# AI-Powered Underwriting System (Multimodal)

# Uses document analysis (OCR + NLP) and computer vision for property risk assessment

import pytesseract

import cv2

import json

import os

from transformers import pipeline

from sklearn.ensemble import RandomForestClassifier

import numpy as np

# -------------------- DOCUMENT ANALYSIS --------------------

def extract\_text\_from\_image(image\_path):

image = cv2.imread(image\_path)

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

text = pytesseract.image\_to\_string(gray)

return text

def analyze\_document(text):

summarizer = pipeline("summarization")

summary = summarizer(text[:1000], max\_length=150, min\_length=40, do\_sample=False)

return summary[0]['summary\_text']

# -------------------- COMPUTER VISION --------------------

def analyze\_property\_image(image\_path):

image = cv2.imread(image\_path)

# Placeholder hazard detection (e.g., damaged roof, pool detection)

height, width, \_ = image.shape

features = {

"pool\_detected": int(np.mean(image[:, :, 0]) > 120),

"roof\_damage": int(np.std(image[:, :, 1]) > 60),

}

return features

# -------------------- RISK ASSESSMENT --------------------

def compute\_risk\_score(doc\_summary, cv\_features):

keywords = ["fire hazard", "flood zone", "asbestos", "foundation issues"]

risk\_points = sum(kw in doc\_summary.lower() for kw in keywords)

total\_risk = risk\_points + sum(cv\_features.values())

risk\_level = "High" if total\_risk > 3 else "Medium" if total\_risk > 1 else "Low"

return risk\_level

# -------------------- MAIN WORKFLOW --------------------

def process\_underwriting(doc\_image\_path, property\_image\_path):

print("Extracting text from document...")

doc\_text = extract\_text\_from\_image(doc\_image\_path)

print("Summarizing key document points...")

doc\_summary = analyze\_document(doc\_text)

print("Analyzing property image...")

cv\_features = analyze\_property\_image(property\_image\_path)

print("Assessing risk...")

risk = compute\_risk\_score(doc\_summary, cv\_features)

report = {

"Document Summary": doc\_summary,

"Property Features": cv\_features,

"Risk Assessment": risk

}

return report

# -------------------- TESTING --------------------

if \_\_name\_\_ == "\_\_main\_\_":

DOC\_IMAGE = "sample\_report\_image.jpg" # Add your scanned document path

PROPERTY\_IMAGE = "property\_photo.jpg" # Add your property image path

result = process\_underwriting(DOC\_IMAGE, PROPERTY\_IMAGE)

print("\n--- Underwriting Report ---")

print(json.dumps(result, indent=2))