**PROJECT DOCUMENTATION**

**Name :** Nivash C

**Course :** Data Analytics

**Batch Number :** 26

**Mode of Study :** Offline

**Roll Number :** 160324CBR004

**Table of Contents**

1. Introduction

2. Aim

3. Project Description

4. Script file functionalities

5. Input Versatility with Error Handling and Exception Handling

6. Results and Outcomes

7. Conclusion

**1. Introduction**

This project involves automating the process of data ingestion, cleaning, and visualization for a loan dataset. By leveraging Python scripts, we aim to streamline these tasks and provide valuable insights from the data. The project integrates with a MySQL database, cleans the data by removing duplicates and null values, identifies significant features, and generates visualizations to understand key aspects of the dataset.

**2. Aim**

The primary objective of this project is to create a streamlined data processing pipeline that ingests raw data from CSV files, cleans the data by removing duplicates and handling missing values, and stores it in a MySQL database. Additionally, the project aims to identify significant features for predicting loan status and visualize the data using various plots.

**3. Project Description**

This project encompasses a complete data processing and visualization pipeline using Python. The process includes:

* Reading raw data from CSV files.
* Appending the data to a MySQL database.
* Cleaning the data by removing duplicates and handling missing values.
* Performing feature selection to identify the most correlated features for loan status prediction.
* Visualizing the data through line plots, histograms, and pie charts.

The technologies used in this project include Python, pandas, SQLAlchemy for database operations, scikit-learn for feature selection, and seaborn and matplotlib for data visualization.

**4. Script File Functionalities**

**4.1 Script File 1**

Description:

This script handles the initial ingestion of data from a CSV file and appends it to a specified table in a MySQL database. It uses pandas for reading the CSV file and SQLAlchemy for database connectivity and operations.

**Code:**

import pandas as pd

from sqlalchemy import create\_engine

def append\_to\_sql(file):

df = pd.read\_csv(file)

path = "mysql://root:Nivash%4003@localhost/Nivash\_c"

engine = create\_engine(path)

print("Appending to SQL...")

while True:

table\_name = input("Enter table name")

try:

df.to\_sql(table\_name, con=engine, index=False)

print("Function completed")

print("Table created in SQL")

print("Table Name: ", table\_name)

break

except ValueError:

print("Table name already exists. Try a different table name")

**4.2 Script File 2**

Description:

This script cleans the data by removing duplicates and handling missing values. It also performs feature selection to identify the most correlated features for predicting loan status. The cleaned data is then stored back in the MySQL database.

**Code:**

from pymysql import connect

import pandas as pd

from sqlalchemy import create\_engine

from sklearn.feature\_selection import SelectKBest, f\_classif

def clean(t\_name):

df = connect(host='localhost', user='root', password='Nivash@03', database='Nivash\_C')

cur = df.cursor()

query = "select \* from " + t\_name

cur.execute(query)

df1 = cur.fetchall()

columns = [desc[0] for desc in cur.description]

df\_1 = pd.DataFrame(df1, columns=columns)

print("Before cleaning the data, the shape of the dataset is ", df\_1.shape)

df\_1.drop\_duplicates(inplace=True)

print("After dropping duplicates, the shape of dataset is ", df\_1.shape)

if df\_1.isnull().sum().sum() > 0:

df\_1.dropna(inplace=True, how='any')

print("After dropping NULL values, the shape of dataset is ", df\_1.shape)

x = df\_1.iloc[:, 4:12]

y = df\_1[' loan\_status']

select = SelectKBest(score\_func=f\_classif, k=2)

fit = select.fit(x, y)

features = x.columns[fit.get\_support()]

print("The most correlated columns for loan\_status are ", features[0], " and ", features[1])

query = "drop table " + t\_name

cur.execute(query)

df.commit()

cur.close()

df.close()

path = "mysql://root:Nivash%4003@localhost/Nivash\_c"

engine = create\_engine(path)

df\_1.to\_sql(t\_name, con=engine, index=False)

print("File cleaned and appended in SQL")

return df\_1

**4.3 Script File 3**

Description:

This script visualizes the cleaned data using various plots. It includes line plots, bar plots, and pie charts to provide insights into the dataset.

**Code:**

from pymysql import connect

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

def visual(t\_name):

df = connect(host='localhost', user='root', password='Nivash@03', database='Nivash\_C')

cur = df.cursor()

query = "select \* from " + t\_name

cur.execute(query)

df1 = cur.fetchall()

columns = [desc[0] for desc in cur.description]

df\_1 = pd.DataFrame(df1, columns=columns)

print("Line Plot")

sns.lineplot(x=' no\_of\_dependents', y=' loan\_amount', data=df\_1)

plt.title("Loan amount based on no. of dependents")

plt.xlabel("No\_of\_dependents")

plt.ylabel('Loan\_amount')

plt.show()

print("Histogram")

    sns.histplot(df\_1[' loan\_amount'],bins=10,kde=True)

    plt.xlabel('Loan\_Amount')

    plt.ylabel('Count')

    plt.show()

print("Pie Chart")

\_count = df\_1[' no\_of\_dependents'].value\_counts()

plt.pie(\_count, labels=\_count.index, autopct='%1.1f%%')

plt.title('Dependents Distribution')

plt.show()

**5. Input Versatility with Error Handling and Exception Handling:**

The project is designed to handle various types of inputs and includes mechanisms for error and exception handling:

* script\_file\_1: Checks for duplicate table names in the database and prompts the user to enter a different name if a duplicate is found.
* script\_file\_2: Handles missing values by dropping rows with null values and removes duplicate entries to ensure data integrity.
* script\_file\_3: Utilizes warnings to suppress unnecessary warnings and focuses on generating clear and informative visualizations.

**6. Results and Outcomes**

The project successfully processes and cleans the data, stores it in a MySQL database, identifies significant features for predicting loan status, and provides insightful visualizations.

Key outcomes include:

* Identification of the most correlated features for loan status prediction.
* Clear visualizations showing relationships and distributions in the dataset.
* Improved data quality through the removal of duplicates and handling of missing values.

**7. Conclusion**

This project demonstrates a comprehensive approach to data processing, feature selection, and visualization using Python. By leveraging pandas, SQLAlchemy, scikit-learn, and seaborn, the project achieves efficient data handling and insightful analysis.