|  |  |
| --- | --- |
| **DATE** | **28/10/2023** |
| **TEAM ID** | **687** |
| **PROJECT NAME** | **AI Driven company registration Trend prediction and Exploration** |
| **DEVELOPMENT** | **01** |

**Project Goal:**

The goal of this project is to process the company registration data for time series forecasting, you'll typically need to perform several steps, including data loading, cleaning, and transformation.

**Data Loading:**

• Import the necessary Python libraries, such as Pandas and NumPy.

• Load the dataset from the provided link. You can use Pandas to read CSV files or other formats.

**Data Pre-processing:**

• Check for and handle any missing values.

• Convert the timestamp column to a datetime data type for time- based analysis.

**Exploratory Data Analysis (EDA):**

• Perform basic data exploration to understand the data's characteristics.

• Visualize the time series data to identify trends and patterns.

**Python Script:**

**Importing libraries**

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import math

**importing dataset**

data=pd.read\_csv("Data\_Gov\_Tamil\_Nadu\_2.csv")

df = data

**Data pre-processing**

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 150871 entries, 0 to 150870

Data columns (total 17 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 CORPORATE\_IDENTIFICATION\_NUMBER 150871 non-null object

1 COMPANY\_NAME 150871 non-null object

2 COMPANY\_STATUS 150871 non-null object

3 COMPANY\_CLASS 150537 non-null object

4 COMPANY\_CATEGORY 150537 non-null object

5 COMPANY\_SUB\_CATEGORY 150537 non-null object

6 DATE\_OF\_REGISTRATION 150832 non-null object

7 REGISTERED\_STATE 150871 non-null object

8 AUTHORIZED\_CAP 150871 non-null float64

9 PAIDUP\_CAPITAL 150871 non-null float64

10 INDUSTRIAL\_CLASS 150561 non-null object

11 PRINCIPAL\_BUSINESS\_ACTIVITY\_AS\_PER\_CIN 150871 non-null object

12 REGISTERED\_OFFICE\_ADDRESS 150781 non-null object

13 REGISTRAR\_OF\_COMPANIES 150697 non-null object

14 EMAIL\_ADDR 112742 non-null object

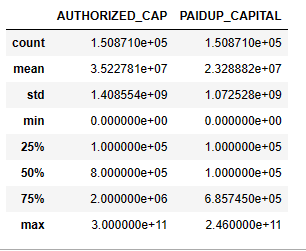
15 LATEST\_YEAR\_ANNUAL\_RETURN 74982 non-null object

16 LATEST\_YEAR\_FINANCIAL\_STATEMENT 75089 non-null object

dtypes: float64(2), object(15)

memory usage: 19.6+ MB

df.describe()



df.columns

Index(['CORPORATE\_IDENTIFICATION\_NUMBER', 'COMPANY\_NAME', 'COMPANY\_STATUS',

'COMPANY\_CLASS', 'COMPANY\_CATEGORY', 'COMPANY\_SUB\_CATEGORY',

'DATE\_OF\_REGISTRATION', 'REGISTERED\_STATE', 'AUTHORIZED\_CAP',

'PAIDUP\_CAPITAL', 'INDUSTRIAL\_CLASS',

'PRINCIPAL\_BUSINESS\_ACTIVITY\_AS\_PER\_CIN', 'REGISTERED\_OFFICE\_ADDRESS',

'REGISTRAR\_OF\_COMPANIES', 'EMAIL\_ADDR', 'LATEST\_YEAR\_ANNUAL\_RETURN',

'LATEST\_YEAR\_FINANCIAL\_STATEMENT'],

dtype='object')

print(f"Total Values : {len(df)}\n")

for x in df.columns:

print(f'{len(df)-df[x].count()} values missing in {x}')

Total Values : 150871

0 values missing in CORPORATE\_IDENTIFICATION\_NUMBER

0 values missing in COMPANY\_NAME

0 values missing in COMPANY\_STATUS

334 values missing in COMPANY\_CLASS

334 values missing in COMPANY\_CATEGORY

334 values missing in COMPANY\_SUB\_CATEGORY

39 values missing in DATE\_OF\_REGISTRATION

0 values missing in REGISTERED\_STATE

0 values missing in AUTHORIZED\_CAP

0 values missing in PAIDUP\_CAPITAL

310 values missing in INDUSTRIAL\_CLASS

0 values missing in PRINCIPAL\_BUSINESS\_ACTIVITY\_AS\_PER\_CIN

90 values missing in REGISTERED\_OFFICE\_ADDRESS

174 values missing in REGISTRAR\_OF\_COMPANIES

38129 values missing in EMAIL\_ADDR

75889 values missing in LATEST\_YEAR\_ANNUAL\_RETURN

75782 values missing in LATEST\_YEAR\_FINANCIAL\_STATEMENT

dropCols = ["LATEST\_YEAR\_FINANCIAL\_STATEMENT","EMAIL\_ADDR", "COMPANY\_NAME","LATEST\_YEAR\_ANNUAL\_RETURN","CORPORATE\_IDENTIFICATION\_NUMBER","REGISTERED\_OFFICE\_ADDRESS"]

df = df.drop(dropCols, axis=1)

df["DATE\_OF\_REGISTRATION"] = df["DATE\_OF\_REGISTRATION"].apply(pd.to\_datetime)

df['REG\_YEAR'] = df['DATE\_OF\_REGISTRATION'].dt.year

df['REG\_MONTH'] = df['DATE\_OF\_REGISTRATION'].dt.month

for x in df.columns:

print(f'{x}: {len(df[x].unique())}\n{df[x].unique()[:20]}\n')

COMPANY\_STATUS: 11

['NAEF' 'ACTV' 'ULQD' 'LIQD' 'AMAL' 'DISD' 'UPSO' 'STOF' 'D455' 'CLLP'

'CLLD']

COMPANY\_CLASS: 4

[nan 'Public' 'Private' 'Private (One Person Company)']

COMPANY\_CATEGORY: 4

[nan 'Company limited by Shares' 'Company Limited by Guarantee'

'Unlimited Company']

COMPANY\_SUB\_CATEGORY: 6

[nan 'Non-govt company' 'Union Govt company'

'Subsidiary of Foreign Company' 'State Govt company'

'Guarantee and Association comp']

DATE\_OF\_REGISTRATION: 13541

['1961-01-12T00:00:00.000000000' 'NaT'

'1982-01-03T00:00:00.000000000' '1995-05-09T00:00:00.000000000'

'1996-11-04T00:00:00.000000000' '1997-04-25T00:00:00.000000000'

'1997-11-06T00:00:00.000000000' '1998-10-27T00:00:00.000000000'

'2000-01-05T00:00:00.000000000' '1999-07-13T00:00:00.000000000'

'1999-02-11T00:00:00.000000000' '2000-06-14T00:00:00.000000000'

'2000-07-17T00:00:00.000000000' '2001-01-24T00:00:00.000000000'

'2001-08-03T00:00:00.000000000' '2001-03-22T00:00:00.000000000'

'2001-08-16T00:00:00.000000000' '2001-11-21T00:00:00.000000000'

'2001-12-24T00:00:00.000000000' '1995-09-23T00:00:00.000000000']

REGISTERED\_STATE : 1

['Tamil Nadu']

AUTHORIZED\_CAP : 1623

[0.000e+00 1.250e+07 1.500e+10 1.500e+08 5.000e+05 2.500e+08 5.000e+06

5.000e+07 1.600e+07 5.500e+07 8.000e+06 3.000e+08 4.000e+08 4.000e+07

1.353e+08 1.600e+08 2.000e+08 1.000e+08 1.000e+05 7.000e+07]

PAIDUP\_CAPITAL : 16294

[0.00000000e+00 6.27350000e+06 1.16730000e+08 3.83500000e+07

4.00000000e+07 1.89066750e+08 4.99656600e+07 1.04117300e+07

4.62420000e+07 3.00000000e+01 2.63300000e+08 2.04715000e+08

2.81676800e+08 3.45500000e+07 1.07000000e+08 1.14525000e+08

1.59827370e+08 4.93480000e+07 7.00000000e+03 2.16140336e+09]

INDUSTRIAL\_CLASS : 1563

[nan '1117' '1119' '1122' '1132' '1133' '1211' '1222' '1409' '1542' '2310'

'2511' '3210' '5001' '8031' '11101' '13206' '14200' '14299' '15100']

PRINCIPAL\_BUSINESS\_ACTIVITY\_AS\_PER\_CIN : 17

['Agriculture & allied' 'Mining and quarrying' 'Manufacturing'

'Electricity gas and water supply' 'Construction'

'Wholesale and retail trade repair of motor vehicles motorcycles and personal and household goods'

'Hotels and restaurants' 'Transport storage and communications'

'Financial intermediation' 'Real estate renting and business activities'

'Health and social work'

'Other community social and personal service activities'

'Extraterritorial organizations and bodies' 'Unclassified'

'Activities of private households as employers and undifferentiated production activities of private households'

'Public administration and defence compulsory social security'

'Education']

REGISTRAR\_OF\_COMPANIES : 5

['ROC\xa0DELHI' 'ROC\xa0COIMBATORE' 'ROC\xa0CHENNAI' nan

'ROC\xa0HYDERABAD']

REG\_YEAR : 131

[1961. nan 1982. 1995. 1996. 1997. 1998. 2000. 1999. 2001. 2005. 2002.

2003. 2004. 2006. 2007. 2008. 2009. 2010. 2011.]

DATE\_OF\_REGISTRATION : 13541

['1961-01-12T00:00:00.000000000' 'NaT'

'1982-01-03T00:00:00.000000000' '1995-05-09T00:00:00.000000000'

'1996-11-04T00:00:00.000000000' '1997-04-25T00:00:00.000000000'

'1997-11-06T00:00:00.000000000' '1998-10-27T00:00:00.000000000'

'2000-01-05T00:00:00.000000000' '1999-07-13T00:00:00.000000000'

'1999-02-11T00:00:00.000000000' '2000-06-14T00:00:00.000000000'

'2000-07-17T00:00:00.000000000' '2001-01-24T00:00:00.000000000'

'2001-08-03T00:00:00.000000000' '2001-03-22T00:00:00.000000000'

'2001-08-16T00:00:00.000000000' '2001-11-21T00:00:00.000000000'

'2001-12-24T00:00:00.000000000' '1995-09-23T00:00:00.000000000']

REGISTERED\_STATE : 1

['Tamil Nadu']

AUTHORIZED\_CAP : 1623

[0.000e+00 1.250e+07 1.500e+10 1.500e+08 5.000e+05 2.500e+08 5.000e+06

5.000e+07 1.600e+07 5.500e+07 8.000e+06 3.000e+08 4.000e+08 4.000e+07

1.353e+08 1.600e+08 2.000e+08 1.000e+08 1.000e+05 7.000e+07]

PAIDUP\_CAPITAL : 16294

[0.00000000e+00 6.27350000e+06 1.16730000e+08 3.83500000e+07

4.00000000e+07 1.89066750e+08 4.99656600e+07 1.04117300e+07

4.62420000e+07 3.00000000e+01 2.63300000e+08 2.04715000e+08

2.81676800e+08 3.45500000e+07 1.07000000e+08 1.14525000e+08

1.59827370e+08 4.93480000e+07 7.00000000e+03 2.16140336e+09]

INDUSTRIAL\_CLASS : 1563

[nan '1117' '1119' '1122' '1132' '1133' '1211' '1222' '1409' '1542' '2310'

'2511' '3210' '5001' '8031' '11101' '13206' '14200' '14299' '15100']

PRINCIPAL\_BUSINESS\_ACTIVITY\_AS\_PER\_CIN : 17

['Agriculture & allied' 'Mining and quarrying' 'Manufacturing'

'Electricity gas and water supply' 'Construction'

'Wholesale and retail trade repair of motor vehicles motorcycles and personal and household goods'

'Hotels and restaurants' 'Transport storage and communications'

'Financial intermediation' 'Real estate renting and business activities'

'Health and social work'

'Other community social and personal service activities'

'Extraterritorial organizations and bodies' 'Unclassified'

'Activities of private households as employers and undifferentiated production activities of private households'

'Public administration and defence compulsory social security'

'Education']

REGISTRAR\_OF\_COMPANIES : 5

['ROC\xa0DELHI' 'ROC\xa0COIMBATORE' 'ROC\xa0CHENNAI' nan

'ROC\xa0HYDERABAD']

REG\_YEAR : 131

[1961. nan 1982. 1995. 1996. 1997. 1998. 2000. 1999. 2001. 2005. 2002.

2003. 2004. 2006. 2007. 2008. 2009. 2010. 2011.]

REG\_MONTH : 13

[ 1. nan 5. 11. 4. 10. 7. 2. 6. 8. 3. 12. 9.]

df["COMPANY\_CLASS"] = df["COMPANY\_CLASS"].apply(lambda x:"Solo" if x == 'Private(One Person Company)' else x) print(df["COMPANY\_CLASS"].unique())

df["REGISTRAR"] = df["REGISTRAR\_OF\_COMPANIES"].apply(lambda x: x.split("ROC\xa0")[-1] if isinstance(x, str) else x) print(df["REGISTRAR"].unique())

df["PRINCIPAL\_BUSINESS"] = df["PRINCIPAL\_BUSINESS\_ACTIVITY\_AS\_PER\_CIN"].apply(lambda x:x.split(" ")[0]) print(df["PRINCIPAL\_BUSINESS"].unique())

def round1(x):

if not math.isnan(x):

return (round(x / 5)) \* 5

else:

return x

def round2(x):

if not math.isnan(x):

return (round(x / 10)) \* 10

else:

return x

def round3(x):

if not math.isnan(x):

return (round(x / 20)) \* 20

else:

return x

df["REG\_YEAR\_5BIN"] = df["REG\_YEAR"].apply(round1)

print(df["REG\_YEAR\_5BIN"].unique())

df["REG\_YEAR\_10BIN"] = df["REG\_YEAR"].apply(round2)

print(df["REG\_YEAR\_10BIN"].unique())

df["REG\_YEAR\_20BIN"] = df["REG\_YEAR"].apply(round3)

print(df["REG\_YEAR\_20BIN"].unique())

[1985. 2015. 2020. 1995. 1970. 2000. 2010. 2005. 1990. 1980. 1945. nan

1975. 1955. 1960. 1920. 1965. 1950. 1940. 1930. 1935. 1900. 1895. 1925.

1915.]

[1980. 2010. 2020. 1990. 2000. 1970. 1940. nan 1960. 1920. 1950. 1930.

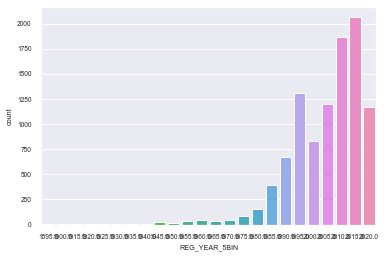
1900. 1910.]

[1980. 2020. 2000. 1960. 1940. nan 1920. 1900.]

**Data visualisation**

sns.countplot(x="REG\_YEAR\_5BIN", data=df)

plt.show()



sns.countplot(x="REG\_YEAR\_10BIN", data=df)

plt.show()

sns.countplot(x="REG\_YEAR\_20BIN", data=df)

plt.show()

