# AI-DRIVEN EXPLORATION AND PREDICTION OF COMPANY REGISTRATION TRENDS WITH REGISTER OF COMPANIES (ROC) IN DATA PRE PROCESSING

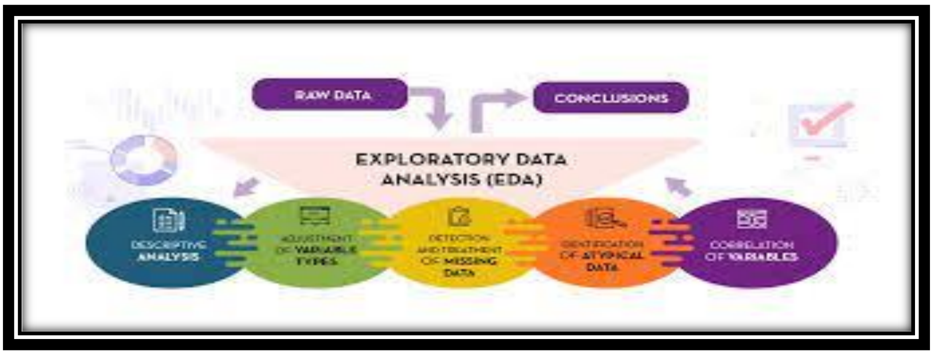
## PHASE-4

## Development Part 2

- In Phase 4 of the "RoC Company Analysis" project, building the AI-driven exploration and prediction system by performing exploratory data analysis (EDA), feature engineering, and predictive modeling. This phase is crucial for extracting valuable insights from the dataset and developing predictive models to anticipate future company registrations. Here's a detailed outline of Phase 4:

### *1. Exploratory Data Analysis (EDA):*

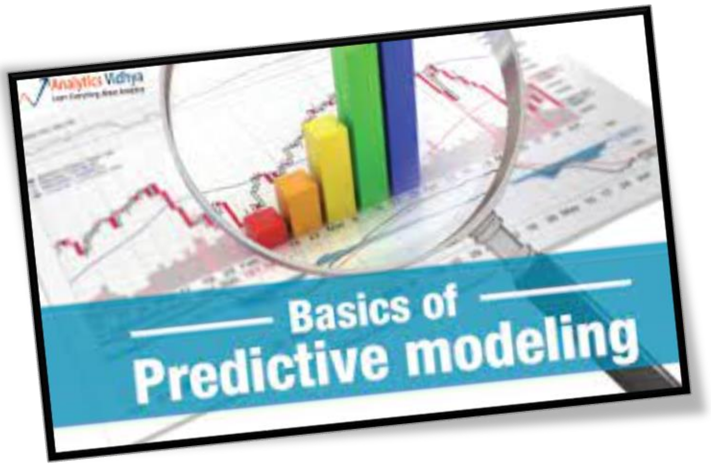
- In this stage, you will delve deeper into the dataset to understand its characteristics and relationships. EDA is essential for gaining insights into the company landscape and identifying hidden patterns.
- Perform statistical analysis to summarize the dataset. This includes calculating summary statistics like mean, median, standard deviation, and more for numerical features.
- **Visualize the data** using various plots, such as histograms, bar charts, scatter plots, and box plots, to reveal data distributions and relationships.
- **Analyze the distribution** of company categories, classes, and registration statuses.
- **Identify any outliers or anomalies i**n the data that might need further attention.


RAW DATA
CONCLUSIONS
EXPLORATORY DATA ANALYSIS (EDA)
DESCRIPTIVE ANALYSIS
ADJUSTMENT OF VARIABLE TYPES
DETECTION AND TREATMENT OF MISSING DATA
IDENTIFICATION OF ATYPICAL DATA
CORRELATION OF VARIABLES

## *2. Feature Engineering:*

- Feature engineering is a critical step in predictive modeling. Create new features or transform existing ones to improve the model's predictive power.
- Consider generating time-related features that capture temporal patterns in the data.
- Encode categorical variables using techniques like one-hot encoding or label encoding, making them suitable for machine learning algorithms.
- Normalize or scale numerical features as necessary to ensure the model's stability and performance.

## *3. Predictive Modeling:*



- Develop and train predictive models using advanced AI algorithms. Since the goal is to anticipate future company registrations, consider time series forecasting methods or ensemble techniques for better accuracy.
- Split the dataset into training and testing sets to assess the model's performance.
- Evaluate and fine-tune the models using appropriate metrics, such as accuracy, precision, recall, F1 score, or RMSE (Root Mean Square Error) for regression tasks.
- Consider using techniques like cross-validation to validate model performance and handle overfitting.

## *4. Model Selection and Deployment:*

- Choose the best-performing model(s) based on evaluation metrics and deploy the model for predictions.
- Ensure that the model is capable of handling new data and making real-time predictions if necessary.
- Save the trained model and its associated parameters for future use.

## *5. Documentation:*

- Document all the code and steps involved in EDA, feature engineering, and predictive modeling.
- Provide clear explanations of the choices made during the process, such as the algorithms selected and the reasons behind those choices.
- Share insights gained from EDA, including any patterns, correlations, or trends discovered in the data.
- Describe the performance of predictive models, discussing the accuracy, precision, and any challenges faced during development.

- **This phase is crucial for turning data into actionable insights and building predictive models that can contribute to informed decision-making for businesses, investors, and policymakers.**

  .

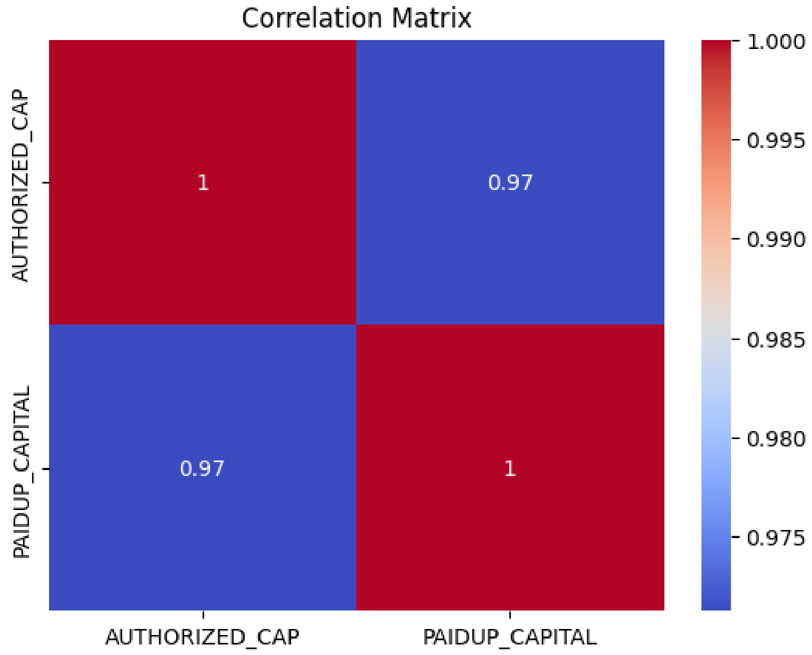
## Program:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report


data=pd.read_csv('//content//sample_data//Data_Gov_Tamil_Nadu.csv',encoding='ISO-8859-1')
```

```
<ipython-input-1-0d79035bc1d0>:9: DtypeWarning: Columns (10) have mixed types. Specify dtype option on import or set low_memory=False.

data=pd.read_csv('//content//sample_data//Data_Gov_Tamil_Nadu.csv',encoding='ISO-8859-1')
```

```
correlation_matrix = data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()
```

```
<ipython-input-2-f70bd4ff3f0a>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  correlation_matrix = data.corr()
```



```
data.fillna(0, inplace=True)
print(data.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(data.head())
```

```
  CORPORATE_IDENTIFICATION_NUMBER  \
0                          F00643
1                          F00721
2                          F00892
3                          F01208
4                          F01218

                                        COMPANY_NAME COMPANY_STATUS  \
0                                   HOCHTIEFF AG,              NAEF
1  SUMITOMO CORPORATION (SUMITOMO SHOJI KAISHA LI...           ACTV
2                       SRILANKAN AIRLINES LIMITED              ACTV
3                             CALTEX INDIA LIMITED              NAEF
4               GE HEALTHCARE BIO-SCIENCES LIMITED              ACTV

  COMPANY_CLASS COMPANY_CATEGORY COMPANY_SUB_CATEGORY DATE_OF_REGISTRATION
\
0             0                0                    0           01-12-1961
1             0                0                    0                    0
2             0                0                    0           01-03-1982
3             0                0                    0                    0
4             0                0                    0                    0

  REGISTERED_STATE  AUTHORIZED_CAP  PAIDUP_CAPITAL INDUSTRIAL_CLASS  \
0       Tamil Nadu             0.0             0.0                0
1       Tamil Nadu             0.0             0.0                0
2       Tamil Nadu             0.0             0.0                0
3       Tamil Nadu             0.0             0.0                0
4       Tamil Nadu             0.0             0.0                0

  PRINCIPAL_BUSINESS_ACTIVITY_AS_PER_CIN  \
0                   Agriculture & allied
1                   Agriculture & allied
2                   Agriculture & allied
3                   Agriculture & allied
4                   Agriculture & allied
```

```
                           REGISTERED_OFFICE_ADDRESS
REGISTRAR_OF_COMPANIES  \
0  AMBLE SIDE, NO.8(OLD NO.30),3RD FLOOR KHADER N...
ROC DELHI
1  FLAT NO. 6, 1st FLOOR, 113/113ARAMA NAICKEN ST...
ROC DELHI
2  SRILANKAN AIRLINES LIMITED, VIJAYA TOWERSNO-4,...
ROC DELHI
3         GOLD CREST 24 55 NORTHUSMAN ROAD T NAGAR
ROC DELHI
4   FF-3 Palani  Centre32 Venkat Naryan Road Nagar
ROC DELHI

               EMAIL_ADDR LATEST_YEAR_ANNUAL_RETURN  \
0                       0                         0
1      shuchi.chug@asa.in                         0
2     shree16us@yahoo.com                         0
3                       0                         0
4  karthick9999@yahoo.com                         0

  LATEST_YEAR_FINANCIAL_STATEMENT
0                               0
1                               0
2                               0
3                               0
4                               0
```

```
X = pd.get_dummies(data[['COMPANY_CLASS', 'COMPANY_CATEGORY',
'INDUSTRIAL_CLASS']], drop_first=True)
X[['AUTHORIZED_CAP', 'PAIDUP_CAPITAL']] = data[['AUTHORIZED_CAP',
'PAIDUP_CAPITAL']]
y = data['COMPANY_STATUS']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)


model = DecisionTreeClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)


print(f"Model Accuracy: {accuracy}")
print(report)
```

```
Model Accuracy: 0.6869262634631317
              precision    recall  f1-score   support

        ACTV       0.72      0.77      0.74     15796
        AMAL       0.12      0.08      0.10       350
        CLLD       0.00      0.00      0.00        24
        CLLP       0.12      0.06      0.08        49
        D455       0.00      0.00      0.00        42
        DISD       0.19      0.11      0.14       174
        LIQD       0.03      0.01      0.02        86
        NAEF       0.17      0.14      0.15       125
        STOF       0.68      0.67      0.67     12731
        ULQD       0.02      0.01      0.01        79
        UPSO       0.05      0.01      0.02       719

    accuracy                           0.69     30175
   macro avg       0.19      0.17      0.18     30175
weighted avg       0.67      0.69      0.68     3017
```