import pandas as pd

import os

from pathlib import Path

import shutil

from fuzzywuzzy import fuzz

from collections import defaultdict

import warnings

warnings.filterwarnings('ignore')

# Define paths

base\_path = r"C:\Users\USER\Desktop\ADONKO\_BITTERS"

output\_path = r"C:\Users\USER\Desktop\ADONKO\_BITTERS\_ANALYSIS"

# Create output directory

os.makedirs(output\_path, exist\_ok=True)

def get\_all\_excel\_files(base\_directory):

"""Get all Excel files from year subfolders"""

excel\_files = []

for year in range(2011, 2023): # 2011 to 2022

year\_folder = os.path.join(base\_directory, str(year))

if os.path.exists(year\_folder):

for file in os.listdir(year\_folder):

if file.endswith(('.xlsx', '.xls')):

excel\_files.append({

'year': year,

'file\_path': os.path.join(year\_folder, file),

'file\_name': file

})

return excel\_files

def extract\_data\_from\_file(file\_path):

"""Extract SSNIT numbers and names from Excel file"""

try:

# Try reading the Excel file

df = pd.read\_excel(file\_path)

# Assume Column A has SSNIT numbers and there's a name column

# Adjust column names based on your actual structure

if len(df.columns) >= 2:

# Get first two columns (assuming A=SSNIT, B=Names)

ssnit\_col = df.iloc[:, 0] # Column A

name\_col = df.iloc[:, 1] # Column B (assuming names are here)

# Clean and extract valid data

data = []

for i in range(len(df)):

ssnit = str(ssnit\_col.iloc[i]).strip() if pd.notna(ssnit\_col.iloc[i]) else None

name = str(name\_col.iloc[i]).strip() if pd.notna(name\_col.iloc[i]) else None

if ssnit and name and ssnit.lower() not in ['nan', 'none', '']:

data.append({

'ssnit': ssnit,

'name': name.upper() # Standardize to uppercase

})

return data

else:

print(f"Warning: {file\_path} has insufficient columns")

return []

except Exception as e:

print(f"Error reading {file\_path}: {str(e)}")

return []

def find\_similar\_names(names\_list, threshold=85):

"""Find similar names using fuzzy matching"""

similar\_groups = []

processed = set()

for i, name1 in enumerate(names\_list):

if name1 in processed:

continue

group = [name1]

processed.add(name1)

for j, name2 in enumerate(names\_list[i+1:], i+1):

if name2 in processed:

continue

similarity = fuzz.ratio(name1, name2)

if similarity >= threshold:

group.append(name2)

processed.add(name2)

if len(group) > 1:

similar\_groups.append(group)

return similar\_groups

# Main analysis

print("Starting SSNIT duplicate analysis...")

print("="\*50)

# Get all Excel files

excel\_files = get\_all\_excel\_files(base\_path)

print(f"Found {len(excel\_files)} Excel files across years 2011-2022")

# Extract all data

all\_data = []

file\_data\_map = {}

for file\_info in excel\_files:

print(f"Processing: {file\_info['year']}/{file\_info['file\_name']}")

data = extract\_data\_from\_file(file\_info['file\_path'])

for record in data:

record['year'] = file\_info['year']

record['file\_name'] = file\_info['file\_name']

record['file\_path'] = file\_info['file\_path']

all\_data.append(record)

file\_data\_map[f"{file\_info['year']}/{file\_info['file\_name']}"] = data

print(f"\nTotal records extracted: {len(all\_data)}")

# Create DataFrame

df\_all = pd.DataFrame(all\_data)

if len(df\_all) > 0:

# Group by name to find multiple SSNIT numbers

name\_ssnit\_groups = df\_all.groupby('name').agg({

'ssnit': lambda x: list(set(x)),

'year': lambda x: list(set(x)),

'file\_name': lambda x: list(set(x))

}).reset\_index()

# Also group by SSNIT to find multiple names (reverse check)

ssnit\_name\_groups = df\_all.groupby('ssnit').agg({

'name': lambda x: list(set(x)),

'year': lambda x: list(set(x)),

'file\_name': lambda x: list(set(x))

}).reset\_index()

# Filter names with multiple SSNIT numbers

duplicates = name\_ssnit\_groups[name\_ssnit\_groups['ssnit'].apply(len) > 1]

# Filter SSNIT numbers with multiple names

ssnit\_duplicates = ssnit\_name\_groups[ssnit\_name\_groups['name'].apply(len) > 1]

print(f"\nFound {len(duplicates)} names with multiple SSNIT numbers")

print(f"Found {len(ssnit\_duplicates)} SSNIT numbers with multiple names")

# Create detailed report for names with multiple SSNIT

detailed\_report = []

for \_, row in duplicates.iterrows():

name = row['name']

ssnit\_numbers = row['ssnit']

# Get detailed info for each SSNIT number

for ssnit in ssnit\_numbers:

matches = df\_all[(df\_all['name'] == name) & (df\_all['ssnit'] == ssnit)]

for \_, match in matches.iterrows():

detailed\_report.append({

'Name': name,

'SSNIT\_Number': ssnit,

'Year': match['year'],

'File\_Name': match['file\_name'],

'Total\_SSNIT\_for\_Name': len(ssnit\_numbers),

'All\_SSNIT\_Numbers': ', '.join(ssnit\_numbers)

})

# Create detailed report for SSNIT with multiple names

ssnit\_detailed\_report = []

for \_, row in ssnit\_duplicates.iterrows():

ssnit = row['ssnit']

names = row['name']

# Get detailed info for each name

for name in names:

matches = df\_all[(df\_all['ssnit'] == ssnit) & (df\_all['name'] == name)]

for \_, match in matches.iterrows():

ssnit\_detailed\_report.append({

'SSNIT\_Number': ssnit,

'Name': name,

'Year': match['year'],

'File\_Name': match['file\_name'],

'Total\_Names\_for\_SSNIT': len(names),

'All\_Names': ', '.join(names)

})

# Create summary report for names with multiple SSNIT

summary\_report = []

for \_, row in duplicates.iterrows():

summary\_report.append({

'Name': row['name'],

'Number\_of\_Different\_SSNIT': len(row['ssnit']),

'SSNIT\_Numbers': ', '.join(row['ssnit']),

'Years\_Found': ', '.join(map(str, sorted(row['year']))),

'Files\_Found': ', '.join(row['file\_name'])

})

# Create summary report for SSNIT with multiple names

ssnit\_summary\_report = []

for \_, row in ssnit\_duplicates.iterrows():

ssnit\_summary\_report.append({

'SSNIT\_Number': row['ssnit'],

'Number\_of\_Different\_Names': len(row['name']),

'Names': ', '.join(row['name']),

'Years\_Found': ', '.join(map(str, sorted(row['year']))),

'Files\_Found': ', '.join(row['file\_name'])

})

# Save reports

detailed\_df = pd.DataFrame(detailed\_report)

summary\_df = pd.DataFrame(summary\_report)

ssnit\_detailed\_df = pd.DataFrame(ssnit\_detailed\_report)

ssnit\_summary\_df = pd.DataFrame(ssnit\_summary\_report)

# Save to Excel

with pd.ExcelWriter(os.path.join(output\_path, 'SSNIT\_Duplicate\_Analysis.xlsx')) as writer:

summary\_df.to\_excel(writer, sheet\_name='Names\_Multiple\_SSNIT', index=False)

detailed\_df.to\_excel(writer, sheet\_name='Names\_Detailed', index=False)

ssnit\_summary\_df.to\_excel(writer, sheet\_name='SSNIT\_Multiple\_Names', index=False)

ssnit\_detailed\_df.to\_excel(writer, sheet\_name='SSNIT\_Detailed', index=False)

# Also save all data for reference

df\_all.to\_excel(writer, sheet\_name='All\_Data', index=False)

# Display summary

print("\n" + "="\*50)

print("DUPLICATE ANALYSIS SUMMARY")

print("="\*50)

print("\n--- NAMES WITH MULTIPLE SSNIT NUMBERS ---")

for \_, row in summary\_df.iterrows():

print(f"\nName: {row['Name']}")

print(f"Different SSNIT Numbers: {row['Number\_of\_Different\_SSNIT']}")

print(f"SSNIT Numbers: {row['SSNIT\_Numbers']}")

print(f"Years Found: {row['Years\_Found']}")

print("-" \* 30)

print("\n--- SSNIT NUMBERS WITH MULTIPLE NAMES ---")

for \_, row in ssnit\_summary\_df.iterrows():

print(f"\nSSNIT Number: {row['SSNIT\_Number']}")

print(f"Different Names: {row['Number\_of\_Different\_Names']}")

print(f"Names: {row['Names']}")

print(f"Years Found: {row['Years\_Found']}")

print("-" \* 30)

print(f"\nReports saved to: {output\_path}")

print("Files created:")

print("- SSNIT\_Duplicate\_Analysis.xlsx with sheets:")

print(" \* Names\_Multiple\_SSNIT (Names with different SSNIT numbers)")

print(" \* Names\_Detailed (Detailed breakdown)")

print(" \* SSNIT\_Multiple\_Names (SSNIT numbers with different names)")

print(" \* SSNIT\_Detailed (Detailed breakdown)")

print(" \* All\_Data (Complete dataset)")

# Additional fuzzy matching for similar names

print("\nChecking for similar names that might be the same person...")

unique\_names = df\_all['name'].unique().tolist()

similar\_name\_groups = find\_similar\_names(unique\_names, threshold=85)

if similar\_name\_groups:

print(f"Found {len(similar\_name\_groups)} groups of similar names:")

similar\_names\_report = []

for i, group in enumerate(similar\_name\_groups):

print(f"\nGroup {i+1}:")

for name in group:

ssnit\_nums = df\_all[df\_all['name'] == name]['ssnit'].unique()

print(f" - {name}: SSNIT {', '.join(ssnit\_nums)}")

similar\_names\_report.append({

'Group': i+1,

'Name': name,

'SSNIT\_Numbers': ', '.join(ssnit\_nums),

'Total\_SSNIT\_Count': len(ssnit\_nums)

})

# Save similar names report

similar\_df = pd.DataFrame(similar\_names\_report)

similar\_df.to\_excel(os.path.join(output\_path, 'Similar\_Names\_Analysis.xlsx'), index=False)

print(f"\nSimilar names report saved to: Similar\_Names\_Analysis.xlsx")

else:

print("No data found in the Excel files. Please check:")

print("1. File paths are correct")

print("2. Excel files contain data in the expected format")

print("3. Column A contains SSNIT numbers and Column B contains names")

print("\nAnalysis completed!")