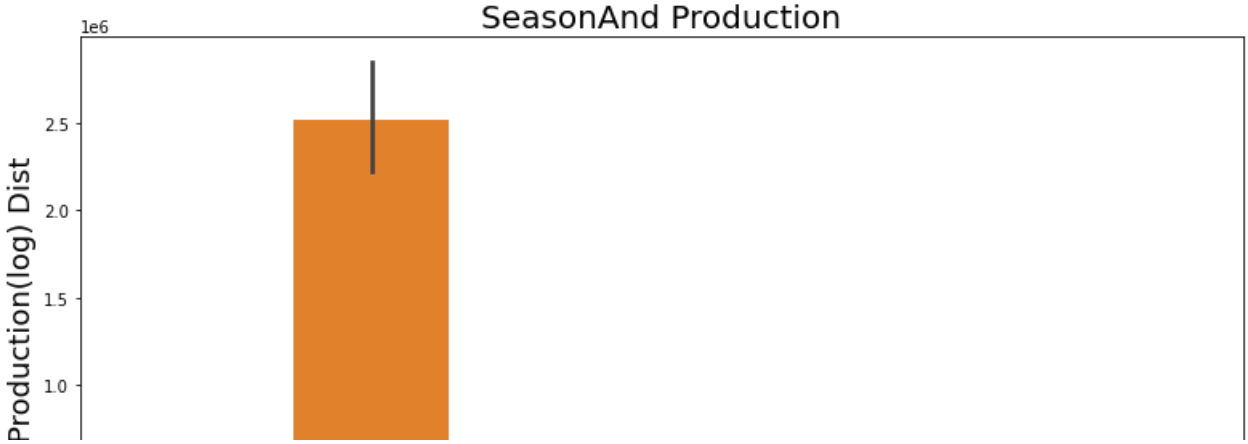
```
# State Name and Production
train_df["Production"] = train_df["Production"].astype('float')

for col in train_df.columns[:6]:
  plt.figure(figsize=(13,6)) #figure size
  #It's another way to plot our data. using a variable that contains the plot parameters
  g1 = sns.barplot(x=col, y='Production',data=train_df[(train_df[col].isin((train_df[col].value_cou
nts()[:10].index.values))) & train_df['Production'] > 0])
  g1.set_title(col+'And Production', fontsize=20) # title and fontsize
  g1.set_xticklabels(g1.get_xticklabels(),rotation=45) # It's the way to rotate the xticks when we
use variable to our graphs
  g1.set_xlabel(col, fontsize=18) # Xlabel
  g1.set_ylabel('Production(log) Dist', fontsize=18) #Ylabel
  plt.show()
```

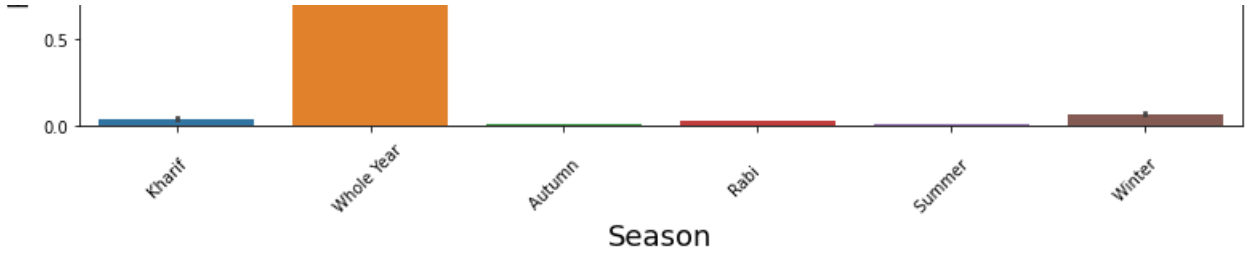


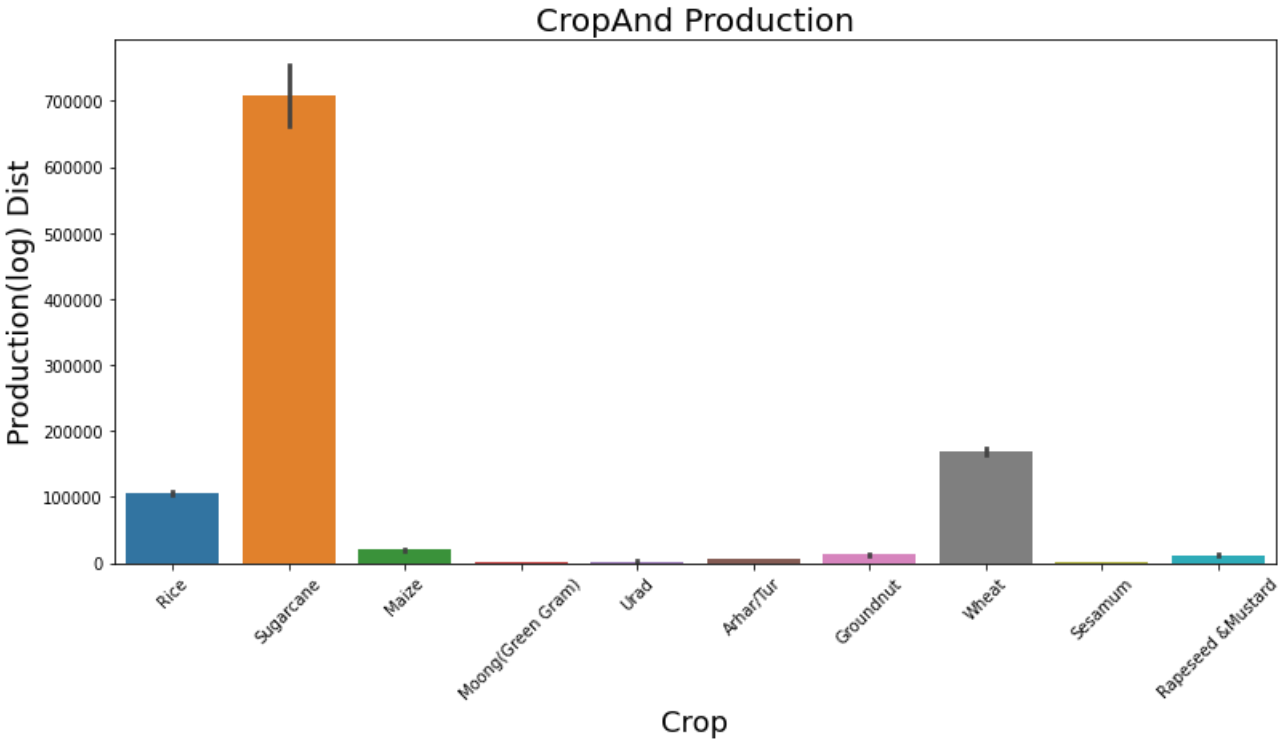

State_Name

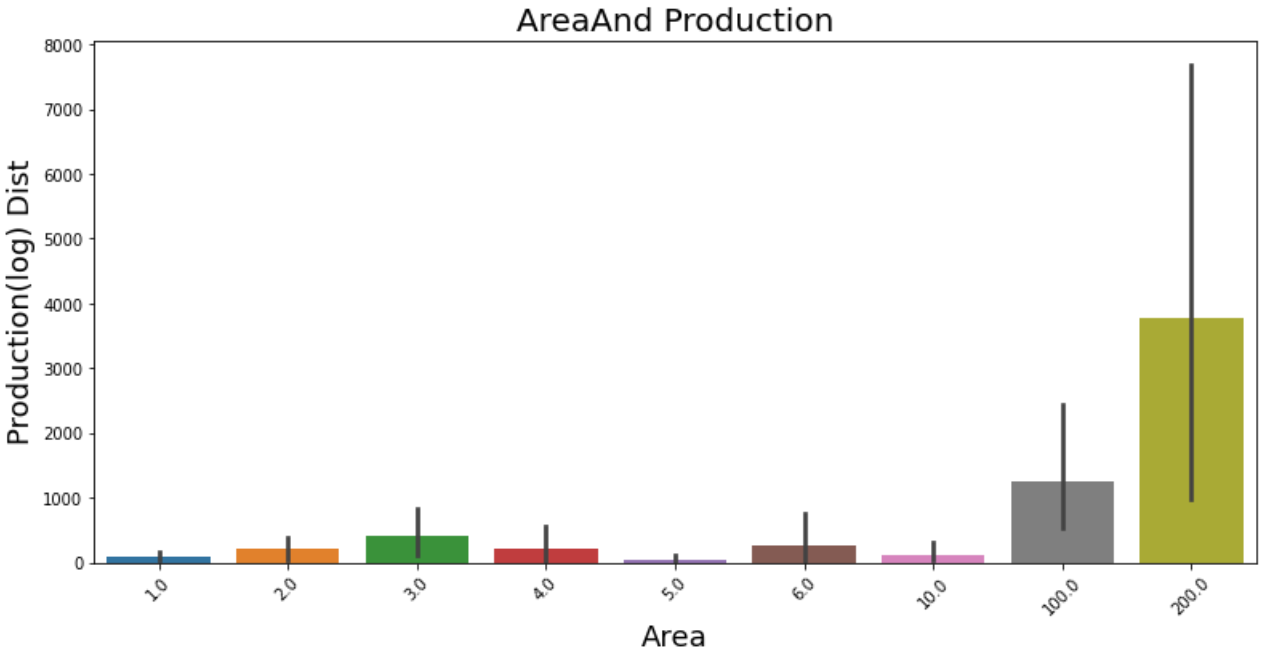

District_NameAnd Production
Production(log) Dist
350000
300000
250000
200000
150000
100000
50000
0
AURANGABAD
BIJAPUR
BELGAUM
BELLARY
CHAMARAJANAGAR
CHITRADURGA
DAVANGERE
HASSAN
HAVERI
TUMKUR
District_Name


Crop_YearAnd Production
1e6
Production(log) Dist
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
Crop_Year


SeasonAnd Production
1e6
Production(log) Dist
2.5
2.0
1.5
1.0







Split the data

In [ ]:

```
import pandas as pd
from sklearn import datasets, linear_model
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
```

In [ ]:

```
train_df["Production"] = train_df["Production"].astype('float')

y=train_df['Production']
X_train, X_test, y_train, y_test = train_test_split(train_df, y, test_size=0.3)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```
(172263, 7) (172263,)
(73828, 7) (73828,)
```

In [ ]:

```
X_train.drop(['Production'], axis=1,inplace=True)
X_test.drop(['Production'],axis=1,inplace=True)
X_test.shape
```

Out[ ]:

```
(73828, 6)
```

Categorical variable to One hot encoding

In [ ]:

```
#One hot encoding of State_Name
col='State_Name'
#Training data
from sklearn.feature_extraction.text import CountVectorizer
train_state=list(X_train['State_Name'].unique())

vectorizer_col = CountVectorizer(vocabulary=train_state, lowercase=False, binary=True)
vectorizer_col.fit(X_train[col].values)
print(vectorizer_col.get_feature_names())
train_state_one_hot = vectorizer_col.transform(X_train[col].values)
print("Shape of matrix after one hot encodig ",train_state_one_hot.shape)
#For Cross validating data
#cv_state_one_hot = vectorizer_col.transform(X_cv[col].values)
#print("Shape of matrix after one hot encodig ",cv_state_one_hot.shape)

  # For testing data
test_state_one_hot= vectorizer_col.transform(X_test[col].values)
print("Shape of matrix after one hot encodig ",test_state_one_hot.shape)
```

```
['Andhra Pradesh', 'Uttarakhand', 'Rajasthan', 'Mizoram', 'Chhattisgarh', 'Odisha', 'Nagaland', 'M
adhya Pradesh', 'Kerala', 'Uttar Pradesh', 'Punjab', 'Haryana', 'Karnataka', 'Assam', 'Bihar', 'Tr
ipura', 'Maharashtra', 'Tamil Nadu', 'Gujarat', 'Himachal Pradesh', 'West Bengal', 'Telangana ', '
Manipur', 'Jammu and Kashmir ', 'Meghalaya', 'Puducherry', 'Jharkhand', 'Sikkim', 'Arunachal
Pradesh', 'Andaman and Nicobar Islands', 'Dadra and Nagar Haveli', 'Chandigarh', 'Goa']
Shape of matrix after one hot encodig  (172263, 33)
Shape of matrix after one hot encodig  (73828, 33)
```

In [ ]:

```
col='District_Name'
#Training data
from sklearn.feature_extraction.text import CountVectorizer
train_district=list(X_train['District_Name'].unique())
vectorizer_col = CountVectorizer(vocabulary=train_district, lowercase=False, binary=True)
vectorizer_col.fit(X_train[col].values)
print(vectorizer_col.get_feature_names())
train_district_one_hot = vectorizer_col.transform(X_train[col].values)
print("Shape of matrix after one hot encodig ",train_district_one_hot.shape)
#For Cross validating data
#cv_district_one_hot = vectorizer_col.transform(X_cv[col].values)
#print("Shape of matrix after one hot encodig ",cv_district_one_hot.shape)

  # For testing data
test_district_one_hot= vectorizer_col.transform(X_test[col].values)
```

```
print("Shape of matrix after one hot encodig ",test_district_one_hot.shape)
```

```
['KRISHNA', 'PAURI GARHWAL', 'BHARATPUR', 'LUNGLEI', 'DANTEWADA', 'SONEPUR', 'WOKHA',
'HOSHANGABAD', 'WEST GODAVARI', 'KANNUR', 'JHALAWAR', 'MAHASAMUND', 'KUSHI NAGAR', 'KANDHAMAL',
'MAYURBHANJ', 'S.A.S NAGAR', 'GURGAON', 'KORAPUT', 'HARDA', 'DHENKANAL', 'SHIMOGA', 'SHAJAPUR', 'P
RAKASAM', 'UDAIPUR', 'MANDSAUR', 'JAIPUR', 'HARDOI', 'BALLIA', 'BETUL', 'BALODA BAZAR', 'KARBI ANG
LONG', 'PRATAPGARH', 'BHOJPUR', 'CACHAR', 'BUDAUN', 'WEST TRIPURA', 'JAISALMER', 'SANGLI',
'SIDDHARTH NAGAR', 'VISAKHAPATANAM', 'KAMRUP', 'BELGAUM', 'NAGPUR', 'WASHIM', 'RUDRA PRAYAG', 'VIR
UDHUNAGAR', 'SEHORE', 'CHAMPHAI', 'RUPNAGAR', 'MOKOKCHUNG', 'PEREN', 'DHARMAPURI', 'UDALGURI', 'MA
HENDRAGARH', 'VALSAD', 'KENDUJHAR', 'KANGRA', 'BANGALORE RURAL', 'MURSHIDABAD', 'MADHEPURA',
'MANDYA', 'KURNOOL', 'THIRUVALLUR', 'VELLORE', 'MUZAFFARPUR', 'KARAULI', 'BANKURA', 'MAHBUBNAGAR',
'ERODE', 'CHANDEL', 'SAHARSA', 'DAUSA', 'PUNE', 'NABARANGPUR', 'NAINITAL', 'BALAGHAT', 'GONDA', 'W
ARDHA', 'SURAT', 'SIVAGANGA', 'GADAG', 'GULBARGA', 'BURHANPUR', 'KOZHIKODE', 'GOPALGANJ',
'SRINAGAR', 'CHHATARPUR', 'FATEHABAD', 'RAIPUR', 'WEST GARO HILLS', 'NADIA', 'CHITTORGARH',
'JODHPUR', 'CHITTOOR', 'REWARI', 'UNNAO', 'SOLAN', 'BARAMULLA', 'KASARAGOD', 'ALWAR', 'DANG', 'JEH
ANABAD', 'BARPETA', 'PITHORAGARH', 'FARRUKHABAD', 'PONDICHERRY', 'ANANTAPUR', 'PURBI CHAMPARAN', '
CHITRADURGA', 'HASSAN', 'CHAMPAWAT', 'BIDAR', 'PURULIA', 'YAMUNANAGAR', 'CHIKMAGALUR', 'NORTH TRIP
URA', 'DHALAI', 'MADURAI', 'GANJAM', 'TONK', 'PERAMBALUR', 'MATHURA', 'SUNDARGARH', 'BASTAR',
'WEST JAINTIA HILLS', 'SANT RAVIDAS NAGAR', 'KHERI', 'GAYA', 'GONDIA', 'BIJNOR', 'JHABUA',
'BHAVNAGAR', 'BHIND', 'MANDLA', 'JALNA', 'SATNA', 'TINSUKIA', 'CHHINDWARA', 'UMARIA', 'JAMNAGAR',
'HAMIRPUR', 'BILASPUR', 'BHAGALPUR', 'BALRAMPUR', 'MARIGAON', 'BARMER', 'PILIBHIT', 'NASHIK', 'CUT
TACK', 'NAMAKKAL', 'SANT KABEER NAGAR', 'BHANDARA', 'DEORIA', 'NANDED', 'KRISHNAGIRI', 'BAREILLY',
'BHIWANI', 'DHANBAD', 'BHARUCH', 'PHEK', 'ALMORA', 'KISHANGANJ', 'AURAIYA', 'ETAH', 'DIBRUGARH',
'DEWAS', 'UNAKOTI', 'MADHUBANI', 'SUPAUL', 'DAVANGERE', 'RAMPUR', 'AMRITSAR', 'JAGATSINGHAPUR', 'J
ANJGIR-CHAMPA', 'CUDDALORE', 'BASTI', 'BAGALKOT', 'KALAHANDI', 'SARAN', 'NALGONDA', 'NARSINGHPUR',
'JORHAT', 'TUENSANG', 'AZAMGARH', 'KOPPAL', 'ERNAKULAM', 'EAST GODAVARI', 'MYSORE', 'BANDA', 'PATA
N', 'JAJAPUR', 'MALKANGIRI', 'MAHOBA', 'KAIMUR (BHABUA)', 'GOALPARA', 'BOUDH', 'DINDIGUL',
'BEGUSARAI', 'UDHAMPUR', 'DURG', 'THANJAVUR', 'UJJAIN', 'DHEMAJI', 'PURNIA', 'ALAPPUZHA',
'RAMANAGARA', 'ETAWAH', 'KODAGU', 'NALBARI', 'JASHPUR', 'SIKAR', 'KAMRUP METRO', 'SHEOPUR',
'THIRUVANANTHAPURAM', 'HAILAKANDI', 'MEDAK', 'KOLLAM', 'VAISHALI', 'BULANDSHAHR', 'MEERUT', 'SPSR
NELLORE', 'SHRAVASTI', 'SIROHI', 'SAGAR', 'KANNAUJ', 'SIVASAGAR', 'RAJKOT', 'KHAGARIA',
'JALPAIGURI', 'JHARSUGUDA', 'NEEMUCH', 'NUAPADA', 'NIZAMABAD', 'AHMEDNAGAR', '24 PARAGANAS SOUTH',
'RATLAM', 'FATEHGARH SAHIB', 'MALAPPURAM', 'MAHARAJGANJ', 'KOLAR', 'HOSHIARPUR', 'AGRA',
'KHORDHA', 'CHAMARAJANAGAR', 'LUCKNOW', 'SAMBALPUR', 'VARANASI', 'THENI', 'SRIKAKULAM', 'DARRANG',
'RAJGARH', 'ALLAHABAD', 'KHARGONE', 'MUZAFFARNAGAR', 'THRISSUR', 'JABALPUR', 'NAGAUR', 'PASHCHIM C
HAMPARAN', 'BUXAR', 'MALDAH', 'KABIRDHAM', 'SOUTH DISTRICT', 'UTTAR KASHI', 'BUNDI',
'KURUKSHETRA', 'GAUTAM BUDDHA NAGAR', 'KORBA', 'ZUNHEBOTO', 'ROHTAS', 'SAHARANPUR', 'KOTTAYAM', 'Y
AVATMAL', 'BHOPAL', 'CHANDRAPUR', 'AJMER', 'PARBHANI', 'RAJSAMAND', 'GAJAPATI', 'GADCHIROLI', 'FAI
ZABAD', 'COIMBATORE', 'JALAUN', 'BANSWARA', 'KATNI', 'MEDINIPUR EAST', 'VIZIANAGARAM', 'JAMMU', 'D
ARBHANGA', 'REWA', 'DHAMTARI', 'KOTA', 'PATHANAMTHITTA', 'KOREA', 'UDAM SINGH NAGAR', 'MORENA', 'K
HEDA', 'ARARIA', 'SOLAPUR', 'LAHUL AND SPITI', 'UDUPI', 'BEED', 'BAGHPAT', 'TAMENGLONG',
'MAINPURI', 'GUNTUR', 'RAJNANDGAON', 'AMROHA', 'MANDI', 'SHIVPURI', 'RAICHUR', 'SIDHI', 'DHULE', '
KHANDWA', 'BARGARH', 'LATUR', 'GORAKHPUR', 'KULLU', 'SEONI', 'CHITRAKOOT', 'SHAHJAHANPUR',
'KARIMNAGAR', 'DINAJPUR UTTAR', 'MEWAT', 'VADODARA', 'DHARWAD', 'BAGESHWAR', 'KULGAM',
'TIKAMGARH', 'AMBEDKAR NAGAR', 'NAVSARI', 'CHANGLANG', 'KOKRAJHAR', 'EAST KAMENG', 'MUNGER',
'ARWAL', 'DATIA', 'UNA', 'JAUNPUR', 'BELLARY', 'CHIRANG', 'WARANGAL', 'SATARA', 'AURANGABAD',
'EAST SIANG', 'WEST KHASI HILLS', 'DHUBRI', 'SABAR KANTHA', 'BIRBHUM', 'NALANDA', 'BAKSA', 'EAST G
ARO HILLS', 'HARIDWAR', 'GANGANAGAR', 'PANCH MAHALS', 'DINDORI', 'BALOD', 'JALGAON',
'HANUMANGARH', 'BHILWARA', 'KINNAUR', 'SANGRUR', 'SONBHADRA', 'JAMUI', 'CHURU', 'RAE BARELI', 'KAT
IHAR', 'HINGOLI', 'SIRMAUR', '24 PARAGANAS NORTH', 'MEDINIPUR WEST', 'KHAMMAM', 'VILLUPURAM',
'RANGAREDDI', 'RAYAGADA', 'WEST DISTRICT', 'VIDISHA', 'DEOGARH', 'ASHOKNAGAR', 'GHAZIPUR',
'AHMADABAD', 'SIWAN', 'KARNAL', 'RAIGARH', 'SITAMARHI', 'BIKANER', 'NAWADA', 'BULDHANA', 'IDUKKI',
'JALANDHAR', 'CHANDAULI', 'ADILABAD', 'TUMKUR', 'BARWANI', 'MAHESANA', 'PATNA', 'ANUPPUR',
'JALORE', 'SINDHUDURG', 'KACHCHH', 'OSMANABAD', 'BIJAPUR', 'LAKHIMPUR', 'THE NILGIRIS',
'TIRUVANNAMALAI', 'LONGLENG', 'JHUNJHUNU', 'BARDHAMAN', 'DARJEELING', 'PATIALA', 'NAYAGARH',
'ALIRAJPUR', 'GOMATI', 'BANAS KANTHA', 'KARIMGANJ', 'PANNA', 'JUNAGADH', 'RI BHOI', 'THIRUVARUR',
'MUKTSAR', 'DAMOH', 'SOUTH WEST KHASI HILLS', 'SURGUJA', 'KOLASIB', 'UPPER SIANG', 'PURI',
'GHAZIABAD', 'KOLHAPUR', 'FARIDABAD', 'RAIGAD', 'KURUNG KUMEY', 'TIRUCHIRAPPALLI', 'NICOBARS', 'KA
RUR', 'SOUTH GARO HILLS', 'HISAR', 'SAMASTIPUR', 'PALAKKAD', 'ANUGUL', 'SAWAI MADHOPUR',
'KONDAGAON', 'SITAPUR', 'REASI', 'NARMADA', 'KANKER', 'DHOLPUR', 'GANDHINAGAR', 'KARAIKAL',
'AMRELI', 'JIND', 'BISHNUPUR', 'UTTAR KANNAD', 'HAVERI', 'ROHTAK', 'CHATRA', 'BALANGIR', 'SOUTH AN
DAMANS', 'PULWAMA', 'KANNIYAKUMARI', 'KENDRAPARA', 'LALITPUR', 'NAGAON', 'DINAJPUR DAKSHIN',
'DAKSHIN KANNAD', 'SHEIKHPURA', 'AMRAVATI', 'HATHRAS', 'PALI', 'YANAM', 'FIROZABAD', 'PALWAL', 'PU
DUKKOTTAI', 'KANPUR NAGAR', 'HOOGHLY', 'DADRA AND NAGAR HAVELI', 'LOWER DIBANG VALLEY', 'CHAMOLI',
'MAHE', 'LOHIT', 'KODERMA', 'ALIGARH', 'SINGRAULI', 'GURDASPUR', 'HAZARIBAGH', 'SONITPUR',
'BARAN', 'SULTANPUR', 'KANPUR DEHAT', 'TIRUPPUR', 'CHAMBA', 'DODA', 'MON', 'BENGALURU URBAN',
'DIMA HASAO', 'RAJAURI', 'SOUTH TRIPURA', 'KANCHIPURAM', 'BONGAIGAON', 'RAISEN', 'AMETHI',
'KAUSHAMBI', 'SHEOHAR', 'DOHAD', 'AIZAWL', 'TEHRI GARHWAL', 'ARIYALUR', 'WAYANAD', 'BHADRAK', 'SAL
EM', 'DUMKA', 'BALESHWAR', 'BANKA', 'SIRSA', 'GWALIOR', 'MAU', 'SURAJPUR', 'BARABANKI', 'JAMTARA',
'INDORE', 'PALGHAR', 'DEHRADUN', 'MAMIT', 'DUNGARPUR', 'KAITHAL', 'AMBALA', 'SIMDEGA',
'RAMANATHAPURAM', 'PANIPAT', 'BAHRAICH', 'TIRUNELVELI', 'LOWER SUBANSIRI', 'KADAPA', 'MUNGELI', 'S
HIMLA', 'TAWANG', 'EAST KHASI HILLS', 'HOWRAH', 'FIROZEPUR', 'ANAND', 'SUKMA', 'SHAHDOL',
'KHOWAI', 'DIMAPUR', 'MORADABAD', 'MIRZAPUR', 'GODDA', 'COOCHBEHAR', 'CHANDIGARH', 'KAPURTHALA', '
NORTH DISTRICT', 'DIBANG VALLEY', 'MOGA', 'GIRIDIH', 'GOLAGHAT', 'EAST DISTRICT', 'GUNA',
'BEMETARA', 'LAKHISARAI', 'TUTICORIN', 'KOHIMA', 'WEST SIANG', 'WEST KAMENG', 'PANCHKULA',
'JHANSI', 'CHURACHANDPUR', 'SHAMLI', 'DHAR', 'EAST JAINTIA HILLS', 'SURENDRANAGAR', 'SAMBHAL', 'KI
```

```
PHIRE', 'IMPHAL WEST', 'NAGAPATTINAM', 'RANCHI', 'RATNAGIRI', 'DEOGHAR', 'FARIDKOT', 'UPPER
SUBANSIRI', 'BATHINDA', 'LATEHAR', 'KASGANJ', 'SONIPAT', 'THANE', 'SENAPATI', 'AKOLA', 'EAST
SINGHBUM', 'NORTH GARO HILLS', 'PAPUM PARE', 'NORTH AND MIDDLE ANDAMAN', 'LEH LADAKH', 'BARNALA',
'SEPAHIJALA', 'LUDHIANA', 'BADGAM', 'POONCH', 'TIRAP', 'JHAJJAR', 'FATEHPUR', 'NANDURBAR',
'YADGIR', 'THOUBAL', 'MANSA', 'GARIYABAND', 'KARGIL', 'SHOPIAN', 'LAWNGTLAI', 'SOUTH WEST GARO HIL
LS', 'SAIHA', 'TAPI', 'BOKARO', 'NORTH GOA', 'SAHEBGANJ', 'UKHRUL', 'CHIKBALLAPUR', 'NARAYANPUR',
'KUPWARA', 'GARHWA', 'LOHARDAGA', 'AGAR MALWA', 'FAZILKA', 'IMPHAL EAST', 'PORBANDAR', 'SARAIKELA
KHARSAWAN', 'PALAMU', 'KATHUA', 'SAMBA', 'TARN TARAN', 'PATHANKOT', 'ANJAW', 'PAKUR', 'HAPUR', 'NA
WANSHAHR', 'SOUTH GOA', 'ANANTNAG', 'GANDERBAL', 'WEST SINGHBHUM', 'RAMGARH', 'KISHTWAR',
'HYDERABAD', 'RAMBAN', 'GUMLA', 'SERCHHIP', 'LONGDING', 'BANDIPORA', 'MUMBAI', 'KHUNTI', 'NAMSAI']
Shape of matrix after one hot encodig  (172263, 646)
Shape of matrix after one hot encodig  (73828, 646)
```

In [ ]:

```
col='Crop'
#Training data
from sklearn.feature_extraction.text import CountVectorizer
train_crop=list(X_train['Crop'].unique())


vectorizer_col = CountVectorizer(vocabulary=train_crop, lowercase=False, binary=True)
vectorizer_col.fit(X_train[col].values)
print(vectorizer_col.get_feature_names())
train_crop_one_hot = vectorizer_col.transform(X_train[col].values)
print("Shape of matrix after one hot encodig ",train_crop_one_hot.shape)
#For Cross validating data
#cv_crop_one_hot = vectorizer_col.transform(X_cv[col].values)
#print("Shape of matrix after one hot encodig ",cv_crop_one_hot.shape)

  # For testing data
test_crop_one_hot= vectorizer_col.transform(X_test[col].values)
print("Shape of matrix after one hot encodig ",test_crop_one_hot.shape)
```

```
['Rice', 'Sugarcane', 'Sannhamp', 'Tobacco', 'Wheat', 'Small millets', 'Other  Rabi pulses', 'Garl
ic', 'Sunflower', 'Turmeric', 'Ragi', 'Moong(Green Gram)', 'Masoor', 'Arhar/Tur', 'Maize',
'Sesamum', 'Potato', 'Safflower', 'Jowar', 'Barley', 'Gram', 'Bajra', 'Linseed', 'Sweet potato', '
Peas & beans (Pulses)', 'other oilseeds', 'Rapeseed &Mustard', 'Onion', 'Cardamom', 'Urad', 'Other
Cereals & Millets', 'Tapioca', 'Horse-gram', 'Groundnut', 'Castor seed', 'Total foodgrain', 'Dry c
hillies', 'Dry ginger', 'Cotton(lint)', 'Jute', 'Banana', 'Other Kharif pulses', 'Pulses total', '
Soyabean', 'Coriander', 'Cabbage', 'Mesta', 'Cowpea(Lobia)', 'Coconut ', 'Niger seed', 'Bhindi', '
Arecanut', 'Brinjal', 'Korra', 'Khesari', 'Papaya', 'Samai', 'Pome Fruit', 'Black pepper',
'Cashewnut', 'Other Fresh Fruits', 'Guar seed', 'Paddy', 'Cashewnut Raw', 'other misc. pulses', 'O
range', 'Oilseeds total', 'Pineapple', 'Tomato', 'Rajmash Kholar', 'Yam', 'Mango', 'Cashewnut Proc
essed', 'Other Vegetables', 'Beans & Mutter(Vegetable)', 'Jack Fruit', 'Moth', 'Pome Granet', 'Bla
ckgram', 'Cauliflower', 'Bitter Gourd', 'Ginger', 'Lentil', 'Sapota', 'Redish', 'Varagu', 'Water M
elon', 'Snak Guard', 'Pump Kin', 'Plums', 'Drum Stick', 'Ash Gourd', 'Turnip', 'Cucumber',
'Carrot', 'Grapes', 'Citrus Fruit', 'Tea', 'Lab-Lab', 'Other Dry Fruit', 'Beet Root', 'Litchi', 'L
emon', 'Bottle Gourd', 'other fibres', 'Bean', 'Other Citrus Fruit', 'Arcanut (Processed)', 'Ribed
Guard', 'Kapas', 'Rubber', 'Ber', 'Jute & mesta', 'Pear', 'Ricebean (nagadal)', 'Colocosia', 'Coff
ee', 'Atcanut (Raw)', 'Perilla', 'Peach', 'Jobster', 'Peas  (vegetable)', 'Cond-spcs other',
'Apple']
Shape of matrix after one hot encodig  (172263, 124)
Shape of matrix after one hot encodig  (73828, 124)
```

In [ ]:

```
col='Season'
#Training data
from sklearn.feature_extraction.text import CountVectorizer
train_season=list(X_train['Season'].unique())


vectorizer_col = CountVectorizer(vocabulary=train_season, lowercase=False, binary=True)
vectorizer_col.fit(X_train[col].values)
print(vectorizer_col.get_feature_names())
train_season_one_hot = vectorizer_col.transform(X_train[col].values)
print("Shape of matrix after one hot encodig ",train_season_one_hot.shape)
#For Cross validating data
#cv_season_one_hot = vectorizer_col.transform(X_cv[col].values)
#print("Shape of matrix after one hot encodig ",cv_season_one_hot.shape)

  # For testing data
test_season_one_hot= vectorizer_col.transform(X_test[col].values)
print("Shape of matrix after one hot encodig ",test_season_one_hot.shape)
```

```
print( Shape of matrix after one hot encodig  ,test_season_one_hot.shape)
```

```
['Rabi       ', 'Whole Year ', 'Summer     ', 'Kharif     ', 'Winter     ', 'Autumn     ']
Shape of matrix after one hot encodig  (172263, 6)
Shape of matrix after one hot encodig  (73828, 6)
```

In [ ]:

```
cropyear_list=list(X_train.Crop_Year.unique())

vectorizer_col = CountVectorizer(vocabulary=cropyear_list, lowercase=False, binary=True)
vectorizer_col.fit(X_train[col].values)
print(vectorizer_col.get_feature_names())
train_cropyear_one_hot = vectorizer_col.transform(X_train[col].values)
print("Shape of matrix after one hot encodig ",train_cropyear_one_hot.shape)
#For Cross validating data
#cv_cropyear_one_hot = vectorizer_col.transform(X_cv[col].values)
#print("Shape of matrix after one hot encodig ",cv_cropyear_one_hot.shape)

  # For testing data
test_cropyear_one_hot= vectorizer_col.transform(X_test[col].values)
print("Shape of matrix after one hot encodig ",test_cropyear_one_hot.shape)
```

```
[2012, 2007, 2000, 2006, 2011, 1998, 2010, 2013, 2009, 1999, 2008, 2005, 2003, 2001, 2004, 1997, 20
14, 2002, 2015]
Shape of matrix after one hot encodig  (172263, 19)
Shape of matrix after one hot encodig  (73828, 19)
```

Normalized Numerical Data

In [ ]:

```
# For Area
from sklearn.preprocessing import Normalizer


normalizer = Normalizer()
normalizer.fit(X_train['Area'].values.reshape(1,-1)) # finding the mean and standard deviation of
this data

# Now standardize the data with above mean and variance.
Area_normalized_train = normalizer.transform(X_train['Area'].values.reshape(1,-1))
# For Training Data
Area_normalized_test= normalizer.transform(X_test['Area'].values.reshape(1,-1))

print("After Area Normalization")
print(Area_normalized_train.shape, y_train.shape)
print(Area_normalized_test.shape, y_test.shape)
print('='*50)
```

```
After Area Normalization
(1, 172263) (172263,)
(1, 73828) (73828,)
==================================================
```

Combine All categorical and Numerical Data

In [ ]:

```
from scipy.sparse import hstack
X_tr=hstack((train_state_one_hot,train_district_one_hot,train_cropyear_one_hot,train_crop_one_hot,
train_season_one_hot,Area_normalized_train.T)).tocsr()
X_te=hstack((test_state_one_hot,test_district_one_hot,test_cropyear_one_hot,test_crop_one_hot,test
_season_one_hot,Area_normalized_test.T)).tocsr()
#X_cv=hstack((cv_state_one_hot,cv_district_one_hot,cv_cropyear_one_hot,cv_crop_one_hot,cv_season_o
t,Area_normalized_cv.T)).tocsr()

print("Final Data Matrix")
print(X_tr.shape, y_train.shape)
print(X_te.shape, y_test.shape)
```

```
print(X_te.shape, y_test.shape)
#print(X_cv.shape, y_cv.shape)
```

```
Final Data Matrix
(172263, 829) (172263,)
(73828, 829) (73828,)
```

In [ ]:

```
def batch_predict(clf, data):
    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the posi
tive class
    # not the predicted outputs

    y_data_pred = []
    tr_loop = data.shape[0] - data.shape[0]%1000
    # consider you X_tr shape is 49041, then your tr_loop will be 49041 - 49041%1000 = 49000
    # in this for loop we will iterate unti the last 1000 multiplier
    for i in range(0, tr_loop, 1000):
        y_data_pred.extend(clf.predict_proba(data[i:i+1000])[:,1])
    # we will be predicting for the last data points
    if data.shape[0]%1000 !=0:
        y_data_pred.extend(clf.predict_proba(data[tr_loop:])[:,1])

    return y_data_pred
```

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as seabornInstance
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
%matplotlib inline
```

Linear Regresssion Model

In [ ]:

```
from sklearn.metrics import mean_squared_error

lm = linear_model.LinearRegression()
model = lm.fit(X_tr, y_train)
predt_test = lm.predict(X_te)
print(mean_squared_error(y_test,predt_test))

print(len(model.coef_))
```

```
241116104525435.3
829
```

In [ ]:

```
import os
import json
import numpy as np
import pandas as pd
from pandas.io.json import json_normalize
import matplotlib.pyplot as plt
import seaborn as sns
color = sns.color_palette()



from plotly import tools
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
```

```
from sklearn import model_selection, preprocessing, metrics

pd.options.mode.chained_assignment = None
pd.options.display.max_columns = 999


import pandas as pd
train_df=pd.read_csv('crop_production.csv')

train_df.head()


# Null value in database
train_df.isnull().any()

#Unique value in database
for col in train_df.columns:
  print(train_df[col].unique())
  print('='*100)


#Replace the value
train_df['Production'].value_counts()
train_df['Production']=train_df['Production'].replace('=','0')

#Missing values in dataset
for col in train_df.columns:
    nullrow=train_df[col].isnull().sum()
    notrow=train_df[col].notnull().sum()
    percentage=(nullrow*100)/(nullrow+notrow)
    print("Column is ",col,percentage,"% Missing Values")

col=train_df.columns[:6]
col

train_df["Production"] = train_df["Production"].astype('float')

for col in train_df.columns[:6]:
  plt.figure(figsize=(13,6)) #figure size
  #It's another way to plot our data. using a variable that contains the plot parameters
  g1 = sns.barplot(x=col, y='Production',data=train_df[(train_df[col].isin((train_df[col].value_cou
nts()[:10].index.values))) & train_df['Production'] > 0])
  g1.set_title(col+'And Production', fontsize=20) # title and fontsize
  g1.set_xticklabels(g1.get_xticklabels(),rotation=45) # It's the way to rotate the xticks when we
use variable to our graphs
  g1.set_xlabel(col, fontsize=18) # Xlabel
  g1.set_ylabel('Production(log) Dist', fontsize=18) #Ylabel
  plt.show()


import pandas as pd
from sklearn import datasets, linear_model
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt



y=train_df['Production']
X_train, X_test, y_train, y_test = train_test_split(train_df, y, test_size=0.3)
#X_train,X_cv,y_train,y_cv=train_test_split(x_train,Y_train,test_size=0.2)
print(X_train.shape, y_train.shape)
#print(X_cv.shape, y_cv.shape)
print(X_test.shape, y_test.shape)


X_train.drop(['Production'], axis=1,inplace=True)
X_test.drop(['Production'],axis=1,inplace=True)
#X_cv.drop(['Production'],axis=1,inplace=True)
X_test.shape


#One hot encoding of State_Name
col='State_Name'
#Training data
from sklearn.feature_extraction.text import CountVectorizer
train_state=list(X_train['State_Name'].unique())
```

```
vectorizer_col = CountVectorizer(vocabulary=train_state, lowercase=False, binary=True)
vectorizer_col.fit(X_train[col].values)
print(vectorizer_col.get_feature_names())
train_state_one_hot = vectorizer_col.transform(X_train[col].values)
print("Shape of matrix after one hot encodig ",train_state_one_hot.shape)
#For Cross validating data
#cv_state_one_hot = vectorizer_col.transform(X_cv[col].values)
#print("Shape of matrix after one hot encodig ",cv_state_one_hot.shape)

  # For testing data
test_state_one_hot= vectorizer_col.transform(X_test[col].values)
print("Shape of matrix after one hot encodig ",test_state_one_hot.shape)

col='District_Name'
#Training data
from sklearn.feature_extraction.text import CountVectorizer
train_district=list(X_train['District_Name'].unique())
vectorizer_col = CountVectorizer(vocabulary=train_district, lowercase=False, binary=True)
vectorizer_col.fit(X_train[col].values)
print(vectorizer_col.get_feature_names())
train_district_one_hot = vectorizer_col.transform(X_train[col].values)
print("Shape of matrix after one hot encodig ",train_district_one_hot.shape)
#For Cross validating data
#cv_district_one_hot = vectorizer_col.transform(X_cv[col].values)
#print("Shape of matrix after one hot encodig ",cv_district_one_hot.shape)

  # For testing data
test_district_one_hot= vectorizer_col.transform(X_test[col].values)
print("Shape of matrix after one hot encodig ",test_district_one_hot.shape)


col='Crop'
#Training data
from sklearn.feature_extraction.text import CountVectorizer
train_crop=list(X_train['Crop'].unique())


vectorizer_col = CountVectorizer(vocabulary=train_crop, lowercase=False, binary=True)
vectorizer_col.fit(X_train[col].values)
print(vectorizer_col.get_feature_names())
train_crop_one_hot = vectorizer_col.transform(X_train[col].values)
print("Shape of matrix after one hot encodig ",train_crop_one_hot.shape)
#For Cross validating data
#cv_crop_one_hot = vectorizer_col.transform(X_cv[col].values)
#print("Shape of matrix after one hot encodig ",cv_crop_one_hot.shape)

  # For testing data
test_crop_one_hot= vectorizer_col.transform(X_test[col].values)
print("Shape of matrix after one hot encodig ",test_crop_one_hot.shape)



col='Season'
#Training data
from sklearn.feature_extraction.text import CountVectorizer
train_season=list(X_train['Season'].unique())


vectorizer_col = CountVectorizer(vocabulary=train_season, lowercase=False, binary=True)
vectorizer_col.fit(X_train[col].values)
print(vectorizer_col.get_feature_names())
train_season_one_hot = vectorizer_col.transform(X_train[col].values)
print("Shape of matrix after one hot encodig ",train_season_one_hot.shape)
#For Cross validating data
#cv_season_one_hot = vectorizer_col.transform(X_cv[col].values)
#print("Shape of matrix after one hot encodig ",cv_season_one_hot.shape)

  # For testing data
test_season_one_hot= vectorizer_col.transform(X_test[col].values)
print("Shape of matrix after one hot encodig ",test_season_one_hot.shape)



cropyear_list=list(X_train.Crop_Year.unique())

vectorizer_col = CountVectorizer(vocabulary=cropyear_list, lowercase=False, binary=True)
vectorizer_col.fit(X_train[col].values)
```

```
vectorizer_col.fit(X_train[col].values)
print(vectorizer_col.get_feature_names())
train_cropyear_one_hot = vectorizer_col.transform(X_train[col].values)
print("Shape of matrix after one hot encodig ",train_cropyear_one_hot.shape)
#For Cross validating data
#cv_cropyear_one_hot = vectorizer_col.transform(X_cv[col].values)
#print("Shape of matrix after one hot encodig ",cv_cropyear_one_hot.shape)

  # For testing data
test_cropyear_one_hot= vectorizer_col.transform(X_test[col].values)
print("Shape of matrix after one hot encodig ",test_cropyear_one_hot.shape)


# For Area
# the cost feature is already in numerical values,  we are going to represent the money, as numeri
cal values within the range 0-1
# normalization sklearn: https://scikit-
learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
from sklearn.preprocessing import Normalizer

# price_normalized = standardScalar.fit(project_data['price'].values)
# this will rise the error
# ValueError: Expected 2D array, got 1D array instead: array=[725.05 213.03 329.   ... 399.   287.
73   5.5 ].
# Reshape your data either using array.reshape(1,-1)

normalizer = Normalizer()
normalizer.fit(X_train['Area'].values.reshape(1,-1)) # finding the mean and standard deviation of
this data

# Now standardize the data with above mean and variance.
Area_normalized_train = normalizer.transform(X_train['Area'].values.reshape(1,-1))
# For Training Data
Area_normalized_test= normalizer.transform(X_test['Area'].values.reshape(1,-1))
# For Validating Data
#Area_normalized_cv= normalizer.transform(X_cv['Area'].values.reshape(1,-1))

print("After Area Normalization")
print(Area_normalized_train.shape, y_train.shape)
print(Area_normalized_test.shape, y_test.shape)
#print(Area_normalized_cv.shape, y_cv.shape)

print('='*50)


from scipy.sparse import hstack
X_tr=hstack((train_state_one_hot,train_district_one_hot,train_cropyear_one_hot,train_crop_one_hot,
train_season_one_hot,Area_normalized_train.T)).tocsr()
X_te=hstack((test_state_one_hot,test_district_one_hot,test_cropyear_one_hot,test_crop_one_hot,test
_season_one_hot,Area_normalized_test.T)).tocsr()
#X_cv=hstack((cv_state_one_hot,cv_district_one_hot,cv_cropyear_one_hot,cv_crop_one_hot,cv_season_o
t,Area_normalized_cv.T)).tocsr()

print("Final Data Matrix")
print(X_tr.shape, y_train.shape)
print(X_te.shape, y_test.shape)
#print(X_cv.shape, y_cv.shape)


def batch_predict(clf, data):
    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability estimates of the posi
tive class
    # not the predicted outputs

    y_data_pred = []
    tr_loop = data.shape[0] - data.shape[0]%1000
    # consider you X_tr shape is 49041, then your tr_loop will be 49041 - 49041%1000 = 49000
    # in this for loop we will iterate unti the last 1000 multiplier
    for i in range(0, tr_loop, 1000):
        y_data_pred.extend(clf.predict_proba(data[i:i+1000])[:,1])
    # we will be predicting for the last data points
    if data.shape[0]%1000 !=0:
        y_data_pred.extend(clf.predict_proba(data[tr_loop:])[:,1])

    return y_data_pred
```

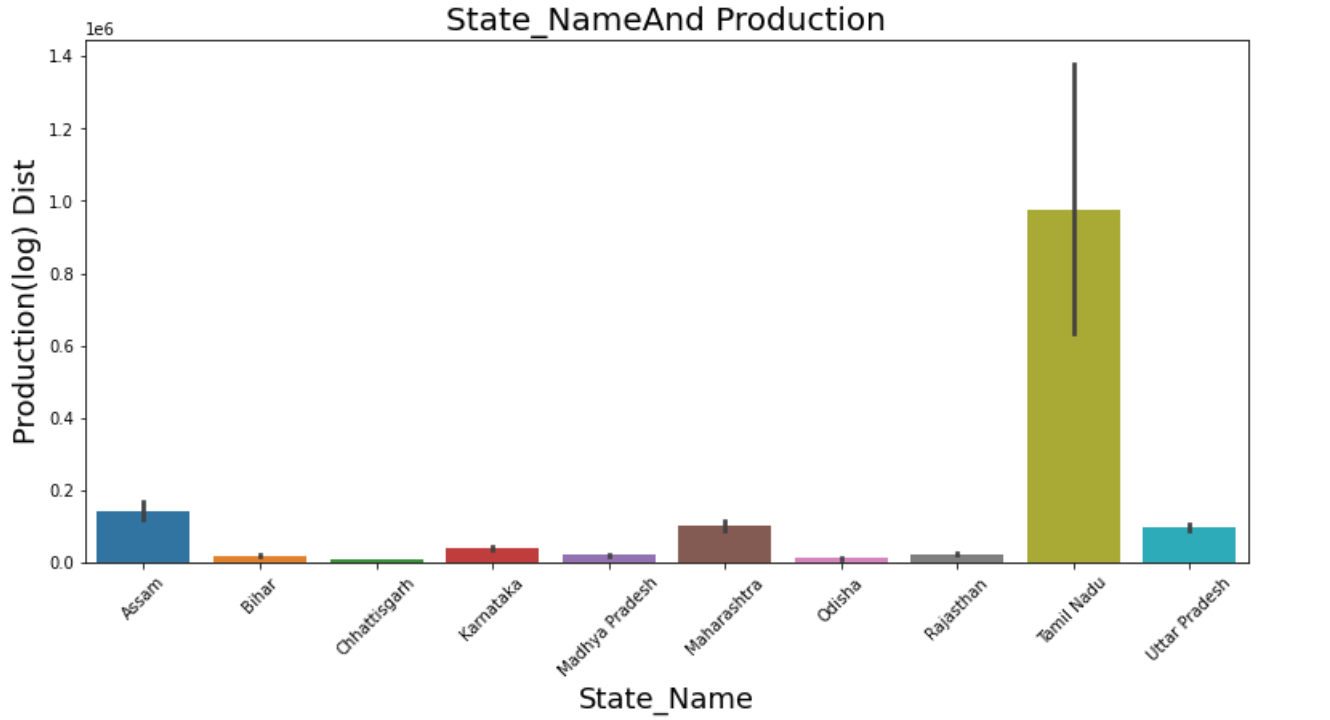
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as seabornInstance
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
#%matplotlib inline


from sklearn.metrics import mean_squared_error

lm = linear_model.LinearRegression()
model = lm.fit(X_tr, y_train)
predt_test = lm.predict(X_te)
print(mean_squared_error(y_test,predt_test))

print(len(model.coef_))


import matplotlib.pyplot as plt
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import roc_auc_score
import math
"""
y_true : array, shape = [n_samples] or [n_samples, n_classes]
Final Data matrix
((22445, 10144), (22445,))
((11055, 10144), (11055,))
((16500, 10144), (16500,))
=========================================================================================================

y_true : array, shape = [n_samples] or [n_samples, n_classes]
True binary labels or binary label indicators.
y_score : array, shape = [n_samples] or [n_samples, n_classes]
Target scores, can either be probability estimates of the positive class, confidence values, or no
n-thresholded measure of
decisions (as returned by "decision_function" on some classifiers).
For binary y_true, y_score is supposed to be the score of the class with greater label.
"""
train_auc = []
cv_auc = []
neigh =linear_model.LinearRegression()
neigh.fit(X_tr, y_train)
y_train_pred =neigh.predict(X_tr)
y_cv_pred =neigh.predict(X_tr)
mean_squared_error(y_train_pred,y_cv_pred)

lm = linear_model.LinearRegression()
model = lm.fit(X_tr, y_train)
predictions = lm.predict(X_te)
mean_squared_error(y_test,predictions)
print(model.coef_)



import xgboost as xgb
model1=xgb.XGBRegressor()
model1.fit(X_tr,y_train)
print(model1)
pred=model.predict(X_te)
print(mean_squared_error(y_test,pred))




from sklearn.model_selection import GridSearchCV
import xgboost as xgb

from sklearn.ensemble import RandomForestClassifier
estimators=[10, 50, 100, 200, 300,500, 1000]
depth=[2, 3, 4,7, 8, 10,12]

for n in estimators:
  for d in depth:
    model =xgb.XGBRegressor(n_estimators=n,learning_rate=0.01,max_depth=d,n_jobs=-1,random_state=2)
    model.fit(X_tr,y_train)
```

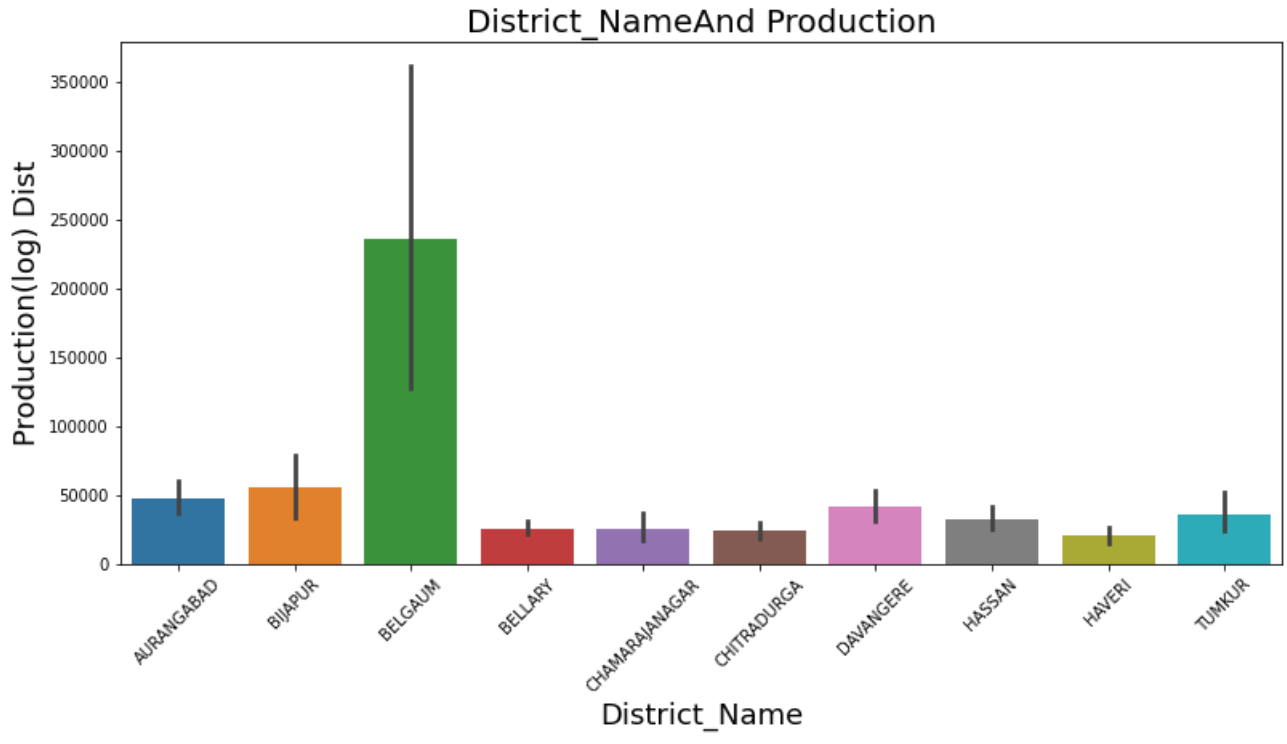
```
    model.fit(x_tr,y_train)
    pred=model.predict(X_te)
    print(mean_squared_error(y_test,pred))
```

```
['Andaman and Nicobar Islands' 'Andhra Pradesh' 'Arunachal Pradesh'
 'Assam' 'Bihar' 'Chandigarh' 'Chhattisgarh' 'Dadra and Nagar Haveli'
 'Goa' 'Gujarat' 'Haryana' 'Himachal Pradesh' 'Jammu and Kashmir '
 'Jharkhand' 'Karnataka' 'Kerala' 'Madhya Pradesh' 'Maharashtra' 'Manipur'
 'Meghalaya' 'Mizoram' 'Nagaland' 'Odisha' 'Puducherry' 'Punjab'
 'Rajasthan' 'Sikkim' 'Tamil Nadu' 'Telangana ' 'Tripura' 'Uttar Pradesh'
 'Uttarakhand' 'West Bengal']
==========================================================================================

['NICOBARS' 'NORTH AND MIDDLE ANDAMAN' 'SOUTH ANDAMANS' 'ANANTAPUR'
 'CHITTOOR' 'EAST GODAVARI' 'GUNTUR' 'KADAPA' 'KRISHNA' 'KURNOOL'
 'PRAKASAM' 'SPSR NELLORE' 'SRIKAKULAM' 'VISAKHAPATANAM' 'VIZIANAGARAM'
 'WEST GODAVARI' 'ANJAW' 'CHANGLANG' 'DIBANG VALLEY' 'EAST KAMENG'
 'EAST SIANG' 'KURUNG KUMEY' 'LOHIT' 'LONGDING' 'LOWER DIBANG VALLEY'
 'LOWER SUBANSIRI' 'NAMSAI' 'PAPUM PARE' 'TAWANG' 'TIRAP' 'UPPER SIANG'
 'UPPER SUBANSIRI' 'WEST KAMENG' 'WEST SIANG' 'BAKSA' 'BARPETA'
 'BONGAIGAON' 'CACHAR' 'CHIRANG' 'DARRANG' 'DHEMAJI' 'DHUBRI' 'DIBRUGARH'
 'DIMA HASAO' 'GOALPARA' 'GOLAGHAT' 'HAILAKANDI' 'JORHAT' 'KAMRUP'
 'KAMRUP METRO' 'KARBI ANGLONG' 'KARIMGANJ' 'KOKRAJHAR' 'LAKHIMPUR'
 'MARIGAON' 'NAGAON' 'NALBARI' 'SIVASAGAR' 'SONITPUR' 'TINSUKIA'
 'UDALGURI' 'ARARIA' 'ARWAL' 'AURANGABAD' 'BANKA' 'BEGUSARAI' 'BHAGALPUR'
 'BHOJPUR' 'BUXAR' 'DARBHANGA' 'GAYA' 'GOPALGANJ' 'JAMUI' 'JEHANABAD'
 'KAIMUR (BHABUA)' 'KATIHAR' 'KHAGARIA' 'KISHANGANJ' 'LAKHISARAI'
 'MADHEPURA' 'MADHUBANI' 'MUNGER' 'MUZAFFARPUR' 'NALANDA' 'NAWADA'
 'PASHCHIM CHAMPARAN' 'PATNA' 'PURBI CHAMPARAN' 'PURNIA' 'ROHTAS'
 'SAHARSA' 'SAMASTIPUR' 'SARAN' 'SHEIKHPURA' 'SHEOHAR' 'SITAMARHI' 'SIWAN'
 'SUPAUL' 'VAISHALI' 'CHANDIGARH' 'BALOD' 'BALODA BAZAR' 'BALRAMPUR'
 'BASTAR' 'BEMETARA' 'BIJAPUR' 'BILASPUR' 'DANTEWADA' 'DHAMTARI' 'DURG'
 'GARIYABAND' 'JANJGIR-CHAMPA' 'JASHPUR' 'KABIRDHAM' 'KANKER' 'KONDAGAON'
 'KORBA' 'KOREA' 'MAHASAMUND' 'MUNGELI' 'NARAYANPUR' 'RAIGARH' 'RAIPUR'
 'RAJNANDGAON' 'SUKMA' 'SURAJPUR' 'SURGUJA' 'DADRA AND NAGAR HAVELI'
 'NORTH GOA' 'SOUTH GOA' 'AHMADABAD' 'AMRELI' 'ANAND' 'BANAS KANTHA'
 'BHARUCH' 'BHAVNAGAR' 'DANG' 'DOHAD' 'GANDHINAGAR' 'JAMNAGAR' 'JUNAGADH'
 'KACHCHH' 'KHEDA' 'MAHESANA' 'NARMADA' 'NAVSARI' 'PANCH MAHALS' 'PATAN'
 'PORBANDAR' 'RAJKOT' 'SABAR KANTHA' 'SURAT' 'SURENDRANAGAR' 'TAPI'
 'VADODARA' 'VALSAD' 'AMBALA' 'BHIWANI' 'FARIDABAD' 'FATEHABAD' 'GURGAON'
 'HISAR' 'JHAJJAR' 'JIND' 'KAITHAL' 'KARNAL' 'KURUKSHETRA' 'MAHENDRAGARH'
 'MEWAT' 'PALWAL' 'PANCHKULA' 'PANIPAT' 'REWARI' 'ROHTAK' 'SIRSA'
 'SONIPAT' 'YAMUNANAGAR' 'CHAMBA' 'HAMIRPUR' 'KANGRA' 'KINNAUR' 'KULLU'
 'LAHUL AND SPITI' 'MANDI' 'SHIMLA' 'SIRMAUR' 'SOLAN' 'UNA' 'ANANTNAG'
 'BADGAM' 'BANDIPORA' 'BARAMULLA' 'DODA' 'GANDERBAL' 'JAMMU' 'KARGIL'
 'KATHUA' 'KISHTWAR' 'KULGAM' 'KUPWARA' 'LEH LADAKH' 'POONCH' 'PULWAMA'
 'RAJAURI' 'RAMBAN' 'REASI' 'SAMBA' 'SHOPIAN' 'SRINAGAR' 'UDHAMPUR'
 'BOKARO' 'CHATRA' 'DEOGHAR' 'DHANBAD' 'DUMKA' 'EAST SINGHBUM' 'GARHWA'
 'GIRIDIH' 'GODDA' 'GUMLA' 'HAZARIBAGH' 'JAMTARA' 'KHUNTI' 'KODERMA'
 'LATEHAR' 'LOHARDAGA' 'PAKUR' 'PALAMU' 'RAMGARH' 'RANCHI' 'SAHEBGANJ'
 'SARAIKELA KHARSAWAN' 'SIMDEGA' 'WEST SINGHBHUM' 'BAGALKOT'
 'BANGALORE RURAL' 'BELGAUM' 'BELLARY' 'BENGALURU URBAN' 'BIDAR'
 'CHAMARAJANAGAR' 'CHIKBALLAPUR' 'CHIKMAGALUR' 'CHITRADURGA'
 'DAKSHIN KANNAD' 'DAVANGERE' 'DHARWAD' 'GADAG' 'GULBARGA' 'HASSAN'
 'HAVERI' 'KODAGU' 'KOLAR' 'KOPPAL' 'MANDYA' 'MYSORE' 'RAICHUR'
 'RAMANAGARA' 'SHIMOGA' 'TUMKUR' 'UDUPI' 'UTTAR KANNAD' 'YADGIR'
 'ALAPPUZHA' 'ERNAKULAM' 'IDUKKI' 'KANNUR' 'KASARAGOD' 'KOLLAM' 'KOTTAYAM'
 'KOZHIKODE' 'MALAPPURAM' 'PALAKKAD' 'PATHANAMTHITTA' 'THIRUVANANTHAPURAM'
 'THRISSUR' 'WAYANAD' 'AGAR MALWA' 'ALIRAJPUR' 'ANUPPUR' 'ASHOKNAGAR'
 'BALAGHAT' 'BARWANI' 'BETUL' 'BHIND' 'BHOPAL' 'BURHANPUR' 'CHHATARPUR'
 'CHHINDWARA' 'DAMOH' 'DATIA' 'DEWAS' 'DHAR' 'DINDORI' 'GUNA' 'GWALIOR'
 'HARDA' 'HOSHANGABAD' 'INDORE' 'JABALPUR' 'JHABUA' 'KATNI' 'KHANDWA'
 'KHARGONE' 'MANDLA' 'MANDSAUR' 'MORENA' 'NARSINGHPUR' 'NEEMUCH' 'PANNA'
 'RAISEN' 'RAJGARH' 'RATLAM' 'REWA' 'SAGAR' 'SATNA' 'SEHORE' 'SEONI'
 'SHAHDOL' 'SHAJAPUR' 'SHEOPUR' 'SHIVPURI' 'SIDHI' 'SINGRAULI' 'TIKAMGARH'
 'UJJAIN' 'UMARIA' 'VIDISHA' 'AHMEDNAGAR' 'AKOLA' 'AMRAVATI' 'BEED'
 'BHANDARA' 'BULDHANA' 'CHANDRAPUR' 'DHULE' 'GADCHIROLI' 'GONDIA'
 'HINGOLI' 'JALGAON' 'JALNA' 'KOLHAPUR' 'LATUR' 'MUMBAI' 'NAGPUR' 'NANDED'
 'NANDURBAR' 'NASHIK' 'OSMANABAD' 'PALGHAR' 'PARBHANI' 'PUNE' 'RAIGAD'
 'RATNAGIRI' 'SANGLI' 'SATARA' 'SINDHUDURG' 'SOLAPUR' 'THANE' 'WARDHA'
 'WASHIM' 'YAVATMAL' 'BISHNUPUR' 'CHANDEL' 'CHURACHANDPUR' 'IMPHAL EAST'
 'IMPHAL WEST' 'SENAPATI' 'TAMENGLONG' 'THOUBAL' 'UKHRUL'
 'EAST GARO HILLS' 'EAST JAINTIA HILLS' 'EAST KHASI HILLS'
 'NORTH GARO HILLS' 'RI BHOI' 'SOUTH GARO HILLS' 'SOUTH WEST GARO HILLS'
 'SOUTH WEST KHASI HILLS' 'WEST GARO HILLS' 'WEST JAINTIA HILLS'
 'WEST KHASI HILLS' 'AIZAWL' 'CHAMPHAI' 'KOLASIB' 'LAWNGTLAI' 'LUNGLEI'
```

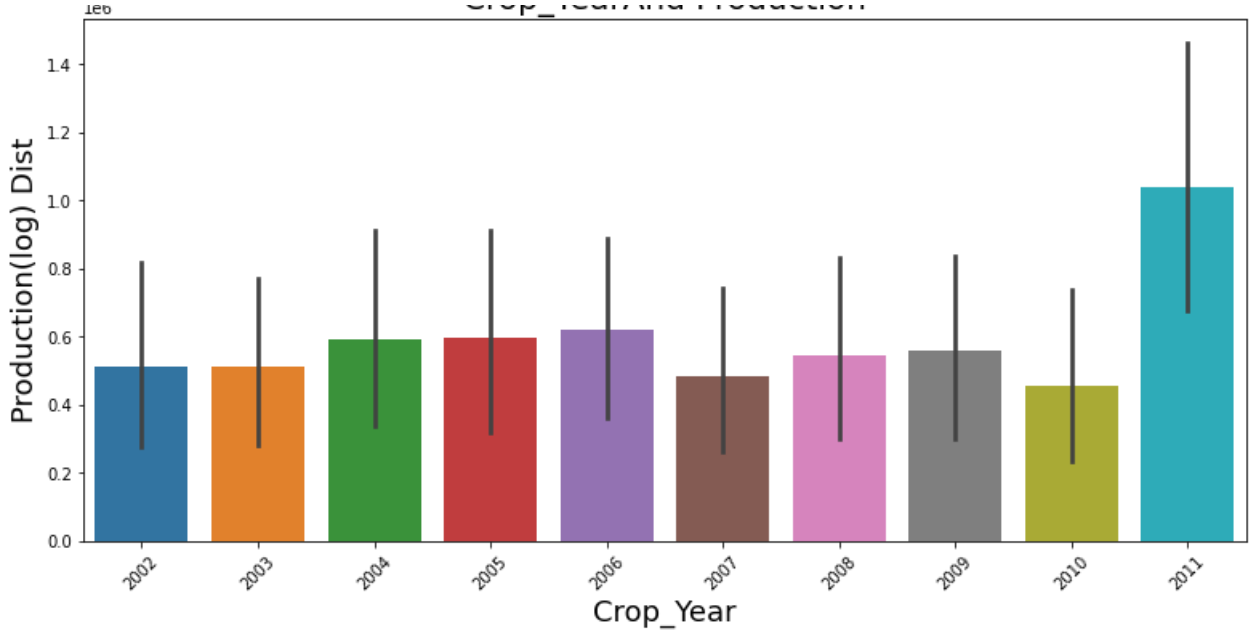
```
 'MAMIT' 'SAIHA' 'SERCHHIP' 'DIMAPUR' 'KIPHIRE' 'KOHIMA' 'LONGLENG'
 'MOKOKCHUNG' 'MON' 'PEREN' 'PHEK' 'TUENSANG' 'WOKHA' 'ZUNHEBOTO' 'ANUGUL'
 'BALANGIR' 'BALESHWAR' 'BARGARH' 'BHADRAK' 'BOUDH' 'CUTTACK' 'DEOGARH'
 'DHENKANAL' 'GAJAPATI' 'GANJAM' 'JAGATSINGHAPUR' 'JAJAPUR' 'JHARSUGUDA'
 'KALAHANDI' 'KANDHAMAL' 'KENDRAPARA' 'KENDUJHAR' 'KHORDHA' 'KORAPUT'
 'MALKANGIRI' 'MAYURBHANJ' 'NABARANGPUR' 'NAYAGARH' 'NUAPADA' 'PURI'
 'RAYAGADA' 'SAMBALPUR' 'SONEPUR' 'SUNDARGARH' 'KARAIKAL' 'MAHE'
 'PONDICHERRY' 'YANAM' 'AMRITSAR' 'BARNALA' 'BATHINDA' 'FARIDKOT'
 'FATEHGARH SAHIB' 'FAZILKA' 'FIROZEPUR' 'GURDASPUR' 'HOSHIARPUR'
 'JALANDHAR' 'KAPURTHALA' 'LUDHIANA' 'MANSA' 'MOGA' 'MUKTSAR' 'NAWANSHAHR'
 'PATHANKOT' 'PATIALA' 'RUPNAGAR' 'S.A.S NAGAR' 'SANGRUR' 'TARN TARAN'
 'AJMER' 'ALWAR' 'BANSWARA' 'BARAN' 'BARMER' 'BHARATPUR' 'BHILWARA'
 'BIKANER' 'BUNDI' 'CHITTORGARH' 'CHURU' 'DAUSA' 'DHOLPUR' 'DUNGARPUR'
 'GANGANAGAR' 'HANUMANGARH' 'JAIPUR' 'JAISALMER' 'JALORE' 'JHALAWAR'
 'JHUNJHUNU' 'JODHPUR' 'KARAULI' 'KOTA' 'NAGAUR' 'PALI' 'PRATAPGARH'
 'RAJSAMAND' 'SAWAI MADHOPUR' 'SIKAR' 'SIROHI' 'TONK' 'UDAIPUR'
 'EAST DISTRICT' 'NORTH DISTRICT' 'SOUTH DISTRICT' 'WEST DISTRICT'
 'ARIYALUR' 'COIMBATORE' 'CUDDALORE' 'DHARMAPURI' 'DINDIGUL' 'ERODE'
 'KANCHIPURAM' 'KANNIYAKUMARI' 'KARUR' 'KRISHNAGIRI' 'MADURAI'
 'NAGAPATTINAM' 'NAMAKKAL' 'PERAMBALUR' 'PUDUKKOTTAI' 'RAMANATHAPURAM'
 'SALEM' 'SIVAGANGA' 'THANJAVUR' 'THE NILGIRIS' 'THENI' 'THIRUVALLUR'
 'THIRUVARUR' 'TIRUCHIRAPPALLI' 'TIRUNELVELI' 'TIRUPPUR' 'TIRUVANNAMALAI'
 'TUTICORIN' 'VELLORE' 'VILLUPURAM' 'VIRUDHUNAGAR' 'ADILABAD' 'HYDERABAD'
 'KARIMNAGAR' 'KHAMMAM' 'MAHBUBNAGAR' 'MEDAK' 'NALGONDA' 'NIZAMABAD'
 'RANGAREDDI' 'WARANGAL' 'DHALAI' 'GOMATI' 'KHOWAI' 'NORTH TRIPURA'
 'SEPAHIJALA' 'SOUTH TRIPURA' 'UNAKOTI' 'WEST TRIPURA' 'AGRA' 'ALIGARH'
 'ALLAHABAD' 'AMBEDKAR NAGAR' 'AMETHI' 'AMROHA' 'AURAIYA' 'AZAMGARH'
 'BAGHPAT' 'BAHRAICH' 'BALLIA' 'BANDA' 'BARABANKI' 'BAREILLY' 'BASTI'
 'BIJNOR' 'BUDAUN' 'BULANDSHAHR' 'CHANDAULI' 'CHITRAKOOT' 'DEORIA' 'ETAH'
 'ETAWAH' 'FAIZABAD' 'FARRUKHABAD' 'FATEHPUR' 'FIROZABAD'
 'GAUTAM BUDDHA NAGAR' 'GHAZIABAD' 'GHAZIPUR' 'GONDA' 'GORAKHPUR' 'HAPUR'
 'HARDOI' 'HATHRAS' 'JALAUN' 'JAUNPUR' 'JHANSI' 'KANNAUJ' 'KANPUR DEHAT'
 'KANPUR NAGAR' 'KASGANJ' 'KAUSHAMBI' 'KHERI' 'KUSHI NAGAR' 'LALITPUR'
 'LUCKNOW' 'MAHARAJGANJ' 'MAHOBA' 'MAINPURI' 'MATHURA' 'MAU' 'MEERUT'
 'MIRZAPUR' 'MORADABAD' 'MUZAFFARNAGAR' 'PILIBHIT' 'RAE BARELI' 'RAMPUR'
 'SAHARANPUR' 'SAMBHAL' 'SANT KABEER NAGAR' 'SANT RAVIDAS NAGAR'
 'SHAHJAHANPUR' 'SHAMLI' 'SHRAVASTI' 'SIDDHARTH NAGAR' 'SITAPUR'
 'SONBHADRA' 'SULTANPUR' 'UNNAO' 'VARANASI' 'ALMORA' 'BAGESHWAR' 'CHAMOLI'
 'CHAMPAWAT' 'DEHRADUN' 'HARIDWAR' 'NAINITAL' 'PAURI GARHWAL'
 'PITHORAGARH' 'RUDRA PRAYAG' 'TEHRI GARHWAL' 'UDAM SINGH NAGAR'
 'UTTAR KASHI' '24 PARAGANAS NORTH' '24 PARAGANAS SOUTH' 'BANKURA'
 'BARDHAMAN' 'BIRBHUM' 'COOCHBEHAR' 'DARJEELING' 'DINAJPUR DAKSHIN'
 'DINAJPUR UTTAR' 'HOOGHLY' 'HOWRAH' 'JALPAIGURI' 'MALDAH'
 'MEDINIPUR EAST' 'MEDINIPUR WEST' 'MURSHIDABAD' 'NADIA' 'PURULIA']
==========================================================================================================

[2000 2001 2002 2003 2004 2005 2006 2010 1997 1998 1999 2007 2008 2009
 2011 2012 2013 2014 2015]
==========================================================================================================

['Kharif     ' 'Whole Year ' 'Autumn     ' 'Rabi       ' 'Summer     '
 'Winter     ']
==========================================================================================================

['Arecanut' 'Other Kharif pulses' 'Rice' 'Banana' 'Cashewnut' 'Coconut '
 'Dry ginger' 'Sugarcane' 'Sweet potato' 'Tapioca' 'Black pepper'
 'Dry chillies' 'other oilseeds' 'Turmeric' 'Maize' 'Moong(Green Gram)'
 'Urad' 'Arhar/Tur' 'Groundnut' 'Sunflower' 'Bajra' 'Castor seed'
 'Cotton(lint)' 'Horse-gram' 'Jowar' 'Korra' 'Ragi' 'Tobacco' 'Gram'
 'Wheat' 'Masoor' 'Sesamum' 'Linseed' 'Safflower' 'Onion'
 'other misc. pulses' 'Samai' 'Small millets' 'Coriander' 'Potato'
 'Other  Rabi pulses' 'Soyabean' 'Beans & Mutter(Vegetable)' 'Bhindi'
 'Brinjal' 'Citrus Fruit' 'Cucumber' 'Grapes' 'Mango' 'Orange'
 'other fibres' 'Other Fresh Fruits' 'Other Vegetables' 'Papaya'
 'Pome Fruit' 'Tomato' 'Rapeseed &Mustard' 'Mesta' 'Cowpea(Lobia)' 'Lemon'
 'Pome Granet' 'Sapota' 'Cabbage' 'Peas  (vegetable)' 'Niger seed'
 'Bottle Gourd' 'Sannhamp' 'Varagu' 'Garlic' 'Ginger' 'Oilseeds total'
 'Pulses total' 'Jute' 'Peas & beans (Pulses)' 'Blackgram' 'Paddy'
 'Pineapple' 'Barley' 'Khesari' 'Guar seed' 'Moth'
 'Other Cereals & Millets' 'Cond-spcs other' 'Turnip' 'Carrot' 'Redish'
 'Arcanut (Processed)' 'Atcanut (Raw)' 'Cashewnut Processed'
 'Cashewnut Raw' 'Cardamom' 'Rubber' 'Bitter Gourd' 'Drum Stick'
 'Jack Fruit' 'Snak Guard' 'Pump Kin' 'Tea' 'Coffee' 'Cauliflower'
 'Other Citrus Fruit' 'Water Melon' 'Total foodgrain' 'Kapas' 'Colocosia'
 'Lentil' 'Bean' 'Jobster' 'Perilla' 'Rajmash Kholar' 'Ricebean (nagadal)'
 'Ash Gourd' 'Beet Root' 'Lab-Lab' 'Ribed Guard' 'Yam' 'Apple' 'Peach'
 'Pear' 'Plums' 'Litchi' 'Ber' 'Other Dry Fruit' 'Jute & mesta']
```

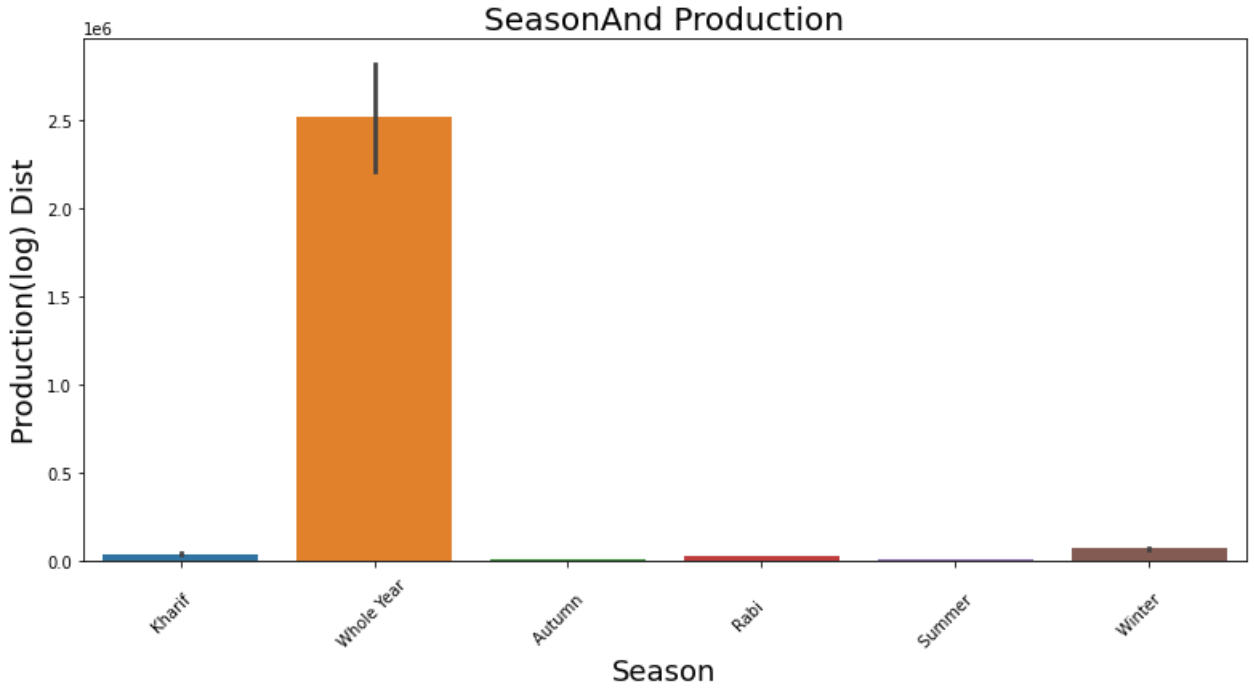
```
==========================================================================================================

[1.25400e+03 2.00000e+00 1.02000e+02 ... 3.02274e+05 1.14930e+04
 2.79151e+05]
==========================================================================================================

[2.00000e+03 1.00000e+00 3.21000e+02 ... 7.29553e+05 7.30136e+05
 5.97899e+05]
==========================================================================================================

Column is  State_Name 0.0 % Missing Values
Column is  District_Name 0.0 % Missing Values
Column is  Crop_Year 0.0 % Missing Values
Column is  Season 0.0 % Missing Values
Column is  Crop 0.0 % Missing Values
Column is  Area 0.0 % Missing Values
Column is  Production 1.5156994770227274 % Missing Values
```

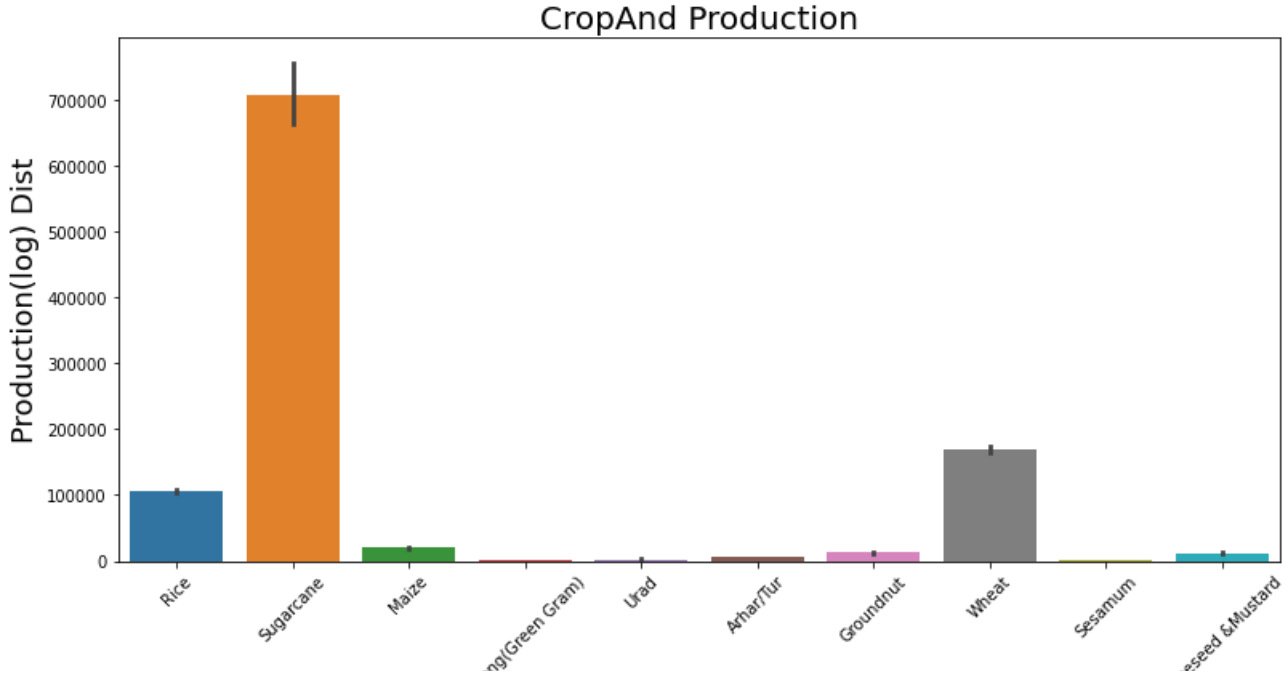


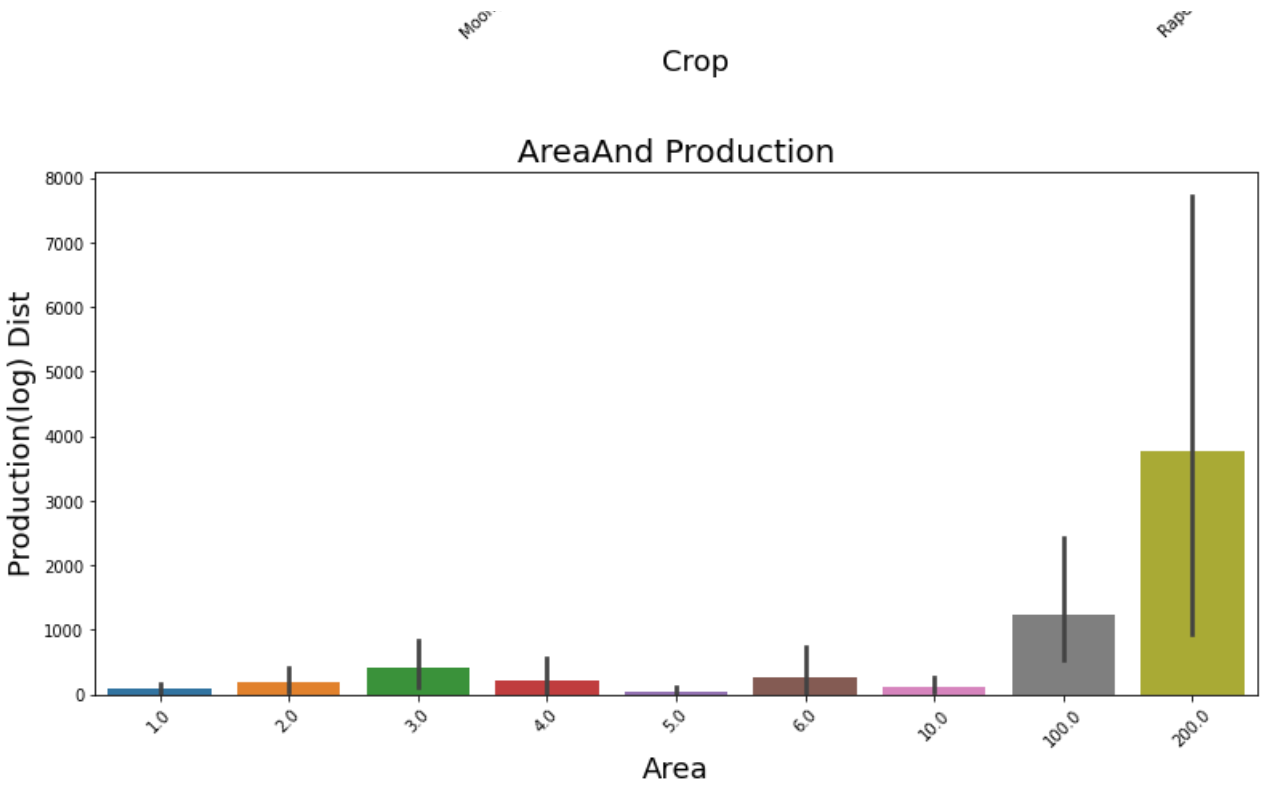


Crop_YearAnd Production


1e6
Production(log) Dist
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
Crop_Year


SeasonAnd Production
1e6
Production(log) Dist
2.5
2.0
1.5
1.0
0.5
0.0
Khanif
Whole Year
Autumn
Rabi
Summer
Winter
Season


CropAnd Production
Production(log) Dist
700000
600000
500000
400000
300000
200000
100000
0
Rice
Sugarcane
Maize
Urad
Arhar/Tur
Groundnut
Wheat
Sesamum




```
(172263, 7) (172263,)
(73828, 7) (73828,)
['Uttar Pradesh', 'West Bengal', 'Rajasthan', 'Odisha', 'Andhra Pradesh', 'Assam', 'Maharashtra',
'Bihar', 'Karnataka', 'Goa', 'Uttarakhand', 'Tamil Nadu', 'Himachal Pradesh', 'Madhya Pradesh', 'T
elangana ', 'Gujarat', 'Nagaland', 'Chhattisgarh', 'Puducherry', 'Kerala', 'Meghalaya', 'Punjab',
'Haryana', 'Jammu and Kashmir ', 'Sikkim', 'Mizoram', 'Arunachal Pradesh', 'Manipur', 'Jharkhand',
'Andaman and Nicobar Islands', 'Tripura', 'Dadra and Nagar Haveli', 'Chandigarh']
Shape of matrix after one hot encodig  (172263, 33)
Shape of matrix after one hot encodig  (73828, 33)
['ETAWAH', 'HATHRAS', 'MEDINIPUR WEST', 'ALWAR', 'BALANGIR', 'VIZIANAGARAM', 'KOKRAJHAR',
'NABARANGPUR', 'JALNA', 'MADHUBANI', 'BHANDARA', 'BAGALKOT', 'RAYAGADA', 'BANKA', 'DINAJPUR UTTAR'
, 'EAST GODAVARI', 'DHUBRI', 'MORADABAD', 'KANNAUJ', 'KODAGU', 'GADCHIROLI', 'BARABANKI', 'SOUTH
GOA', 'JAMUI', 'CHAMOLI', 'KANPUR NAGAR', 'TUTICORIN', 'SHAHJAHANPUR', 'GAUTAM BUDDHA NAGAR',
'THANE', 'PURULIA', 'MANDI', 'TUMKUR', 'SEONI', 'BIDAR', 'KARIMNAGAR', 'PERAMBALUR', 'CHANDAULI',
'DHULE', 'GAJAPATI', 'SHAHDOL', 'DEORIA', 'AZAMGARH', 'SABAR KANTHA', 'KOHIMA', 'TUENSANG', 'DHAMT
ARI', 'KHAGARIA', 'TIKAMGARH', 'BHOPAL', 'BUNDI', 'JODHPUR', 'JAMNAGAR', 'CHAMARAJANAGAR',
'KARIMGANJ', 'NALBARI', 'KHANDWA', 'JANJGIR-CHAMPA', 'SAHARSA', 'JAGATSINGHAPUR', 'MAYURBHANJ',
'GANGANAGAR', 'MAHE', 'KATNI', 'ETAH', 'GOLAGHAT', 'UDUPI', 'KRISHNA', 'THIRUVANANTHAPURAM', 'KADA
PA', 'MAHBUBNAGAR', 'FIROZABAD', 'RAMPUR', 'MURSHIDABAD', 'KHORDHA', 'LAKHIMPUR', 'GANJAM',
'JHALAWAR', 'PATHANAMTHITTA', 'EAST JAINTIA HILLS', 'AHMEDNAGAR', 'BASTAR', 'SINGRAULI',
'PATHANKOT', 'BANGALORE RURAL', 'BIJAPUR', 'PANCHKULA', 'BARPETA', 'KENDUJHAR', 'MIRZAPUR',
'KURNOOL', 'MADHEPURA', 'KUPWARA', 'KOLLAM', 'RAICHUR', 'SRIKAKULAM', 'MEDINIPUR EAST', 'BANKURA',
'MALDAH', 'NORTH GOA', 'LALITPUR', 'BETUL', 'LUDHIANA', 'PURBI CHAMPARAN', 'SOUTH DISTRICT', 'CHHA
TARPUR', 'KANKER', 'ARARIA', 'SHIMOGA', 'MUNGER', 'AIZAWL', 'KANDHAMAL', 'LUCKNOW', 'SUNDARGARH',
'TONK', 'BASTI', '24 PARAGANAS SOUTH', 'KRISHNAGIRI', 'DARBHANGA', 'JHABUA', 'CHAMBA',
'CHIKMAGALUR', 'NAGAON', 'DEHRADUN', 'CACHAR', 'PANNA', 'SITAMARHI', 'UDAIPUR', 'DINAJPUR
DAKSHIN', 'SOUTH GARO HILLS', 'MAINPURI', 'BHARUCH', 'MAHOBA', 'DINDORI', 'MYSORE',
'TIRUCHIRAPPALLI', 'KARUR', 'PASHCHIM CHAMPARAN', 'MANSA', 'BARWANI', 'BALRAMPUR', 'AMRELI', 'WASH
IM', 'PUDUKKOTTAI', 'ADILABAD', 'RI BHOI', 'MOKOKCHUNG', 'DHAR', 'UPPER SIANG', 'HAPUR', 'GONDIA',
'KHEDA', 'KOPPAL', 'NANDED', 'RATNAGIRI', 'KOZHIKODE', 'PRATAPGARH', 'TAMENGLONG', 'LOHARDAGA',
'SIWAN', 'BELGAUM', 'DAMOH', 'KOREA', 'KAITHAL', 'PITHORAGARH', 'RAMANATHAPURAM', 'NICOBARS', 'JAI
PUR', 'SOUTH TRIPURA', 'DHANBAD', 'BURHANPUR', 'RAISEN', 'MEDAK', 'BILASPUR', 'AURANGABAD',
'HASSAN', 'GOALPARA', 'NAMAKKAL', 'DIMAPUR', 'SAMASTIPUR', 'FAIZABAD', 'AGRA', 'HAMIRPUR',
'PONDICHERRY', 'NAWADA', 'KAMRUP METRO', 'SONITPUR', 'MAHESANA', 'BOUDH', 'NADIA', 'BHIND', 'JEHAN
ABAD', 'BHAGALPUR', 'SIRMAUR', 'RAJKOT', 'HAILAKANDI', 'ALMORA', 'NUAPADA', 'SAMBALPUR',
'TIRUNELVELI', 'CHITRADURGA', 'DIMA HASAO', 'RUDRA PRAYAG', 'ALIGARH', 'VALSAD', 'HOWRAH',
'UDALGURI', 'CHURACHANDPUR', 'WARANGAL', 'PORBANDAR', 'BARAN', 'BENGALURU URBAN', 'UJJAIN',
'GUNA', 'KARBI ANGLONG', 'NANDURBAR', 'SALEM', 'BHILWARA', 'SIVAGANGA', 'ALLAHABAD', 'JAISALMER',
'BARGARH', 'DANTEWADA', 'KAMRUP', 'THENI', 'CHITRAKOOT', 'BIRBHUM', 'ANJAW', 'ZUNHEBOTO',
'FATEHABAD', 'BALODA BAZAR', 'PALI', 'BULDHANA', 'REWA', 'SONBHADRA', 'JAJAPUR', 'PEREN', 'SANT RA
VIDAS NAGAR', 'CHITTORGARH', 'LUNGLEI', 'JUNAGADH', 'SARAIKELA KHARSAWAN', 'GULBARGA', 'RATLAM', '
KISHANGANJ', 'KORBA', 'KHAMMAM', 'KALAHANDI', 'HOSHANGABAD', 'RAIGAD', 'NALGONDA', 'DODA',
'GURGAON', 'MON', 'GOMATI', 'EAST KHASI HILLS', 'RAJSAMAND', 'HANUMANGARH', 'JHARSUGUDA',
'GUNTUR', 'GURDASPUR', 'UTTAR KASHI', 'DHENKANAL', 'VILLUPURAM', 'COOCHBEHAR', 'KANPUR DEHAT',
'CHIRANG', 'JALGAON', 'BEGUSARAI', 'BEMETARA', 'BANAS KANTHA', 'CUDDALORE', 'KATIHAR', 'BIKANER',
'MUZAFFARNAGAR', 'KHERI', 'DHARMAPURI', '24 PARAGANAS NORTH', 'CHITTOOR', 'JALPAIGURI',
'SHAJAPUR', 'WEST KAMENG', 'NIZAMABAD', 'ALIRAJPUR', 'SURGUJA', 'DINDIGUL', 'ANUGUL',
```

'SHAOPUR', 'WEST KAMENG', 'NIZAMABAD', 'ALIRAJPUR', 'SURGUJA', 'DINDIGUL', 'ANUGUL',
'TIRUVANNAMALAI', 'KANNUR', 'KANGRA', 'JHANSI', 'YANAM', 'KASGANJ', 'NEEMUCH', 'FARRUKHABAD', 'KAN
NIYAKUMARI', 'GOPALGANJ', 'JALAUN', 'WEST GODAVARI', 'ANUPPUR', 'KAIMUR (BHABUA)', 'NARSINGHPUR',
'INDORE', 'SHEIKHPURA', 'UMARIA', 'KOLASIB', 'JORHAT', 'SULTANPUR', 'PARBHANI', 'AKOLA',
'RAJNANDGAON', 'LAKHISARAI', 'VADODARA', 'JASHPUR', 'COIMBATORE', 'PAURI GARHWAL', 'JIND', 'WEST
KHASI HILLS', 'GONDA', 'DAKSHIN KANNAD', 'KHARGONE', 'MANDLA', 'MUZAFFARPUR', 'BIJNOR', 'NALANDA',
'TINSUKIA', 'MANDSAUR', 'THANJAVUR', 'AMROHA', 'SIDHI', 'DARJEELING', 'GANDHINAGAR', 'BAREILLY', '
HARDA', 'DHALAI', 'GAYA', 'GORAKHPUR', 'BALLIA', 'CHANDRAPUR', 'SIKAR', 'VELLORE', 'SHEOHAR',
'EAST DISTRICT', 'HINGOLI', 'BISHNUPUR', 'BHIWANI', 'BARMER', 'SIROHI', 'BARDHAMAN', 'PRAKASAM', '
LONGLENG', 'JALORE', 'DARRANG', 'MEERUT', 'DHARWAD', 'SATNA', 'NASHIK', 'ASHOKNAGAR', 'SOLAPUR', '
FARIDKOT', 'KARAULI', 'SITAPUR', 'AMBALA', 'THRISSUR', 'MANDYA', 'EAST SIANG', 'DADRA AND NAGAR HA
VELI', 'THIRUVARUR', 'KORAPUT', 'CHHINDWARA', 'SHIMLA', 'MATHURA', 'NAVSARI', 'KIPHIRE', 'PURI', 'M
ADURAI', 'PALAKKAD', 'FARIDABAD', 'MEWAT', 'JAUNPUR', 'BANDA', 'VARANASI', 'BAKSA', 'WEST SIANG',
'BULANDSHAHR', 'VIRUDHUNAGAR', 'BHADRAK', 'HAVERI', 'PANCH MAHALS', 'IDUKKI', 'HISAR',
'BONGAIGAON', 'EAST GARO HILLS', 'SOLAN', 'BAGHPAT', 'PALWAL', 'RAIPUR', 'BALAGHAT', 'UTTAR
KANNAD', 'KANCHIPURAM', 'JHUNJHUNU', 'DEWAS', 'DHOLPUR', 'KINNAUR', 'KOTA', 'SIVASAGAR',
'RAJGARH', 'NAINITAL', 'KOLHAPUR', 'SONEPUR', 'SURAT', 'SAWAI MADHOPUR', 'KENDRAPARA', 'AJMER', 'S
HIVPURI', 'WOKHA', 'DHEMAJI', 'ROHTAK', 'CHAMPAWAT', 'AMRITSAR', 'VAISHALI', 'WEST GARO HILLS', 'S
IDDHARTH NAGAR', 'FATEHPUR', 'LATUR', 'RAIGARH', 'PURNIA', 'SANGLI', 'OSMANABAD', 'SONIPAT',
'RANGAREDDI', 'DATIA', 'UKHRUL', 'AURAIYA', 'GWALIOR', 'GHAZIPUR', 'GHAZIABAD', 'GADAG', 'SAGAR',
'DAVANGERE', 'THIRUVALLUR', 'MORENA', 'DAUSA', 'YAVATMAL', 'MARIGAON', 'PANIPAT', 'SEHORE', 'ARWAL
', 'MALAPPURAM', 'SURENDRANAGAR', 'DANG', 'WEST TRIPURA', 'KUSHI NAGAR', 'KASARAGOD', 'KARNAL', 'V
ISAKHAPATANAM', 'IMPHAL EAST', 'NAGAUR', 'JABALPUR', 'HARDOI', 'SUPAUL', 'BAHRAICH', 'SANGRUR', 'K
OLAR', 'SHAMLI', 'SIRSA', 'MAU', 'RAE BARELI', 'ALAPPUZHA', 'BELLARY', 'LONGDING', 'RUPNAGAR', 'BA
LOD', 'SPSR NELLORE', 'MOGA', 'UNAKOTI', 'BOKARO', 'AMRAVATI', 'MUNGELI', 'SARAN', 'DUNGARPUR', 'M
AHASAMUND', 'UDAM SINGH NAGAR', 'RAJAURI', 'YAMUNANAGAR', 'UDHAMPUR', 'KODERMA', 'KABIRDHAM',
'BALESHWAR', 'SHEOPUR', 'GARHWA', 'KAPURTHALA', 'NORTH DISTRICT', 'MAHARAJGANJ', 'UNA', 'SOUTH WES
T GARO HILLS', 'POONCH', 'SOUTH WEST KHASI HILLS', 'BUDAUN', 'TIRUPPUR', 'KARAIKAL', 'AMETHI', 'AH
MADABAD', 'NARMADA', 'BATHINDA', 'PHEK', 'LOWER SUBANSIRI', 'SATARA', 'TIRAP', 'LOWER DIBANG VALLE
Y', 'BHARATPUR', 'KATHUA', 'NAGPUR', 'NARAYANPUR', 'CUTTACK', 'KACHCHH', 'TAWANG', 'TEHRI
GARHWAL', 'KULLU', 'JAMMU', 'ERNAKULAM', 'ANANTAPUR', 'MALKANGIRI', 'HARIDWAR', 'SANT KABEER NAGAR
', 'NAWANSHAHR', 'BHAVNAGAR', 'GARIYABAND', 'WAYANAD', 'CHAMPHAI', 'CHANDEL', 'ROHTAS', 'HOOGHLY',
'DIBRUGARH', 'UNNAO', 'ERODE', 'PATIALA', 'BADGAM', 'EAST KAMENG', 'PATAN', 'BUXAR', 'DOHAD', 'ANA
NTNAG', 'NAGAPATTINAM', 'BAGESHWAR', 'LEH LADAKH', 'MAHENDRAGARH', 'DIBANG VALLEY', 'DURG', 'NORTH
TRIPURA', 'PATNA', 'CHURU', 'BEED', 'UPPER SUBANSIRI', 'SHRAVASTI', 'VIDISHA', 'THE NILGIRIS', 'LO
HIT', 'ANAND', 'TAPI', 'YADGIR', 'JHAJJAR', 'HAZARIBAGH', 'BARNALA', 'KAUSHAMBI', 'BANSWARA',
'TARN TARAN', 'PILIBHIT', 'PUNE', 'SAHARANPUR', 'SOUTH ANDAMANS', 'KOTTAYAM', 'JALANDHAR',
'KURUKSHETRA', 'SUKMA', 'REWARI', 'WARDHA', 'WEST DISTRICT', 'NAYAGARH', 'SURAJPUR', 'FIROZEPUR',
'CHATRA', 'DEOGHAR', 'SENAPATI', 'RAMANAGARA', 'AMBEDKAR NAGAR', 'RANCHI', 'LAWNGTLAI', 'PAPUM PAR
E', 'KARGIL', 'KISHTWAR', 'DEOGARH', 'MUKTSAR', 'BHOJPUR', 'SERCHHIP', 'SRINAGAR', 'PULWAMA',
'CHANGLANG', 'WEST JAINTIA HILLS', 'GUMLA', 'GIRIDIH', 'ARIYALUR', 'DUMKA', 'CHIKBALLAPUR',
'KONDAGAON', 'KURUNG KUMEY', 'PALAMU', 'SIMDEGA', 'RAMBAN', 'HOSHIARPUR', 'NORTH AND MIDDLE
ANDAMAN', 'THOUBAL', 'SAIHA', 'SAMBHAL', 'SINDHUDURG', 'WEST SINGHBHUM', 'LAHUL AND SPITI',
'SAMBA', 'SAHEBGANJ', 'KHOWAI', 'CHANDIGARH', 'BANDIPORA', 'SEPAHIJALA', 'LATEHAR', 'MAMIT',
'S.A.S NAGAR', 'GODDA', 'IMPHAL WEST', 'FATEHGARH SAHIB', 'BARAMULLA', 'FAZILKA', 'HYDERABAD',
'PAKUR', 'NORTH GARO HILLS', 'REASI', 'EAST SINGHBUM', 'KULGAM', 'AGAR MALWA', 'SHOPIAN',
'GANDERBAL', 'JAMTARA', 'KHUNTI', 'RAMGARH', 'PALGHAR', 'MUMBAI', 'NAMSAI']
Shape of matrix after one hot encodig (172263, 646)
Shape of matrix after one hot encodig (73828, 646)
['Urad', 'Arhar/Tur', 'Maize', 'Onion', 'Rice', 'Castor seed', 'Niger seed', 'Groundnut',
'Linseed', 'Gram', 'Small millets', 'Sesamum', 'Bajra', 'Moong(Green Gram)', 'Black pepper', 'Sunf
lower', 'Rapeseed &Mustard', 'Dry chillies', 'Wheat', 'Guar seed', 'Other Kharif pulses',
'Masoor', 'Cowpea(Lobia)', 'Sannhamp', 'Turmeric', 'Garlic', 'Sweet potato', 'Coriander', 'Potato'
, 'Other Rabi pulses', 'Horse-gram', 'Sugarcane', 'Banana', 'Papaya', 'Jowar', 'Cotton(lint)', 'P
eas & beans (Pulses)', 'Coconut ', 'Jute', 'Soyabean', 'Korra', 'Barley', 'Arecanut', 'Tobacco', '
Orange', 'Oilseeds total', 'Pulses total', 'Khesari', 'Moth', 'Dry ginger', 'Jack Fruit', 'Other V
egetables', 'Beans & Mutter(Vegetable)', 'other oilseeds', 'Ragi', 'Mesta', 'Mango', 'Brinjal', 'S
afflower', 'Drum Stick', 'Bitter Gourd', 'Water Melon', 'Samai', 'Pome Fruit', 'Cardamom', 'Jute &
mesta', 'Paddy', 'Other Cereals & Millets', 'Other Fresh Fruits', 'Cashewnut', 'Tapioca',
'Colocosia', 'Cabbage', 'Pineapple', 'Cucumber', 'Rubber', 'Ginger', 'Rajmash Kholar', 'Bottle Gou
rd', 'Blackgram', 'Yam', 'Grapes', 'Tomato', 'Bhindi', 'Total foodgrain', 'Citrus Fruit',
'Redish', 'Sapota', 'Ash Gourd', 'Pump Kin', 'Arcanut (Processed)', 'other misc. pulses',
'Cashewnut Raw', 'Varagu', 'Snak Guard', 'Ber', 'Ribed Guard', 'Pome Granet', 'Turnip', 'Lab-Lab',
'Coffee', 'other fibres', 'Atcanut (Raw)', 'Plums', 'Tea', 'Cauliflower', 'Beet Root', 'Lentil', '
Jobster', 'Lemon', 'Carrot', 'Cond-spcs other', 'Bean', 'Peach', 'Kapas', 'Other Citrus Fruit', 'P
ear', 'Cashewnut Processed', 'Apple', 'Perilla', 'Litchi', 'Peas (vegetable)', 'Ricebean
(nagadal)']
Shape of matrix after one hot encodig (172263, 123)
Shape of matrix after one hot encodig (73828, 123)
['Kharif   ', 'Whole Year ', 'Winter   ', 'Autumn   ', 'Rabi   ', 'Summer   ']
Shape of matrix after one hot encodig (172263, 6)
Shape of matrix after one hot encodig (73828, 6)
[2000, 2001, 2004, 2008, 2005, 2003, 2009, 2002, 2011, 2007, 2013, 2012, 1999, 2006, 2010, 1997, 19
98, 2014, 2015]
Shape of matrix after one hot encodig (172263, 19)
Shape of matrix after one hot encodig (73828, 19)
After Area Normalization
(1, 172263) (172263,)

```
(1, 172263) (172263,)
(1, 73828) (73828,)
==================================================
Final Data Matrix
(172263, 828) (172263,)
(73828, 828) (73828,)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-27-54e2c9986a96> in <module>()
    249
    250 lm = linear_model.LinearRegression()
--> 251 model = lm.fit(X_tr, y_train)
    252 predt_test = lm.predict(X_te)
    253 print(mean_squared_error(y_test,predt_test))

/usr/local/lib/python3.6/dist-packages/sklearn/linear_model/_base.py in fit(self, X, y,
sample_weight)
    490         n_jobs_ = self.n_jobs
    491         X, y = check_X_y(X, y, accept_sparse=['csr', 'csc', 'coo'],
--> 492                          y_numeric=True, multi_output=True)
    493
    494         if sample_weight is not None:

/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py in check_X_y(X, y,
accept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd,
multi_output, ensure_min_samples, ensure_min_features, y_numeric, warn_on_dtype, estimator)
    756     if multi_output:
    757         y = check_array(y, 'csr', force_all_finite=True, ensure_2d=False,
--> 758                         dtype=None)
    759     else:
    760         y = column_or_1d(y, warn=True)

/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py in check_array(array,
accept_sparse, accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd,
ensure_min_samples, ensure_min_features, warn_on_dtype, estimator)
    576         if force_all_finite:
    577             _assert_all_finite(array,
--> 578                                allow_nan=force_all_finite == 'allow-nan')
    579
    580     if ensure_min_samples > 0:

/usr/local/lib/python3.6/dist-packages/sklearn/utils/validation.py in _assert_all_finite(X,
allow_nan, msg_dtype)
     58                     msg_err.format
     59                     (type_err,
---> 60                      msg_dtype if msg_dtype is not None else X.dtype)
     61             )
     62     # for object dtype data, we only check for NaNs (GH-13254)

ValueError: Input contains NaN, infinity or a value too large for dtype('float64').
```

Linear Regression Model

In [ ]:

```
import matplotlib.pyplot as plt
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import roc_auc_score
import math
"""
y_true : array, shape = [n_samples] or [n_samples, n_classes]
Final Data matrix
((22445, 10144), (22445,))
((11055, 10144), (11055,))
((16500, 10144), (16500,))
==========================================================================================

y_true : array, shape = [n_samples] or [n_samples, n_classes]
True binary labels or binary label indicators.
y_score : array, shape = [n_samples] or [n_samples, n_classes]
Target scores, can either be probability estimates of the positive class, confidence values, or no
n-thresholded measure of
decisions (as returned by "decision_function" on some classifiers).
```

```
For binary y_true, y_score is supposed to be the score of the class with greater label.
"""
train_auc = []
cv_auc = []
neigh =linear_model.LinearRegression()
neigh.fit(X_tr, y_train)
y_test_pred =neigh.predict(X_te)
#y_cv_pred =neigh.predict(X_cv)
mean_squared_error(y_test,y_test_pred)
```

Out[ ]:

241116104525435.3

In [ ]:

```
lm = linear_model.LinearRegression()
model = lm.fit(X_tr, y_train)
predictions = lm.predict(X_te)
mean_squared_error(y_test,predictions)
print(model.coef_)
print(mean_squared_error(y_test,predictions))
```

```
[ 0.00000000e+00 -1.50099316e+06 -1.33461461e+06 -1.08065839e+06
 -1.03788102e+06 -1.15888400e+06 -1.02323732e+06  0.00000000e+00
  2.23823370e+07  0.00000000e+00 -1.36413284e+06 -1.41026356e+06
 -9.87606754e+05 -9.26256551e+05 -1.26436520e+06 -1.37981678e+06
 -1.52060913e+06  0.00000000e+00 -1.68044314e+06  0.00000000e+00
  0.00000000e+00  0.00000000e+00 -9.15519955e+05  0.00000000e+00
 -8.03635164e+05  2.60577794e+05 -1.02031503e+06 -1.39492368e+06
  0.00000000e+00  0.00000000e+00  0.00000000e+00 -4.34680187e+05
  1.62489128e+06 -6.98806898e+05  0.00000000e+00 -3.48108639e+05
 -5.79308610e+04 -4.41228242e+05  7.38473192e+03 -1.07267605e+05
 -1.51017784e+06  0.00000000e+00  1.08926685e+07 -3.02759325e+05
 -4.50285929e+05  0.00000000e+00  8.90336965e+04 -1.72231808e+05
  0.00000000e+00  1.69964844e+04 -7.80555664e+04 -1.40996899e+06
  6.39461176e+04 -3.64376298e+04 -1.62226505e+06 -1.20724656e+06
 -3.19527314e+05 -1.61319863e+06 -6.40440236e+05 -1.37062094e+06
 -1.38684324e+06 -1.60459338e+06  0.00000000e+00  0.00000000e+00
 -1.20099341e+06  8.87344653e+03  1.08524805e+05 -1.35130096e+06
  0.00000000e+00 -6.72269662e+05  1.77416788e+05  0.00000000e+00
  6.28528755e+04  2.37055979e+05 -2.41322399e+05 -5.88754467e+04
 -2.49495380e+04  0.00000000e+00 -1.61754846e+05 -1.61114046e+06
 -1.04436294e+05  7.14512497e+04 -7.23708544e+04  2.82209747e+04
 -6.24354347e+05  4.12436117e+04 -1.70991219e+05  4.85553476e+05
 -8.45293785e+04 -1.45673792e+06  0.00000000e+00 -1.44913204e+06
  9.33104457e+04 -2.07534496e+05 -1.53259625e+06 -8.96534243e+05
  5.55195823e+04 -1.13894478e+04 -1.54708317e+05 -1.29530377e+06
 -1.49095408e+06 -2.41846030e+05  2.31425656e+05  1.44161605e+05
 -3.56697047e+05  1.22615823e+05 -6.08143881e+04  1.38246266e+05
 -1.47561552e+06 -1.32509332e+06 -1.17303025e+05  5.75346293e+05
 -6.35149729e+05 -4.70636516e+05 -6.88490074e+05 -1.36170156e+06
  2.86471428e+07  7.93795478e+04 -1.43108950e+06 -1.54707487e+06
 -1.64035195e+05 -5.11800291e+05  0.00000000e+00 -1.60698691e+06
 -3.55790637e+05 -1.01393275e+06 -7.17921465e+05 -8.23436089e+04
 -1.35779713e+06 -1.42742115e+06 -1.39172974e+06  3.81726637e+06
 -3.99053491e+05  3.61047145e+05 -1.94123994e+04  1.39985430e+05
 -4.64377925e+04 -1.27085076e+06 -2.45207599e+05 -1.31623787e+06
  0.00000000e+00 -3.22992431e+05 -1.74967414e+05  5.07948050e+04
 -3.40433577e+05 -1.34201552e+06  4.00682861e+04 -4.36208055e+04
  0.00000000e+00  5.45094576e+03 -4.49382689e+05 -1.67113312e+05
 -3.18686118e+05 -9.01838885e+05 -1.47525926e+06 -8.76278808e+04
 -4.45277748e+05  0.00000000e+00  0.00000000e+00 -1.14799567e+06
 -2.44720553e+04  2.25604677e+05 -7.84824741e+05 -1.49582940e+06
 -5.04647038e+04 -1.52453952e+06 -1.49983381e+06 -1.72837999e+05
 -1.55085722e+06 -5.00470439e+04 -1.61272672e+06 -1.40533842e+06
 -1.05435950e+05 -1.41298973e+06 -7.57885772e+05  4.50711628e+04
 -1.07344277e+06  1.90243209e+05 -1.43493049e+06 -1.13118962e+06
 -5.67434789e+04 -7.31047452e+04 -5.59888915e+05  0.00000000e+00
  2.85518963e+05 -1.31925930e+06 -1.38809227e+05  2.93762113e+05
 -1.21810173e+06 -3.58264330e+05  1.82138208e+05  2.75941221e+05
 -1.67632056e+05 -8.59513035e+04  1.23861631e+05 -1.30929159e+06
 -1.36856920e+06  8.16025917e+04 -1.56379663e+06  1.42913548e+05
 -6.08897804e+04  2.42382518e+04 -2.74899483e+05 -1.27200655e+06
 -4.02034540e+05 -2.23038573e+04  0.00000000e+00 -6.07064472e+05
```

```
-1.31956022e+06 -4.45741364e+05 -1.17003432e+05  4.55868391e+04
-1.46622530e+06 -1.52611119e+06  1.57067923e+05 -2.29063957e+05
-1.44461611e+06 -5.15457068e+05 -1.93107867e+06  0.00000000e+00
-1.98179290e+05 -1.49262308e+06 -1.83944698e+05 -5.42706237e+04
 6.81884721e+03 -1.40958695e+06  0.00000000e+00  4.78769996e+04
 1.17299117e+04  7.82284201e+05  8.63364786e+04 -1.33157353e+06
-5.48865555e+05  3.76306889e+06 -1.64062689e+06  2.95780705e+04
 2.91559412e+04 -3.74767411e+06 -3.34347830e+04 -1.34923161e+06
 7.10275266e+05  1.48040444e+05 -3.98578535e+05 -5.47070693e+05
 0.00000000e+00 -1.44129677e+06  1.87797605e+07  1.30624963e+05
-1.25964031e+06 -1.81290736e+06  7.88002478e+04 -1.40029174e+06
-1.05781558e+06  0.00000000e+00 -1.29479281e+06 -2.65979380e+05
-1.69668263e+06 -1.26548182e+06  1.55917716e+05 -2.97334787e+05
 7.20838974e+03 -1.17994835e+06  8.36533036e+04 -1.31551875e+06
 8.66238724e+03 -1.30474810e+06 -1.00765256e+05  0.00000000e+00
-1.53720654e+06  0.00000000e+00  2.56783185e+07 -1.23496783e+06
-3.03388834e+05 -2.90247825e+04 -1.44912394e+06 -1.15319213e+05
-2.56391738e+04 -1.28960408e+06  2.30502268e+04 -1.31276422e+06
 1.05968906e+06  2.43086526e+06 -6.26394775e+04 -1.65044333e+06
-1.50133094e+06 -1.55831006e+06 -7.60486578e+05  1.07356035e+07
-1.52607354e+06 -9.40462508e+05  0.00000000e+00 -1.34971245e+05
-1.36667248e+06 -4.47525989e+05  0.00000000e+00  0.00000000e+00
-2.45771338e+05  7.45339958e+04  0.00000000e+00 -3.82465420e+05
-8.19202304e+04 -1.00966500e+05 -1.11249142e+06 -1.17391203e+07
-3.53613700e+05 -1.42634869e+06 -1.09927039e+05 -3.61582179e+05
-2.11950544e+05 -7.89227722e+04 -3.25200050e+04  1.77171495e+05
-1.36592527e+06  5.76993936e+06 -1.43954458e+06 -3.21204266e+05
-1.45371620e+06  0.00000000e+00 -4.88333223e+05 -1.41565067e+06
 1.11966854e+05 -1.53763238e+06 -4.30841455e+05 -2.19468570e+05
-1.36423019e+07 -4.12894944e+05  0.00000000e+00 -1.47940883e+06
 1.80100041e+05  2.75619444e+04  7.46763770e+04  0.00000000e+00
 2.57468028e+05 -1.00461080e+05 -1.12880767e+06 -1.67843462e+04
-1.31413133e+06 -1.16472973e+06 -5.09054293e+05 -1.09402094e+06
-1.39899203e+06 -1.59656395e+06 -5.57677136e+05 -1.46332581e+06
 2.74293108e+04 -1.54893205e+06 -6.72578857e+04 -7.33534879e+04
-1.38883041e+06 -1.45641164e+06 -1.61940085e+06 -1.40847082e+06
-1.29492382e+06 -1.44082594e+06  0.00000000e+00 -8.84346157e+04
 1.15631778e+05 -3.08683423e+05  3.74806847e+04 -1.54936256e+06
-1.42134940e+06  0.00000000e+00  5.46598928e+05 -1.22614832e+06
-6.97824969e+03  0.00000000e+00 -2.93006237e+04  1.18933883e+04
-1.41059923e+06 -1.34604536e+06 -1.40554026e+06 -3.50997280e+05
 8.60285988e+04 -1.40643361e+06  1.38976337e+05 -1.71749032e+05
 0.00000000e+00  0.00000000e+00  5.44613574e+04  0.00000000e+00
-1.31340807e+06 -7.21589345e+04  6.53973072e+04  0.00000000e+00
 3.61891710e+05 -5.38649587e+05  0.00000000e+00 -1.48132270e+06
-4.13729565e+05 -2.03541429e+05 -6.57123254e+05 -2.34190827e+05
-1.25482460e+06 -6.04360679e+05 -1.32090372e+06  4.85232441e+03
-1.23751966e+06  0.00000000e+00  7.37484959e+03 -1.87510779e+04
-1.40533539e+06  0.00000000e+00  0.00000000e+00 -1.06053614e+06
-7.17674908e+05 -1.21455328e+06  5.81452057e+04  0.00000000e+00
-1.74237764e+06  4.26202881e+04 -1.49463275e+06 -1.42867271e+06
 8.20654611e+04  2.81284176e+04 -1.16567967e+05 -4.48167771e+05
 1.03641860e+05 -8.09772465e+05 -3.64884260e+04 -2.50639607e+05
-1.83000264e+07 -2.31407189e+05 -1.38454265e+06 -1.33173031e+06
-2.62785652e+05 -1.40466278e+06 -5.20617819e+04 -1.57821420e+04
-1.43738363e+06 -5.57786264e+05  2.83006186e+05 -8.47271473e+04
-1.94319276e+05 -6.31107378e+05  1.01345536e+04  0.00000000e+00
-8.99565636e+05 -1.67627547e+05 -5.85564522e+05 -1.46104549e+06
-9.36596082e+05 -5.07181233e+05 -6.20648031e+04 -1.52634258e+06
-3.58845147e+03  0.00000000e+00  1.09094575e+04 -1.49769534e+06
-7.96638775e+04  0.00000000e+00 -1.31441591e+05 -2.85069787e+05
-1.48186672e+06  0.00000000e+00 -4.84601234e+05 -1.14852740e+05
 0.00000000e+00 -2.56593082e+04 -1.22091790e+06  5.27360342e+05
 2.51803016e+04  1.54912845e+05  0.00000000e+00 -4.72459474e+05
 5.99221662e+06 -6.67975645e+05  0.00000000e+00 -2.32963043e+05
 1.91057629e+03 -5.16371980e+06  7.29022660e+04  0.00000000e+00
-4.79118488e+05 -1.24362235e+06 -1.21288190e+06  5.57824063e+05
-4.33016516e+05 -2.57629218e+05  2.03504601e+05 -4.09515346e+05
-3.31592186e+04 -1.36762623e+05 -2.02362138e+05  0.00000000e+00
-3.66382978e+05  9.76678867e+03 -1.71627395e+05 -6.82916259e+04
 0.00000000e+00 -1.46198146e+06  8.42209545e+05 -8.84559810e+04
-1.50369275e+06  7.39717188e+05  0.00000000e+00  0.00000000e+00
-2.04212146e+04 -1.51335685e+05 -1.32856024e+06 -4.63493695e+05
-3.14650320e+05 -1.33655885e+06  7.98021165e+04 -1.00874668e+05
 0.00000000e+00 -1.16839284e+06  0.00000000e+00  0.00000000e+00
 1.19307294e+04  1.22995106e+06 -1.38525698e+06 -9.52536772e+04
-1.38800183e+06 -1.49104403e+06 -1.35986488e+05 -6.01567729e+04
```

```
 1.97087834e+05 -2.34496630e+05 -1.42885096e+06  0.00000000e+00
 2.78569016e+06 -1.43032488e+06 -1.39318081e+06 -1.83612624e+05
 0.00000000e+00  0.00000000e+00 -1.39482613e+06  0.00000000e+00
-8.15401618e+05  7.21114684e+04 -1.60934560e+06 -1.35081666e+06
-1.21012037e+06  1.03363532e+05  1.32173024e+04 -7.14629281e+04
 0.00000000e+00 -8.11850846e+05 -1.98315945e+07 -8.79647703e+04
-2.50744359e+05 -1.98493349e+05 -1.62242401e+05 -1.22929997e+05
-2.89794751e+05 -1.37029745e+06 -1.30649790e+06 -3.97393527e+05
-1.38575829e+06  4.97814654e+04 -1.53307540e+06  2.41207997e+05
 3.86075753e+04 -1.94173596e+05 -2.51101723e+05 -7.08468083e+04
 2.78820171e+03 -1.03726802e+05 -5.87716631e+05 -5.34242553e+03
-1.41398776e+06  4.70789286e+04  0.00000000e+00 -1.20697685e+06
-4.65473453e+05 -1.34037469e+06 -1.37695764e+06  0.00000000e+00
-9.71839115e+05 -6.68728681e+05  3.23803814e+05 -4.09655247e+05
-1.43171474e+06  3.04687174e+04 -9.24507041e+04 -1.25008901e+06
-1.42941217e+06 -1.35064555e+05 -1.14059744e+06 -4.34680187e+05
-3.61937141e+04  0.00000000e+00  0.00000000e+00 -4.39436125e+05
 1.57548683e+05  2.36078929e+05  0.00000000e+00 -1.62338695e+06
-4.15556203e+05 -1.57035583e+05 -8.96386896e+05  2.04742385e+03
 0.00000000e+00  0.00000000e+00  9.41350123e+04 -1.43920414e+06
-1.16597506e+05 -1.16846103e+06 -1.64015652e+06  0.00000000e+00
-8.05670397e+04 -1.30080409e+06  4.84398615e+04  0.00000000e+00
-3.91957404e+05 -2.18570966e+05  1.79102981e+05  3.04041608e+05
-1.78151257e+05  0.00000000e+00 -5.02886034e+05 -6.85279215e+04
-1.24655513e+06 -5.73722482e+04  1.43499231e+05 -3.49182270e+05
-2.16610334e+05  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00 -4.19397030e+05  1.68373804e+04
-5.76198087e+05 -1.42380204e+06 -1.26542979e+06 -1.46784951e+06
 2.01840835e+04 -1.34790392e+06  1.14710866e+05 -5.36697011e+05
-2.74255508e+05 -1.83959841e+05 -3.90707793e+05 -1.67262112e+06
-1.95570062e+06  1.89046043e+04  0.00000000e+00 -3.23012341e+05
 4.24738764e+05 -7.96363453e+02  0.00000000e+00 -1.29843060e+05
-9.03300466e+04 -2.65450805e+05 -3.53773547e+05 -1.16469864e+06
 3.54354920e+04  4.20981706e+03  0.00000000e+00 -5.34505786e+05
 0.00000000e+00  2.90657907e+05  0.00000000e+00 -5.81350160e+04
-1.38112561e+06 -1.44298011e+06  0.00000000e+00  5.30289764e+05
-1.23128983e+06  4.63829649e+04 -9.82895645e+05  5.77174836e+04
 0.00000000e+00 -1.42226223e+06 -1.17441663e+06  0.00000000e+00
-5.59861559e+05 -1.52014822e+06  3.91530507e+05 -1.05440702e+06
-1.32511213e+05 -2.33694236e+05 -1.68439729e+06 -1.42631006e+06
 4.57609711e+05 -2.91325267e+05 -1.15079154e+06  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00 -3.33272618e+06 -1.42409240e+06
-1.16951424e+06 -1.64894357e+06 -1.99804707e+06  0.00000000e+00
 0.00000000e+00 -1.17900136e+06 -1.40690141e+06 -2.90324931e+06
-1.79246027e+06  0.00000000e+00 -1.33141606e+06  0.00000000e+00
-1.45976944e+06 -1.81915427e+06 -1.37223748e+06 -1.18691283e+06
-1.66755757e+06 -1.19232081e+06 -1.49022524e+06 -1.65202418e+06
-1.24879858e+06  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00 -1.35738886e+06 -1.03843251e+07 -1.42907699e+06
 0.00000000e+00 -6.06825616e+06  0.00000000e+00 -1.55621046e+06
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00 -1.53320340e+06 -3.42993765e+06  0.00000000e+00
 0.00000000e+00 -1.77650604e+06 -1.30277349e+06 -1.59791109e+06
-1.45820892e+06  0.00000000e+00  0.00000000e+00  0.00000000e+00
-4.58104158e+06 -6.00558344e+06 -3.68726320e+06 -2.05147358e+06
-1.36110954e+06 -3.84366510e+06 -2.18617671e+06  0.00000000e+00
 0.00000000e+00 -7.73706771e+06  0.00000000e+00  0.00000000e+00
-2.03501604e+06  0.00000000e+00  0.00000000e+00 -1.73326960e+06
 0.00000000e+00 -6.18597329e+06 -1.58234652e+06  0.00000000e+00
-2.88316255e+06 -5.65261944e+06  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00 -1.16113871e+06  0.00000000e+00
-6.50869585e+06 -1.46565871e+06  0.00000000e+00 -1.45025592e+06
-1.27995166e+06 -2.22010648e+06 -2.34047730e+06 -2.16410631e+06
 0.00000000e+00  0.00000000e+00  0.00000000e+00 -3.18137709e+06
 0.00000000e+00  0.00000000e+00 -1.74855246e+06 -2.00119939e+06
-1.97868240e+06 -1.55028344e+06  0.00000000e+00 -6.28827435e+06
 0.00000000e+00  0.00000000e+00  0.00000000e+00 -2.33666547e+06
-2.04831407e+06  0.00000000e+00  0.00000000e+00 -1.40527642e+06
 0.00000000e+00  0.00000000e+00  0.00000000e+00 -1.44970381e+06
-2.49759324e+07 -2.86497213e+06  0.00000000e+00 -2.54802066e+06
 0.00000000e+00 -1.39331673e+06 -9.10595098e+06  0.00000000e+00
-1.40997679e+06 -2.91345665e+06 -1.39665448e+06  0.00000000e+00
 0.00000000e+00 -3.04211804e+06  0.00000000e+00  0.00000000e+00
```

```
  0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
  3.56757308e+08]
241116104525435.3
```

In [ ]:

```
import xgboost as xgb
model1=xgb.XGBRegressor(n_estimators=1000,learning_rate=0.01,max_depth=2,n_jobs=-1,random_state=2)
model1.fit(X_tr,y_train)
print(model1)
pred=model.predict(X_te)
print(mean_squared_error(y_test,pred))
```

```
[04:10:43] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0,
             importance_type='gain', learning_rate=0.01, max_delta_step=0,
             max_depth=2, min_child_weight=1, missing=None, n_estimators=1000,
             n_jobs=-1, nthread=None, objective='reg:linear', random_state=2,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
             silent=None, subsample=1, verbosity=1)
391840535195630.0
```

Hyper Parameter Tunning through GridSearchCV in XGboost

In [ ]:

```
from sklearn.model_selection import GridSearchCV
import xgboost as xgb

from sklearn.ensemble import RandomForestClassifier
estimators=[10, 50, 100, 200, 300,500, 1000]
depth=[2, 3, 4,7, 8, 10,12]

for n in estimators:
  for d in depth:
    model =xgb.XGBRegressor(n_estimators=n,learning_rate=0.01,max_depth=d,n_jobs=-1,random_state=2)
    model.fit(X_tr, y_train)
    pred=model.predict(X_te)
    print(mean_squared_error(y_test,pred))
    print(n,d,"\n")
```

```
[03:38:26] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
225067863552662.62
10 2

[03:38:27] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
217301479360221.53
10 3

[03:38:28] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
228412472408679.06
10 4

[03:38:29] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
243196367600026.12
10 7

[03:38:30] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
245614334119734.53
10 8

[03:38:32] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
```

```
246761065991174.84
10 10

[03:38:33] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
246752001746062.5
10 12

[03:38:35] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
152868017179835.1
50 2

[03:38:37] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
133369534652106.03
50 3

[03:38:40] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
166302612392029.28
50 4

[03:38:44] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
223450884657524.75
50 7

[03:38:48] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
223849002567361.6
50 8

[03:38:54] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
235879262572992.9
50 10

[03:39:00] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
237170592703307.47
50 12

[03:39:08] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
120599102062034.08
100 2

[03:39:12] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
117652394309401.02
100 3

[03:39:17] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
147866317446275.6
100 4

[03:39:23] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
237589907735752.28
100 7

[03:39:33] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
231199198416576.66
100 8

[03:39:43] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
242435539547009.72
100 10

[03:39:55] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
243752317804734.84
100 12
```

```
[03:40:10] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
107263651055402.53
200 2

[03:40:18] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
121968772088607.14
200 3

[03:40:28] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
158894023116353.47
200 4

[03:40:40] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
270656272794578.84
200 7

[03:40:58] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
288937494398943.94
200 8

[03:41:18] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
278869241020103.25
200 10

[03:41:42] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
281369463087329.88
200 12

[03:42:11] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
100801812347773.66
300 2

[03:42:23] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
123667808480964.78
300 3

[03:42:37] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
152354109749146.66
300 4

[03:42:55] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
294055935680657.5
300 7

[03:43:22] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
342048529007394.4
300 8

[03:43:51] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
310134096460826.7
300 10

[03:44:28] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
307502806735197.25
300 12

[03:45:13] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
89493749900162.8
500 2

[03:45:33] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
```

```
n favor of reg:squarederror.
118343885601093.73
500 3

[03:45:57] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
153211625910697.72
500 4

[03:46:26] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
334420801524084.56
500 7

[03:47:10] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
381835286309740.5
500 8

[03:48:01] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
359021013219500.2
500 10

[03:49:06] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
361499237317695.0
500 12

[03:50:23] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
98291586734419.34
1000 2

[03:51:03] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
121957924907711.25
1000 3

[03:51:53] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
165984237356534.0
1000 4

[03:52:52] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
342221048586180.0
1000 7

[03:54:23] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
397894198144352.94
1000 8

[03:56:06] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
395645936730518.3
1000 10

[03:58:12] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated i
n favor of reg:squarederror.
391840535195630.0
1000 12
```

Conclusion

Steps of Creating this model

1. Check Null and Missing Values in Dataset
2. Do EDA
3. Convert Categorical variable into one hot encoding
4. normalized the numerical data

5. Create a simple model
6. Get RMSE score

I just want to know that the score that is got is valid or not. And is there any procedure is missing or tell me right way to create a model.

Where I am getting wrong ! tell me.....