# Data Analysis and Machine Learning Projects

## Table of Contents

* [Introduction](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#introduction)
* [Analysis 1: Basic Information about the Dataset](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#analysis-1-basic-information-about-the-dataset)
* [Analysis 2: Data Cleaning and Preprocessing](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#analysis-2-data-cleaning-and-preprocessing)
* [Analysis 3: Descriptive Statistics](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#analysis-3-descriptive-statistics)
* [Analysis 4: Visualizations](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#analysis-4-visualizations)
* [Analysis 5: Grouping and Aggregation](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#analysis-5-grouping-and-aggregation)
* [Analysis 6: Principal Component Analysis (PCA)](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#analysis-6-principal-component-analysis-pca)
* [Analysis 7: Correlation Matrix](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#analysis-7-correlation-matrix)
* [Machine Learning Project](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#machine-learning-project)
  + [Data Preprocessing](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#data-preprocessing)
  + [Label Encoding](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#label-encoding)
  + [Data Splitting and Model Training](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#data-splitting-and-model-training)
  + [Model Evaluation](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#model-evaluation)
  + [Hyperparameter Tuning](https://chat.openai.com/c/476af419-6d60-4b72-a096-9a1906316d8b#hyperparameter-tuning)

## Introduction

This document provides a detailed overview of the data analysis and machine learning projects. Each section focuses on a specific aspect of the analysis and provides code snippets along with their respective analyses.

## Analysis 1: Basic Information about the Dataset

import pandas as pd # Load the CSV file with specified encoding df = pd.read\_csv('Data\_Gov\_Tamil\_Nadu.csv', encoding='ISO-8859-1')

**Analysis:** The code snippet loads a CSV file containing data about companies registered in Tamil Nadu. A warning is issued regarding mixed data types in column 10.

.

## Analysis 2: Data Cleaning and Preprocessing

# Example: Handling missing values

df.fillna(0, inplace=True)

**Analysis:** This code snippet demonstrates a method for handling missing values by replacing them with zeros.

## Analysis 3: Descriptive Statistics

# Example: Summary statistics for numeric columns

print("\nSummary Statistics for Numeric Columns:")

print(df.describe())

**Analysis:** The code provides summary statistics for numeric columns, including count, mean, standard deviation, minimum, maximum, and quartiles.

## Machine Learning Project

# Assuming other relevant columns are potential features x = df.drop([ 'CORPORATE\_IDENTIFICATION\_NUMBER', 'COMPANY\_NAME', 'COMPANY\_STATUS', 'DATE\_OF\_REGISTRATION'], axis=1) # Assuming 'COMPANY\_CATEGORY' is the column representing the trend category you want to predict y = df['REGISTRAR\_OF\_COMPANIES']

**Analysis:** This section focuses on preparing the data for machine learning, including feature selection and target variable identification.

### Data Preprocessing

## Analysis 4: Visualizations

import matplotlib.pyplot as plt # Example: Histogram of Authorized Capital plt.figure(figsize=(10, 6)) plt.hist(df['AUTHORIZED\_CAP'], bins=20, color='skyblue') plt.title('Histogram of Authorized Capital') plt.xlabel('Authorized Capital') plt.ylabel('Frequency') plt.show()

**Analysis:** This code snippet generates a histogram to visualize the distribution of authorized capital.

## Analysis 5: Grouping and Aggregation

# Example: Average Paid-up Capital by Company Category avg\_paidup\_capital\_by\_category = df.groupby('COMPANY\_CATEGORY')['PAIDUP\_CAPITAL'].mean() print("\nAverage Paid-up Capital by Company Category:") print(avg\_paidup\_capital\_by\_category)

**Analysis:** This code calculates the average paid-up capital for each company category.

## Analysis 6: Principal Component Analysis (PCA)

from sklearn.decomposition import PCA # Assuming 'features' is a DataFrame containing only numeric columns features = df.select\_dtypes(include=['float64', 'int64']) pca = PCA(n\_components=2) principal\_components = pca.fit\_transform(features) principal\_df = pd.DataFrame(data=principal\_components, columns=['PC1', 'PC2']) plt.figure(figsize=(10, 8)) plt.scatter(principal\_df['PC1'], principal\_df['PC2']) plt.title('PCA') plt.xlabel('Principal Component 1') plt.ylabel('Principal Component 2') plt.show()

**Analysis:** This code snippet performs Principal Component Analysis (PCA) on numeric features, reducing them to two principal components for visualization.

## Machine Learning Project (Continued)

### Model Evaluation

from sklearn.metrics import accuracy\_score, confusion\_matrix y\_pred = model.predict(X\_test) accuracy = accuracy\_score(y\_test, y\_pred) conf\_matrix = confusion\_matrix(y\_test, y\_pred) print(f"Accuracy: {accuracy}") print(f"Confusion Matrix:\n{conf\_matrix}")

**Analysis:** This section evaluates the performance of the machine learning model using accuracy and a confusion matrix

### Hyperparameter Tuning

from sklearn.model\_selection import GridSearchCV param\_grid = {'n\_estimators': [50, 100, 200], 'max\_depth': [None, 10, 20]} grid\_search = GridSearchCV(model, param\_grid, cv=5) grid\_search.fit(X\_train, y\_train) best\_model = grid\_search.best\_estimator\_ best\_params = grid\_search.best\_params\_ print(f"Best Parameters: {best\_params}") y\_pred = best\_model.predict(X\_test) accuracy = accuracy\_score(y\_test, y\_pred) conf\_matrix = confusion\_matrix(y\_test, y\_pred) print(f"Accuracy: {accuracy}") print(f"Confusion Matrix:\n{conf\_matrix}")

**Analysis:** This code snippet performs hyperparameter tuning using GridSearchCV to optimize the Random Forest Classifier.

## Conclusion

This documentation provides a comprehensive overview of the data analysis and machine learning projects. Each section includes code snippets with detailed analyses. The machine learning project encompasses data preprocessing, model training, evaluation, and hyperparameter tuning, resulting in an optimized model.