

```

import kagglehub
import pandas as pd
import numpy as np
import os
from datetime import datetime
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Step 1: Download dataset from Kaggle
path = kagglehub.dataset_download("lakshmi25npathi/online-retail-dataset")
print("Path to dataset files:", path)

# Step 2: Confirm the Excel file name
data_file = os.path.join(path, "online_retail_II.xlsx")

# Step 3: Load Excel file
df = pd.read_excel(data_file)

# Step 4: Clean and prepare the data
df = df.dropna(subset=['Customer ID']) # Use correct column name
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])

# Filter only positive quantity and price
df = df[df['Quantity'] > 0]
df = df[df['Price'] > 0] # Use 'Price' instead of 'UnitPrice'

# Total transaction amount
df['TotalPrice'] = df['Quantity'] * df['Price']

# Define a snapshot date (1 day after last purchase)
snapshot_date = df['InvoiceDate'].max() + pd.Timedelta(days=1)

# Step 5: RFM (Recency, Frequency, Monetary) calculation
rfm = df.groupby('Customer ID').agg({
    'InvoiceDate': lambda x: (snapshot_date - x.max()).days, # Recency
    'Invoice': 'nunique', # Frequency
    'TotalPrice': 'sum' # Monetary
})

# Rename columns
rfm.rename(columns={
    'InvoiceDate': 'Recency',
    'Invoice': 'Frequency',
    'TotalPrice': 'Monetary'
}, inplace=True)

# Step 6: Prepare features and labels
X = rfm[['Recency', 'Frequency']]
y = rfm['Monetary']

# Step 7: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 8: Model training and evaluation
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

# Step 9: Results
print(f"RMSE: {rmse:.2f}")
print(f"R2 Score: {r2:.2f}")

📄 Path to dataset files: /kaggle/input/online-retail-dataset
RMSE: 5118.53
R2 Score: 0.06

```

