Hardware Requirements:

Computing Power:

- High-performance multi-core CPUs for data processing and modeling.
- Depending on the project's complexity, consider GPUs (Graphics Processing Units) for deep learning tasks.

Memory (RAM):

- A minimum of 16 GB of RAM for handling medium-sized datasets. More RAM may be required for larger datasets or complex modeling tasks.

Storage:

- Fast SSD storage for efficient data access and modeling.
- Sufficient storage capacity to store datasets, intermediate results, and trained models.

Network:

- Reliable internet connectivity for data access, library updates, and potential cloud services usage.

Software Requirements:

Operating System:

- Preferred operating systems include Linux (e.g., Ubuntu), macOS, or Windows.
- Linux is often preferred for its compatibility with data science libraries and tools.

Python Environment:

- Python 3.x as the programming language.
- Use virtual environments (e.g., **venv**, **conda**) to manage Python packages and dependencies.

Package Management:

- **pip** for installing Python packages.
- **conda** for managing packages and environments if using Anaconda distribution.

Integrated Development Environment (IDE):

- Popular IDEs include Jupyter Notebook, JupyterLab, PyCharm, or Visual Studio Code for coding, debugging, and documentation.

Data Requirements:

Registrar of Companies (RoC) Data:

- Access to historical and up-to-date RoC data, which should include company registration details, dates, types of companies, industries, and geographical locations.

Auxiliary Data:

- Access to relevant auxiliary data, such as economic indicators, demographic information, or industry-specific data that may impact registration trends.

Libraries and Frameworks:

Data Analysis and Machine Learning Libraries:

- Libraries like pandas, numpy, scikit-learn, TensorFlow, and PyTorch for data manipulation, analysis, and machine learning modeling.

Visualization Libraries:

- Libraries such as matplotlib, seaborn, plotly, or bokeh for data visualization.

NLP and Text Processing Libraries (if dealing with text data):

- Libraries like spaCy, nltk, gensim, or fastText for natural language processing and text analysis.

Time Series Analysis Libraries (if dealing with time series data):

- Libraries like statsmodels, Prophet, or pmdarima for time series analysis and forecasting.

Web Frameworks (if developing web-based interfaces or APIs):

- Flask, FastAPI, or Django for creating web-based applications or APIs.

Version Control:

- Git for version control, with a repository hosted on platforms like GitHub, GitLab, or Bitbucket.

Documentation:

- Tools like Sphinx, MkDocs, or Jupyter Notebook for documenting project code and processes.