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

Team Member

510521104052: Suriya Vignesh M

PHASE-1:DOCUMENTSUBMISSION

OBJECTIVIE:

The problem is to perform an AI-driven exploration and predictiveanalysis on the master details of companies registered with the Registrarof Companies (RoC). The objective is to uncover hidden patterns, gaininsightsintothe company landscape, andforecastfuture registrationtrends.

PHASE-1:ProblemDefinitionand Design Thinking

DataSource:Utilizethedatasetcontaininginformationaboutregistered companies,includingcolumnslikecompany name,status,class, category, registration date, authorized capital, paid-up capital, andmore.

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

1.DataSource;

F00643	HOCHTIEFFAG,	NAEF	NA	NA	NA	########	Tamil Nadu	0
F00721	SUMITOMO CORPORATION(SUMITOM OSHOJIKAISHA LIMITED)	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F00892	SRILANKANAIRLINESLIMITED	ACTV	NA	NA	NA	1/3/1982	Tamil Nadu	0
F01208	CALTEXINDIALIMITED	NAEF	NA	NA	NA	NA	Tamil Nadu	0
F01218	GEHEALTHCAREBIO-SCIENCES LIMITED	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F01265	CAIRNENERGYINDIAPTY. LIMITED	NAEF	NA	NA	NA	NA	Tamil Nadu	0
F01269	TORIELLIS.R.L	ACTV	NA	NA	NA	5/9/1995	Tamil Nadu	0
F01311	HARDYEXPLORATION& PRODUCTION(INDIA)INC..	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F01314	HOCHTIOFAKTIENGESELLSHARFF VORMGFBRHELFMANN	ACTV	NA	NA	NA	########	Tamil Nadu	0
F01412	EPSONSINGAPOREPVTLTD	ACTV	NA	NA	NA	25-04-1997	Tamil Nadu	0
F01426	CARGOLUXAIRLINES INTERNATIONALS A	ACTV	NA	NA	NA	########	Tamil Nadu	0
F01468	CHOHEUNGELECTRIC INDUSTRIALCOMPANYLIMITED	NAEF	NA	NA	NA	NA	Tamil Nadu	0
F01543	NYCOMED ASIAPACIFICPTE LIMITED	ACTV	NA	NA	NA	27-10-1998	Tamil Nadu	0
F01544	CHERRINGTONASIALTD	ACTV	NA	NA	NA	1/5/2000	Tamil Nadu	0
F01563	SHIMADZUASIAPACIFICPTE LIMITED	NAEF	NA	NA	NA	NA	Tamil Nadu	0

F01565	CORKINTERNATIONALPTYLIMITED	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F01566	ERBISENGGCOMPANYLIMITED	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F01589	RALFSCHNEIDERHOLDINGGMBH	NAEF	NA	NA	NA	NA	Tamil Nadu	0
F01593	MITRAJAYATRADINGPRIVATE LIMITED	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F01618	HEATANDCONTROLPTYLIMITED	ACTV	NA	NA	NA	13-07-1999	Tamil Nadu	0
F01628	DIREXSYSTEMSLIMITED	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F01641	NMB-MINEBEATHAILIMITED	NAEF	NA	NA	NA	NA	Tamil Nadu	0

F01643	ARROWINTERNATIONALINC	ACTV	NA	NA	NA	########	Tamil Nadu	0
F01694	GAMBROCHINA LTD	ACTV	NA	NA	NA	14-06-2000	Tamil Nadu	0
F01703	OBARACORPORATION	NAEF	NA	NA	NA	17-07-2000	Tamil Nadu	0
F01752	CIPTAWAWASONMAJU ENGINEERINGSDMBHD	ACTV	NA	NA	NA	24-01-2001	Tamil Nadu	0
F01753	AUCHANINTERNATIONALS.A.	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F01767	TOSHIBAPLANTSYSTEMSAND SERVICESCORPORATION	NAEF	NA	NA	NA	8/3/2001	Tamil Nadu	0
F01768	YAMAZENCORPORATION	NAEF	NA	NA	NA	NA	Tamil Nadu	0
F01770	OWLINTERNATIONALPTELTD	ACTV	NA	NA	NA	22-03-2001	Tamil Nadu	0
F01826	LEXMARKINTERNATIONAL (SINGAPORE)PTELIMITED	ACTV	NA	NA	NA	16-08-2001	Tamil Nadu	0
F01830	FLUIDENERGYCONTROLSINC.	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F01861	WATCHGUARDTECHNOLOGIES INC	ACTV	NA	NA	NA	21-11-2001	Tamil Nadu	0
F01878	SINARJERUIHSDNBHD	ACTV	NA	NA	NA	24-12-2001	Tamil Nadu	0
F01918	SIPLECINTERNATIONALLIMITED	ACTV	NA	NA	NA	23-09-1995	Tamil Nadu	0
F01935	INTELSATGLOBALSERVICES CORPORATION	ACTV	NA	NA	NA	20-05-2005	Tamil Nadu	0
F01940	PGSGEOPHYSICALA.S	ACTV	NA	NA	NA	27-05-2002	Tamil Nadu	0
F01987	SEVERNGLOCONLIMITED	ACTV	NA	NA	NA	29-08-2002	Tamil Nadu	0
F02028	LAGERWEYWINDTURBINEBV	ACTV	NA	NA	NA	24-10-2002	Tamil Nadu	0
F02061	SOCAMMANAGEMENTSERVICES SINGAPOREPTELIMITED	NAEF	NA	NA	NA	NA	Tamil Nadu	0
F02098	JANDENULNV	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F02104	BUCKMANLABORATORIES(ASIA) PTE.LIMITED	ACTV	NA	NA	NA	5/2/2003	Tamil Nadu	0
F02110	ZWICKASIAPTELIMITED	ACTV	NA	NA	NA	13-02-2002	Tamil Nadu	0
F02122	INVETHAILANDLIMITED	NAEF	NA	NA	NA	NA	Tamil Nadu	0

F02126	SUNLEYFASHIONSFAREAST LIMITED	ACTV	NA	NA	NA	########	Tamil Nadu	0
F02143	ROTHEERDEGMBH	NAEF	NA	NA	NA	NA	Tamil Nadu	0
F02157	RANGASWAMYANDASSOCIATES INC	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F02189	EASTMANFILMSINC	ACTV	NA	NA	NA	18-08-2003	Tamil Nadu	0
F02222	XAMBALAINCORPORATED	NAEF	NA	NA	NA	NA	Tamil Nadu	0
F02235	DAINTEELIMITED	ACTV	NA	NA	NA	########	Tamil Nadu	0
F02253	COLUMBIA SPORTSWEARCOMPANY	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F02261	KISTLER INSTRUMENTS PTELIMITED	NAEF	NA	NA	NA	NA	Tamil Nadu	0
F02262	AJINOMOTOCOINC	NAEF	NA	NA	NA	21-01-2004	Tamil Nadu	0
F02297	DANKOTUWAPROCELAINLIMITED	ACTV	NA	NA	NA	15-04-2004	Tamil Nadu	0
F02337	PUNCAK NAGAHOLDINGSBERHAD	ACTV	NA	NA	NA	26-07-2004	Tamil Nadu	0
F02339	SIGMACORPORATION	NAEF	NA	NA	NA	NA	Tamil Nadu	0
F02372	CARGOCOMMUNITYNETWORK PTELTD	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F02378	HETTIGODADISTRIBUTORS PRIVATELIMITED	ACTV	NA	NA	NA	17-09-2004	Tamil Nadu	0
F02394	PROPLUSSYSTEMSINC	ACTV	NA	NA	NA	NA	Tamil Nadu	0
F02418	DEUTSCHEWOOLWORTH SOURCINGHKLIMITED	ACTV	NA	NA	NA	NA	Tamil Nadu	0

2.DataPreprocessing:

Cleaning and preprocessing data is a crucial step in the data preparationprocessbeforeyoucan use itformachine learningor analysis. Belowarethesteps youcanfollowtocleanandpreprocessyourdata,includinghandling missing values and converting categorical features intonumericalrepresentations.

1. **ImportLibraries**
   StartbyimportingthenecessaryPythonlibrariesfordataman ipulation and preprocessing, such as Pandas, NumPy, andScikit-Learn.

```
python
importpandasaspdi
mportnumpyasnp
fromsklearn.preprocessingimportLabelEncoder,OneHotEncoderfr
omsklearn.imputeimportSimpleImputer
```

2. Load Your Dataset Load your dataset into a Pandas DataFrame.Replace'your_data.csv'withtheactualfilepathorURL ofyourdataset.

```
python
data=pd.read_csv('your_data.csv')
```

3. HandlingMissingValues
Dealwithmissingvaluesinyourdataset.Depending on the nature of the data, you can choose one of thefollowingmethods:

- **ImputationwithMean/Median/Mode**:Fill missingvalueswiththe mean,median,ormodeof therespectivecolumn.

```
python
imputer=SimpleImputer(strategy='mean')#Youcanalsouse'median'or'mo
st_frequent'
data['column_name']=imputer.fit_transform(data[['column_name']])
```

- **DroppingRows**:Removerowswithmissingvaluesifthenumberof missing values is small and doesn't significantly affect yourdataset.

pythondata.dropna(inplace=Tru
e)

4. HandlingCategoricalFeatures

Ifyourdatasetcontainscategoricalfeatures,youneedtoconvertthemintonu merical representations.Thiscanbedoneinseveralways:

- **Label Encoding**: Use label encoding to convert categoricalvariablesintoordinalintegers.Thisissuitablewhenther eisanordinalrelationship between categories.

```
python
label_encoder =
LabelEncoder()data['categorical_column']
=label_encoder.fit_transform(data['categorical_column'])
```

- **One-HotEncoding**:Useone-hotencodingtoconvertcategoricalvariables into binary columns. Each category becomes a newbinarycolumnwith 0s and1s.

```
python
one_hot_encoder=OneHotEncoder()e
ncoded_categories=
one_hot_encoder.fit_transform(data[['categorical_column']]).toarray()enc
oded_df =
pd.DataFrame(encoded_categories,columns=one_hot_encoder.get_featur
e_names(['categorical_column']))data = pd.concat([data, encoded_df],
axis=1)data.drop(['categorical_column'],axis=1,inplace=True)
```

5. StandardizationorNormalization(ifnecessary)

Depending on your machine learning algorithm, you might want tostandardize or normalize your numerical features to have a consistentscale.YoucanusetechniqueslikeMin-Maxscalingor StandardScalerfromScikit-Learn.

```
python
fromsklearn.preprocessingimportStandardScaler,MinMaxScaler

scaler = StandardScaler()# or MinMaxScalerdata[['numerical_column1', 'numerical_column2']]
=scaler.fit_transform(data[['numerical_column1','numerical_column2']])
```

6. SaveProcessedData(Optional)

Ifyouwanttosaveyourcleanedand preprocesseddata forfutureuse,you can use the to_csv method in Pandas or other appropriate fileformats.

```
python
data.to_csv('preprocessed_data.csv',index=False)
```

By following these steps, you can clean and preprocess your data, handlemissing values, and convert categorical features into numericalrepresentations suitable for machine learning or analysis. Make sure tocustomize these steps according to your specific dataset andrequirements.

3.ExploratoryDataAnalysis:

ExploratoryDataAnalysis(EDA)isacrucialstepinunderstandingyourdata and extracting valuable insights from it. In this example, we'llassume you have a dataset containing information about registeredcompanies. Here's how you can perform EDA to understand thedistribution,relationships,anduniquecharacteristicsofthesecompanies:

1. ImportLibraries
StartbyimportingthenecessaryPythonlibrariesfordataanalysis and visualization.

```
python
importpandasaspdi
mportnumpyasnp
importmatplotlib.pyplotaspltі
mportseabornas sns
```

2. LoadYourDataset LoadyourdatasetintoaPandasDataFrameifyou haven't already (you can reuse the data DataFrame from thepreviousexample).

```
python
data=pd.read_csv('your_data.csv')
```

3. BasicDataExploration

- **PreviewData**:Usedata.head()todisplaythefirstfew rowsofyourdatasettogetaninitialsenseof thedata'sstructure.

```
pythonprint(data.hea
d())
```

- **SummaryStatistics**:Getsummarystatisticsfornumericalcol umnstounderstandcentraltendenciesandspreads.

```
python
```

print(data.describe())

4. DataVisualization

- **Histograms**:Createhistogramstovisualizethedistributionofnu mericalvariables.

python
```
data['numerical_column'].plot(kind='hist',bins=20,edgecolor='k')plt.xlabe
l('NumericalColumn')
plt.ylabel('Frequency')
plt.title('HistogramofNumericalColumn')p
lt.show()
```

- **Box Plots**: Use box plots to identify outliers and understand thedistributionofnumericalvariables.

python
```
sns.boxplot(x='categorical_column',y='numerical_column',data=data)
plt.xlabel('CategoricalColumn')
plt.ylabel('NumericalColumn')
plt.title('BoxPlotofNumericalColumn
byCategory')plt.xticks(rotation=90)
plt.show()
```

- **CountPlots**:Createcountplotstovisualizethedistributionofcateg oricalvariables.

python
```
sns.countplot(x='categorical_column',data=data)
plt.xlabel('Categorical
Column')plt.ylabel('Count')
plt.title('CountPlotofCategoricalColumn')pl
t.xticks(rotation=90)
plt.show()
```

5. RelationshipsandCorrelations

- **CorrelationMatrix**:Computeandvisualizethecorrelationbet weennumericalvariables.

python
correlation_matrix=data.corr()sns.heatmap(correlation_matrix,a nnot=True,cmap='coolwarm',linewidths=0.5)
plt.title('CorrelationMatrix')
plt.show()

- **Pairplots**:Createpairplotstovisualizepairwiserelationshipsbet weennumericalvariables.

python
sns.pairplot(data,
hue='categorical_column')plt.suptitle('Pairpl otofNumericalVariables')plt.show()

6. UniqueCharacteristics

- **UniqueValues**:Exploretheuniquevaluesincategoricalcolumnstoid entifyuniquecharacteristics.

python
unique_values =
data['categorical_column'].unique()print("UniqueValuesinCategor icalColumn:",unique_values)

- **ValueCounts**:
Getthecountofeachuniquevalueinacategoricalcolumn.

python
value_counts=data['categorical_column'].value_counts()p rint("ValueCounts:\n",value_counts)

These are some common EDA techniques to get a better understandingofyourdata.You cancustomizeand expandyouranalysisbasedonthespecificquestionsyouwanttoanswerandth echaracteristicsofyour

4.Feature engineering:

Feature engineering involves creating new features or transforming existing onestoimprovetheperformanceofpredictivemodels.Thegoalistoprovidethemodelwith more relevant and informative input data. Here are some techniques andexamples for featureengineering:

1. EncodingCategoricalVariables:

- We've discussed this in the data preprocessing section. You can usetechniqueslikeone-hotencodingorlabelencodingtoconvertcategoricalvariables intonumerical representations.

2. DateandTime Features:

- Extractmeaningfulinformationfromdateandtimevariablessuchasyear,month, day, day of the week, or time of day. These can be useful in time-series analysisor when time-relatedpatterns matter.

```
python
data['year'] =
data['date'].dt.yeardata['month']=dat
a['date'].dt.month
data['day_of_week']=data['date'].dt.dayofweek
```

3. AggregationandSummaryStatistics:

- Createnewfeaturesbyaggregatingorsummarizingexistingones.Forexample, calculate the mean, sum, or standard deviation of numericalvariables foreachcategoryinacategoricalcolumn.

```
python
#Calculatethemeanofanumericalcolumnforeach categoryinacategoricalcolumn
mean_by_category
=data.groupby('categorical_column')['numerical_column'].mean()data['mean_numeri
cal_by_category'] =data['categorical_column'].map(mean_by_category)
```

4. InteractionFeatures:

- Create new features by combining existing ones to capture interactions orrelationships between them. This can be useful in cases where the interactionhaspredictive power.

```
python
data['interaction_feature']=data['feature1']*data['feature2']
```

5. PolynomialFeatures:

- Create polynomial features to capture non-linear relationships in the data.This is particularly useful in polynomial regression or when you suspect thathigher-ordertermsaresignificant.

```
python
fromsklearn.preprocessingimportPolynomialFeatures

poly=PolynomialFeatures(degree=2)
X_poly=poly
```

5.PredictiveModelling:

To develop predictive models for future company registrations, you can followthesesteps:

1.DataPreparation:

- **Ensure your dataset is** cleaned, preprocessed, and contains the relevantfeatures asdiscussedearlier.

-Splityourdataintotrainingandtestingsetstoevaluatethemodel'sperformance.

```python
fromsklearn.model_selectionimporttrain_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. ModelSelection:**

- Chooseappropriatemachinelearningalgorithmsbasedonthenatureofyourproblem.Common choices forpredictivemodeling include:

  - **LinearRegression**:Forregressiontaskswhenthetargetvariableiscontinuous.

  - **LogisticRegression**:Forbinaryclassificationtasks.

  - **RandomForest**,**GradientBoosting**,**XGBoost**:Forbothregressionand classification tasks,andtheyoften performwell.

- **NeuralNetworks**:Forcomplexproblemswithlargedatasets.

- **Support Vector Machines (SVM)**: For classification and regression tasks,especiallywhendealing withhigh-dimensionaldata.

3.ModelTraining:

- Trainyourchosenmachinelearningmodelsusingthetrainingdata.

```python
from sklearn.ensemble import RandomForestClassifier# Replace with theappropriatemodel

model=RandomForestClassifier()#Initializethemodelmodel.fit(X_
train,y_train)# Trainthemodel
```

4.ModelEvaluation:

- Assess the model's performance using appropriate evaluation metrics. Forclassification, common metrics include accuracy, precision, recall, F1-score, andROC-AUC. For regression, you can use metrics like mean squared error (MSE), R-squared,andmeanabsolute error(MAE).

```python
fromsklearn.metricsimportaccuracy_score,classification_report,mean_squared_error

#Forclassification
y_pred=model.predict(X_test)
```

accuracy = accuracy_score(y_test,

y_pred)report=classification_report(y_test,y_pr

ed)

#Forregression

y_pred=model.predict(X_test)

mse=mean_squared_error(y_test,y_pred)
```

**5.HyperparameterTuning:**

- Optimizeyourmodel'shyperparameterstoimproveitsperformance.Youcanusete chniqueslike GridSearchorRandomSearch.

```python

fromsklearn.model_selectionimportGridSearchCV

param_grid={'n_estimators':[100,200,300],'max_depth':[None,10,20]}grid_

search = GridSearchCV(RandomForestClassifier(), param_grid,

cv=5)grid_search.fit(X_train,y_train)

best_params=grid_search.best_params_

# 6.Modelevaluation:

Model evaluation is a crucial step in assessing the performance of your predictivemodels.Thechoiceofevaluationmetricsdependsonthenatureoftheproblemyouare trying to solve (classification, regression, etc.). Below, I'll provide examples ofhow to evaluate predictive models using common metrics for classification andregressiontasks:

**ClassificationMetrics:**

1. **Accuracy:**Itmeasurestheproportionofcorrectlypredictedinstancesoutofthetotalinstances.

```
python
fromsklearn.metricsimportaccuracy_score

y_true=[0,1,1,0,1]
y_pred =[0,1,0,0,1]
accuracy=accuracy_score(y_true,y_pred)p
rint("Accuracy:",accuracy)
```

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

```
python
fromsklearn.metricsimportprecision_score

precision=precision_score(y_true,y_pred)print("
Precision:",precision)
```

3. **Recall (Sensitivity or True Positive Rate):** It measures the proportion oftruepositives correctlypredicted amongallactualpositives.

```
python
fromsklearn.metricsimportrecall_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 whenyou wanttobalanceprecision andrecall.

```
python
fromsklearn.metricsimportf1_score

f1 = f1_score(y_true,
y_pred)print("F1-Score:",f1)
```

5. **Confusion Matrix:** It provides a detailed breakdown of the model'spredictions,includingtruepositives,truenegatives,falsepositives,andfalsenegatives.

```
python
fromsklearn.metricsimportconfusion_matrix

conf_matrix = confusion_matrix(y_true,
y_pred)print("ConfusionMatrix:\n",conf_matrix)
```

6. **Receiver Operating Characteristic (ROC) Curve and Area Under theCurve (AUC):** Useful for binary classification problems with a probabilityscore.

```
python
fromsklearn.metricsimportroc_curve,roc_auc_score

y_probs=model.predict_proba(X_test)[:,1]fpr,tp
r,thresholds=roc_curve(y_true,y_probs)roc_auc
=roc_auc_score(y_true,y_probs)

#
PlotROCCurveplt.figur
e(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('FalsePositive
Rate')plt.ylabel('TruePositiveR
ate')
plt.title('ReceiverOperatingCharacteristic(ROC)')plt.legend(loc='l
owerright')
```

plt.show()

RegressionMetrics:

1. **Mean Absolute Error (MAE):** It measures the average absolute differencebetweenpredictedandactual values.

python
fromsklearn.metricsimportmean_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)p
rint("MAE:",mae)

2. **Mean Squared Error (MSE):** It measures the average of the squareddifferences betweenpredictedand actualvalues.

python
fromsklearn.metricsimportmean_squared_error

mse=mean_squared_error(y_true,y_pred)p
rint("MSE:",mse)

3. **Root Mean Squared Error (RMSE):** It is the square root of MSE andprovidestheerror in the sameunitsas the target variable.

python
importnumpyasnp

rmse =
np.sqrt(mse)print("R
MSE:",rmse)

4. **R-squared(R2):**Itmeasurestheproportionofthevarianceinthedependentvariablethatispredictable fromthe independentvariables.

python
fromsklearn.metricsimportr2_score

r2 = r2_score(y_true,
y_pred)print("R-
squared:",r2)

When evaluating predictive models, choose the evaluation metrics that are mostrelevant to your specific problem and consider the trade-offs between them. It'softenagood practicetouseacombinationofmetricstoget acomprehensiveviewofthe model'sperformance.