Report on AI code fixing

This report focuses on how various AI tools rectify errors in a given broken code. The tools used were ChatGPT and Copiolet.

Prompt: Fix the code

# Broken code:

import numpy as np

import pandas as pd

import random

def generate\_random\_number(min\_num, max\_num):

    num = random.randint(min\_num, max\_num)

    print("Random number is: " + num)

def calc\_average(num\_list):

    total = sum(num\_list)

    return total / lenght(num\_list)

def check\_prime(start, end):

    prime\_list = []

    for i in range(start, end):

        if i % 2 == 0:

            prime\_list.append(i)

    return prime\_list

def load\_data(filepath):

    data = pd.read\_csv(filepath)

    return data

def main():

    num\_list = [10, 20, 30, "forty", 50]

    print("The average is: ", calc\_average(num\_list))

    print("Prime numbers: ", check\_prime(1, 10))

    file\_path = "data.csv"

    data = load\_data(file\_path)

    print("Data loaded: ", data)

    random\_num = generate\_random\_number(1, 100)

    print("Generated Random Number: ", random\_num)

    try:

        print("Result of division: ", 10 / 0)

    except ZeroDivisionError:

        print("Can't divide by zero")

    numbers = [x for x in range(100) if x % 3 == 0 and x % 5 == 0]

    print("Numbers divisible by 3 and 5 are: ", numbers)

    undefined\_function\_call()

main()

# Copiolet response:

import numpy as np

import pandas as pd

import random

def generate\_random\_number(min\_num, max\_num):

num = random.randint(min\_num, max\_num)

print("Random number is: " + str(num)) # Convert number to string before concatenation

def calc\_average(num\_list):

# Filter out non-numeric values

num\_list = [num for num in num\_list if isinstance(num, (int, float))]

total = sum(num\_list)

return total / len(num\_list) # Correct the spelling of 'length'

def check\_prime(start, end):

prime\_list = []

for i in range(start, end):

if i > 1: # Correct prime number check

for j in range(2, int(i\*\*0.5) + 1):

if (i % j) == 0:

break

else:

prime\_list.append(i)

return prime\_list

def load\_data(filepath):

try:

data = pd.read\_csv(filepath)

return data

except FileNotFoundError:

print("File not found. Please check the file path.")

return None

def main():

num\_list = [10, 20, 30, 50] # Removed "forty" to avoid errors in calculation

print("The average is: ", calc\_average(num\_list))

print("Prime numbers: ", check\_prime(1, 10))

file\_path = "data.csv"

data = load\_data(file\_path)

if data is not None:

print("Data loaded: \n", data)

random\_num = generate\_random\_number(1, 100)

print("Generated Random Number: ", random\_num)

try:

print("Result of division: ", 10 / 0)

except ZeroDivisionError:

print("Can't divide by zero")

numbers = [x for x in range(100) if x % 3 == 0 and x % 5 == 0]

print("Numbers divisible by 3 and 5 are: ", numbers)

# Commented out undefined function call to prevent error

# undefined\_function\_call()

main()

Suggestions Given:

Here are the fixes I made:

1. Converted the random number to a string before concatenating it in generate\_random\_number.
2. Filtered out non-numeric values in calc\_average and fixed the spelling of length.
3. Corrected the prime number check logic in check\_prime.
4. Added error handling for file loading in load\_data.
5. Removed the non-numeric value "forty" from num\_list.
6. Commented out the undefined\_function\_call() to prevent an error.

# ChatGPT response

Code it gave:

import numpy as np

import pandas as pd

import random

def generate\_random\_number(min\_num, max\_num):

num = random.randint(min\_num, max\_num)

return num

def calc\_average(num\_list):

num\_list = [x for x in num\_list if isinstance(x, (int, float))]

total = sum(num\_list)

return total / len(num\_list)

def check\_prime(start, end):

prime\_list = []

for i in range(start, end):

if i > 1:

for j in range(2, int(i\*\*0.5) + 1):

if i % j == 0:

break

else:

prime\_list.append(i)

return prime\_list

def load\_data(filepath):

data = pd.read\_csv(filepath)

return data

def main():

num\_list = [10, 20, 30, "forty", 50]

print("The average is: ", calc\_average(num\_list))

print("Prime numbers: ", check\_prime(1, 10))

file\_path = "data.csv"

data = load\_data(file\_path)

print("Data loaded: ", data)

random\_num = generate\_random\_number(1, 100)

print("Generated Random Number: ", random\_num)

try:

print("Result of division: ", 10 / 0)

except ZeroDivisionError:

print("Can't divide by zero")

numbers = [x for x in range(100) if x % 3 == 0 and x % 5 == 0]

print("Numbers divisible by 3 and 5 are: ", numbers)

main()

Suggestions made:

 **In the generate\_random\_number function**, you're trying to concatenate a string with an integer. This will cause an error. You need to convert the integer num to a string before concatenating it.

python

CopyEdit

print("Random number is: " + str(num))

 **In the calc\_average function**, there's a typo in the word length. It should be len(num\_list) instead of lenght(num\_list). Also, since your num\_list contains a string ("forty"), this will cause an error during the sum() operation, as it can only sum numerical values. You should filter out non-numeric values.

python

CopyEdit

return total / len(num\_list) # Fix typo

Before summing, filter out the non-numeric values like this:

python

CopyEdit

num\_list = [x for x in num\_list if isinstance(x, (int, float))]

 **In the check\_prime function**, you are checking if a number is divisible by 2 to determine if it is prime. This isn't correct because primes are numbers greater than 1 that are divisible only by 1 and themselves. You need a better prime-checking logic.

Here's a corrected version:

python

CopyEdit

def check\_prime(start, end):

prime\_list = []

for i in range(start, end):

if i > 1:

for j in range(2, int(i\*\*0.5) + 1):

if i % j == 0:

break

else:

prime\_list.append(i)

return prime\_list

 **In the main function**, you are trying to print the return value of generate\_random\_number(), but it doesn't return anything; it just prints. To fix this, you can modify the function to return the number and then print it in main():

python

CopyEdit

def generate\_random\_number(min\_num, max\_num):

num = random.randint(min\_num, max\_num)

return num # Instead of print

 **In the main function**, you're calling undefined\_function\_call(), but this function doesn't exist, so it will raise an error. You should either define the function or remove the call.

 **ZeroDivisionError** is caught properly using the try-except block, so this is fine.

The code given by chatgpt when run in google colab gave an error, while that of copiolet ran smoothly. Copiolet is better for fixing codes in my optinion