

Sprint 1

Sprints for AI Use Case For Demand Forecasting Using POS Transaction Data

Team

Name	Group Name
Zameer Ahmad Mir	Dynamo
Amjad Ali Malik	Dynamo
Tufail Irshad	Dynamo
Irfan Ahmad Mutoo	Dynamo
Asif Ahmad Najar	Dynamo

1) **Generate Synthetic POS Transaction Records Using Python**

Here's the complete script to generate synthetic POS transaction records with missing values:

```
import csv

import random

from faker import Faker

# Initialize Faker
fake = Faker()

# Define constants for the dataset
ITEMS = ['Laptop', 'Smartphone', 'Tablet', 'Headphones', 'Smartwatch', 'Camera', 'Printer', 'Monitor']
CURRENCIES = ['USD', 'EUR', 'GBP', 'JPY', 'AUD', 'CAD']
TRANSACTION_MODES = ['Credit Card', 'Debit Card', 'Cash', 'Mobile Payment']

# Probability of missing value
missing_value_probability = 0.1

# List to store the data
data = []

# Function to randomly assign missing values
def randomly_assign_missing_value(value):
    if random.random() < missing_value_probability:
        return None
    return value

# Generate 10,000 rows of data
for _ in range(10000):
    date_of_transaction = randomly_assign_missing_value(fake.date_this_decade())
    item_purchased = randomly_assign_missing_value(random.choice(ITEMS))
    unit_price = randomly_assign_missing_value(round(random.uniform(10.0, 1000.0), 2))
    tax = round(unit_price * 0.1, 2) if unit_price is not None else None # Assuming a 10% tax rate
```

```

transaction_amount = round(unit_price + tax, 2) if unit_price is not None and tax is not None else
None

transaction_mode = randomly_assign_missing_value(random.choice(TRANSACTION_MODES))

transaction_currency = randomly_assign_missing_value(random.choice(CURRENCIES))


row = [
    date_of_transaction,
    item_purchased,
    unit_price,
    tax,
    transaction_amount,
    transaction_mode,
    transaction_currency
]

data.append(row)

# Print the first 10 rows to see the data
def head(data, n=10):
    for row in data[:n]:
        print(row)

head(data)

# Open CSV file for writing
csv_file = 'pos_dataset_with_missing_values.csv'
with open(csv_file, mode='w', newline='') as file:
    writer = csv.writer(file)

    writer.writerow(['Date of Transaction', 'Item Purchased', 'Unit Price', 'Tax', 'Transaction Amount',
'Transaction Mode', 'Transaction Currency'])

    writer.writerows(data)

print(f'10,000 rows of POS data with missing values have been written to {csv_file}')

```

Explanation Of Code

This Python code generates synthetic point-of-sale (POS) data and writes it into a CSV file. Let's go through it step by step:

1. `!pip install faker`: This line is installing the Faker library using pip. Faker is a Python library that generates fake data. It's useful for generating test data or in this case, synthetic data for simulating a point-of-sale system.
2. `import csv`: This imports Python's built-in CSV module, which allows reading and writing CSV files.
3. `import random`: This imports Python's random module, which is used for generating random numbers.
4. `from faker import Faker`: This imports the Faker class from the faker module.
5. `fake = Faker()`: This creates an instance of the Faker class. We'll use this instance to generate fake data.
6. `ITEMS, CURRENCIES, and TRANSACTION_MODES`: These are lists containing possible items, currencies, and transaction modes respectively.
7. `csv_file = 'pos_dataset.csv'`: This specifies the name of the CSV file to which we'll write our generated data.
8. `with open(csv_file, mode='w', newline='') as file::` This opens the CSV file in write mode. `newline=""` is used to prevent extra blank lines being inserted between rows.
9. `writer = csv.writer(file)`: This creates a CSV writer object, which will be used to write data to the CSV file.
10. `writer.writerow([...])`: This writes the header row into the CSV file, specifying the column names.
11. The loop `for _ in range(10000)`: iterates 10,000 times, generating a row of data for each iteration.

Inside the loop:

- a. `date_of_transaction = fake.date_this_decade()`: This generates a fake date within the current decade.
 - b. `item_purchased = random.choice(ITEMS)`: This randomly selects an item from the ITEMS list.
 - c. `unit_price = round(random.uniform(10.0, 1000.0), 2)`: This generates a random unit price between \$10 and \$1000, rounded to 2 decimal places.
 - d. `tax = round(unit_price * 0.1, 2)`: This calculates a tax of 10% on the unit price.
 - e. `transaction_amount = round(unit_price + tax, 2)`: This calculates the total transaction amount by adding the unit price and tax.
 - f. `transaction_mode = random.choice(TRANSACTION_MODES)`: This randomly selects a transaction mode from the TRANSACTION_MODES list.
 - g. `transaction_currency = random.choice(CURRENCIES)`: This randomly selects a currency from the CURRENCIES list.
12. `writer.writerow([...])`: This writes the generated data as a row into the CSV file.
13. Finally, it prints a confirmation message indicating that the data has been written to the CSV file.

2) Introduction to Missing Values and Outliers in the Synthetic Data

1. **Missing Values:** Missing values are data entries that are not recorded or are otherwise unavailable. They can occur for various reasons, such as data entry errors, sensor malfunctions, or privacy concerns. Handling missing values is crucial because they can lead to biased estimates and reduced statistical power.
2. **Outliers:** Outliers are data points that significantly differ from other observations. They can occur due to measurement errors, data entry errors, or genuine anomalies. Outliers can skew and mislead the data analysis, leading to incorrect conclusions.
3. In synthetic data, missing values and outliers are often deliberately introduced to simulate real-world data imperfections. This helps in testing the robustness of data processing and analysis algorithms.

3) Synthesis, Manipulation, and Analysis of POS Transaction Data

Synthesis:

The script above generates synthetic POS transaction data with missing values.

Manipulation:

Data manipulation involves cleaning and transforming the data to make it suitable for analysis. This includes handling missing values, correcting errors, and formatting the data.

Analysis:

Data analysis involves exploring and interpreting the data to extract useful insights. This can include statistical analysis, data visualization, and applying machine learning models.

4) **Draft an Outline for the Preprocessing and Model Development**

Outline for Preprocessing and Model Development

1. Data Preprocessing:

- Handling Missing Values:
 - Drop rows or columns with excessive missing values.
 - Impute missing values using mean, median, mode, or other advanced methods.
- Handling Outliers:
 - Identify outliers using statistical methods or visualization.
 - Decide whether to remove or cap outliers.
- Data Transformation:
 - Normalize or standardize numerical features.
 - Encode categorical features using one-hot encoding or label encoding.
- Feature Engineering:
 - Create new features that may improve model performance.
 - Select important features based on statistical tests or feature importance scores.

2. Model Development:

- Split the Data:
 - Split the dataset into training and testing sets.
- Model Selection:
 - Choose appropriate machine learning models based on the problem (e.g., classification, regression).
- Model Training:
 - Train the models using the training data.
- Model Evaluation:
 - Evaluate the models using the testing data.
 - Use metrics such as accuracy, precision, recall, F1-score, or RMSE.

5) **Document the Data Generation and Initial Cleaning Process**

Data Generation Process:

1. Setup: Initialize the Faker library and define constants for the dataset.
2. Data Generation: Use loops and random functions to generate synthetic data for each transaction record.
3. Introduce Missing Values: Randomly replace certain values with None to simulate missing data.
4. Save Data: Write the generated data to a CSV file.

Initial Cleaning Process:

1. Load Data: Read the CSV file into a DataFrame using pandas.
2. Inspect Data: Use `df.info()`, `df.describe()`, and `df.head()` to inspect the data structure and summary statistics.
3. Check Missing Values: Use `df.isnull().sum()` to check for missing values in each column.
4. Visualize Data: Use histograms and bar plots to visualize the distribution of numerical and categorical features.
5. Handle Missing Values: Decide on a strategy to handle missing values (e.g., imputation, removal).
6. Handle Outliers: Identify and handle outliers based on visual inspection or statistical methods.

import pandas as pd

Read the CSV file into a DataFrame

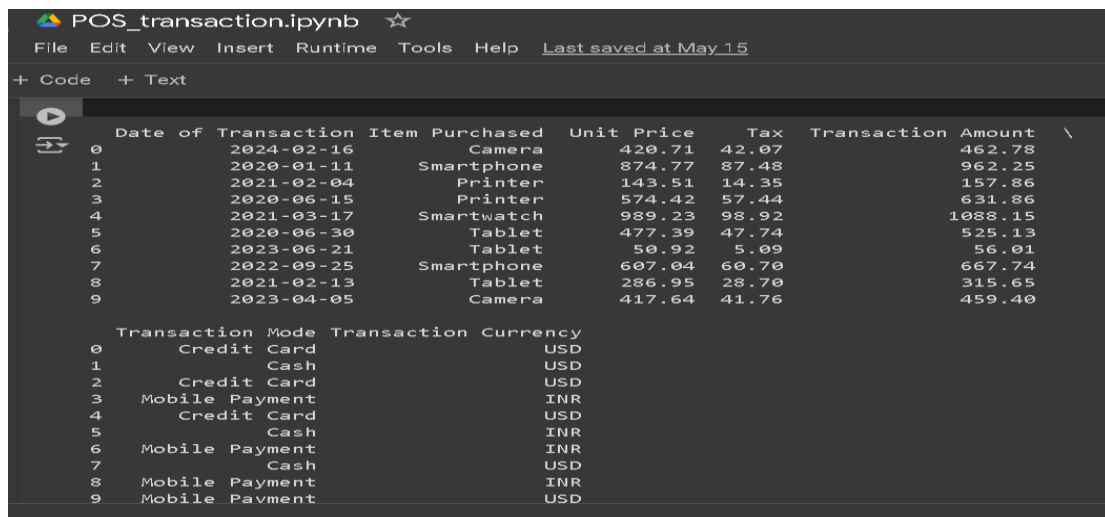
csv_file = 'pos_dataset.csv'

df = pd.read_csv(csv_file)

Print the first 10 rows of the DataFrame

print(df.head(10))

explain this code



	Date of Transaction	Item Purchased	Unit Price	Tax	Transaction Amount
0	2024-02-16	Camera	420.71	42.07	462.78
1	2020-01-11	Smartphone	874.77	87.48	962.25
2	2021-02-04	Printer	143.51	14.35	157.86
3	2020-06-15	Printer	574.42	57.44	631.86
4	2021-03-17	Smartwatch	989.23	98.92	1088.15
5	2020-06-30	Tablet	477.39	47.74	525.13
6	2023-06-21	Tablet	50.92	5.09	56.01
7	2022-09-25	Smartphone	607.04	60.70	667.74
8	2021-02-13	Tablet	286.95	28.70	315.65
9	2023-04-05	Camera	417.64	41.76	459.40

	Transaction Mode	Transaction Currency
0	Credit Card	USD
1	Cash	USD
2	Credit Card	USD
3	Mobile Payment	INR
4	Credit Card	USD
5	Cash	INR
6	Mobile Payment	INR
7	Cash	USD
8	Mobile Payment	INR
9	Mobile Payment	USD