import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import os  
from email.message import EmailMessage  
import ssl  
import smtplib  
  
*#Bewegungsdaten für Mai aus der Excel-Datei lesen*df = pd.read\_excel(  
 r"C:\Users\kamal\OneDrive\Desktop\Bewerbung\Badawi\_Showcase\_Bargeld Einzahlung\Bewegungsdaten\Kontoumsaetze\_2023\_05.xlsx",  
 dtype={"Art": "category"},  
 sheet\_name="Fact\_Data",  
 usecols=["Zeitstempel", "Kundennummer", "Art", "Betrag"])  
  
*#Berechnung neuer Spalten*df["Zeitstempel"] = pd.to\_datetime(df['Zeitstempel'])  
df["Datum"] = df['Zeitstempel'].dt.date  
df["Monat"] = df['Zeitstempel'].dt.month  
df["Ist\_Einzahlung"] = df["Art"].map(  
 lambda x: True if x in ["Bargeldeinzahlung", "Gehalt", "Überweisung (Ein)"] else False)  
  
  
def ab\_eingang(art):  
 if art in ["Bargeldeinzahlung", "Gehalt", "Überweisung (Ein)"]:  
 return 1  
 else:  
 return -1  
  
def ist\_bargeld\_einzahlung(art):  
 if art in ["Bargeldeinzahlung", "Überweisung (Ein)"]:  
 return 1  
 else:  
 return 0  
  
df["Betrag\_Minus\_Plus"] = df["Betrag"] \* df["Art"].map(ab\_eingang)  
df["Bargeldeinzahlung"] = df["Betrag\_Minus\_Plus"].map(lambda x: 0 if float(x) <= 0 else float(x)) \* df["Art"].map(ist\_bargeld\_einzahlung)  
  
*# 1-Monatiger kum. Galdeingang*df["KUM\_GELDEINGANG"] = (df  
.set\_index('Zeitstempel')  
.groupby('Kundennummer')['Bargeldeinzahlung']  
.transform(lambda d: d.rolling('30D').sum())).reset\_index()["Bargeldeinzahlung"]  
  
  
*#Limit-Überschreitung Validierung*df["Limit\_Ueberschritten"] = df["KUM\_GELDEINGANG"].map(lambda x: x if x >=10000 else 0)  
  
*#print(df[["Art","Betrag","Ist\_Einzahlung","Betrag\_Minus\_Plus","Bargeldeinzahlung"]].tail(50))  
  
#Kunden Infos*kunden\_daten = pd.read\_excel(  
 r"C:\Users\kamal\OneDrive\Desktop\Bewerbung\Badawi\_Showcase\_Bargeld Einzahlung\Stammdaten\Kundendaten.xlsx",  
 sheet\_name="Kundendaten",  
 usecols=["Kundennummer", "Vorname","Nachname", "Geschlecht", "Strasse","Hausnummer","PLZ","Wohnort","email"])  
  
*#Daten nach überschrittenen Geldeingängen filtern*ueberschrittene\_geldeingaenge = df[df["Limit\_Ueberschritten"]>=10000].drop\_duplicates( subset=["Kundennummer"],  
 keep="last")  
  
*#Daten mit Kunden Infos joinen*ueberschrittene\_geldeingaenge\_infos = ueberschrittene\_geldeingaenge.merge(kunden\_daten,  
 on="Kundennummer",  
 how="left")  
  
*#BEstimmmte Spalten auswählen*ueberschrittene\_geldeingaenge\_emails = ueberschrittene\_geldeingaenge\_infos[["Kundennummer",  
 "Vorname",  
 "Nachname",  
 "Geschlecht",  
 "Strasse",  
 "Hausnummer",  
 "PLZ",  
 "Wohnort",  
 "email",  
 "KUM\_GELDEINGANG"]]  
  
  
*################################  
###############################  
##############################  
# Email Schicken*def send\_mail(empfaenger, geschlecht, name, kundennummer, betrag):  
 absender = r"kimo.utube.69@gmail.com"  
 kennwort = "qgsweepjcyioohyo"  
 empfaenger = empfaenger  
  
 grenze = 10000  
 betreff = f"MusterBank (Nachweispflicht bei Bareinzahlungen über {grenze:,.2f} €) Kundennummer: {kundennummer}"  
 anrede = "geehrter Herr" if geschlecht.upper() == "M" else "geehrte Frau"  
 inhalt = f""" Sehr {anrede} {name},  
  
 Seit dem 8. August 2021 gelten neue Regeln der Finanzaufsicht BaFin.   
 Bei Bargeld-Einzahlungen über 10.000 Euro müssen Banken und Sparkassen von Kunden einen sogenannten Herkunftsnachweis verlangen.  
  
 Sie haben in der letzten Zeit {betrag:,.2f} € auf Ihrem Konto eingezahlt. Aus diesem Grund sind Sie uns verpflichtet, einen Nachweis über die Herkunft des Gelds innerhalb von 2 Wochen einzureichen.  
  
 Im Online Banking:  
 1) Melden Sie sich mit Ihren Zugangsdaten an, um Ihren Nachweis bei Bareinzahlungen einreichen zu können.  
 2) Füllen Sie die Felder entsprechend aus.  
 3) Wählen Sie die getätigte Transaktion, z. B. "Bareinzahlung Euro", den Tag sowie den Betrag der Einzahlung aus.  
 4) Wählen Sie die Art des Herkunftsnachweises aus, z. B. "Barauszahlungsquittung" und laden Sie den entsprechenden Nachweis hoch.  
 5) Mit Klick auf "Weiter" sehen Sie eine Zusammenfassung Ihrer eingegebenen Daten.  
 6) Klicken Sie auf "Senden", um das Einreichen Ihres Nachweises freizugeben.  
  
  
 Aussagekräftige Belege nach Auskunft der BaFin:  
 ◉ Aktueller Kontoauszug des Kundenkontos bei einer anderen Bank.  
 ◉ Barauszahlungsquittungen einer anderen Bank oder Sparkasse.  
 ◉ Sparbuch, aus dem die Barauszahlung hervorgeht.  
 ◉ Verkaufs- und Rechnungsbelege (z. B. Belege zu einem Auto-, oder Warenverkauf, Verkauf von Dienstleistungen).  
 ◉ Letztwillige vom Nachlassgericht eröffnete Verfügungen, Erbschein oder ähnliche Erbnachweise.  
 ◉ Schenkungsverträge oder Schenkungsanzeigen.  
  
 Hinweis:  
 Die Auflistung der Belege ist nicht abschließend. In Einzelfällen ist zu prüfen, ob ein vorgelegter Nachweis ausreichend ist und die Herkunft plausibel dargelegt werden kann.  
  
 Mit freundlichen Grüßen  
  
 Ihre MusterBank  
 """  
  
 em = EmailMessage()  
  
 em["From"] = absender  
 em["To"] = empfaenger  
 em["subject"] = betreff  
 em.set\_content(inhalt)  
  
 context = ssl.create\_default\_context()  
  
 with smtplib.SMTP\_SSL("smtp.gmail.com", 465, context=context) as smtp:  
 smtp.login(absender, kennwort)  
 smtp.sendmail(absender, empfaenger, em.as\_string())  
  
*#Email an diese Kunden schicken*for index,i in enumerate(ueberschrittene\_geldeingaenge\_emails.to\_numpy()):  
 email = i[8]  
 geschlecht = i[3]  
 nachname = i[2]  
 kundennummer = i[0]  
 betrag = i[9]  
 send\_mail(email,geschlecht,nachname,kundennummer, betrag) if index %2 ==0 else 0  
  
*#heutiges Datum berechnen*heute = pd.to\_datetime('today').date()  
  
*#Ordner für die Visualisierungen erstellen*path = rf"C:\Users\kamal\OneDrive\Desktop\Bewerbung\Badawi\_Showcase\_Bargeld Einzahlung\Ergebnisdaten\{heute}"  
isExist = os.path.exists(path)  
if not isExist:  
 os.makedirs(path)  
  
  
*#Ergebnis in Excel speichern*ueberschrittene\_geldeingaenge\_emails.to\_excel(  
 path+rf"\Kontos mit ueberschrittenen Geldeingängen am {heute}.xlsx",  
 index=False)  
  
*#Kontoanalyse nach überschrittenen Konten filtern*umsaetze\_ueberschritten = df[df["Kundennummer"].isin(ueberschrittene\_geldeingaenge["Kundennummer"])]  
  
*#Kontoanalyse nur Zeitstempel, Kundennummer und Betrag*umsaetze\_ueberschritten = umsaetze\_ueberschritten[["Zeitstempel","Kundennummer","Betrag\_Minus\_Plus"]]  
  
*#Kontostände am Anfang des Monats laden und eine Dummy-Spalte für Zeitstempel erstellen*Kontostaende\_ueberschritten = pd.read\_excel(  
 r"C:\Users\kamal\OneDrive\Desktop\Bewerbung\Badawi\_Showcase\_Bargeld Einzahlung\Bewegungsdaten\Kontostaende\_2023\_05.xlsx",  
 sheet\_name="Kontostand",  
 usecols=["Kundennummer", "Kontostand"]).rename({"Kontostand" : "Betrag\_Minus\_Plus"},axis=1)  
  
Kontostaende\_ueberschritten["Zeitstempel"] = pd.to\_datetime("05.01.2023 00:00:01")  
  
*#Kontostände nach überschrittenen Konten filtern*Kontostaende\_ueberschritten= Kontostaende\_ueberschritten[Kontostaende\_ueberschritten["Kundennummer"].isin(ueberschrittene\_geldeingaenge["Kundennummer"])]  
  
  
*#Spalten neu anordnen*Kontostaende\_ueberschritten = Kontostaende\_ueberschritten.iloc[:,[2,0,1]]  
umsaetze\_ueberschritten = umsaetze\_ueberschritten.iloc[:,[0,1,2]]  
  
*#Kontostände und Kontoumsätze afür überschrittene aneinander fügen*datenanaylse\_ueberschritten = pd.concat([Kontostaende\_ueberschritten,umsaetze\_ueberschritten],  
 ignore\_index=True,  
 axis=0)  
  
  
*#Kum. Summe für die Datenanalyse*datenanaylse\_ueberschritten["Kontostand"] = datenanaylse\_ueberschritten.groupby(["Kundennummer"])["Betrag\_Minus\_Plus"].cumsum()  
datenanaylse\_ueberschritten["Datum"] = datenanaylse\_ueberschritten['Zeitstempel'].dt.date  
  
*#Ergebnis speichern*datenanaylse\_ueberschritten.to\_excel(  
 path+rf"\Datenanalyse für die überschrittenen Konten (Uhrzeitsweise) am {heute}.xlsx"  
 ,index=False)  
  
  
datenanaylse\_ueberschritten\_plot = datenanaylse\_ueberschritten.drop\_duplicates(subset=["Datum","Kundennummer"],  
 keep="last",  
 ignore\_index=True)  
datenanaylse\_ueberschritten\_plot = datenanaylse\_ueberschritten\_plot[["Kundennummer","Datum","Kontostand"]]  
  
*#Ergebnis speichern*datenanaylse\_ueberschritten\_plot.to\_excel(  
 path+rf"\Datenanalyse für die überschrittenen Konten (Tagesweise) am {heute}.xlsx",  
 index=False)  
  
*#Ordner für die Visualisierungen erstellen*path = rf"C:\Users\kamal\OneDrive\Desktop\Bewerbung\Badawi\_Showcase\_Bargeld Einzahlung\Ergebnisvisualisierungen\{heute}"  
isExist = os.path.exists(path)  
if not isExist:  
 os.makedirs(path)  
  
*#Daten für 5 zufällig ausgewählte Kunden visualisieren*anzahl\_u\_konten = len(datenanaylse\_ueberschritten\_plot["Kundennummer"].drop\_duplicates())  
n= 5  
kunden\_ueberschritten\_5 = datenanaylse\_ueberschritten\_plot["Kundennummer"].drop\_duplicates().sample(n=n).to\_numpy()  
data = datenanaylse\_ueberschritten\_plot[datenanaylse\_ueberschritten\_plot["Kundennummer"].isin(kunden\_ueberschritten\_5)]  
plt.figure(figsize=(14,7))  
sns.lineplot(x= data["Datum"],  
 y=data["Kontostand"],  
 hue=data["Kundennummer"])  
  
plt.legend()  
plt.title(f"Kontoanalyse am Tagesende\n (Sample = {n} von {anzahl\_u\_konten} überschrittenen Konten)")  
plt.savefig(path+rf"\Kontoanalyse (Sample = {n}) am {heute}")  
plt.show()