Data Exploration and Visualization

1. Import Libraries:
- The code begins by importing the necessary libraries: `pandas`, `matplotlib`, and `seaborn`. These libraries are used for data manipulation and visualization.

2. Load Data:
- The dataset is loaded from a CSV file located at the specified file path. The `encoding='latin1'` argument is used to handle special characters in the CSV file.

3. Display First Few Rows:
- The code prints the first few rows of the dataset to provide an initial overview of the data.

4. Basic Statistics:
- Summary statistics of the numerical attributes in the dataset are generated using the `describe()` method. This includes information such as mean, standard deviation, and quartiles, and the summary statistics are then printed.

5. Data Distribution Histograms:
- Numerical attributes are selected from the dataset, and histograms are created to visualize the distribution of these attributes. The histograms are displayed using Matplotlib.

6. Correlation Matrix:
- A correlation matrix is calculated for the numerical attributes, showing how different numerical attributes are correlated. The correlation matrix is displayed as a heatmap using Seaborn.

7. Countplot for Categorical Variables:
- For specific categorical columns, count plots are generated to show the distribution of different categories within each column. These plots are also created using Seaborn.

8. Boxplot for Numerical Variables:
- Box plots are generated for specific numerical columns, providing a visual representation of the distribution, central tendency, and variability of the data within each column.

9. Time Series Analysis:
- If the dataset contains a 'DATE_OF_REGISTRATION' column, this section performs time series analysis. It converts the date column to a datetime format, sets it as the index, and resamples the data by month, visualizing it as a time series plot. This is useful for understanding trends over time.

Data Feature Engineering

1. Import Libraries:
- In this section, additional libraries are imported for data preprocessing. These include `LabelEncoder`, `StandardScaler`, `MinMaxScaler`, and `SimpleImputer`.

2. Load the Dataset:
- The dataset is loaded again, using the same file path and encoding method as in Section 1.

3. Define Numerical and Categorical Columns:
- Numerical and categorical columns in the dataset are explicitly defined for further data preprocessing.

4. Scaling of Numerical Columns:
- Numerical columns specified earlier are standardized using `StandardScaler`. This transformation ensures that these columns have a mean of 0 and a standard deviation of 1.

5. Label Encoding of Categorical Columns:
- Categorical columns specified earlier are encoded using `LabelEncoder`, which converts categorical values into numerical representations.

6. One-Hot Encoding for Categorical Variables:
- One-hot encoding is applied to categorical columns. This transforms categorical variables into binary columns, with each binary column representing a category. The `drop_first=True` parameter avoids multicollinearity.

7. Date Feature Engineering:
- The 'DATE_OF_REGISTRATION' column is converted into separate features such as 'Year', 'Month', 'Day', and 'DayOfWeek'. This feature engineering is helpful for time-based analysis.

8. Interaction Feature:
- An 'Authorized_Paidup_Ratio' feature is created by dividing 'AUTHORIZED_CAP' by 'PAIDUP_CAPITAL'.

9. Feature Scaling for Specific Columns:
- The 'AUTHORIZED_CAP' column is scaled using Min-Max scaling, which maps the values to a range between 0 and 1.

10. Feature Imputation:
- Missing values in the 'PAIDUP_CAPITAL' column are imputed by filling them with the mean value.

11. Display the Updated Dataset:
- The code prints the first few rows of the dataset after all the preprocessing steps have been applied.

Predictive Modeling

In this section we load a dataset, clean it by filling missing values, convert categorical features to numerical, split the data into training and testing sets, and train a machine learning model using a Random Forest Classifier. The code then evaluates the model's performance in terms of accuracy, F1-score, and provides a classification report.

1. Loading and Cleaning Data:
- The code begins by loading a dataset from a CSV file and checks for missing values.

2. Filling Missing Values:
- Missing values in specific columns are filled using linear interpolation.

3. Converting Categorical Features to Numerical:
- Categorical features such as 'COMPANY_CLASS,' 'COMPANY_CATEGORY,' and 'COMPANY_SUB_CATEGORY' are converted to numerical values using Label Encoding.

4. Splitting the Data:
- The dataset is split into training and testing sets. Categorical features are encoded, and the data is divided into features and the target variable.

5. Model Selection and Training:
- A Random Forest Classifier is used for multi-class classification. The model is initialized and trained on the training data.

6. Model Evaluation:
- The trained model is used to make predictions on the test set. Model performance is evaluated using accuracy, weighted F1-score, and a classification report, providing detailed metrics for each class.

.