

Applications of GenAI for Business Analytics

1. Introduction

In today's fast-paced and increasingly digital business landscape, Generative Artificial Intelligence (GenAI) has the potential to be a transformative force in operations across various industries. GenAI could disrupt traditional business operations, enabling companies to operate more efficiently and effectively. It has begun to redefine how businesses strategize and execute their operational objectives, paving the way for increased productivity, cost savings, and a competitive edge in the market.

This document describes some of the key use cases of GenAI in operations, including the following:

- **Data Pipelines:** GenAI can automate data entry tasks, efficiently detecting data entry errors. It can be leveraged to assist in data cleaning by automatically identifying errors in datasets that may be challenging for humans to detect.
- **Data Analysis:** GenAI can assist in data analysis, suggesting appropriate models for learning from datasets and offering capabilities to visualize data effectively.
- **Optimization:** GenAI can formulate sophisticated optimization problems using mathematical modeling. Traditionally a task reserved for optimization experts, GenAI can identify suitable models, set up variables, constraints, and objectives of the model, and, by integrating with mathematical optimization software, directly optimize these models.
- **Software Engineering:** While GenAI is perhaps best known for generating natural language text, it is nearly as proficient at generating software code. Primary uses include streamlining the generation of common code structures, debugging software, creating test examples, and assisting in looking up functions and libraries.

Central to these use cases is the emerging integration of GenAI with other business software tools. This integration presents an avenue for generating tremendous value by streamlining current business processes:

- **Spreadsheet integration:** Plugins currently allow for GenAI tools to be used as a plugin within spreadsheets (Khadka, 2023). GenAI can then be directly leveraged for data cleaning and entry.
- **Optimization:** GenAI can leverage the full power of Wolfram's Mathematica (Wolfram, 2023), enabling it to solve sophisticated optimization problems arising in operations. It is likely GenAI is integrated with other optimization tools such as linear programming, integer programming and other mathematical programming tools.
- **Development Frameworks:** GenAI can be integrated into software development frameworks. For example, CoPilot, a tool integrating ChatGPT into Visual Studio (Github Copilot, 2023) Like integrating GenAI into word processing software, it can suggest code, look up functions, and streamline the software engineering process.

1.1 Document Organization

The document is organized as follows. In Section 2, we discuss how data pipelines can leverage GenAI. In Section 3 we discuss GenAI's uses for optimization. Section 4 discusses GenAI's uses in software engineering. Section 3 will assume background in optimization, especially linear programming, and Section 4 assumes some prior knowledge in coding. Section 5 overviews the limitations and shortcomings of using GenAI.

2. Streamlining Data Pipelines and Data Analysis

GenAI is a powerful tool that can significantly streamline data entry, cleaning, and analysis processes. Here are some common use cases, along with illustrative examples.

Data Entry Automation

- **Automated Information Extraction:** As documents are generated, GenAI can extract pertinent information and automatically load it into data logs or databases (Horse, 2023). This reduces manual data entry and minimizes errors.
- **Script and Macro Generation:** GenAI can assist in creating scripts or macros that automate repetitive data entry tasks (Dhana, 2023).
 - **Example:** GenAI can help write a script that automatically populates specific fields in a spreadsheet based on predefined rules, saving time, and reducing manual input.

Data Cleaning and Validation

- **Format Verification:** GenAI can validate the format of data entries, such as checking if an email address is in the correct format, verifying the validity of a date, or confirming that a credit card number passes the Luhn check (Zhang, 2023).
- **Standardized Formatting:** GenAI can assist in formatting data according to specific standards, ensuring consistency across datasets.
 - **Example:** It can convert dates to a uniform format, capitalize names properly, or format phone numbers consistently.
- **Error Identification and Correction Suggestions:** GenAI can help identify potential errors in your data, such as outliers or inconsistencies, and suggest possible corrections (Elfman, 2023). This ensures the integrity and reliability of your data.

Integration with Online Spreadsheet Tools

- **Real-Time Data Interaction with Spreadsheet Software:** By integrating with spreadsheet software, an online spreadsheet tool, GenAI can directly interact with real data. This integration allows for dynamic data manipulation and analysis without leaving the spreadsheet environment.
 - **Example:** ChatGPT can be set up to automatically update a sales report in Google Sheets whenever new sales data is entered, ensuring that the report is always current and accurate (Khadka, 2023).

We next illustrate some examples for how it can be used. In the examples, ChatGPT is used along with integration with Google Sheets. By using this integration, ChatGPT can directly interact with real data. There are currently several ways to link Google Sheets to ChatGPT. ChatGPT Plus, a paid premium service, directly offers integration using their apps. This is the product used for these examples. Additionally, there are applications offered that integrate ChatGPT directly into Google Sheets. For example, SheetGPT (SheetGPT, 2023). Similar products are available for Excel. For example, ExcelBrew (ExcelBrew, 2023).

Example on Improving Data Entry and Cleaning

We consider several use cases of in the data pipeline where GenAI can be of use. The running example will be on the following data set.

| | A | B | C | D | E | F | G |
|----|-------|---------|-------|---------------|---------|-----------|---|
| 1 | Sales | Date | Promo | SchoolHoliday | Month | DayOfWeek | |
| 2 | 5530 | 1/2/13 | 0 | 1 | January | Wednesday | |
| 3 | 4327 | 1/3/13 | 0 | 1 | January | Thursday | |
| 4 | 4486 | 1/4/13 | 0 | 1 | January | Friday | |
| 5 | 4997 | 1/5/13 | 0 | 1 | January | Saturday | |
| 6 | 7176 | 1/7/13 | 1 | 1 | January | Monday | |
| 7 | 5580 | 1/8/13 | 1 | 1 | January | Tuesday | |
| 8 | 5471 | 1/9/13 | 1 | 1 | January | Wednesday | |
| 9 | 4892 | 1/10/13 | 1 | 1 | January | Thursday | |
| 10 | 4881 | 1/11/13 | 1 | 1 | January | Friday | |
| 11 | 4952 | 1/12/13 | 0 | 0 | January | Saturday | |
| 12 | 4717 | 1/14/13 | 0 | 0 | January | Monday | |
| 13 | 3900 | 1/15/13 | 0 | 0 | January | Tuesday | |
| 14 | 4008 | 1/16/13 | 0 | 0 | January | Wednesday | |
| 15 | 4044 | 1/17/13 | 0 | 0 | January | Thursday | |
| 16 | 4127 | 1/18/13 | 0 | 0 | January | Friday | |
| 17 | 5182 | 1/19/13 | 0 | 0 | January | Saturday | |
| 18 | 5394 | 1/21/13 | 1 | 0 | January | Monday | |
| 19 | 5720 | 1/22/13 | 1 | 0 | January | Tuesday | |
| 20 | 5578 | 1/23/13 | 1 | 0 | January | Wednesday | |

This data set contains sales data from Rossman stores. Rossman is a drug store. The data set contains information on the total number of sales on a given date, whether it was a school holiday on the date, and whether a promotion was run on the day.

This data set was uploaded into a Google Sheet and then ChatGPT was used to read the data set via a ChatGPT plugin. This can be read by ChatGPT by sharing a link to the GoogleSheets data.

Used Access Google Sheet

The Google Sheets document you provided contains a table with the following columns:

1. Index
2. Sales
3. Date
4. Promo
5. SchoolHoliday
6. Month
7. DayOfWeek

Here are the first few rows of data:

| Index | Sales | Date | Promo | SchoolHoliday | Month | DayOfWeek |
|-------|-------|--------|-------|---------------|---------|-----------|
| 2 | 5530 | 1/2/13 | 0 | 1 | January | Wednesday |
| 3 | 4327 | 1/3/13 | 0 | 1 | January | Thursday |
| 4 | 4486 | 1/4/13 | 0 | 1 | January | Friday |
| 5 | 4997 | 1/5/13 | 0 | 1 | January | Saturday |

Please let me know what you would like to do with this data.

Consider the following example on catching errors in the data. In this case, an error was introduced into the data sets. In line 7, the random text “sa” was put into the number of sales on 1/8/2013.

| 1 | Sales | Date | Promo | SchoolHoliday | Month | DayOfWeek |
|---|-------|--------|-------|---------------|---------|-----------|
| 2 | 5530 | 1/2/13 | 0 | 1 | January | Wednesday |
| 3 | 4327 | 1/3/13 | 0 | 1 | January | Thursday |
| 4 | 4486 | 1/4/13 | 0 | 1 | January | Friday |
| 5 | 4997 | 1/5/13 | 0 | 1 | January | Saturday |
| 6 | 7176 | 1/7/13 | 1 | 1 | January | Monday |
| 7 | sa | 1/8/13 | 1 | 1 | January | Tuesday |
| 8 | 5471 | 1/9/13 | 1 | 1 | January | Wednesday |

ChatGPT was then asked to determine if there were any errors in the entire data set over hundreds of days. ChatGPT was able to detect this error and states that “sa” is not numerical and this is likely a mistake for a sales column. Likely there was an error on data entry and this should be checked. See the following interaction that shows how simple it is to work with ChatGPT.



Are there any days where the sales data entered has an error

https://docs.google.com/spreadsheets/d/1Q2U9VeTAA19nHpfDLV27jYeujZ7LYiYw7DIZGs_Ausp=sharing



Used **Access Google Sheet**



From the data provided, it appears there is an error in the sales data for the date "1/8/13". The sales data is recorded as "sa" which is not a valid numerical value for sales.

Please review this entry in your Google Sheets document and correct it to a valid numerical value. Let me know if you need further assistance.

This example illustrates how GenAI can be used for data cleaning without any sophisticated input from the user regarding what errors may exist in the data set.

Next ChatGPT was able to do some simple data analysis. For example, asking for the total number of sales in the month of June can be automatically calculated using ChatGPT.

Another example of how it can be used is in determining the right function to use. For instance, say a user is interested in a formula to input into the excel sheet to compute the total number of sales on a given day. One could search documentation for this from the software company (in

this case, regarding Google Sheets). Alternatively, ChatGPT can be directly asked to produce such a formula, streamlining the process. When doing this, ChatGPT outputs the following results.



what formula should i use to calculate total sales on saturdays



To calculate the total sales on Saturdays in Google Sheets, you can use the `SUMIF` function. This function adds all numbers in a range of cells based on a given condition.

Here's the formula you can use:

```
=SUMIF(range_criteria, "Saturday", range_sum)
```

In this formula:

- `range_criteria` is the range of cells where the criteria (in this case, the day of the week) are defined. For example, if your days of the week are in column F from row 2 to 100, this would be `F2:F100`.
- `"Saturday"` is the criteria. The function will look for this value in the `range_criteria`.
- `range_sum` is the range of cells to sum if the criteria is met. For example, if your sales data is in column B from row 2 to 100, this would be `B2:B100`.

So, if your days of the week are in column F and your sales data is in column B, the formula would look like this:

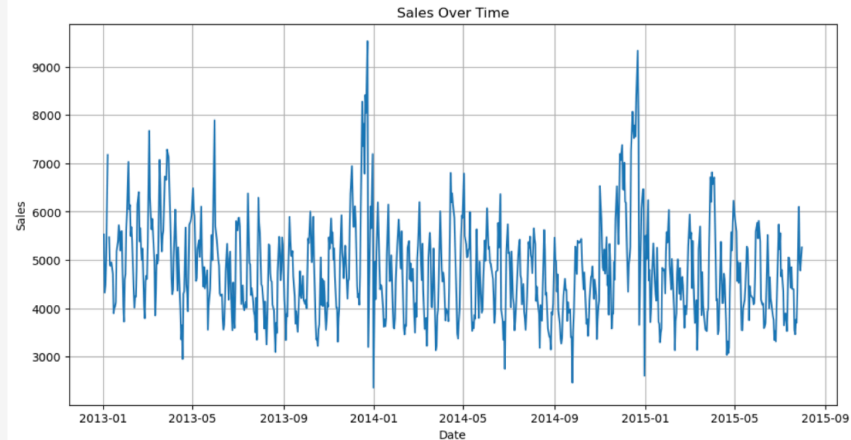
```
=SUMIF(F2:F100, "Saturday", B2:B100)
```

You can adjust the ranges according to your actual data. Just type this formula into a cell, and it will calculate the total sales on Saturdays.

Here one can be seen that it identifies the correct function to use. Moreover, it describes the inputs to the function and suggests the correct inputs based on the given Google Sheet file.

Finally, GenAI can further streamline data analysis using visualization. For example, it may be of interest to plot the sales data over time. Instead of directly coding this, one can leverage GenAI to do it directly. For instance, on the above data set one can ask ChatGPT to plot the data and it will produce the following result.

Here is the plot of sales over time from your data set:



Overall, GenAI holds the promise of streamlining the data pipeline. From entry to cleaning to visualization, it offers tools to quickly get the task done.

3. Optimization

Optimization is a fundamental tool used in operations. This section will overview use cases of GenAI in optimization. This section assumes some background in optimization, especially linear programming. The section shows how GenAI can generate sophisticated, but fairly standardized tasks within optimization.

Traditionally, an optimization expert works with domain experts to understand a real work decision problem. This problem is then modeled mathematically where there is a set of variables corresponding to decisions to be made, a set of constraints on the variables corresponding to the problems underlying constraints and an objective that assigns a score to a set of decisions. Critical, is for an expert to define the variables, constraints and objective.

Remarkably, GenAI is emerging as a key tool for modeling and solving optimization problems. Showcasing how it can be used, here several examples are illustrated. In these examples, the integration of ChatGPT and Mathematica is demonstrated.

Example in Optimization:

Consider the following problem.

Formal problem definition: A farmer owns 60 acres of land. She can plant either wheat or corn. Based on current predictions, wheat will yield a profit of \$200/acre while corn yields a profit of \$300/acre. Each planted acre uses fertilizer and labor, wheat using 2 tons of fertilizer and 3 hours of worker time; corn using 4 tons of fertilizer and 2 hours of worker time. There are 100 hours of worker time and 120 tons of fertilizer available (costs of which are already included in the profit calculation).

This problem represents a linear programming optimization problem. To model it properly, first one needs to determine that linear programming is the right tool to solve the problem. Then the variables, constraints and objective need to be defined.

Putting this problem directly into ChatGPT produces the following response.

Response: To set up a linear programming model for the farmer's land allocation problem, we need to define the decision variables, objective function, and constraints. Let's denote the following:

Let:

x = Number of acres of land planted with wheat

y = Number of acres of land planted with corn

Objective function: Maximize the total profit (in dollars)

$$Z = 200x + 300y$$

The objective is to find the values of x and y that maximize the total profit Z while satisfying all the given constraints. This model can be solved using various optimization techniques to find the optimal allocation of land for planting wheat and corn.

Subject to the following constraints:

Total land constraint: The farmer owns 60 acres of land, so the total land used for planting wheat and corn cannot exceed 60 acres.

$$x + y \leq 60$$

Fertilizer constraint: The total amount of fertilizer used for planting wheat and corn cannot exceed 120 tons.

$$2x + 4y \leq 120$$

Worker time constraint: The total hours of worker time used for planting wheat and corn cannot exceed 100 hours.

$$3x + 2y \leq 100$$

Non-negativity constraint: The number of acres of land for each crop must be non-negative.

$$x \geq 0, y \geq 0$$

Analysis of the Response: ChatGPT determines the correct variables for the problem. Moreover, it uses these variables to model the constraints and the objective correctly. This is exactly a model one would need to construct.

This kind of problem is difficult for some beginning students in business and operations research to model correctly. It is remarkable that ChatGPT can create such a model. In general, GenAI holds promise for streamlining the generation of mathematical models. This can improve the efficiency and accuracy of models used in the field.

Optimizing the Model: ChatGPT was able to produce the following linear program.

Maximize $Z = 200x + 300y$

Subject to:

$x + y \leq 60$

$2x + 4y \leq 120$

$3x + 2y \leq 100$

$x \geq 0$

$y \geq 0$

If prompted to solve this model, it will return an incorrect result. This will be discussed further in the limitations section of this document. The issue is that GenAI is not designed to solve optimization problems and specialized software and algorithms must be used.

However, while GenAI is insufficient to optimize this problem, it can be integrated with optimization software. One example is the currently available integration of ChatGPT with Wolfram's Mathematica. Asking ChatGPT to optimize with Mathematica does produce the correct result, which is the following.

The optimal solution is:

$x = 20$ acres (planted with wheat)

$y = 20$ acres (planted with corn)

The profit is 10,000

This is indeed the optimal solution to the problem. When solving, ChatGPT acknowledges that it used Wolfram to arrive at this solution.

Integrating of GenAI with optimization software holds the promise of streamlining decision making promises in operations. At the current state, it can be used to more efficiently model problems and arrive at solutions. In the future, it should be expected that GenAI can interact with more sophisticated software such as CPLEX and Gurobi offering tools for business to use for effectively any decision-making problem.

4. Software Engineering

GenAI is rapidly becoming an integral part of daily tasks in software engineering. It serves multiple functions, proving to be an effective tool in various aspects of the development process. These include code generation, debugging, and test case creation. This section overviews uses cases in software engineering and assumes some background in coding.

GenAI excels in generating programming languages as effectively as it does with natural languages. For common functions, such as data structures like heaps or priority queues, GenAI can swiftly produce the necessary code. When building a website, GenAI can generate standard templates, streamlining the design process. Beyond templates, GenAI can propose entire functions or modules based on a natural language description of the desired functionality. Traditionally, a software engineer might search the internet for examples, reuse internal code, or write new code from scratch. While engineers can generate code independently, it can be time-consuming and costly. GenAI offers a faster, more efficient alternative, as will be further illustrated in the example below.

Debugging is often cited as one of the most time-consuming tasks for software engineers, and here too, GenAI shines. It can analyze error messages and logs to help pinpoint the root cause of a problem. GenAI interprets cryptic error messages, suggesting their likely meanings and identifying the probable sections of the codebase responsible. It aids in code reviews by flagging potential issues, such as uninitialized variables, potential null pointer exceptions, or potential out-of-bounds array access. GenAI highlights suspicious or anomalous code patterns likely to be bug sources and, based on the identified issues, suggests potential fixes, including code snippets that resolve the problem. Moreover, GenAI can generate test cases designed to expose bugs, including edge cases that developers might overlook. For instance, it might recommend testing a function with extreme values, null values, or invalid inputs.

Highlighting GenAI's impact on software engineering, development environments have begun integrating tools like CoPilot. This integration seamlessly incorporates GenAI into the standard toolset used by today's software developers.

Examples in Software Engineering

Often when coding, one is faced with programming tasks that are common. For instance, a data structure such as a binary search tree. Instead of locating such codes elsewhere, GenAI can automatically generate them. For example, requesting ChatGPT to generate code for a binary search tree produces the following.

Sure, I can provide you with a simple implementation of a binary search tree in Python. Here's a basic version:

```
python Copy code  
  
class Node:  
    def __init__(self, key):  
        self.left = None  
        self.right = None  
        self.val = key  
  
def insert(root, key):  
    if root is None:  
        return Node(key)  
    else:  
        if root.val < key:  
            root.right = insert(root.right, key)  
        else:  
            root.left = insert(root.left, key)  
    return root  
  
def inorder(root):  
    if root:  
        inorder(root.left)  
        print(root.val),  
        inorder(root.right)
```

Another example of GenAI's usage is debugging code. For example, directly giving the code on the left to ChatGPT and asking what is wrong produces the code on the right. Here it can correctly determine the error and offer the correct code as a solution.

```

public class Test {

    private static void printLength(String str) {
        System.out.println(str.length());
    }

    public static void main(String []args){

        String myString = null;
        printLength(myString);
    }
}

The above code is not running. Could you tell me what the problem might be?

```

The code will result in a 'NullPointerException' at runtime when you call the 'printLength' method with 'myString' variable, which is set to null.

The 'NullPointerException' occurs because the 'length()' method is being called on a null object reference. To fix this issue, you can add a null check before calling the 'length()' method.

Here is the updated code:

```

public class Test {

    private static void printLength(String str) {
        if (str != null) {
            System.out.println(str.length());
        } else {
            System.out.println("String is null.");
        }
    }

    public static void main(String[] args) {
        String myString = null;
        printLength(myString);
    }
}

```

Finally, a common challenge faced in practice is generating test cases, especially for boundary conditions. Here GenAI can directly be used to read code and suggest test cases. The above code was placed into ChatGPT and it produced the following output.

Certainly! Here are some test cases you can use to test the 'printLength' method:

1. Test with a non-null string:

```

String myString = "Hello";
printLength(myString);
//Expected Output: 5

```

2. Test with a null string:

```

String myString = null;
printLength(myString);
//Expected Output: String is null

```

3. Test with an empty string:

```

String myString = "";
printLength(myString);
//Expected Output: 0

```

5. Limitations and Challenges using GenAI

This document highlights positive examples that demonstrate how GenAI can be leveraged effectively. While the technology is indeed exciting, it is important to acknowledge its limitations.

In its current state, GenAI can produce incorrect answers. Complicating matters further, these incorrect answers often appear to be correct at first glance, and GenAI provides justifications for these answers. However, a domain expert can identify errors in the solutions provided. Examples of such errors include generating code that will not execute, incorrectly solving an optimization problem and outputting an incorrect solution or recommending an inappropriate software library.

Because of these potential inaccuracies, it is essential to use GenAI with appropriate guardrails in place. For GenAI to be effectively implemented in practice, an expert must review the

solutions produced and determine their correctness. As such, GenAI does not replace the need for experts; rather, it serves as a valuable addition to the tools that experts already have at their disposal.

In the following section, we present a detailed example where GenAI fails to produce the correct solution. Despite this, GenAI provides a compelling argument for why the solution it suggests is correct, further illustrating the need for expert oversight.

Example Challenge for GenAI:

Here we illustrate that ChatGPT can offer seemingly good solutions, but they are incorrect. This illustrates that GenAI has the risk of misleading users. At the current stage, an expert is required to ensure that the solutions returned are correct as desired. In particular, GenAI is not a replacement for expert, but augments an expert's toolkit.

Recall the following linear program that ChatGPT modeled.

Maximize $Z = 200x + 300y$

Subject to:

$x + y \leq 60$

$2x + 4y \leq 120$

$3x + 2y \leq 100$

$x \geq 0$

$y \geq 0$

Consider solving this without using specialized optimization software. If the Mathematica plugin for ChatGPT is not used, the following result is given as a response.

Reponses: To solve this linear programming problem, we can use any optimization technique, such as the Simplex method or graphical method. However, I'll use a solver tool to quickly find the optimal solution. Please note that the solver tool may not be available in this text-based environment, so I'll describe the results:

The optimal solution is:

$x = 20$ acres (planted with wheat)

$y = 40$ acres (planted with corn)

With this allocation, the farmer should plant 20 acres of wheat and 40 acres of corn to achieve the maximum profit. Let's calculate the total profit:

Total profit = (Profit per acre of wheat * Number of acres of wheat) + (Profit per acre of corn * Number of acres of corn)

Total profit = $(200 * 20) + (300 * 40) = 4000 + 12000 = \$16,000$

Therefore, by dividing the land optimally with 20 acres of wheat and 40 acres of corn, the farmer can make a total profit of \$16,000.

Analysis of the Response: This response is quite detailed. On the surface, it seems correct. However, it is not the optimal solution to the problem. Indeed, $x=20$ and $y=40$ does not satisfy the constraint $2x + 4y \leq 120$. Thus, it offers a totally wrong infeasible solution. This shows the

importance of checking the results produced by GenAI, especially when it is used for sophisticated tasks.

6. Conclusion

GenAI and large language models serve as a versatile tool across various domains in operations, especially in technology and data science. In data analysis, it aids in interpreting complex datasets, generating insightful visualizations, and offering statistical guidance, thereby enabling users to make data-driven decisions with confidence. For data cleaning, GenAI can suggest methodologies to identify, rectify, or remove inconsistencies and inaccuracies in datasets, ensuring the integrity and reliability of the data. In the realm of optimization, GenAI can assist in formulating and solving optimization problems, providing strategies to improve efficiency and resource utilization in various contexts. Furthermore, in software engineering, GenAI acts as a virtual pair programmer, offering code suggestions, debugging assistance, and best practice recommendations. Thus, GenAI is not just a conversational agent, but a comprehensive assistant that enhances productivity and fosters innovation in the ever-evolving landscape of technology and data science.

Bibliography

- (2023). Retrieved from ExcelBrew: <https://gptexcel.uk/>
- Dhana, F. (2023, July 8). *Writing advanced Excel macros with ChatGPT-4*. Retrieved from <https://www.journalofaccountancy.com/news/2023/jul/writing-advanced-excel-macros-with-chatgpt-4.html>
- Elfman, L. (2023, May 9). *Can ChatGPT identify and handle data quality issues?* Retrieved from Bigeye: <https://www.bigeye.com/blog/can-chatgpt-identify-and-handle-data-quality-issues>
- Github Copilot*. (2023). Retrieved from <https://github.blog/2023-03-22-github-copilot-x-the-ai-powered-developer-experience/>
- Horse, J. (2023, August 10). Retrieved from <https://www.geeky-gadgets.com/chatgpt-data-entry-automation/>
- Khadka, N. (2023, July 20). *How to Integrate ChatGPT with Google Sheets Using Google Apps Script*. Retrieved from FreeCodeCamp: <https://www.freecodecamp.org/news/create-chat-gpt-formulas-in-google-sheets/>
- SheetGPT*. (2023). Retrieved from <https://sheetgpt.ai/>
- Wolfram*. (2023). Retrieved from <https://www.wolfram.com/wolfram-plugin-chatgpt/>
- Zhang, K. (2023, June 12). *Data Validation with ChatGPT: Trials and Insights*. Retrieved from Qxf2 BLOG: <https://qxf2.com/blog/data-validation-chatgpt-trials-insights/>