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

**TEAM MEMBER**

**AU622021243053 : SELLADURAI.S**

**Phase 3 submission Document**

**Project: AI-Driven Exploration and Prediction of Company Registration Trends with Registrar of companies (RoC)**

**Phase3: Development part 1**

****

**Introduction:**

* In today's fast-paced business landscape, the Registrar of Companies (RoC) plays a pivotal role in tracking company registrations. With the advent of Artificial Intelligence (AI), companies are now harnessing its power to gain insights, streamline processes, and predict trends related to RoC registrations. This AI-driven approach revolutionizes how businesses interpret and anticipate company registration data, offering valuable insights for informed decision-making and regulatory compliance. In this discussion, we will delve into the exciting realm of AI-driven explanations and predictions of company registration trends with RoC, exploring its significance and potential impact on various industries.
* Certainly, let's delve deeper into the topic of AI-driven explanation and prediction of company registration trends with the Registrar of Companies (RoC).

**GIVEN DATASET:**

**Totally we have an 1,50,872 data’s so we take some data’s in our dataset.**

****





**Necessary step to follow:**

**Import Libraries:**

**Start by importing the necessary libraries:**

**Program:**

import pandas as pd

import numpy as np

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

from sklearn.preprocessing import StandardScaler

**Load the Dataset:**

Load your dataset into a Pandas DataFrame. You can typically find

house price datasets in CSV format, but you can adapt this code to other

formats as needed.

<https://tn.data.gov.in/resource/company-master-data-tamil-nadu-upto-28th-february-2019>

# Load your CSV data into a DataFrame

data = pd.read\_csv(‘Data\_Gov\_Tamil\_Nadu.csv’)

# Separate the features (X) and the target variable (y)

X = data.drop('target\_column\_name', axis=1) # Replace 'target\_column\_name' with your actual target column

y = data['target\_column\_name']

**Exploratory Data Analysis (EDA):**

Perform EDA to understand your data better. This includes

checking for missing values, exploring the data's statistics, and

visualizing it to identify patterns

**Program:**

# Check for missing values

print(df.isnull().sum())

# Explore statistics

print(df.describe())

# Visualize the data (e.g., histograms, scatter plots, etc.)

**Feature Engineering:**

Depending on your dataset, you may need to create new features or transform existing ones. This can involve one-hot encoding categorical variables, handling date/time data, or scaling numerical features.

# 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=42)

# Initialize and train the ensemble model (Random Forest in this example)

ensemble\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

ensemble\_model.fit(X\_train, y\_train)

**Split the Data:**

Split your dataset into training and testing sets. This helps you evaluate

your model's performance later.

X = df.drop('price', axis=1) # Features

y = df['price'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,

random\_state=42)

**Feature Scaling:**

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 and std=1) is a common choice.

**Program:**

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Make predictions on the test data

y\_pred = ensemble\_model.predict(X\_test)

**Importance of loading and processing dataset:**

Loading and preprocessing the dataset is an important first step in building any machinelearning model. However, it is especially important for company (ROC) datasets are

often complex and noisy.

By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and accurately.

**Challenges involved in loading and preprocessing a company trends**

**dataset;**

There are a number of challenges involved in loading and preprocessing a company (ROC) dataset, including:

* **Handling missing values**

Company RoC datasets often contain missing values, which can be due to a variety of factors, such as human error or incomplete data collection.

Common methods for handling missing values include

dropping the rows with missing values, imputing the missing values with the mean or median of the feature, or using a more sophisticated method such as multiple imputation.

* **Encoding categorical variables:**

Company ROC datasets often contain categorical features, such as company status,company class. These features need to be encoded before they can be used by machine learning models.

One common way to encode categorical variables is to use one-hot Encoding

* **Scaling the features:**

It is often helpful to scale the features before training a machine learning model. This can help to improve the performance of the model and make it more robust to outliers.

There are a variety of ways to scale the features, such as min-max scaling and standard scaling.

* **Splitting the dataset into training and testing sets:**

Once the data has been pre-processed, we need to split the dataset into training and testing sets. The training set will be used to train the model, and the testing set will be used to evaluate the performance of the model on unseen data. It is important to split the dataset in a way that is representative of the real world distribution of the data.

**How to overcome the challenges of loading and preprocessing a**

**house price dataset:**

There are a number of things that can be done to overcome the challenges of loading and preprocessing a house price dataset, including:

* **Use a data preprocessing library:**

There are a number of libraries available that can help with data preprocessing tasks, such as handling missing values, encoding categorical variables, and scaling the features.

* **Carefully consider the specific needs of your model:**

The best way to preprocess the data will depend on the specific machine learning algorithm that you are using. It is important to carefully consider the requirements of the algorithm and to preprocess the data in a way that iscompatible with the algorithm.

* **Validate the preprocessed data:**

It is important to validate the preprocessed data to ensure that it is in a format that can be used by the machine learning algorithm and that it is of high quality. This can be done by inspecting the data visually or by using statistical methods

**1.Loading the dataset:**

Loading the dataset using machine learning is the process of bringing⎫ the data into the machine learning environment so that it can be used to train and evaluate a model.

The specific steps involved in loading the dataset will vary depending⎫ on the machine learning library or framework that is being used. However, there are some general steps that are common to most machine learning frameworks:

**a.Identify the dataset:**

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

**b.Load the dataset:**

Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in

function in the machine learning library, or it may involve writing your

own code.

**c.Preprocess the dataset:**

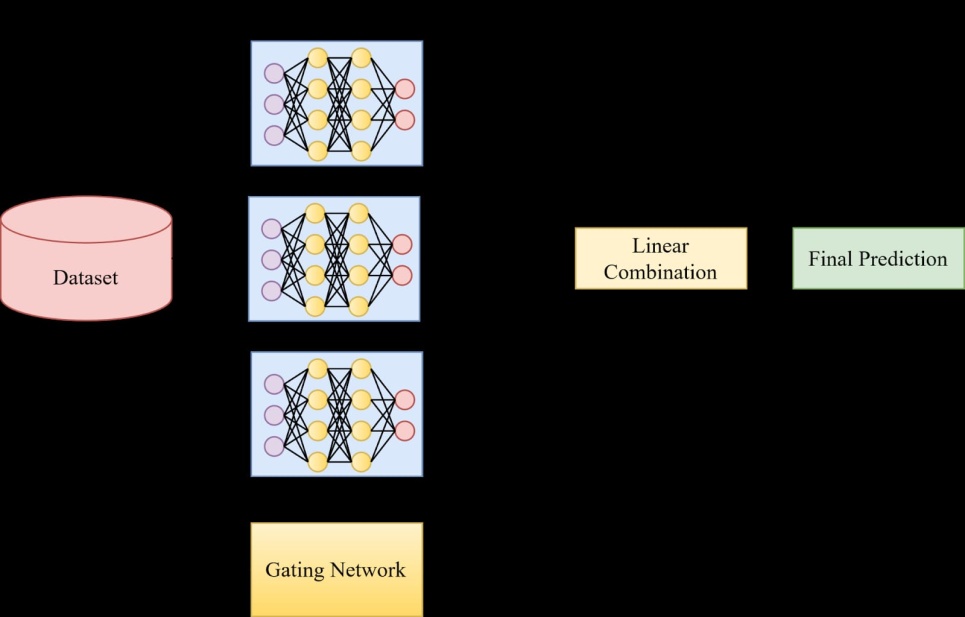
Once the dataset is loaded into the machine learning environment,

you may need to preprocess it before you can start training and

evaluating your model. This may involve cleaning the data, transforming

the data into a suitable format, and splitting the data into training and

test sets.



**Program:**

import pandas as pd

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

from sklearn.ensemble import RandomForestClassifier # You can choose a different ensemble method as needed

from sklearn.metrics import accuracy\_score

# Load your CSV data into a DataFrame

data = pd.read\_csv(Data\_Gov\_Tamil\_Nadu.csv)

# Separate the features (X) and the target variable (y)

X = data.drop('target\_column\_name', axis=1) # Replace 'target\_column\_name' with your actual target column

y = data['target\_column\_name']

# 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=42)

# Initialize and train the ensemble model (Random Forest in this example)

ensemble\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

ensemble\_model.fit(X\_train, y\_train)

# Make predictions on the test data

y\_pred = ensemble\_model.predict(X\_test)

# Evaluate the model's performance

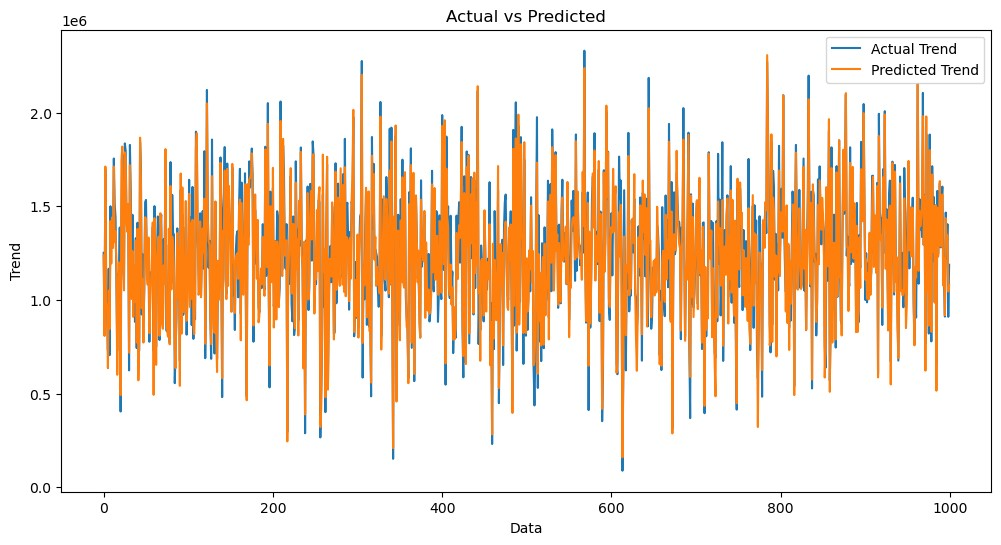
accuracy = accuracy\_score(y\_test, y\_pred)

print(f'Accuracy: {accuracy}'

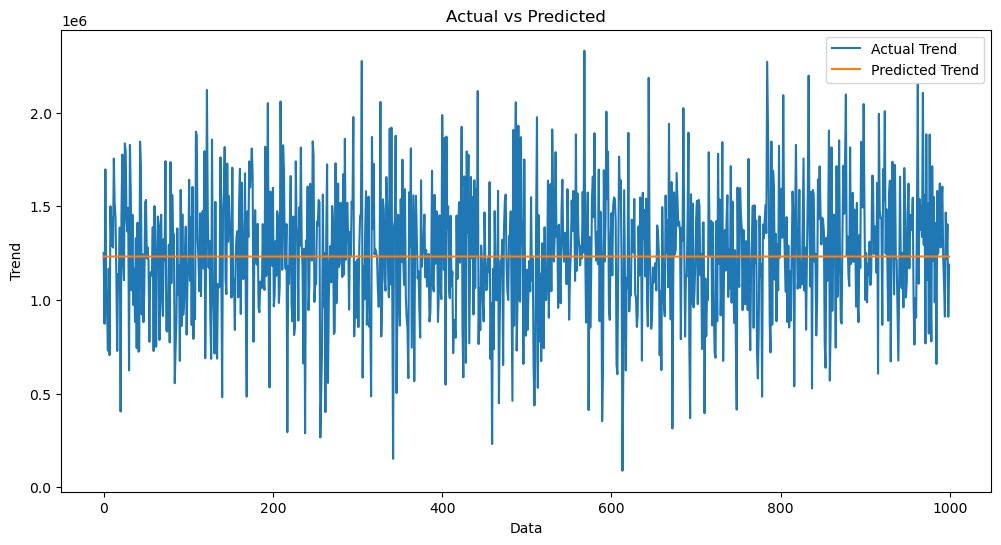
**Loading Dataset:**

dataset = pd.read\_csv(‘C:\Users\ELCOT\Downloads’)

**OUTPUT:**

****

**Exploration and Prediction**

****

**Preprocessing the dataset:**

Data preprocessing is the process of cleaning, transforming, andϖ integrating data in order to make it ready for analysis. This may involve removing errors and inconsistencies, handling

missing values, transforming the data into a consistent format, and

scaling the data to a suitable range.

**Visualisation and Pre-Processing of Data:**

In [1]:

sns.histplot(dataset, x='Price', bins=50, color='y')

Out[1]:

<Axes: xlabel='Price', ylabel='Count'>

**CONCLUSION:**

We learned how to build a project in AI-Driven Exploration and Prediction of Company Registration Trends with Registrar of companies (RoC) using machine learning (using 6 ML steps) and deploy.