**AI-Driven Exploration and Prediction of Company Registration**

**Trends with Registrar of Companies (RoC)**

**TEAM LEADER**

510521104013:Gopik

**PHASE-5: Documentation & Submission**



**OBJECTIVIE:**

The problem is to perform an AI-driven exploration and predictive analysis on the master details of companies registered with the Registrar of Companies (RoC). The objective is to uncover hidden patterns, gain insights into the company landscape, and forecast future registration

trends.

**Phase 5: Project Documentation & Submission**

**Documentation**

. Clearly outline the problem statement, design thinking process, and the phases of development.

. Describe the dataset used, data preprocessing steps, and AI algorithms applied.

. Explain the insights gained from exploratory data analysis and the performance of predictive models

**Submission**

. Compile all the code files, including the data preprocessing, EDA, and predictive modeling code.

. Provide a well-structured README file that explainshow to run the code and any dependencies.

. Share the submission on platforms like GitHub or personal portfolio for others to access and review.

**Dataset Link: [https://tn.data.gov.in/resource/company-master-data-](https://tn.data.gov.in/resource/company-master-data-tamil-nadu-upto-28th-february-2019) [tamil-nadu-upto-28th-february-2019](https://tn.data.gov.in/resource/company-master-data-tamil-nadu-upto-28th-february-2019)**

**Problem Statement**

The Registrar of Companies (RoC) is a government authority

responsible for maintaining a registry of companies in a given

jurisdiction. Understanding and predicting company registration trends is vital for policymakers, businesses, and investors to make informed decisions. The problem statement is to create an AI-driven system that explores historical data from the RoC to predict future company

registration trends. This system should assist in identifying patterns, potential growth areas, and regulatory changes that may impact

company registrations.

**Design Thinking Process:**

1. **Empathize:**

o Understand the stakeholders'needs, including government agencies, businesses, investors, and researchers.

o Gather requirements and pain points related to company registration data analysis and prediction.

2. **Define:**

o Clearly define the problem: Predicting company registration trends using AI.

o Identify the data sources (RoC records, economic indicators, etc.).

o Set specific goals and metrics (e.g., prediction accuracy, early trend detection).

3. **Ideate:**

o Brainstorm AI-driven solutions and methodologies for trend prediction.

o Explore various AI algorithms and technologies suitable for time-series data analysis.

o Consider data preprocessing and feature engineering approaches.

4. **Prototype:**

o Develop a prototype system:

. Collect and preprocess historical RoC data.

. Implement AI models for trend prediction (e.g., time- series forecasting, regression).

. Create auser interface or API for interaction.

. Test the prototype with a subset of data to assess its performance.

5. **Test:**

o Evaluate the prototype's performance against defined metrics.

o Collect feedback from stakeholders and make necessary adjustments.

o Ensure the system can handle real-time data updates and adapt to changing trends.

6. **Implement:**

o Deploy the AI-driven system to a production environment.

o Connect it to the RoC database or data sources for automatic data updates.

o Develop documentation and training materials for users.

7. **Monitor:**

o Setup continuous monitoring of the system's predictions and accuracy.

o Implement alerting mechanisms for significant deviations or anomalies.

o Regularly update the AI models and data sources.

8. **Iterate:**

o Continuously improve the system based on user feedback and changing requirements.

o Explore advanced AI techniques and data sources to enhance prediction accuracy.

**Phases of Development:**

1. **Data Collection:**

o Acquire historical RoC registration data, economic indicators, and relevant external data.

o Ensure data quality, cleanliness, and completeness.

2. **Data Preprocessing:**

o Clean and normalize the data.

o Perform feature engineering to extract relevant features for prediction.

o Handle missing data and outliers.

3. **Model Development:**

o Choose appropriate AI models for time-series prediction (e.g., ARIMA, LSTM, Prophet).

o Train and validate the models using historical data.

4. **Deployment:**

o Deploy the AI system to a server or cloud platform.

o Setup data pipelines for real-time updates.

o Develop a user-friendly interface for stakeholders.

5. **Monitoring and Maintenance:**

o Implement monitoring for model performance and data quality.

o Regularly retrain models with new data.

o Address issues and update the system as needed.

6. **Feedback and Improvement:**

o Gather user feedback and insights.

o Continuously improve the system's accuracy and usability.

o Explore new technologies and data sources to enhance predictions.

The AI-driven system for exploring and predicting company registration trends with the RoC should bean adaptable, valuable tool for various

stakeholders in making informed decisions and staying ahead of market trends.

**Dataset:**

1. **Historical RoC Data:** This dataset contains historical records of company registrations, including information such as the

company's name, registration date, type, location, and possibly additional details like industry classification codes.

2. **Economic Indicators:** To enhance the prediction model, you can incorporate relevant economic indicators such as GDP growth,

unemployment rates, inflation, and industry-specific data that might influence company registration trends.

**Data Preprocessing Steps:**

Data preprocessing is a critical step to ensure the dataset is suitable for AI-driven exploration and prediction:

1. **Data Cleaning:**

o Handle missing values in the dataset.

o Remove duplicate records.

o Address outliers, if any.

2. **Feature Engineering:**

o Extract relevant features from the RoC data, such as the company's registration date, industry classification, and geographic location.

o Calculate additional features, like the number of registrations per month or quarter.

3. **Data Transformation:**

o Convert categorical variables into numerical representations (e.g., one-hot encoding for industry types).

o Normalize or scale numerical features, especially if different features have different scales.

4. **Time-Series Preparation:**

o If your prediction task involves time-series forecasting,

organize the data as time-series, where the registration date is the time index.

o Perform time-based resampling or aggregation to obtain periodic data, such as monthly or quarterly.

5. **Feature Selection:**

o Use techniques like correlation analysis or feature importance from AI models to select the most relevant features for

prediction.

**AI Algorithms Applied:**

Several AI algorithms can be applied for the exploration and prediction of company registration trends:

1. **Time-Series Forecasting Models:**

o **ARIMA (AutoRegressive Integrated Moving Average):** Suitable for univariate time series forecasting and capturing trends, seasonality, and noise in data.

o **LSTM (Long Short-Term Memory):** Deep learning model for sequential data, useful for capturing complex time

dependencies.

2. **Regression Models:**

o **Linear Regression:** Simple and interpretable for predicting numeric values.

o **Random Forest Regression:** Effective for handling complex feature interactions and providing feature importance scores.

3. **Prophet:**

o Facebook Prophet is a specialized time series forecasting tool designed to handle datasets with strong seasonal patterns and holidays.

4. **XGBoost or LightGBM:**

o Gradient boosting algorithms can be used for regression tasks, providing high accuracy and the ability to handle various data types.

5. **Neural Networks:**

o Custom neural network architectures can be designed for complex, non-linear relationships in the data.

**Exploratory Data Analysis (EDA):**

1. **Temporal Trends:** EDA allows you to identify temporal patterns and trends in company registrations overtime. For example, you can discover if there are seasonal variations, long-term growth, or short-term fluctuations.

2. **Geographical Insights:** EDA can reveal regional variations in company registrations. You may find that certain areas or cities exhibit higher registration rates, indicating regional economic disparities or growth opportunities.

3. **Industry Analysis:** By analyzing the distribution of registrations across different industry categories, you can identify which sectors are experiencing growth or decline. This can be valuable for

investors and policymakers.

4. **Correlation Analysis:** EDA helps identify relationships between company registrations and economic indicators. For example, you may find that company registrations tend to increase during

periods of high GDP growth or decline during economic recessions.

5. **Outliers Detection:** EDA can help identify unusual spikes or

drops in registration counts, which maybe indicative of significant events or anomalies. This information can be crucial for

understanding external factors affecting registrations.

6. **Data Quality Assessment:** EDA allows you to check data quality, detect missing values, and ensure the dataset is clean and suitable for modeling.

**Performance of Predictive Models:**

1. **Prediction Accuracy:** The primary performance metric for

predictive models is the accuracy of the predictions. You can

assess how well the model forecasts future company registration trends by comparing its predictions to actual data.

2. **Root Mean Squared Error (RMSE):** RMSE is a common metric for time-series forecasting models. It quantifies the prediction

errors and helps measure how closely the predicted values align with the actual values.

3. **Mean Absolute Percentage Error (MAPE):** MAPE provides a percentage-based view of the prediction errors and is useful for assessing the relative accuracy of predictions.

4. **Model Comparison:** If you've experimented with multiple AI models, you can compare their performance using metrics like RMSE or MAPE to select the best-performing model.

5. **Cross-Validation:** Use cross-validation techniques to assess how well the model generalizes to unseen data. This helps to avoid

overfitting and ensures the model's robustness.

6. **Feature Importance:** Analyze the importance of different features in the predictive models. Understanding which factors contribute

the most to the predictions can provide insights into what drives company registration trends.

7. **Model Stability:** Overtime, evaluate the stability of the predictive models. Ensure they continue to perform well and adapt to

changing patterns in the registration data.

8. **Early Warning Systems:** If your model is used for trend

detection, assess its ability to provide early warnings of potential changes in registration trends, allowing stakeholders to react

proactively.

Insights gained from EDA can inform the choice of predictive models and help in feature selection and engineering. EDA allows you to better understand the data, identify potential data anomalies, and uncover

relationships that can be leveraged by the AI models. The performance of predictive models, on the other hand, provides a quantitative

assessment of how well the system is at forecasting future company

registration trends. It allows stakeholders to make data-driven decisions and plan accordingly, whether they are government policymakers,

business investors, or researchers studying economic trends.

6. **Ensemble Methods:**

o Combining multiple models (e.g., stacking different

algorithms) can lead to improved prediction accuracy.

The choice of AI algorithms should depend on the specific

characteristics of your dataset and the performance metrics you aim to achieve. It's often a good practice to experiment with multiple

algorithms and select the one that performs best based on validation and testing results.

Once the AI models are developed, they should be trained on the

preprocessed data, validated using appropriate evaluation metrics, and then deployed for ongoing prediction and exploration of company

registration trends with the Registrar of Companies (RoC). Regular model retraining and monitoring should be established to ensure the system's reliability and accuracy.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data S** | **ource;** | | | | | | | |
| F006  43 | HOCHTIEFF AG, | NAE F | N  A | N  A | N  A | ###### ## | Tam il  Nad  u | 0 |
| F007  21 | SUMITOMO  CORPORATION  (SUMITOMO SHOJI KAISHA LIMITED) | ACT V | N  A | N  A | N  A | NA | Tam il  Nad  u | 0 |
| F008  92 | SRILANKAN  AIRLINES LIMITED | ACT V | N  A | N  A | N  A | 1/3/198 2 | Tam il  Nad  u | 0 |
| F012  08 | CALTEX INDIA  LIMITED | NAE F | N  A | N  A | N  A | NA | Tam il  Nad  u | 0 |
| F012  18 | GE HEALTHCARE BIO-SCIENCES  LIMITED | ACT V | N  A | N  A | N  A | NA | Tam il  Nad  u | 0 |
| F012  65 | CAIRN ENERGY INDIA PTY.  LIMITED | NAE F | N  A | N  A | N  A | NA | Tam il  Nad  u | 0 |
| F012  69 | TORIELLI S.R.L | ACT V | N  A | N  A | N  A | 5/9/199 5 | Tam il  Nad  u | 0 |
| F013  11 | HARDY  EXPLORATION & PRODUCTION  (INDIA) INC.. | ACT V | N  A | N  A | N  A | NA | Tam il  Nad  u | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F013  14 | HOCHTIOF  AKTIENGESELLSH ARFF VORM GFBR  HELFMANN | ACT V | N  A | N  A | N  A | ###### ## | Tam il  Nad  u | 0 |
| F014  12 | EPSON  SINGAPORE PVT LTD | ACT V | N  A | N  A | N  A | 25-04-  1997 | Tam il  Nad  u | 0 |
| F014  26 | CARGOLUX  AIRLINES  INTERNATIONAL S A | ACT V | N  A | N  A | N  A | ###### ## | Tam il  Nad  u | 0 |
| F014  68 | CHO HEUNG  ELECTRIC  INDUSTRIAL  COMPANY  LIMITED | NAE F | N  A | N  A | N  A | NA | Tam il  Nad  u | 0 |
| F015  43 | NYCOMED ASIA PACIFIC PTE  LIMITED | ACT V | N  A | N  A | N  A | 27-10-  1998 | Tam il  Nad  u | 0 |
| F015  44 | CHERRINGTON  ASIA LTD | ACT V | N  A | N  A | N  A | 1/5/200 0 | Tam il  Nad  u | 0 |
| F015  63 | SHIMADZU ASIA PACIFIC PTE  LIMITED | NAE F | N  A | N  A | N  A | NA | Tam il  Nad  u | 0 |
|  | | |  | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F01565 | CORK  INTERNATIONAL  PTY LIMITED | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| F01566 | ERBIS ENGG  COMPANY LIMITED | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| F01589 | RALF SCHNEIDER HOLDING GMBH | NAEF | NA | NA | NA | NA | Tamil Nadu | 0 |
| F01593 | MITRAJAYA  TRADING PRIVATE LIMITED | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| F01618 | HEAT AND  CONTROL PTY  LIMITED | ACTV | NA | NA | NA | 13-07-  1999 | Tamil Nadu | 0 |
| F01628 | DIREX SYSTEMS  LIMITED | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| F01641 | NMB-MINEBEA THAI LIMITED | NAEF | NA | NA | NA | NA | Tamil Nadu | 0 |
| F01643 | ARROW  INTERNATIONAL  INC | ACTV | NA | NA | NA | ######## | Tamil Nadu | 0 |
| F01694 | GAMBRO CHINA LTD | ACTV | NA | NA | NA | 14-06-  2000 | Tamil Nadu | 0 |
|  |  |  |  |  |  |  |  |  |
| OBARA  F01703 CORPORATION | | NAEF | NA | NA | NA | 17-07-  2000 | Tamil Nadu | 0 |
| F01752 | CIPTA WAWASON  MAJU ENGINEERING SDM BHD | ACTV | NA | NA | NA | 24-01-  2001 | Tamil Nadu | 0 |
| F01753 | AUCHAN  INTERNATIONAL  S.A. | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| TOSHIBA PLANT  F01767 SYSTEMS AND | | NAEF | NA | NA | NA | 8/3/2001 | Tamil Nadu | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SERVICES  CORPORATION | |  |  |  |  |  |  |  |
| YAMAZEN  F01768 CORPORATION | | NAEF | NA | NA | NA | NA | Tamil Nadu | 0 |
| F01770 | OWL  INTERNATIONAL  PTE LTD | ACTV | NA | NA | NA | 22-03-  2001 | Tamil Nadu | 0 |
| F01826 | LEXMARK  INTERNATIONAL (SINGAPORE) PTE LIMITED | ACTV | NA | NA | NA | 16-08-  2001 | Tamil Nadu | 0 |
| FLUID ENERGY F01830 CONTROLS INC. | | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| WATCH GUARD  F01861 TECHNOLOGIES INC | | ACTV | NA | NA | NA | 21-11-  2001 | Tamil Nadu | 0 |
| SINAR JERUIH SDN F01878 BHD | | ACTV | NA | NA | NA | 24-12-  2001 | Tamil Nadu | 0 |
| F01918 | SIPLEC  INTERNATIONAL  LIMITED | ACTV | NA | NA | NA | 23-09-  1995 | Tamil Nadu | 0 |
| F01935 | INTELSAT GLOBAL SERVICES  CORPORATION | ACTV | NA | NA | NA | 20-05-  2005 | Tamil Nadu | 0 |
| PGS GEOPHYSICAL F01940 A.S | | ACTV | NA | NA | NA | 27-05-  2002 | Tamil Nadu | 0 |
| F01987 | SEVERN GLOCON LIMITED | ACTV | NA | NA | NA | 29-08-  2002 | Tamil Nadu | 0 |
| F02028 | LAGERWEY  WINDTURBINE B V | ACTV | NA | NA | NA | 24-10-  2002 | Tamil Nadu | 0 |
| F02061 | SOCAM  MANAGEMENT  SERVICES  SINGAPORE  PTELIMITED | NAEF | NA | NA | NA | NA | Tamil Nadu | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F02098 | JAN DE NUL NV | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| F02104 | BUCKMAN  LABORATORIES  (ASIA) PTE. LIMITED | ACTV | NA | NA | NA | 5/2/2003 | Tamil Nadu | 0 |
| F02110 | ZWICK ASIA PTE  LIMITED | ACTV | NA | NA | NA | 13-02-  2002 | Tamil Nadu | 0 |
| F02122 | INVE THAILAND  LIMITED | NAEF | NA | NA | NA | NA | Tamil Nadu | 0 |
| F02126 | SUNLEY FASHIONS FAR EAST LIMITED | ACTV | NA | NA | NA | ######## | Tamil Nadu | 0 |
| F02143 | ROTHE ERDE GMBH | NAEF | NA | NA | NA | NA | Tamil Nadu | 0 |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| F02157 | RANGASWAMY AND ASSOCIATES INC | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| F02189 | EASTMAN FILMS INC | ACTV | NA | NA | NA | 18-08-  2003 | Tamil Nadu | 0 |
| F02222 | XAMBALA  INCORPORATED | NAEF | NA | NA | NA | NA | Tamil Nadu | 0 |
| F02235 | DAINTEE LIMITED | ACTV | NA | NA | NA | ######## | Tamil Nadu | 0 |
| F02253 | COLUMBIA  SPORTSWEAR  COMPANY | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| F02261 | KISTLER  INSTRUMENTS PTE LIMITED | NAEF | NA | NA | NA | NA | Tamil Nadu | 0 |
| F02262 | AJINOMOTO CO INC | NAEF | NA | NA | NA | 21-01-  2004 | Tamil Nadu | 0 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F02297 | DANKOTUWA  PROCELAIN LIMITED | ACTV | NA | NA | NA | 15-04-  2004 | Tamil Nadu | 0 |
| F02337 | PUNCAK NAGA  HOLDINGS BERHAD | ACTV | NA | NA | NA | 26-07-  2004 | Tamil Nadu | 0 |
| F02339 | SIGMA  CORPORATION | NAEF | NA | NA | NA | NA | Tamil Nadu | 0 |
| F02372 | CARGO  COMMUNITY  NETWORK PTE LTD | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| F02378 | HETTIGODA  DISTRIBUTORS  PRIVATE LIMITED | ACTV | NA | NA | NA | 17-09-  2004 | Tamil Nadu | 0 |
| F02394 | PROPLUS SYSTEMS INC | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |
| F02418 | DEUTSCHE  WOOLWORTH  SOURCING HK  LIMITED | ACTV | NA | NA | NA | NA | Tamil Nadu | 0 |

**2.Data Preprocessing:**

Cleaning and preprocessing data is a crucial step in the data preparation process before you can use it for machine learning or analysis. Below

are the steps you can follow to clean and preprocess your data, including

handling missing values and converting categorical features into numerical representations.

**1. Import Libraries**

Start by importing the necessary Python libraries for data manipulation and preprocessing, such as Pandas, NumPy, and Scikit-Learn.

python

import pandas aspd

import numpy asnp

from sklearn.preprocessing import LabelEncoder, OneHotEncoder from sklearn.impute import SimpleImputer

**2. Load Your Dataset** Load your dataset into a Pandas DataFrame.

Replace 'your\_data.csv' with the actual file path or URL of your dataset.

python

data = pd.read\_csv('your\_data.csv')

**3. Handling Missing Values** Deal with missing values in your dataset. Depending on the nature of the data, you can choose one of the

following methods:

. **Imputation with Mean/Median/Mode**: Fill missing values with the mean, median, or mode of the respective column.

python

imputer = SimpleImputer(strategy='mean') # You can also use 'median' or 'most\_frequent'

data['column\_name'] = imputer.fit\_transform(data[['column\_name']])

. **Dropping Rows**: Remove rows with missing values if the number of missing values is small and doesn't significantly affect your

dataset.

python

data.dropna(inplace=True)

**4. Handling Categorical Features**

If your dataset contains categorical features, you need to convert them into numerical representations. This can be done in several ways:

. **Label Encoding**: Use label encoding to convert categorical

variables into ordinal integers. This is suitable when there is an ordinal relationship between categories.

python

label\_encoder = LabelEncoder()

data['categorical\_column'] =

label\_encoder.fit\_transform(data['categorical\_column'])

. **One-Hot Encoding**: Use one-hot encoding to convert categorical variables into binary columns. Each category becomes a new

binary column with 0s and 1s.

python

one\_hot\_encoder = OneHotEncoder()

encoded\_categories =

one\_hot\_encoder.fit\_transform(data[['categorical\_column']]).toarray() encoded\_df = pd.DataFrame(encoded\_categories,

columns=one\_hot\_encoder.get\_feature\_names(['categorical\_column'])) data = pd.concat([data, encoded\_df], axis=1)

data.drop(['categorical\_column'], axis=1, inplace=True)

**5. Standardization or Normalization (if necessary)**

Depending on your machine learning algorithm, you might want to

standardize or normalize your numerical features to have a consistent scale. You can use techniques like Min-Max scaling or StandardScaler from Scikit-Learn.

python

from sklearn.preprocessing import StandardScaler, MinMaxScaler

scaler = StandardScaler() # or MinMaxScaler

data[['numerical\_column1', 'numerical\_column2']] =

scaler.fit\_transform(data[['numerical\_column1', 'numerical\_column2']])

**6. Save Processed Data (Optional)**

If you want to save your cleaned and preprocessed data for future use, you can use the to\_csv method in Pandas or other appropriate file

formats.

python

data.to\_csv('preprocessed\_data.csv', index=False)

By following these steps, you can clean and preprocess your data, handle missing values, and convert categorical features into numerical

representations suitable for machine learning or analysis. Make sure to

customize these steps according to your specific dataset and requirements.

**3.Exploratory Data Analysis:**

Exploratory Data Analysis (EDA) is a crucial step in understanding your

data and extracting valuable insights from it. In this example, we'll assume you have a dataset containing information about registered companies. Here'show you can perform EDA to understand the

distribution, relationships, and unique characteristics of these companies:

**1. Import Libraries** Start by importing the necessary Python libraries for data analysis and visualization.

python

import pandas aspd

import numpy asnp

import matplotlib.pyplot asplt

import seaborn as sns

**2. Load Your Dataset** Load your dataset into a Pandas DataFrame if you haven't already (you can reuse the data DataFrame from the

previous example).

python

data = pd.read\_csv('your\_data.csv')

**3. Basic Data Exploration**

. **Preview Data**: Use data.head() to display the first few rows of your dataset to get an initial sense of the data's structure.

python

print(data.head())

. **Summary Statistics**: Get summary statistics for numerical columns to understand central tendencies and spreads.

python

print(data.describe())

**4. Data Visualization**

. **Histograms**: Create histograms to visualize the distribution of numerical variables.

python

data['numerical\_column'].plot(kind='hist', bins=20, edgecolor='k') plt.xlabel('Numerical Column')

plt.ylabel('Frequency')

plt.title('Histogram of Numerical Column')

plt.show()

. **Box Plots**: Use box plots to identify outliers and understand the distribution of numerical variables.

python

sns.boxplot(x='categorical\_column', y='numerical\_column', data=data) plt.xlabel('Categorical Column')

plt.ylabel('Numerical Column')

plt.title('Box Plot of Numerical Column by Category')

plt.xticks(rotation=90)

plt.show()

. **Count Plots**: Create count plots to visualize the distribution of categorical variables.

python

sns.countplot(x='categorical\_column', data=data)

plt.xlabel('Categorical Column')

plt.ylabel('Count')

plt.title('Count Plot of Categorical Column')

plt.xticks(rotation=90)

plt.show()

**5. Relationships and Correlations**

. **Correlation Matrix**: Compute and visualize the correlation between numerical variables.

python

correlation\_matrix = data.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)

plt.title('Correlation Matrix')

plt.show()

. **Pairplots**: Create pairplots to visualize pairwise relationships between numerical variables.

python

sns.pairplot(data, hue='categorical\_column')

plt.suptitle('Pairplot of Numerical Variables')

plt.show()

**6. Unique Characteristics**

. **Unique Values**: Explore the unique values in categorical columns to identify unique characteristics.

python

unique\_values = data['categorical\_column'].unique()

print("Unique Values in Categorical Column:", unique\_values)

. **Value Counts**: Get the count of each unique value in a categorical column.

python

value\_counts = data['categorical\_column'].value\_counts()

print("Value Counts:\n", value\_counts)

These are some common EDA techniques to get a better understanding of your data. You can customize and expand your analysis based on the specific questions you want to answer and the characteristics of your

**4.Feature engineering:**

Feature engineering involves creating new features or transforming

existing ones to improve the performance of predictive models. The goal is to provide the model with more relevant and informative input data.

Here are some techniques and examples for feature engineering:

**1. Encoding Categorical Variables:**

. We've discussed this in the data preprocessing section. You can

use techniques like one-hot encoding or label encoding to convert categorical variables into numerical representations.

**2. Date and Time Features:**

. Extract meaningful information from date and time variables such as year, month, day, day of the week, or time of day. These can be useful in time-series analysis or when time-related patterns matter.

python

data['year'] = data['date'].dt.year

data['month'] = data['date'].dt.month

data['day\_of\_week'] = data['date'].dt.dayofweek

**3. Aggregation and Summary Statistics:**

. Create new features by aggregating or summarizing existing ones. For example, calculate the mean, sum, or standard deviation of

numerical variables for each category in a categorical column.

python

# Calculate the mean of a numerical column for each category in a categorical column

mean\_by\_category =

data.groupby('categorical\_column')['numerical\_column'].mean() data['mean\_numerical\_by\_category'] =

data['categorical\_column'].map(mean\_by\_category)

**4. Interaction Features:**

. Create new features by combining existing ones to capture

interactions or relationships between them. This can be useful in cases where the interaction has predictive power.

python

data['interaction\_feature'] = data['feature1'] \* data['feature2']

**5. Polynomial Features:**

. Create polynomial features to capture non-linear relationships in the data. This is particularly useful in polynomial regression or when you suspect that higher-order terms are significant.

python

from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree=2)

X\_poly = poly

**5.Predictive Modelling:**

To develop predictive models for future company registrations, you can follow these steps:

**\*\*1. Data Preparation:\*\***

**- Ensure your dataset is** cleaned, preprocessed, and contains the relevant features as discussed earlier.

- Split your data into training and testing sets to evaluate the model's performance.

```python

from sklearn.model\_selection import train\_test\_split

X = data.drop('target\_variable', axis=1)

y = data['target\_variable']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

```

2. **Model Selection:\*\***

- Choose appropriate machine learning algorithms based on the nature of your problem. Common choices for predictive modeling include:

- \*\*Linear Regression\*\*: For regression tasks when the target variable is continuous.

- \*\*Logistic Regression\*\*: For binary classification tasks.

- \*\*Random Forest\*\*, \*\*Gradient Boosting\*\*, \*\*XGBoost\*\*: For both regression and classification tasks, and they often perform well.

- \*\*Neural Networks\*\*: For complex problems with large datasets.

- \*\*Support Vector Machines (SVM)\*\*: For classification and

regression tasks, especially when dealing with high-dimensional data.

\*\*3. Model Training:\*\*

- Train your chosen machine learning models using the training data.

```python

from sklearn.ensemble import RandomForestClassifier # Replace with the appropriate model

model = RandomForestClassifier() # Initialize the model

model.fit(X\_train, y\_train) # Train the model

```

\*\*4. Model Evaluation:\*\*

- Assess the model's performance using appropriate evaluation metrics. For classification, common metrics include accuracy, precision, recall,

F1-score, and ROC-AUC. For regression, you can use metrics like mean squared error (MSE), R-squared, and mean absolute error (MAE).

```python

from sklearn.metrics import accuracy\_score, classification\_report, mean\_squared\_error

# For classification

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

report = classification\_report(y\_test, y\_pred)

# For regression

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

```

\*\*5. Hyperparameter Tuning:\*\*

- Optimize your model's hyperparameters to improve its performance. You can use techniques like Grid Search or Random Search.

```python

from sklearn.model\_selection import GridSearchCV

param\_grid = {'n\_estimators': [100, 200, 300], 'max\_depth': [None, 10, 20]}

grid\_search = GridSearchCV(RandomForestClassifier(), param\_grid,

grid\_search.fit(X\_train, y\_train)

best\_params = grid\_search.best\_params\_

**6.Model evaluation:**

Model evaluation is a crucial step in assessing the performance of your predictive models. The choice of evaluation metrics depends on the

nature of the problem you are trying to solve (classification, regression, etc.). Below, I'll provide examples of how to evaluate predictive models using common metrics for classification and regression tasks:

**Classification Metrics:**

1. **Accuracy:** It measures the proportion of correctly predicted instances out of the total instances.

python

from sklearn.metrics import accuracy\_score

y\_true = [0, 1, 1, 0, 1]

y\_pred = [0, 1, 0, 0, 1]

accuracy = accuracy\_score(y\_true, y\_pred)

print("Accuracy:", accuracy)

2. **Precision:** It measures the proportion of true positive predictions among all positive predictions.

python

from sklearn.metrics import precision\_score

precision = precision\_score(y\_true, y\_pred)

print("Precision:", precision)

3. **Recall (Sensitivity or True Positive Rate):** It measures the

proportion of true positives correctly predicted among all actual positives.

python

from sklearn.metrics import recall\_score

recall = recall\_score(y\_true, y\_pred)

print("Recall:", recall)

4. **F1-Score:** It is the harmonic mean of precision and recall and is useful when you want to balance precision and recall.

python

from sklearn.metrics import f1\_score

f1 = f1\_score(y\_true, y\_pred)

print("F1-Score:", f1)

5. **Confusion Matrix:** It provides a detailed breakdown of the

model's predictions, including true positives, true negatives, false positives, and false negatives.

python

from sklearn.metrics import confusion\_matrix

conf\_matrix = confusion\_matrix(y\_true, y\_pred)

print("Confusion Matrix:\n", conf\_matrix)

6. **Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC):** Useful for binary classification

problems with a probability score.

python

from sklearn.metrics import roc\_curve, roc\_auc\_score

y\_probs = model.predict\_proba(X\_test)[:, 1]

fpr, tpr, thresholds = roc\_curve(y\_true, y\_probs)

roc\_auc = roc\_auc\_score(y\_true, y\_probs)

# Plot ROC Curve

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, label='ROC curve (area = {:.2f})'.format(roc\_auc))

plt.plot([0, 1], [0, 1], 'k--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic (ROC)')

plt.legend(loc='lower right')

plt.show()

**Regression Metrics:**

1. **Mean Absolute Error (MAE):** It measures the average absolute difference between predicted and actual values.

python

from sklearn.metrics import mean\_absolute\_error

y\_true = [3.0, 4.5, 2.0, 5.1, 6.3]

y\_pred = [2.8, 4.2, 2.2, 5.0, 6.0]

mae = mean\_absolute\_error(y\_true, y\_pred)

print("MAE:", mae)

2. **Mean Squared Error (MSE):** It measures the average of the squared differences between predicted and actual values.

python

from sklearn.metrics import mean\_squared\_error

mse = mean\_squared\_error(y\_true, y\_pred)

print("MSE:", mse)

3. **Root Mean Squared Error (RMSE):** It is the square root of MSE and provides the error in the same units as the target variable.

python

import numpy asnp

rmse = np.sqrt(mse)

print("RMSE:", rmse)

4. **R-squared (R2):** It measures the proportion of the variance in the dependent variable that is predictable from the independent

variables.

python

from sklearn.metrics import r2\_score

r2 = r2\_score(y\_true, y\_pred)

print("R-squared:", r2)

When evaluating predictive models, choose the evaluation metrics that are most relevant to your specific problem and consider the trade-offs

between them. It's often a good practice to use a combination of metrics to get a comprehensive view of the model's performance.