**Project Title: AI-Driven Exploration and Prediction of Company Registration Trends with Registrar of Companies.**

**PROBLEM STATEMENT:**

The problem at hand is to conduct an AI-driven exploration and predictive analysis on the master details of companies registered with the Registrar of Companies (ROC). The primary objective is to uncover hidden patterns, gain insights into the company landscape, and forecast future registration trends. This project aims to develop predictive models using advanced Artificial Intelligence techniques to anticipate future company registrations, thereby supporting informed decision-making for businesses, investors, and policymakers.

**DESIGN THINKING:**

**Step 1: Data Collection**

* Data Source: Acquire access to a comprehensive dataset containing information about registered companies. The dataset should include relevant columns such as company name, status, class, category, registration date, authorized capital, paid-up capital, and more. Ensure the dataset is regularly updated.

Tools: Data can be collected from the ROC if available or from third-party sources. Web scraping tools like Beautiful Soup or data APIs can be useful.

**Step 2: Data Preprocessing**

* Data Cleaning: Identify and handle missing values, outliers, and data inconsistencies. Ensure data quality by applying data cleaning techniques.
* Categorical to Numerical Conversion: Convert categorical features (e.g., company status, category) into numerical representations using techniques like one-hot encoding or label encoding.

Tools: Python libraries such as pandas can be used for data preprocessing.

**Step 3: Exploratory Data Analysis (EDA)**

* Descriptive Statistics: Calculate summary statistics (mean, median, standard deviation) for numerical features to understand their distributions.
* Data Visualization: Create visualizations (e.g., histograms, scatter plots) to explore data distribution, relationships, and potential patterns.
* Correlation Analysis: Examine the correlation between different features to identify potential dependencies.

Tools: Python libraries like matplotlib, seaborn, or Plotly for data visualization.

**Step 4: Feature Engineering**

* Feature Creation: Develop new features or transform existing ones to capture meaningful information. For instance, extract features from company names or calculate registration trends over time.

Tools: Python libraries like scikit-learn can be used for feature engineering.

**Step 5: Predictive Modelling**

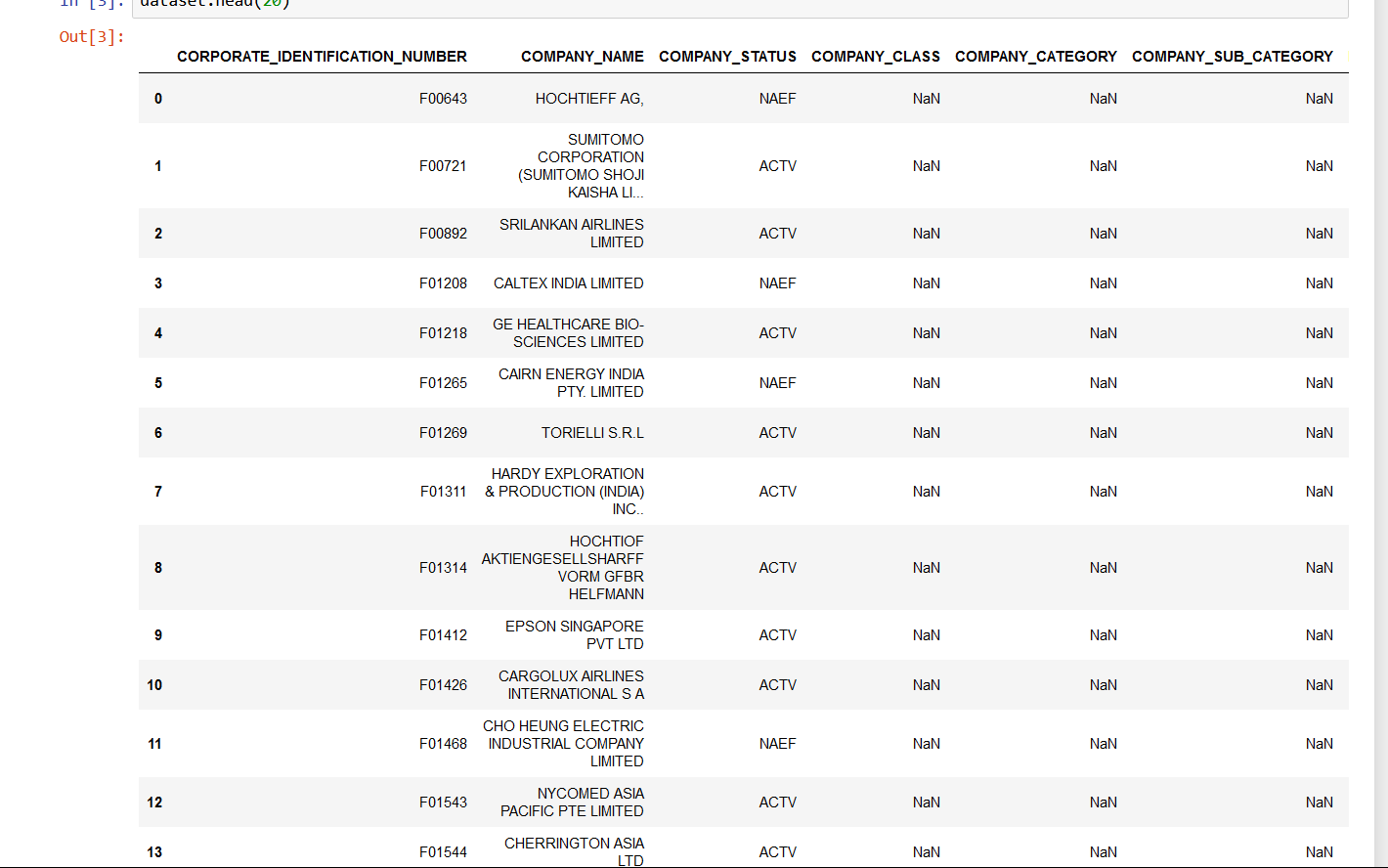
* Model Selection: Choose appropriate machine learning or deep learning algorithms based on the predictive task. Depending on the problem, regression, classification, or time series forecasting models may be applicable.
* Training and Validation: Split the dataset into training and validation sets for model training and evaluation. Implement cross-validation techniques to ensure model generalization.
* Hyperparameter Tuning: Fine-tune model hyperparameters using techniques like grid search or random search.

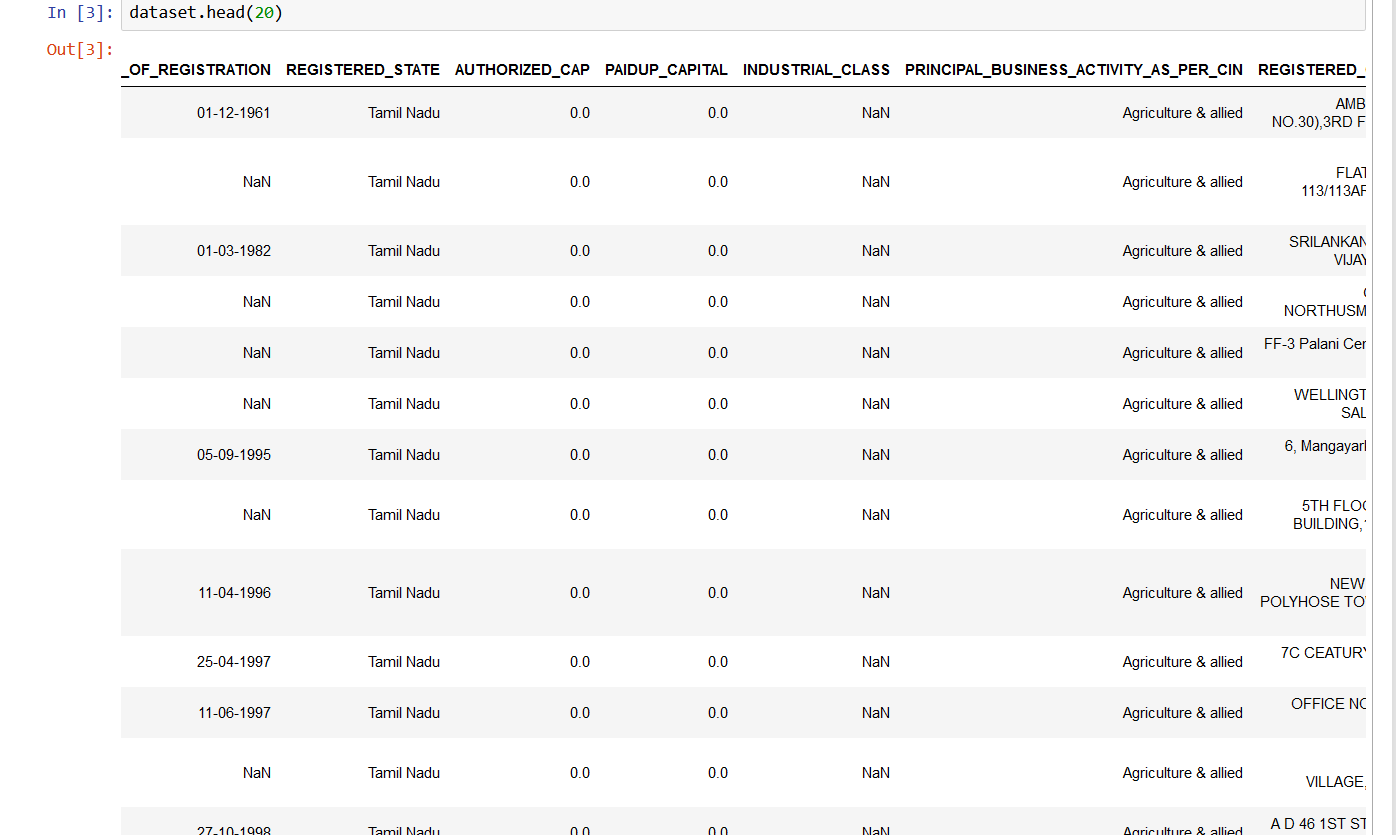
Tools: Python libraries like scikit-learn, TensorFlow, or PyTorch for machine learning.

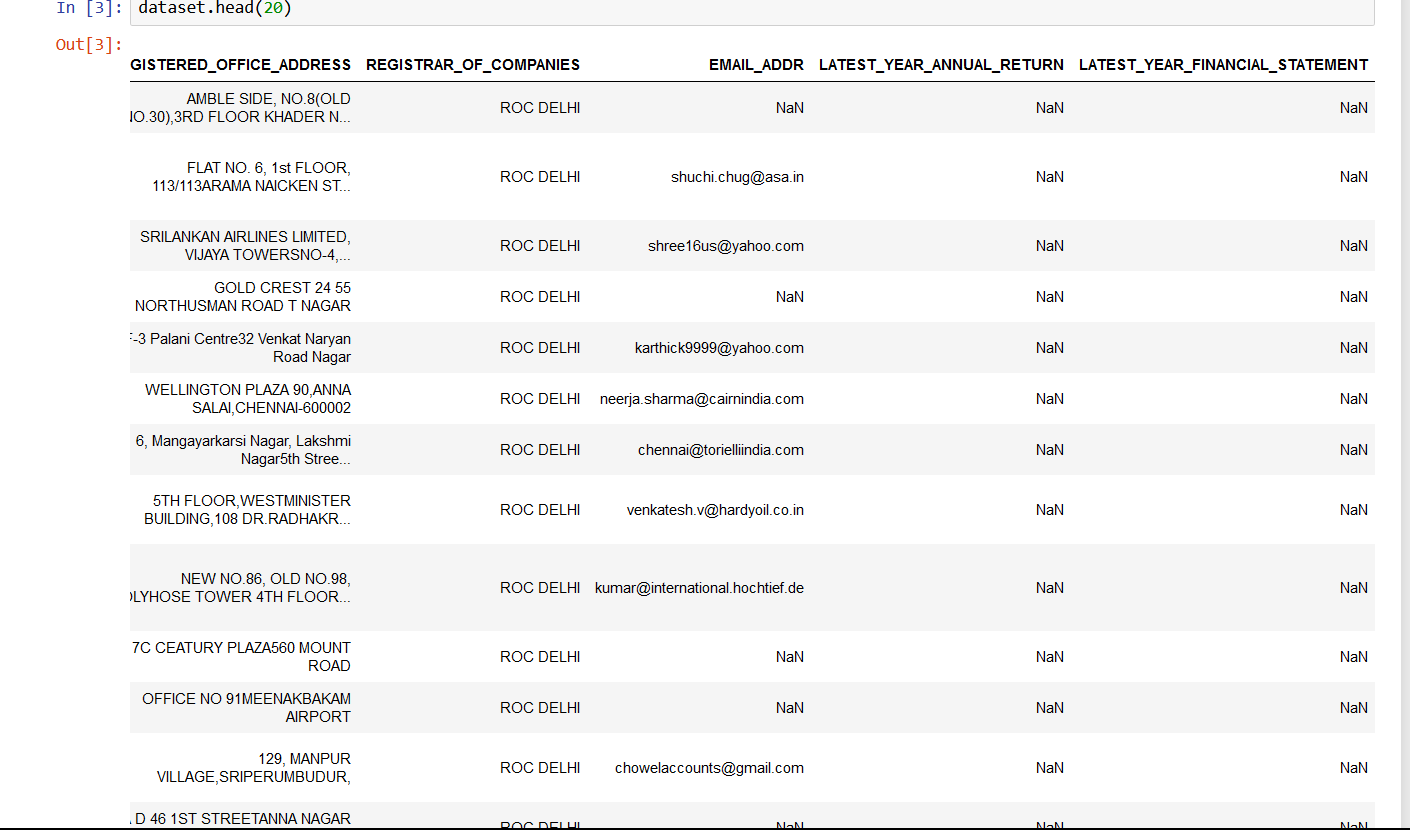
**Step 6: Model Evaluation**

* Performance Metrics: Evaluate model performance using relevant metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, tailored to the specific prediction task.
* Cross-Validation: Use cross-validation to assess model generalization and mitigate overfitting.
* Model Comparison: Compare the performance of different models to select the most suitable one for predicting company registration trends. Tools: Python libraries like scikit-learn for metric calculations.

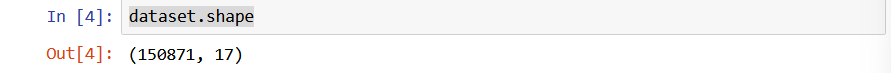
**Data set of master registration of Tamil Nadu Government up to 2019**







We used the data set of Tamil Nadu government master registration of up to 2019. It contains company names, registration dates, classes, category, sub category etc…



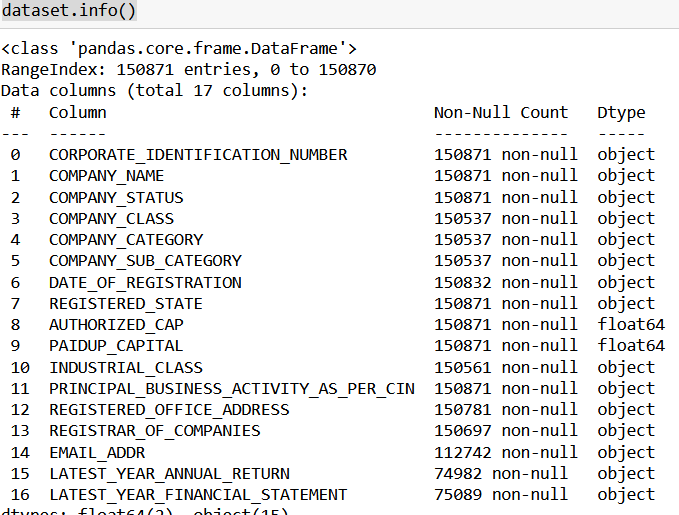
The data set contains 150871(data), 17(columns)of company Master registration.

The columns contain names are

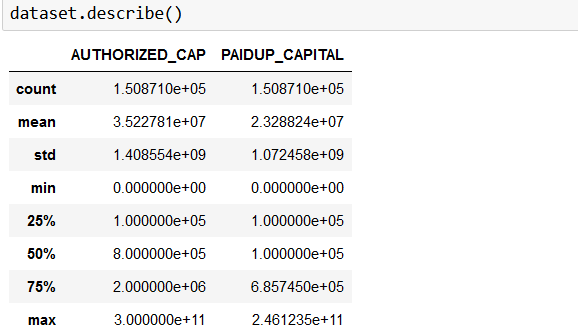
(['CORPORATE\_IDENTIFICATION\_NUMBER','COMPANY\_NAME','COMPANY\_STATUS',’COMPANY\_CLASS', 'COMPANY\_CATEGORY', 'COMPANY\_SUB\_CATEGORY',DATE\_OF\_REGISTRATION', 'REGISTERED\_STATE', 'AUTHORIZED\_CAP',‘PAIDUP\_CAPITAL','INDUSTRIAL\_CLASS','PRINCIPAL\_BUSINESS\_ACTIVITY\_AS\_PER\_CIN', 'REGISTERED\_OFFICE\_ADDRESS','REGISTRAR\_OF\_COMPANIES', 'EMAIL\_ADDR', 'LATEST\_YEAR\_ANNUAL\_RETURN','LATEST\_YEAR\_FINANCIAL\_STATEMENT'],dtype='object')

The dataset contains information about each column names and their null values, non- null values, Dtype info () and describe()

**INFO ()**



**DESCRIBE** **()**

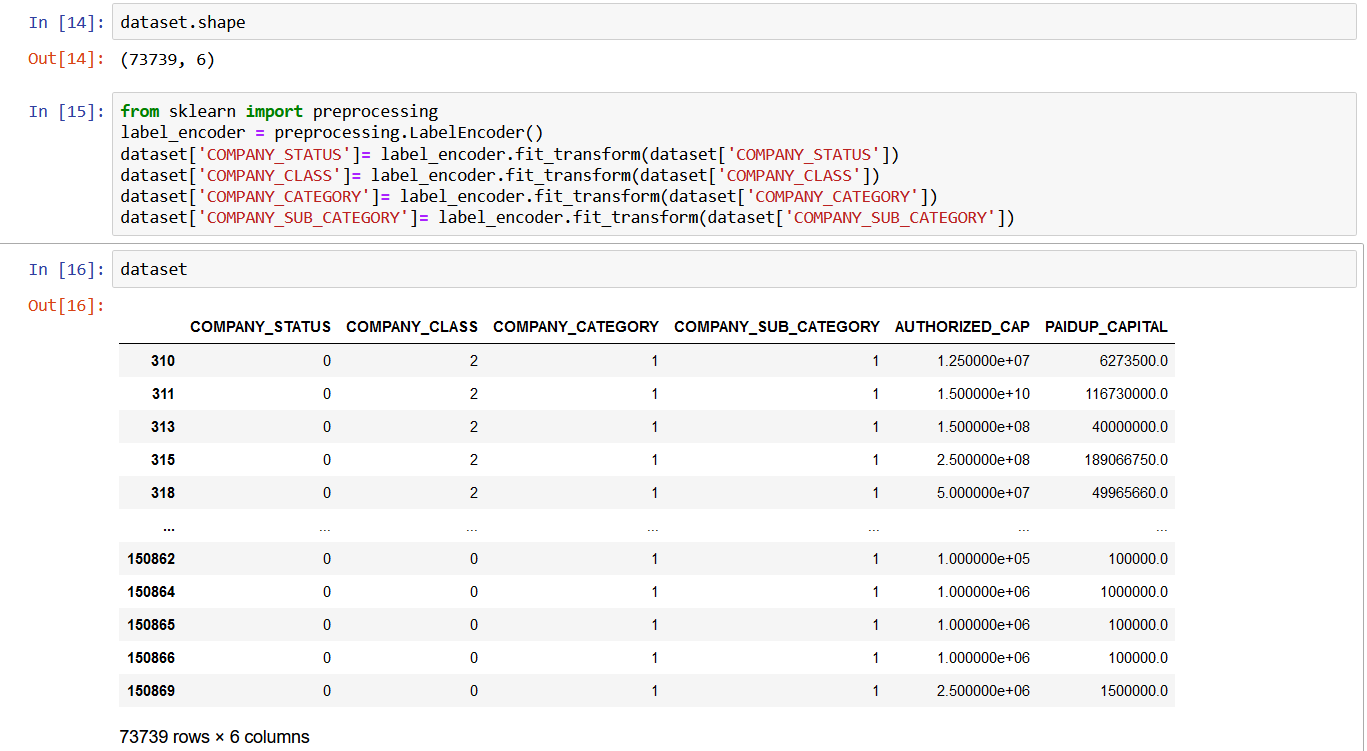


​This data set is used to develop predictive models using advanced Artificial Intelligence techniques to anticipate future company registrations, thereby supporting informed decision-making for businesses, investors, and policymakers and analyse future registration of company after few years also possible to predict how many members will join as per company dataset analysis.

**Data Preprocessing**

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. It refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models.

Data preprocessing import libraries, coding and output



In this technique find is null(), label encoding, handling missing data, values, dropna() ,Splitting, Feature scaling

**ISNULL()**

dataset.isnull().sum()

CORPORATE\_IDENTIFICATION\_NUMBER 0

COMPANY\_NAME 0

COMPANY\_STATUS 0

COMPANY\_CLASS 334

COMPANY\_CATEGORY 334

COMPANY\_SUB\_CATEGORY 334

DATE\_OF\_REGISTRATION 39

REGISTERED\_STATE 0

AUTHORIZED\_CAP 0

PAIDUP\_CAPITAL 0

INDUSTRIAL\_CLASS 310

PRINCIPAL\_BUSINESS\_ACTIVITY\_AS\_PER\_CIN 0

REGISTERED\_OFFICE\_ADDRESS 90

REGISTRAR\_OF\_COMPANIES 174

EMAIL\_ADDR 38129

LATEST\_YEAR\_ANNUAL\_RETURN 75889

LATEST\_YEAR\_FINANCIAL\_STATEMENT 75782

dtype: int64

After data cleaning process is null values are

COMPANY\_STATUS 0

COMPANY\_CLASS 334

COMPANY\_CATEGORY 334

COMPANY\_SUB\_CATEGORY 334

PAIDUP\_CAPITAL 0

dtype: int64

**DROPNA()**

dataset.dropna(inplace=True)

| **CORPORATE\_IDENTIFICATION\_NUMBER** | **COMPANY\_NAME** | **COMPANY\_STATUS** | **COMPANY\_CLASS** | **COMPANY\_CATEGORY** | **COMPANY\_SUB\_CATEGORY** | **DATE\_OF\_REGISTRATION** | **REGISTERED\_STATE** | **AUTHORIZED\_CAP** | **PAIDUP\_CAPITAL** | **INDUSTRIAL\_CLASS** | **PRINCIPAL\_BUSINESS\_ACTIVITY\_AS\_PER\_CIN** | **REGISTERED\_OFFICE\_ADDRESS** | **REGISTRAR\_OF\_COMPANIES** | **EMAIL\_ADDR** | **LATEST\_YEAR\_ANNUAL\_RETURN** | **LATEST\_YEAR\_FINANCIAL\_STATEMENT** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **310** | L01117TZ1943PLC000117 | NEELAMALAI AGRO INDUSTRIES LIMITED | ACTV | Public | Company limited by Shares | Non-govt company | 21-04-1943 | Tamil Nadu | 1.250000e+07 | 6273500.0 | 01117 | Agriculture & allied | KATARY ESTATEKATARY POSTCOONOOR | ROC COIMBATORE | secneelamalai@avtplantations.co.in | 31-03-2019 |
| **311** | L01119TN1986PLC013473 | ABAN OFFSHORE LIMITED | ACTV | Public | Company limited by Shares | Non-govt company | 25-09-1986 | Tamil Nadu | 1.500000e+10 | 116730000.0 | 01119 | Agriculture & allied | 'JANPRIYA CREST'96, PANTHEON ROAD,EGMORE | ROC CHENNAI | secretarial@aban.com | 31-03-2019 |

**LABEL ENCODING**:

from sklearn import preprocessing

label\_encoder = preprocessing.LabelEncoder()

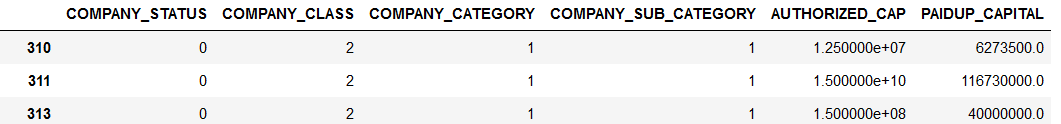
dataset['COMPANY\_STATUS']= label\_encoder.fit\_transform(dataset['COMPANY\_STATUS'])

dataset['COMPANY\_CLASS']= label\_encoder.fit\_transform(dataset['COMPANY\_CLASS'])

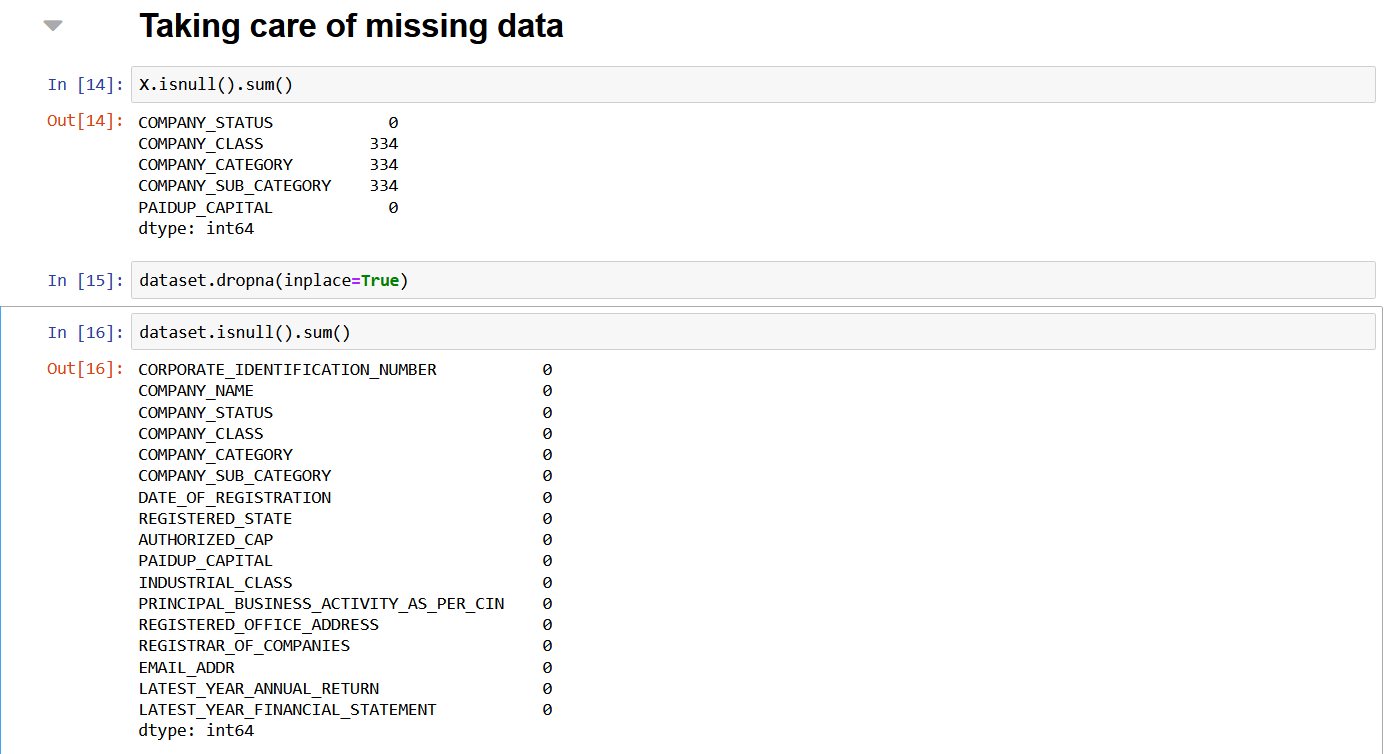
dataset['COMPANY\_CATEGORY']= label\_encoder.fit\_transform(dataset['COMPANY\_CATEGORY'])

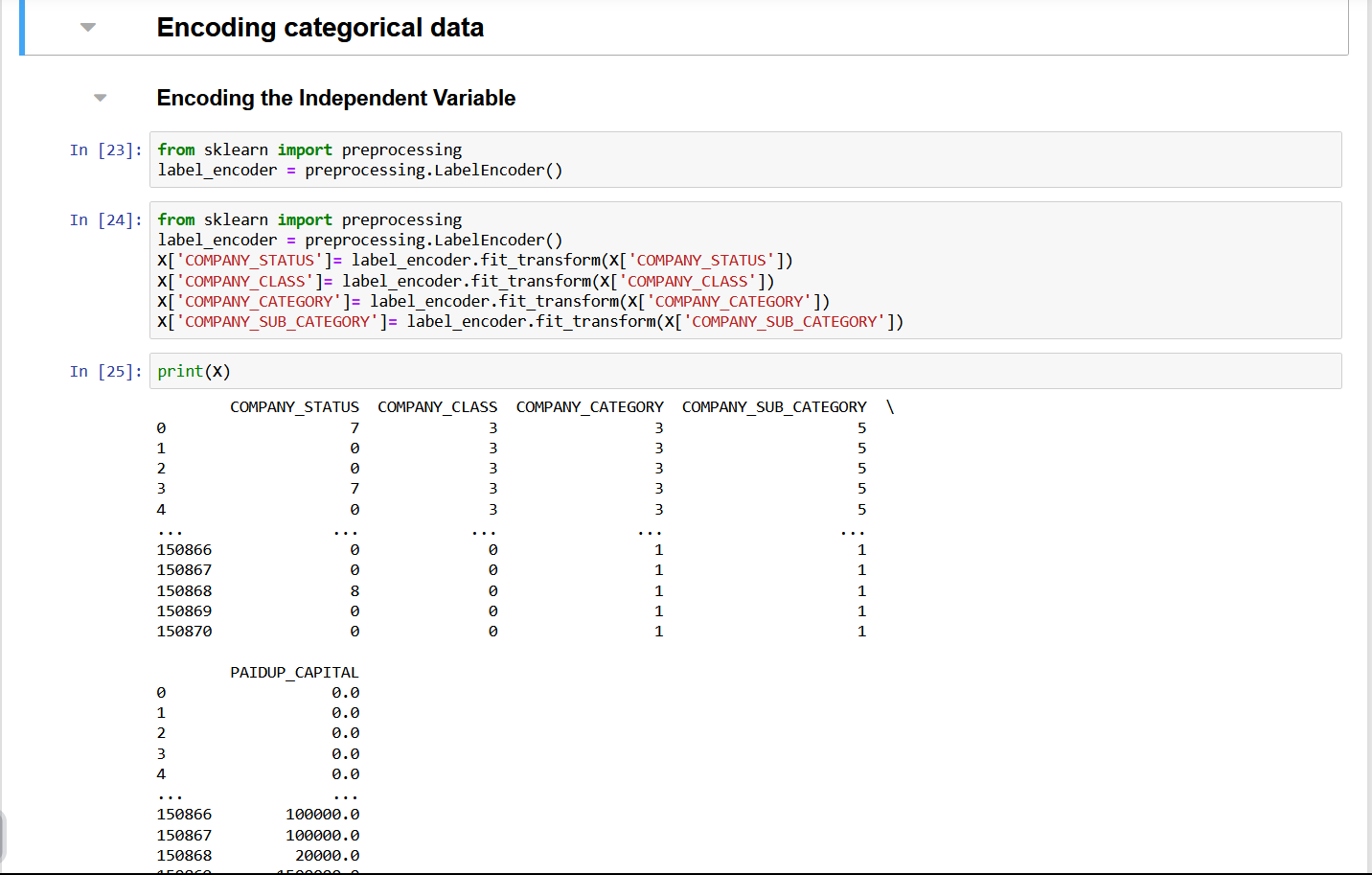
dataset['COMPANY\_SUB\_CATEGORY']= label\_encoder.fit\_transform(dataset['COMPANY\_SUB\_CATEGORY'])

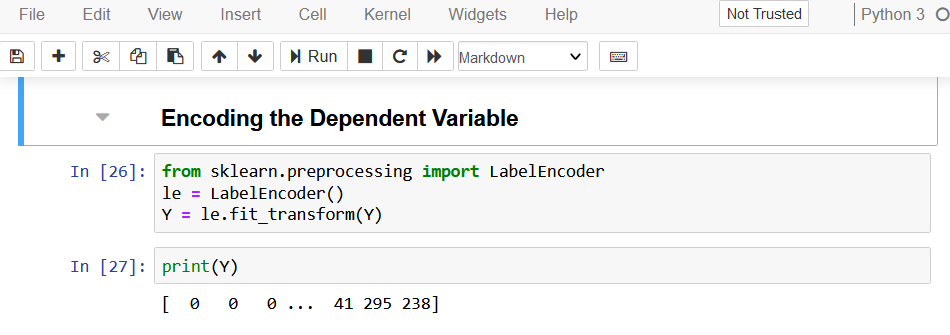
OUTPUT:



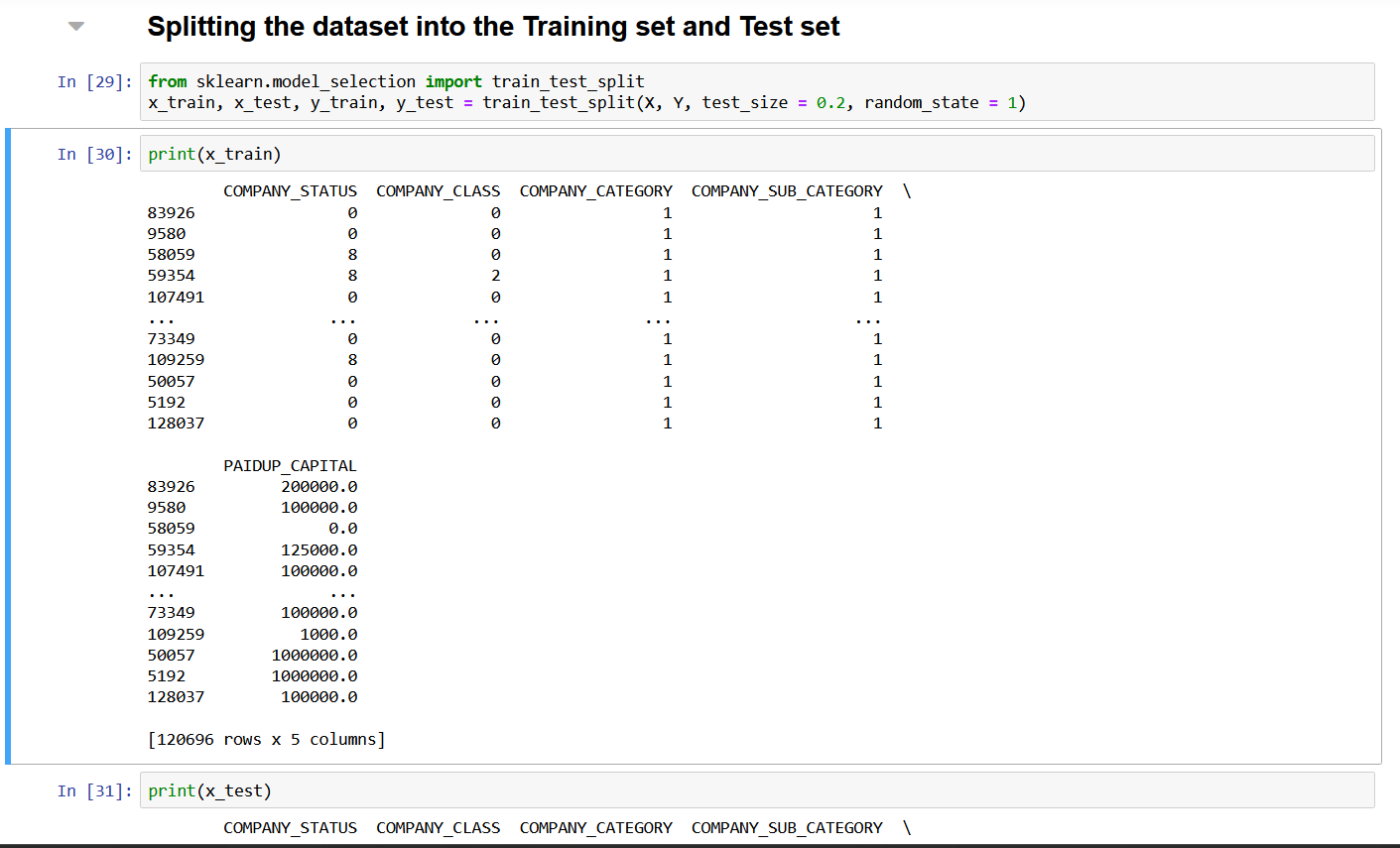
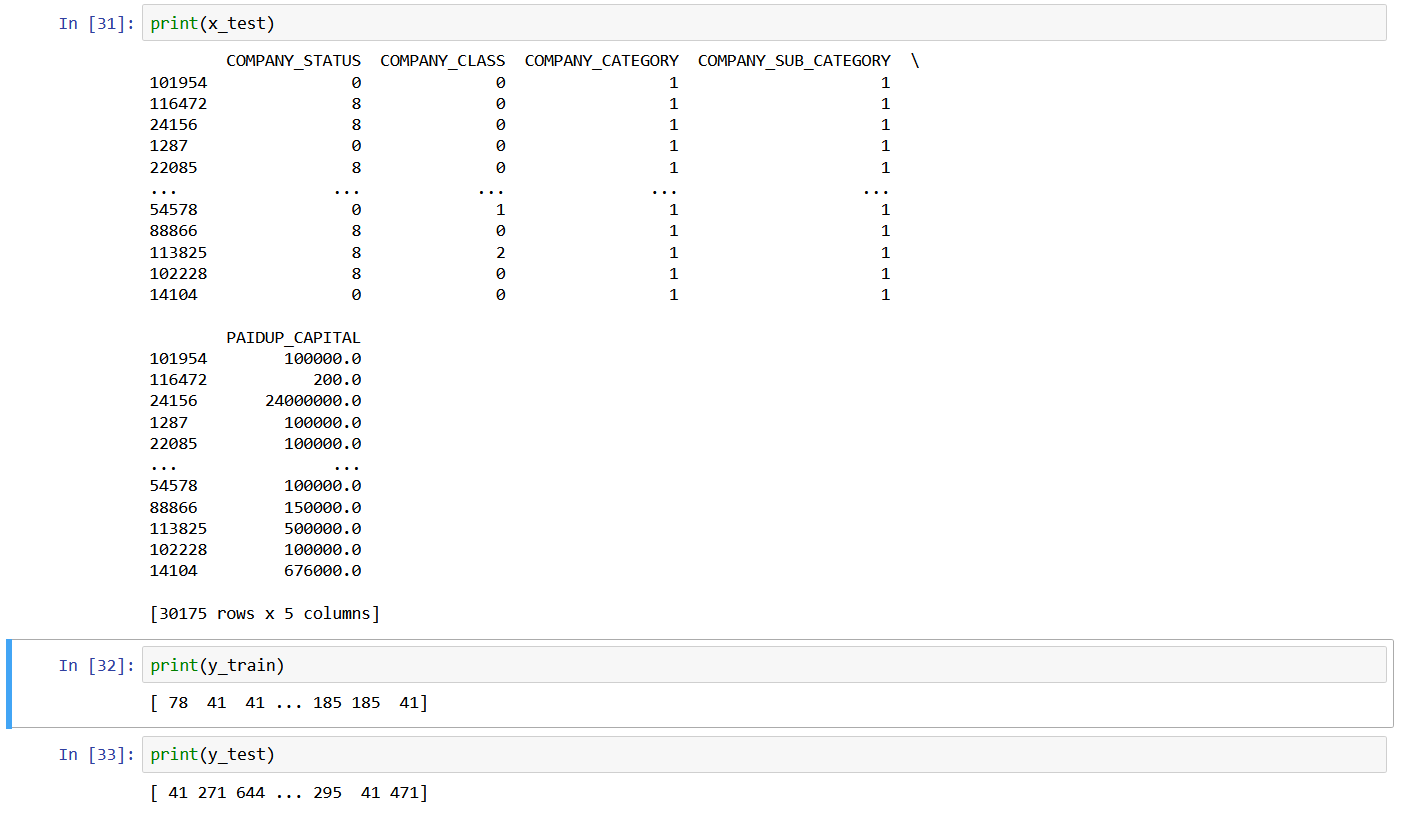
Missing data, dependent variable and independent variable coding and outputs







Splitting the dataset and training the dataset using sklearn library files

Feature scaling



AI ALGORITHM:

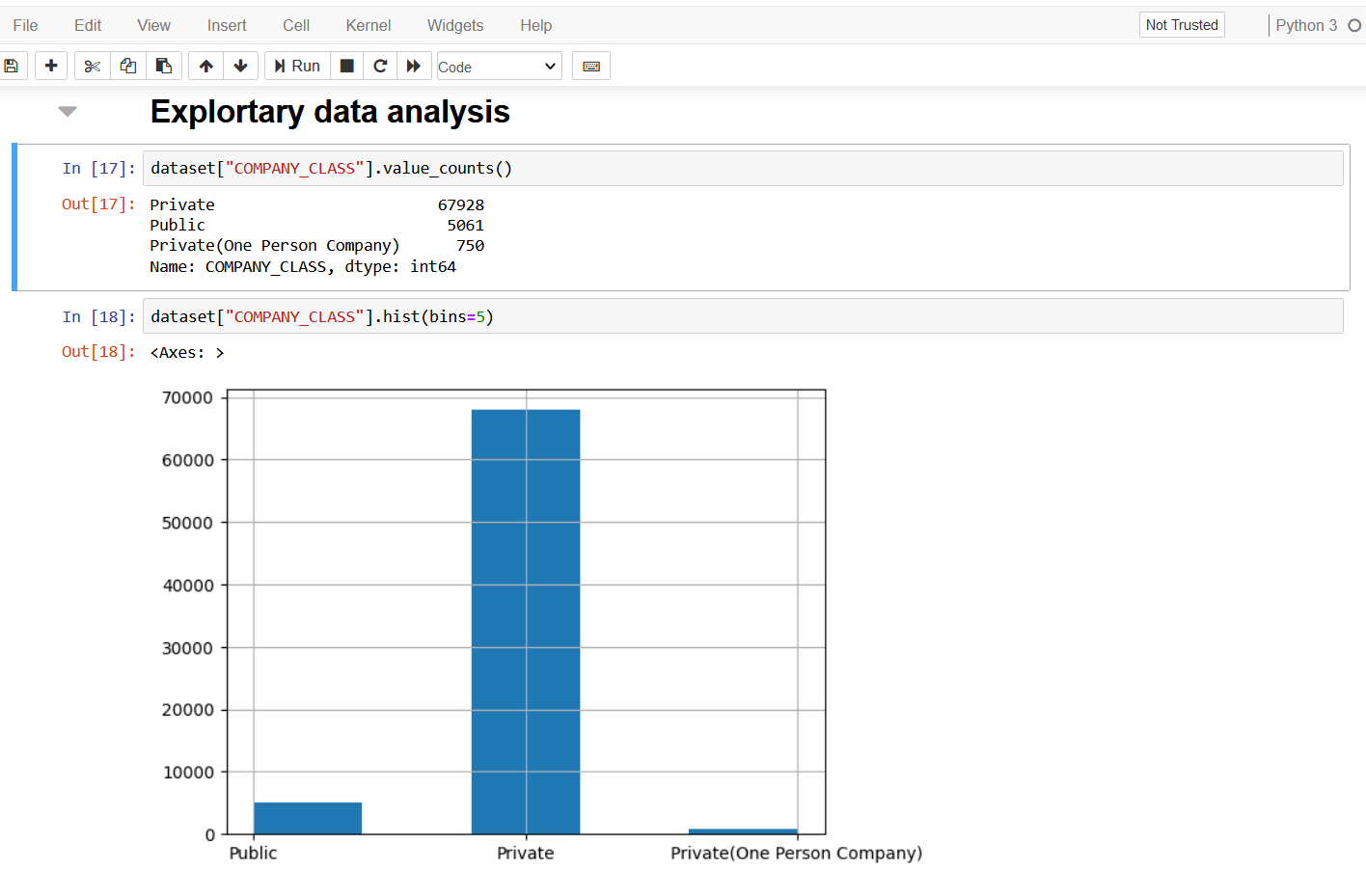
Here we used Logistic algorithm for model evaluation. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of certain classes based on some dependent variables. In short, the logistic regression model computes a sum of the input features (in most cases, there is a bias term), and calculates the logistic of the result.

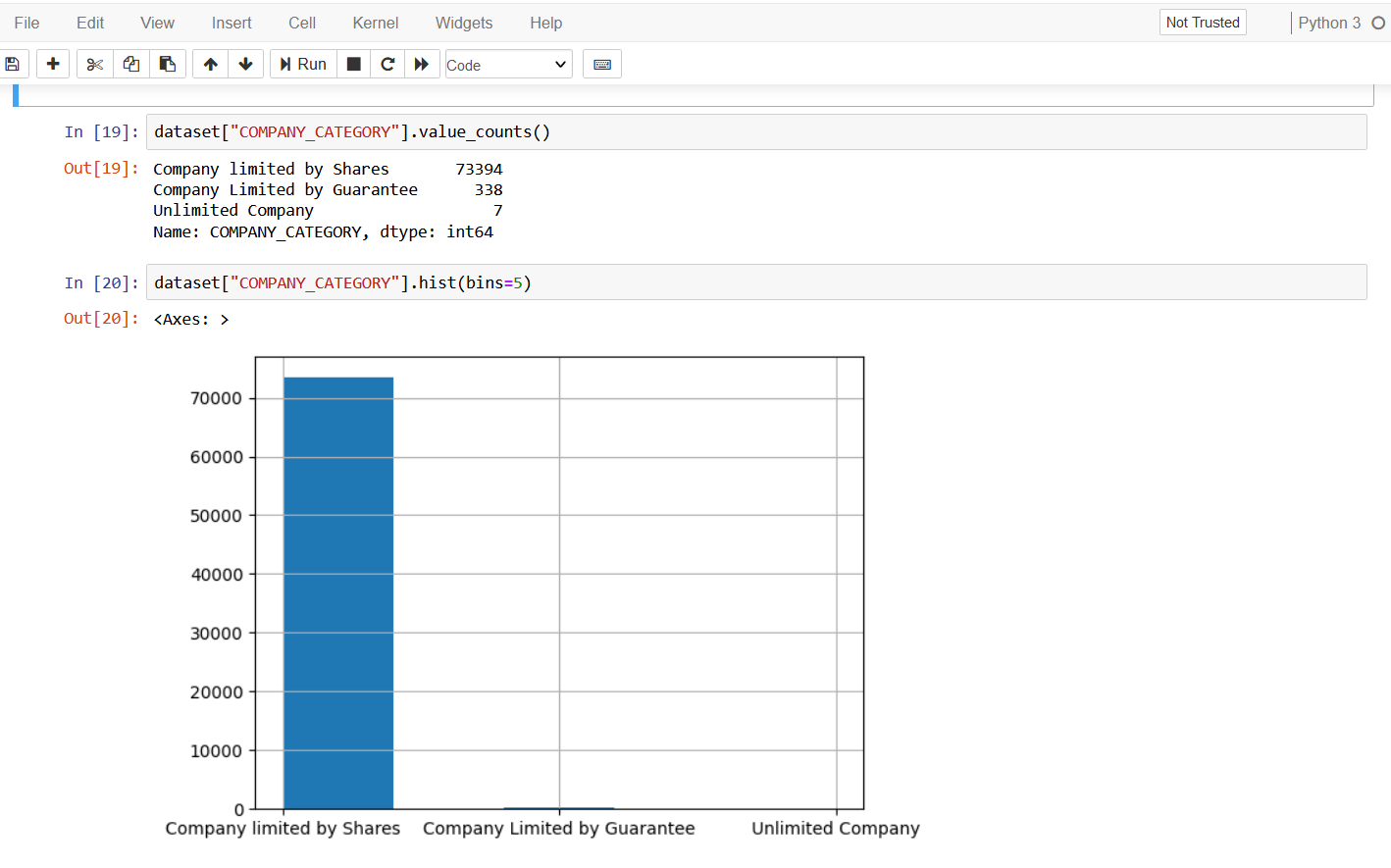
Gain Insights: Through data analysis, understand the characteristics of registered companies, including their status, class, category, registration date, authorized capital, paid-up capital, and more.

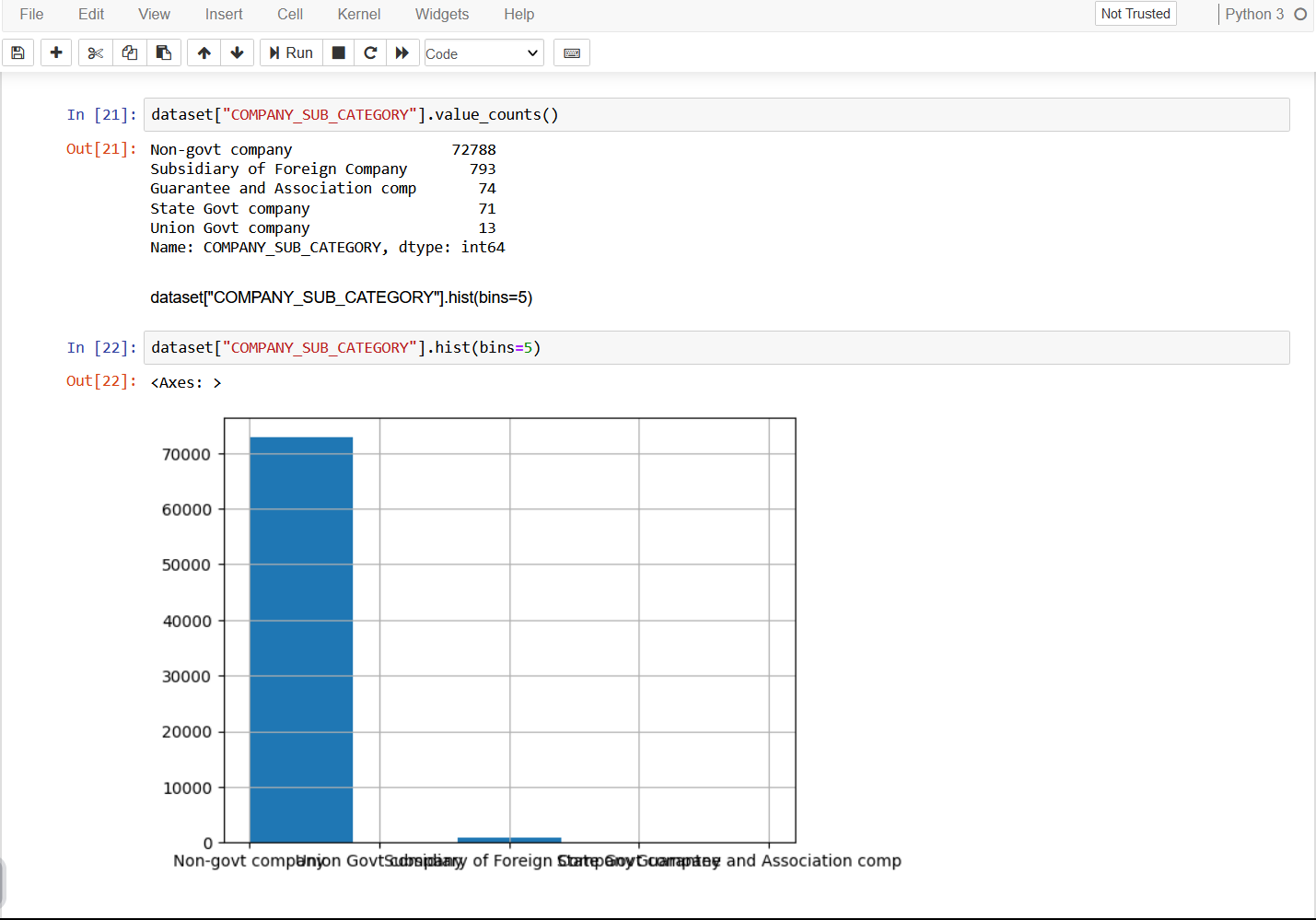
Company name is used identify the company how many of them registered it, company status indicates Company Status indicates the present operating condition of a company, class category like private or public, registration date consist when employee register and start to work, authorized capital and paid up capital is that authorized capital is the maximum amount of capital a company is legally permitted to raise by selling its shares, whereas paid up capital is the actual amount a company has received after selling its shares. Likewise industrial class and sub class category, status, registration date also.

**EXPLOTARY DATA ANALYSIS**

A method used to analyze and summarize data sets. Exploratory data analysis (EDA) is used to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.



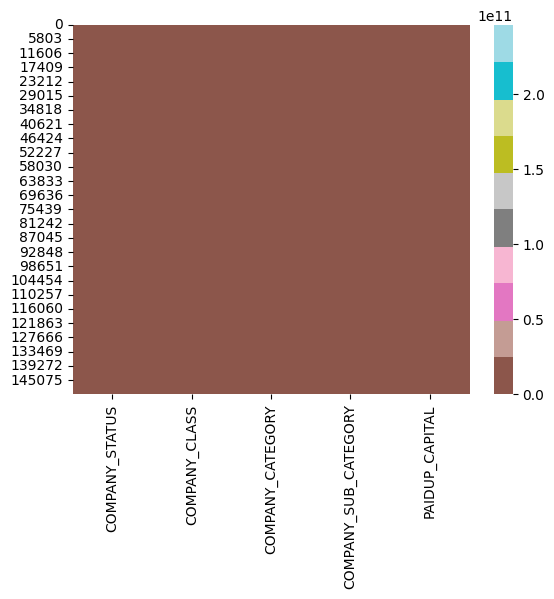




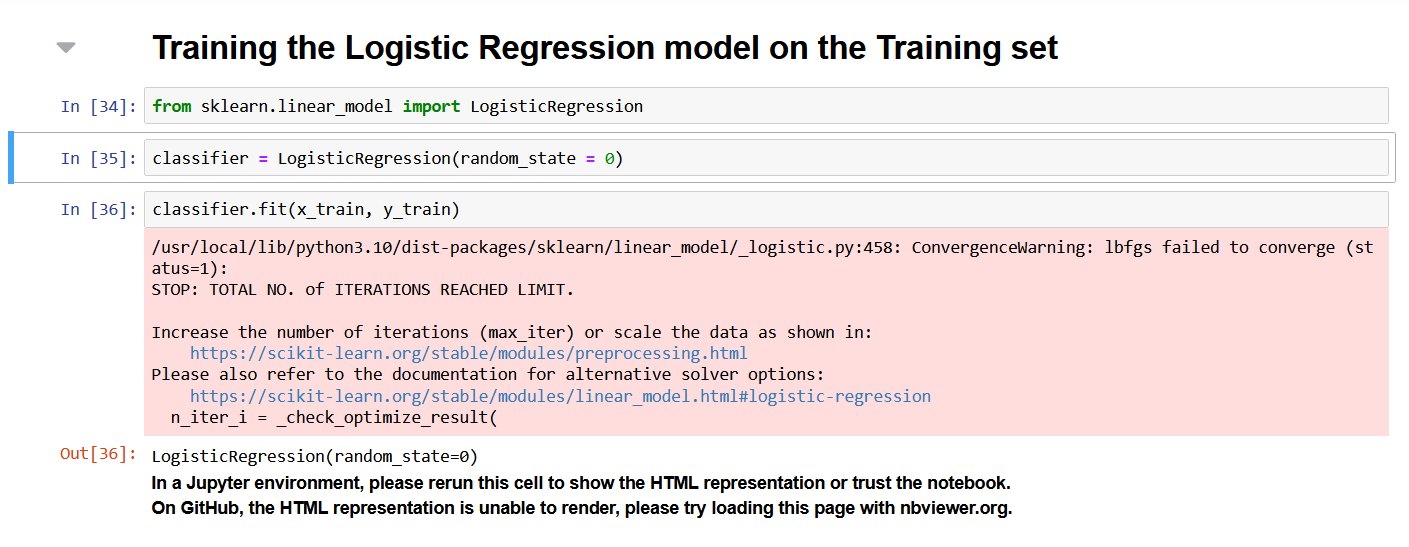
**HEAT MAP**

By analysing the heat map, the following diagram…

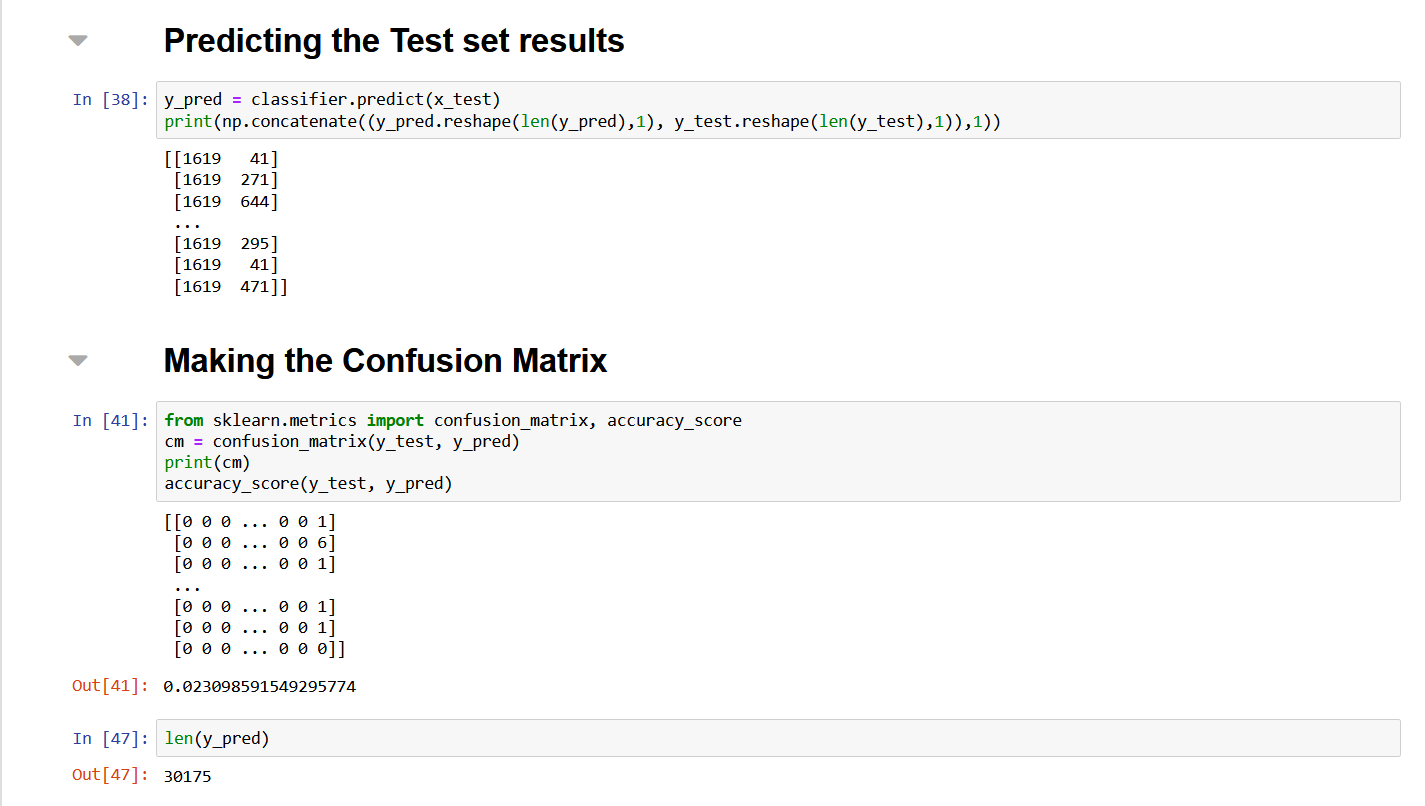
sns.heatmap(X,cmap = "tab20",center = 0)



**TRAINING THE LOGISTIC REGRESSION MODEL OF TRAINING DATASET**



**PREDECTIVE MODEL PERFORMANCE AND RESULT**



**CONFUSION MATRIX:**

from sklearn import metrics

cm = metrics.confusion\_matrix(y\_test[:100], y\_pred[:100])

cm\_display = metrics.ConfusionMatrixDisplay(confusion\_matrix=cm,

display\_labels=[False, True])

cm\_display.plot()

plt.show()

