SURYA GROUP OF INSTITUTIONS

NAAN MUDHALVAN

IBM - ARTIFICIAL INTELLIGENCE

LATCHUMANAN M

422221104019

TEAM-08

AI-Driven Exploration and Prediction of Company Registration Trends with Registrar of Companies (RoC)

Problem Definition :

I found a project on **GitHub** that might be of interest to you. It's called AI-driven exploration and prediction of company registration trends with the Registrar of Companies (RoC)

Data Source :

There are a few web sources that might help you with your project. One of them is a GitHub repository by jjsuryanehru, who has uploaded a PDF file of his project report on the same topic1. He has used Python to perform data analysis and visualization, as well as machine learning models to predict the number of company registrations in different states and sectors. He has also provided some recommendations based on his findings.

A third source is a Forbes article that talks about unlocking the power of predictive analytics with AI3. The article highlights how AI can enhance the accuracy and timeliness of forecasting by using advanced techniques such as deep learning, natural language processing and computer vision. It also gives some examples of how AI can help businesses improve their decision making and performance. This might be relevant for your project, as you can use AI to improve your predictive models and generate insights from the RoC data.

Ranging from new methods to measure and increase return on investment (ROI) to new uses in the "metaverse," companies that capitalize on these trends can grow their bottom lines and fulfill their broader corporate purpose. If you use them well, they can be part of your foundation for sustainable, long-term growth.

AI will fully converge with data and cloud demanding a new management approach :

it needs to be organized, analyzed and used at scale — which AI can do. For this kind of AI investment to really pay off, it needs to be embedded in application systems that can work 24 hours a day, 365 days a year. These systems, in turn, need cloud-based computing power that can scale up and down to cost-effectively meet ever changing demands. With these imperatives, it's clear why leading companies increasingly invest in and manage data, AI and cloud (DAC) as a unified whole.

The project employs various AI techniques such as natural language processing (NLP), machine learning (ML), deep learning (DL) and computer vision (CV) to process, analyze and visualize the data. Some of the tasks performed by the project are:

- Data cleaning and preprocessing: The project uses NLP and ML to remove duplicates, outliers, errors and missing values from the data. It also performs feature engineering and normalization to make the data suitable for further analysis.
- Data exploration and visualization: The project uses CV and ML to create interactive dashboards and charts that show the trends and patterns of company registrations across different dimensions such as time, sector, industry, region, state, city, etc. It also uses NLP and ML to generate natural language summaries and insights from the data.
- Data modeling and prediction: The project uses DL and ML to build predictive models that can forecast the number of company registrations for a given period, sector, industry or region. It also uses NLP and ML to generate natural language explanations for the predictions and recommendations for action.

Sample Program :

Creating a complete AI-driven program in Python is a complex task, but I can provide you with a simplified example of an AI-driven program that uses a basic machine learning model for a simple task. In this example, we'll create a program that uses a machine learning model to predict whether a given fruit is an apple or an orange based on its weight and color.

First, you'll need to install the necessary libraries if you haven't already. You can use pip for this:

```
pip install scikit-learn
```

The Python program:

```
# Import necessary libraries

from sklearn import tree


# Define a dataset of fruit features (weight in grams and color)

# The label '0' represents 'apple' and '1' represents 'orange'
```

```
features = [[140, 1], [130, 1], [150, 0], [170, 0]]
labels = [0, 0, 1, 1]

# Create a decision tree classifier
clf = tree.DecisionTreeClassifier()

# Train the classifier on the dataset
clf = clf.fit(features, labels)

# Input from the user
weight = int(input("Enter the weight of the fruit (grams): "))
color = int(input("Enter the color (0 for red, 1 for orange): "))

# Make a prediction using the trained model
prediction = clf.predict([[weight, color]])

# Map the prediction back to the fruit name
fruit_name = "apple" if prediction[0] == 0 else "orange"
```

```
# Display the prediction

print(f"The fruit is a {fruit_name}")
```