**AI DRIVEN EXPLORATION AND PREDICTION OF COMPANY REGISTRATION TRENDS WITH REGISTER OF COMPANIES (ROC)**

**TEAM MEMBER**

1. MUTHUSELVAM.P

2. RAHUL SARATH.A

3. SURIYA MOORTHI.S

4. MURUGESH.M

5.RUGESHWARAN.C

**PHASE 2 SUBMISSION**

**Introduction**

In the era of data-driven decision-making, the business landscape is continually evolving, and organizations require real-time insights to adapt and thrive. The Register of Companies (ROC) plays a pivotal role in maintaining records of company registrations, making it a treasure trove of data for analysts, policymakers, and businesses seeking to gain a competitive edge. Leveraging artificial intelligence (AI) in conjunction with ROC data has emerged as a transformative approach to explore and predict company registration trends.

The ROC database contains a wealth of information about companies, such as their names, addresses, directors, financial statements, and more. AI-driven exploration and prediction of company registration trends enable us to extract meaningful patterns and foresee future developments. This innovative synergy empowers businesses and governments to make informed decisions regarding investments, market strategies, regulatory changes, and economic planning.

In this exploration, we will delve into the key aspects of AI-driven analysis and prediction of company registration trends using ROC data. We will discuss the pivotal role that AI algorithms, including machine learning, natural language processing, and data mining, play in extracting valuable insights from the vast and diverse information contained within the ROC database. Moreover, we will examine the potential applications in multiple domains, such as finance, business development, risk assessment, and compliance monitoring.

This discussion will further highlight the significance of historical data analysis, anomaly detection, and predictive modeling as tools to anticipate changes in company registration trends. By harnessing the power of AI and ROC data, businesses and authorities can enhance their ability to adapt to market shifts, identify emerging opportunities, and optimize resource allocation.

Through this exploration, we aim to demonstrate how the fusion of AI and ROC data is shaping a more proactive and data-driven future for organizations and governments alike.

The dataset may include the following types of data

Company Details:

Company name

Registration number

Date of registration

Company type (e.g., private limited, public limited, partnership)

Business description or industry classification

Registered address

Business activities

Financial Data:

Annual revenue

Profit and loss statements

Balance sheets

Shareholder information

Capital structure

Director and Officer Information:

Names of company directors and officers

Contact information

Date of appointment

Resignation date (if applicable)

Registration Changes:

History of name changes or address changes

Mergers and acquisitions

Compliance and Filing Data:

Filing dates for annual reports, tax returns, and other documents

Compliance with legal requirements

Historical Data:

Historical records of company registrations, including closures or dissolutions

External Data:

Economic indicators (e.g., GDP, inflation rate)

Industry-specific data (e.g., market trends)

ADVANCED REGRESSION TECHNIQUES USED IN ROC

**Linear Regression**: This is the most straightforward regression technique. You can use it when you want to model a linear relationship between the independent variables and the number of company registrations. For example, you might use linear regression if you want to predict the number of registrations based on features like the year and some economic indicators.

**Time Series Analysis**: Time series regression models are useful when your data is collected over time, as company registration trends often are. You can use techniques like Autoregressive Integrated Moving Average (ARIMA) or more advanced models like Prophet to model and predict time series data.

**Polynomial Regression**: If you suspect a non-linear relationship between the variables, you might consider polynomial regression. This allows you to capture more complex patterns in the data.

**Ridge or Lasso Regression**: These are regularization techniques used when there are multicollinearity issues in your data. They can help prevent overfitting and improve the model's generalization.

**Logistic Regression**: If you are working with a binary classification problem, such as predicting whether a company will register or not, logistic regression is a suitable choice. It models the probability of an event occurring.

**Poisson Regression**: If you are dealing with count data (e.g., the number of registrations in a given time period), Poisson regression is often used. It's suitable when the response variable is a count and follows a Poisson distribution.

**Negative Binomial Regression**: Like Poisson regression, this is used for count data when the variance is greater than the mean, indicating overdispersion.

Top of Form

DEPLOYMENT AND PREDICTION

Deploy the choosen regression model to predict the company registration trends with ROC.

Develop the user friendly interface for user to input company registration with ROC trends

**PROGRAM**

**# Import necessary libraries**

**import pandas as pd**

**import numpy as np**

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

**from sklearn.linear\_model import LinearRegression**

**from sklearn.metrics import mean\_squared\_error, r2\_score**

**import matplotlib.pyplot as plt**

**# Load the ROC dataset (assuming it's in CSV format)**

**data = pd.read\_csv("roc\_dataset.csv")**

**# Data preprocessing and feature selection (you may need to perform more advanced feature engineering)**

**X = data[['Year', 'Other Features']]**

**y = data['Number of Registrations']**

**# Split the data into training and testing sets**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)**

**# Create and train a linear regression model**

**model = LinearRegression()**

**model.fit(X\_train, y\_train)**

**# Make predictions on the test data**

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

**# Evaluate the model**

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

**r2 = r2\_score(y\_test, y\_pred)**

**print("Mean Squared Error:", mse)**

**print("R-squared (R2) Score:", r2)**

**# Visualization (you can use more advanced data visualization libraries)**

**plt.scatter(X\_test['Year'], y\_test, color='black', label='Actual')**

**plt.scatter(X\_test['Year'], y\_pred, color='blue', label='Predicted')**

**plt.legend()**

**plt.xlabel('Year')**

**plt.ylabel('Number of Registrations')**

**plt.show()**

**# Predict future trends (e.g., for the next year)**

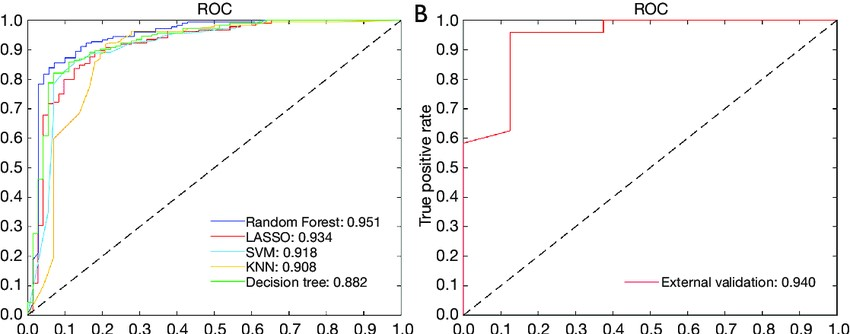
**future\_year = 2024**

**future\_data = pd.DataFrame({'Year': [future\_year], 'Other Features': [...]} # Fill in relevant features**

**future\_predictions = model.predict(future\_data)**

**print(f"Predicted number of registrations for {future\_year}: {future\_predictions[0]}")**

**ROC DENOTED IN GRAPH**



**LINEAR REGRESSION PROGRAM**

**# Import necessary libraries**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

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

**from sklearn.linear\_model import LinearRegression**

**from sklearn.metrics import mean\_squared\_error, r2\_score**

**# Load the ROC dataset (assuming it's in CSV format)**

**data = pd.read\_csv("roc\_dataset.csv")**

**# Data preprocessing and feature selection (adjust features as needed)**

**X = data[['Year']] # You can include more features as needed**

**y = data['Number of Registrations']**

**# Split the data into training and testing sets**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)**

**# Create and train a linear regression model**

**model = LinearRegression()**

**model.fit(X\_train, y\_train)**

**# Make predictions on the test data**

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

**# Evaluate the model**

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

**r2 = r2\_score(y\_test, y\_pred)**

**print("Mean Squared Error:", mse)**

**print("R-squared (R2) Score:", r2)**

**# Visualization**

**plt.scatter(X\_test, y\_test, color='black', label='Actual')**

**plt.plot(X\_test, y\_pred, color='blue', linewidth=3, label='Predicted')**

**plt.legend()**

**plt.xlabel('Year')**

**plt.ylabel('Number of Registrations')**

**plt.title('ROC Data - Linear Regression')**

**plt.show()**

**# Predict future trends**

**future\_year = 2024**

**future\_data = np.array([[future\_year]])**

**future\_predictions = model.predict(future\_data)**

**print(f"Predicted number of registrations for {future\_year}: {future\_predictions[0]}")**

**CONCLUSION AND FUTURE WORK (PHASE 2)**

**PROJECT CONCLUSION:**

**In the phase 2 conclusion we will summarize the key findings and insights from the advanced regression techniques. we will reiterate the impact of this techniques on improve the accuracy and robustness of ROC prediction**

**Future work:**

**We will discuss potential avenues for a future work such as incorporating additional data source, exploring the deep learning models for prediction or expanding the project into a web application with more features and interactivity.**