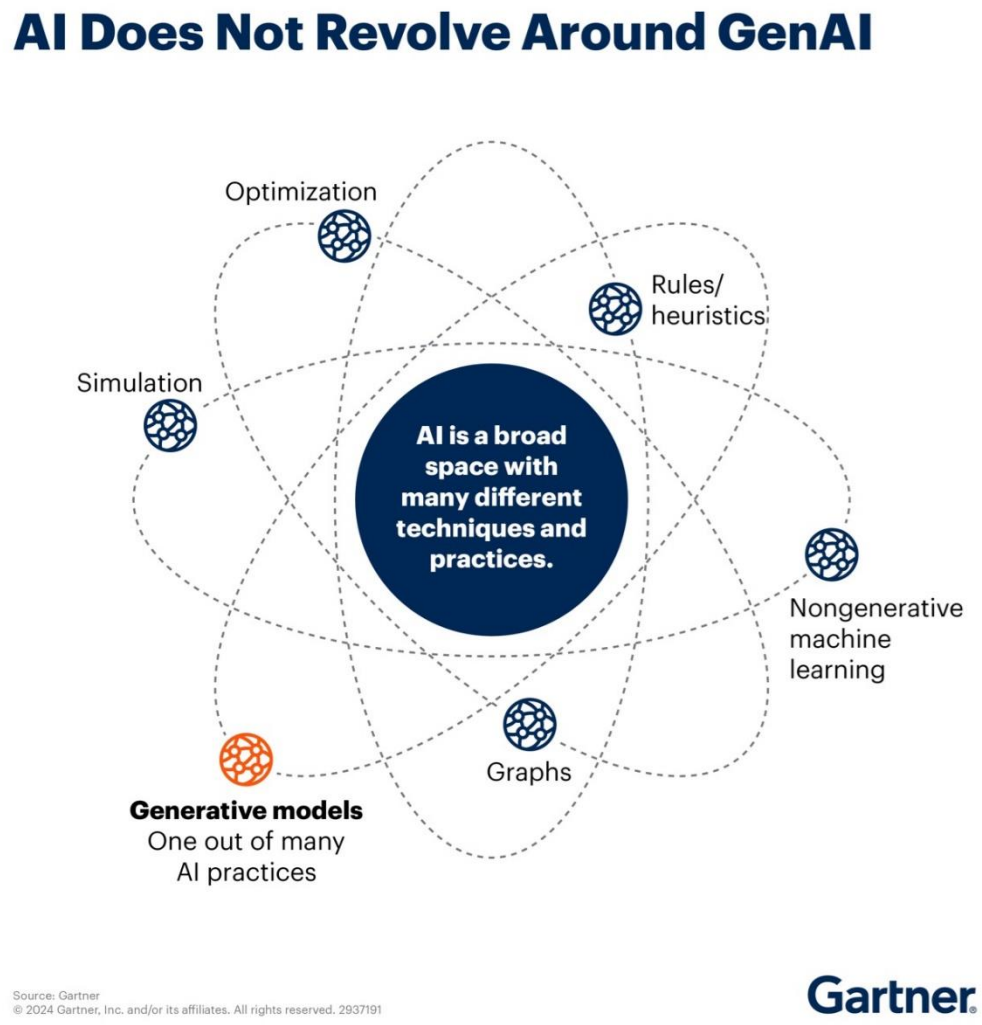
Challenge January 2025

Christmas Word Search

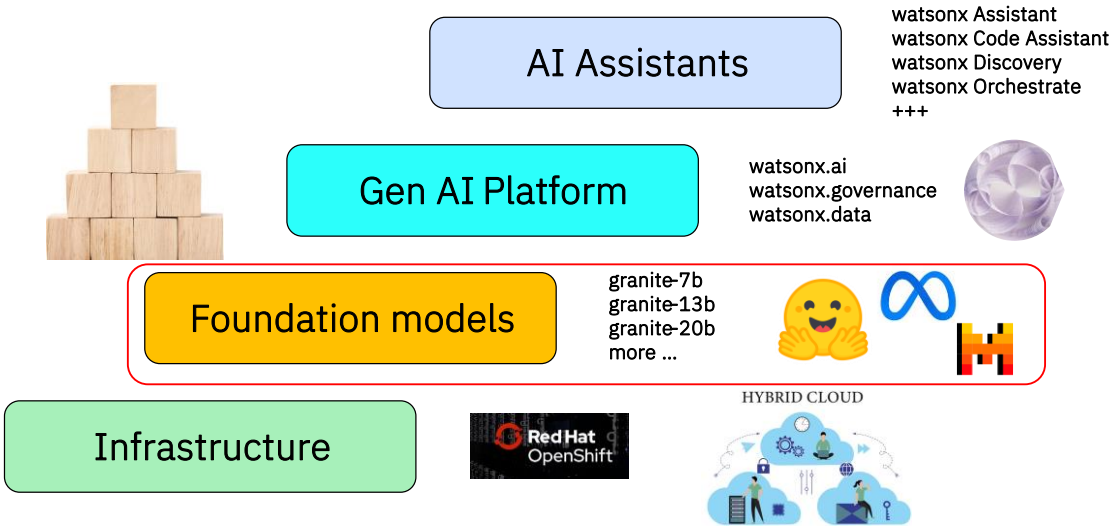
A solution with IBM watsonx AI Llama and watsonx code assistant
by Alex Fleischer afleischer@fr.ibm.com

After solving this puzzle a few days ago with optimization (which is a fundamental AI technique as Gartner wrote) , I gave it a try with IBM watsonx generative AI platform and watsonx code assistant



Watsonx.AI is IBM Generative AI platform and watsonx code assistant is IBM code AI Assistant. (“Code smarter not harder”)

The generative AI Stack



In 3 steps , 3 prompts and NOT A SINGLE LINE OF CODE, I was able to count “XMAS”.

Prompt1 in watsonx.ai llama3

Out of the 10 times 10 matrix
"MMMSXXMASMMSAMXMSMSAAMXSXMAAMMMSAMASMSMX
XMASAMXAMMXXAMMXXAMASMSMSASXSSSAXAMASAAAMA
MMMXMMMMMXMXAXMASX"
write a python program which generates a list of strings with all the 10 rows, all the 10 columns,
the 19 main diagonals and the 19 secondary diagonals

Prompt2 in watsonx code assistant in VSCode

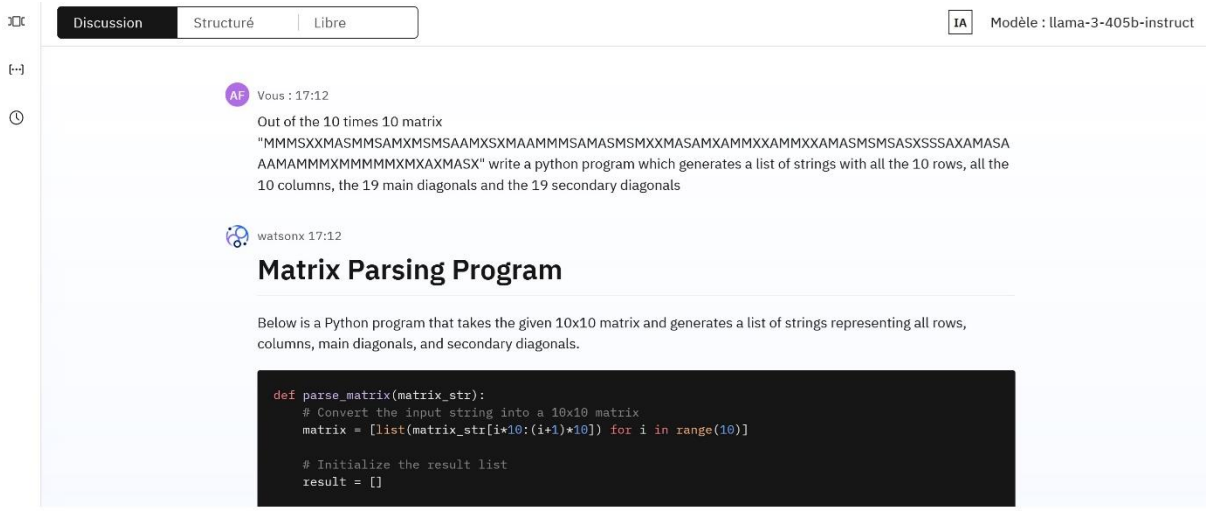
result is a list of strings , write python code to append to result all the reversed strings out of the ones already in result

Prompt3 in watsonx code assistant in VSCode

"result" is a list of strings, write a python program
that counts how many times we can read "XMAS" in those strings

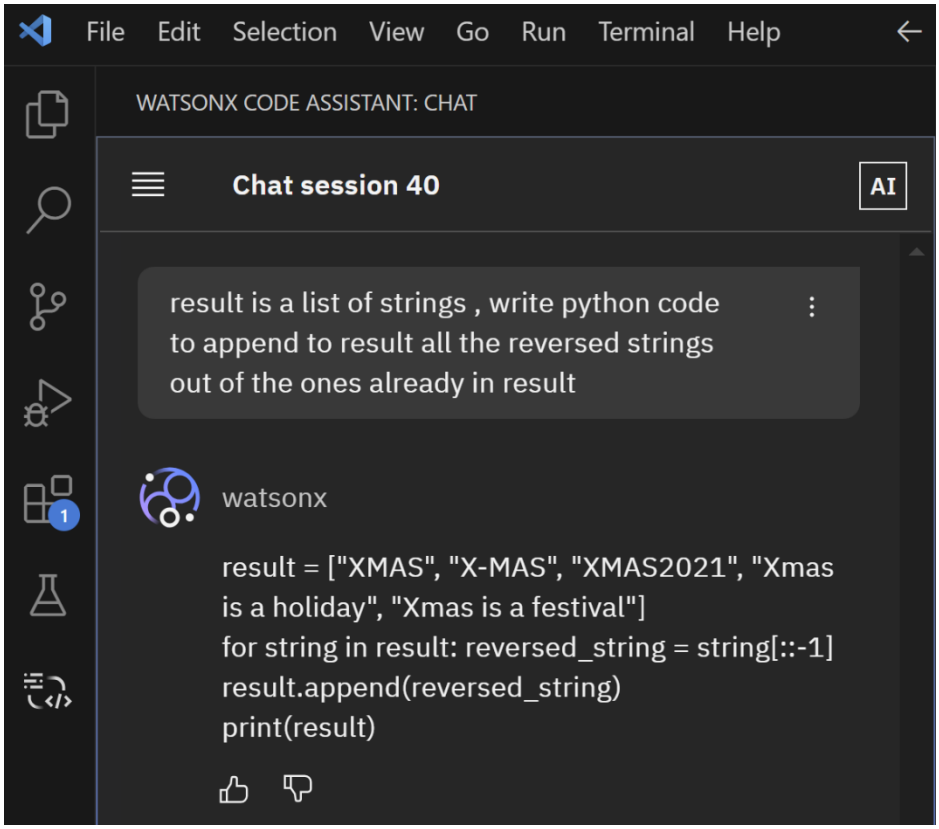
Let’s review 3 steps with more details:

Step 1: Use llama3 from Meta within watsonx.ai through a discussion

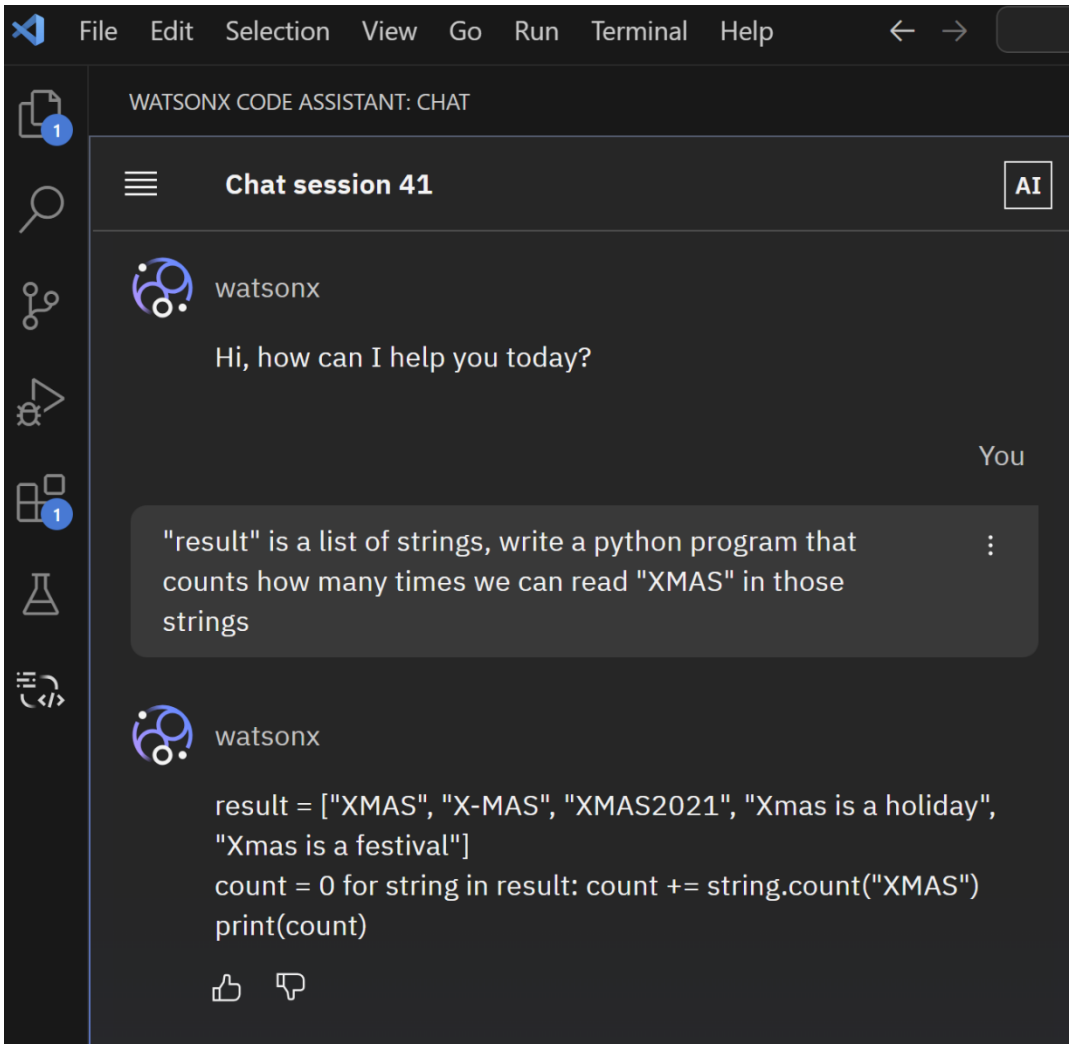


Copy the python code to VSCode

Step 2: Use IBM watsonx code assistant within Microsoft VSCode in order to read backwards too.



Step 3: Use IBM watsonx code assistant within Microsoft VSCode in order to count how many”XMAS” we have.



So in a nutshell I wrote 3 prompts and no code. Here is the python code with the 3 prompts:

"""

With llama 405 B in watsonx ai I used

Out of the 10 times 10 matrix "MMMSXXMASMMSAMXMSMSAAMXSXMAAMMMSAMASMSMXXMASAMXAMMXXAMMXXAMASMSMSASXSSSAXAMASAAAMAMMMXMMMMMXMXAXMASX"

write a python program which generates a list of strings with all the 10 rows, all the 10 columns,
 the 19 main diagonals and the 19 secondary diagonals

"""

```
def parse_matrix(matrix_str):
    # Convert the input string into a 10x10 matrix
    matrix = [list(matrix_str[i*10:(i+1)*10]) for i in range(10)]

    # Initialize the result list
    result = []

    # Rows
    for row in matrix:
        result.append(''.join(row))

    # Columns
    for col_idx in range(10):
        result.append(''.join([row[col_idx] for row in matrix]))

    # Main Diagonals
    for k in range(-9, 10):
        diagonal = []
        for i in range(10):
            j = i + k
            if 0 <= j < 10:
                diagonal.append(matrix[i][j])
        result.append(''.join(diagonal))

    # Secondary Diagonals
    for k in range(-9, 10):
        diagonal = []
        for i in range(10):
            j = 9 - i + k
            if 0 <= j < 10:
                diagonal.append(matrix[i][j])
        result.append(''.join(diagonal))

    return result
```

```
matrix_str =
"MMMSXXMASMMSAMXMSMSAAMXSXMAAMMMSAMASMSMX
XMASAMXAMMXXAMMXXAMASMSMSASXSSSAXAMASAAAMA
MMMXMMMMMMXMXAXMASX"
result = parse_matrix(matrix_str)
for string in result:
    print(string)

"""
result is a list of strings , write python code to append to result all
 the reversed strings out of the ones already in result
"""

# Assisted by watsonx Code Assistant
#result = ["hello", "world", "IBM", "watsonx"]

# Create a list of reversed strings
reversed_strings = [string[::-1] for string in result]

# Append the reversed strings to the list "result"
result.extend(reversed_strings)

# Print the list "result"
print(result)

"""
"result" is a list of strings, write a python program
that counts how many times we can read "XMAS" in those strings
"""

# Assisted by watsonx Code Assistant

#result = ["XMAS", "X-MAS", "XMAS2021", "Xmas is a holiday",
"Xmas is a festival"]

count = 0
for string in result:
    count += string.count("XMAS")
```

```
print(count)

"""
which gives
18
"""
```