PharmacoEconomics - Open
https://doi.org/10.1007/s41669-024-00477-8

ORIGINAL RESEARCH ARTICLE

Artificial Intelligence to Automate Health Economic Modelling: A Case Study to Evaluate the Potential Application of Large Language Models

Tim Reason[1] · William Rawlinson[1] · Julia Langham[1] · Andy Gimblett[1] · Bill Malcolm[2] · Sven Klijn[3]

Accepted: 1 February 2024

Abstract

Background Current generation large language models (LLMs) such as Generative Pre-Trained Transformer 4 (GPT-4) have achieved human-level performance on many tasks including the generation of computer code based on textual input. This study aimed to assess whether GPT-4 could be used to automatically programme two published health economic analyses.

Methods The two analyses were partitioned survival models evaluating interventions in non-small cell lung cancer (NSCLC) and renal cell carcinoma (RCC). We developed prompts which instructed GPT-4 to programme the NSCLC and RCC models in R, and which provided descriptions of each model's methods, assumptions and parameter values. The results of the generated scripts were compared to the published values from the original, human-programmed models. The models were replicated 15 times to capture variability in GPT-4's output.

Results GPT-4 fully replicated the NSCLC model with high accuracy: 100% (15/15) of the artificial intelligence (AI)-generated NSCLC models were error-free or contained a single minor error, and 93% (14/15) were completely error-free. GPT-4 closely replicated the RCC model, although human intervention was required to simplify an element of the model design (one of the model's fifteen input calculations) because it used too many sequential steps to be implemented in a single prompt. With this simplification, 87% (13/15) of the AI-generated RCC models were error-free or contained a single minor error, and 60% (9/15) were completely error-free. Error-free model scripts replicated the published incremental cost-effectiveness ratios to within 1%.

Conclusion This study provides a promising indication that GPT-4 can have practical applications in the automation of health economic model construction. Potential benefits include accelerated model development timelines and reduced costs of development. Further research is necessary to explore the generalisability of LLM-based automation across a larger sample of models.

1 Introduction

We are living through a golden age of innovations and the development of new treatments for many diseases. However, this is occurring at a time of increasing demand, primarily due to an ageing population with complex health needs, together with constrained healthcare resources and budgets. Health economic models, which provide evidence of the relative costs and benefits of new health technologies compared with existing technologies [1], are vital tools for informing health decision making, particularly health technology assessments that inform national decisions for market access and reimbursement [2].

Key Points for Decision Makers

GPT-4, a current generation large language model (LLM), automatically replicated two published health economic models with high accuracy, based on instructions about how the models should be designed and what input values should be used.

This is a promising early indication that LLMs could be used to automate building health economic models, which could reduce the costs of health economic analysis, accelerate model development timelines and reduce the risk of error in modelling.

✉ Tim Reason
tim.reason@estima-sci.com

[1] Estima Scientific, Mediaworks, 191 Wood Ln, London W12 7FP, UK

[2] Bristol Myers Squibb, Uxbridge, UK

[3] Bristol Myers Squibb, Princeton, NJ, USA

Published online: 10 February 2024

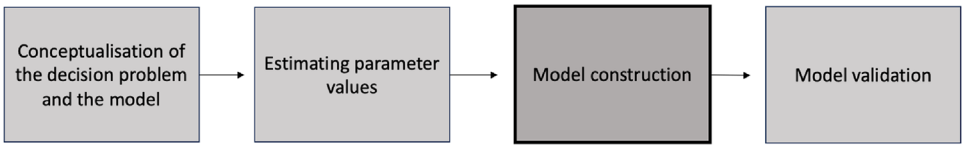

Fig. 1 The four phases of developing a health economic model

To ensure prompt market access to medicines, there is a demand for timely and reliable health economic analysis. However, existing methods for model development are expensive, time-consuming and prone to human-error [3]. There is therefore a need for research to enhance the efficiency and quality of health economic modelling. Automation of some aspects of economic modelling using artificial intelligence (AI) could accelerate development timelines, reduce costs and reduce the risk of technical errors, which are present in virtually all human-built models [4], ultimately improving access to medicines and outcomes for patients.

The development of a health economic model typically involves four phases: conceptualisation of the model, estimating parameter values, constructing the model and validating the model, as shown in Fig. 1 [2]. During the model construction phase, a health economist programmes the model in a software such as R or Excel [5], based on a previously specified design.

Large language models (LLMs), such as Generative Pre-Trained Transformer 4 (GPT-4), are mathematical models that work by repeatedly predicting the next word [7, 8]. LLMs enable automated generation of text content, including computer code, based on input (prompts) [8]. Therefore, LLMs offer a potential route to automating health economic model construction. Theoretically, we could provide an LLM with a series of text-based prompts describing a models' design, and ask it to generate code to programme the model in a software such as R. However, the potential of LLMs in automating model construction has not yet been explored.

LLM-based model construction is a promising idea for several reasons. Firstly, health economists usually produce a text-based summary of a model's design prior to model construction (a specification document). Secondly, several aspects of model construction are suited to automation: model construction involves programming a large number of simple formulae, which is time consuming, repetitive and prone to human error; health economic models are typically based on a limited set of well-established methodologies, and there are objectively correct and incorrect ways of programming a model provided the model is conceptualised (designed) in sufficient detail [3].

In this paper, we report a case study that aimed to assess whether an LLM, GPT-4, could be used to automatically construct and replicate the results of two published health economic analyses based on text prompts describing the model's assumptions, methods and parameter values.

2 Methods

2.1 Economic Models used in the Case Study

The two published health economic analyses were chosen because we had access to complete information on the methodology used, and both models were three-state partitioned survival models, which is a very commonly used model type in oncology modelling. Both published models were built in Microsoft Excel. One model assessed the cost-effectiveness of nivolumab versus docetaxel in patients with non-small cell lung cancer (NSCLC) previously treated with platinum-based chemotherapy from a US payer perspective (the NSCLC model), and the other assessed the cost effectiveness of nivolumab plus ipilimumab versus both sunitinib and pazopanib for the first-line treatment of unresectable advanced renal cell carcinoma (RCC) in Switzerland (the RCC model) [10, 11]. Key characteristics of each model are presented in Table 1.

For this study, we did not have access to individual patient data that were used in the published models to fit overall survival, progression-free survival and time-to-discontinuation curves. Therefore, these extrapolated curves were used directly as parameters in the AI-generated models. To constrain the scope of our case study, we generated only the base case analyses, and sensitivity and scenario analyses were not included.

2.2 Overview of the LLM-Based Automation of Model Construction

An overview of the LLM-based automation of model construction, including the prompt development process, is shown in Fig. 2.

2.2.1 Prompt Development Process

LLMs generate text content based on inputs known as 'prompts'. Text-based prompts can use any text-based form, including questions or instructions in natural language, and should convey the nature of the output that the user wishes to elicit from the LLM. An example of a prompt is, 'write me an essay on Hamlet'. The output of an LLM can vary significantly depending on the style and quality of a prompt [11]. Numerous studies have investigated 'effective' prompting, where 'effective' prompts are those most likely to produce

Table 1 Models replicated in the case study

Specification	Model	
	NSCLC	RCC
Model type	Three-state PSM	Three-state PSM
Treatments	Nivolumab, docetaxel	Nivolumab + ipilimumab, pazopanib, sunitinib
Health states	Progression-free, progressed disease, death	Progression-free, progressed disease, death
Time horizon and cycle length	20 years/1 week	40 years/1 week
Cost categories	Drug acquisition	Drug acquisition
	Drug administration	Drug administration
	Drug monitoring	Treatment initiation (upon starting treatment)
	Subsequent therapy drug acquisition, drug administration and drug monitoring (upon death or progression for nivolumab, upon finishing first-line treatment for docetaxel)	Disease management
		Subsequent therapy drug acquisition, drug administration (upon finishing first-line treatment)
		End-of-life care (upon death)
	Disease management	
	Adverse events (upon starting treatment)	
	End-of-life care (upon death)	
Utility categories	Adverse events utility decrement	Treatment-specific health-state utilities
	Health-state utilities	

NSCLC non-small cell lung cancer, *PSM* partitioned survival model, *RCC* renal cell carcinoma

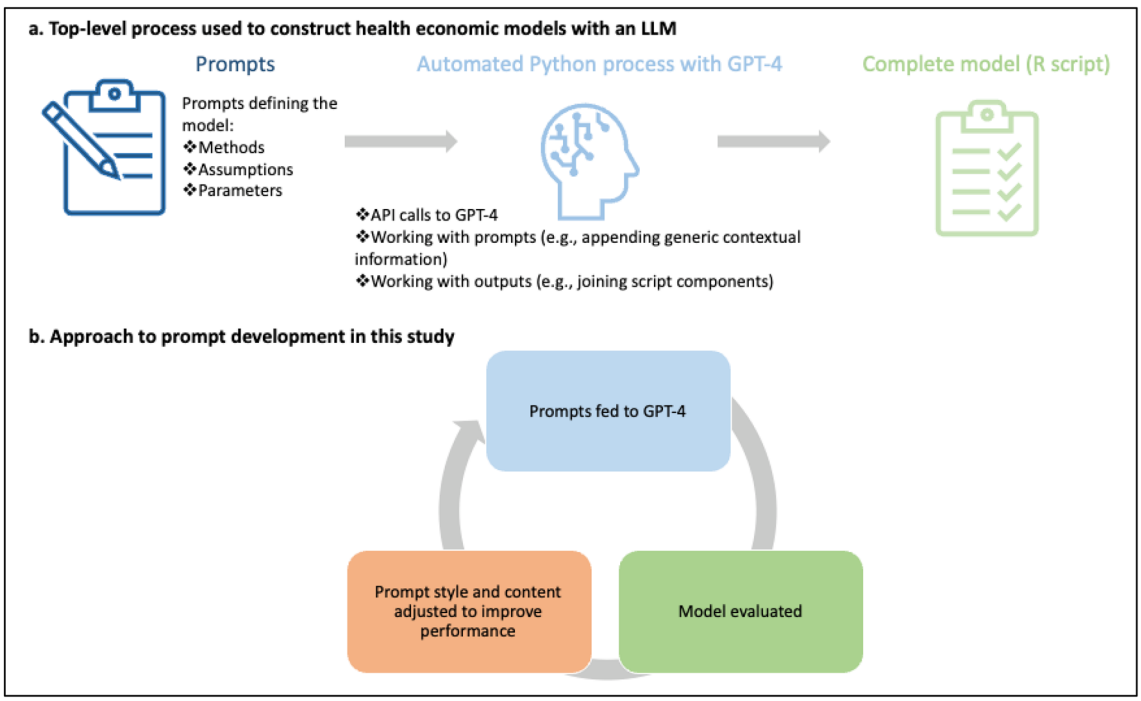

Fig. 2 Diagram showing (**a**) the top-level process used to construct health economic models using an LLM and (**b**) the iterative prompt development process used in this study. *API* application programming interface, *GPT-4* generative pre-trained transformer 4, *LLM* large language model

an output of the desired form and quality, and this is a highly active area of research that is rapidly progressing [13–16]. A scientific process is required for best outcomes. Numerous strategies such as 'chain of thought' prompting and the inclusion of key phrases (e.g. 'let's think step by step') have been assessed on benchmark problem sets and have been demonstrated to significantly improve performance [16, 17]. Iterative optimisation methods have also been shown to produce improvements in outcomes for given task sets [18].

Given the impact of prompting strategies on performance, it was important that we developed effective prompts for our case study to fairly assess GPT-4's capabilities in model construction. As no existing studies had investigated how to effectively prompt LLMs to construct health economic models, we opted to use an iterative method to develop the prompts. It should be noted that an alternative prompting strategy may yield superior outcomes; however, the iterative method provided satisfactory outcomes for this study. This functioned as follows (Fig 2b): for each model initial prompts were developed; these were submitted to GPT-4 and the generated models were evaluated and based on these insights the prompts were adjusted. The adjusted prompts were then submitted back into GPT-4 for further testing and evaluation; the process continued until no further improvements could be made through reasonable adjustments to the content and style of the prompts, and final prompts were reached for each model.

The prompts we developed instructed GPT-4 to code the NSCLC and RCC models in R, and provided descriptions of each model's methods, assumptions and parameter values as supporting information.

2.2.2 LLM Interaction

There are a variety of methods to submit prompts to an LLM and receive an output. ChatGPT is a web application that allows prompts to be submitted to an LLM online, in a dialogue format [19], a method that is readily accessible and popular. However, it is not suited to automation, as it requires manual entry of prompts into the web application, and manual extraction of the response.

For this study, we used application programming interface (API) calls to submit prompts to GPT-4 and receive output. API calls transmit a request to a server (in this case, transmitting a prompt to the GPT-4 servers) and return a response (in this case, returning the text output from GPT-4). Importantly, API calls can be embedded into code, such as a Python script. This enables automation of complex, multi-step interactions with LLMs. For example, a computer programme can be written to automate a series of prompt–output interactions with an LLM, and subsequently manipulate the LLM's outputs.

2.3 Prompting Methods and Key Learnings

Several key insights were uncovered through iterative prompt development, which shaped the form of the final prompts, as described below.

2.3.1 Using Multiple Prompts

A token is a unit of text that can be processed and generated by an LLM. GPT-4 had a token limit of 8192 at the time of the study. This restricted the quantity of text in a prompt–response pair to roughly 4000 words. The base case analyses of the models were found to require more than 15,000 tokens to specify in R. Therefore, the models could not be generated using a single prompt. In addition, GPT-4 was observed to have significantly better performance when instructed to build a single element of the models (such as a particular input calculation, or survival analysis) than when instructed to build a full model in one go.

Therefore, we developed multiple prompts for each model, each instructing GPT-4 to generate a separate section of the R script. We split the scripts into sections as follows:

- Parameter definition sections—each of these sections defined a set of model parameters.
- Input calculation sections—each of these sections calculated a cost or utility from the model parameters, which was later applied in the model trace.
- Model trace sections—each of these sections defined a part of the model trace, using functions from the Heemod R package [20].
- Other sections—these sections contained routine code, such as code to run the model or load R packages.

Generating the scripts in sections posed challenges. When generating a section of the R script, GPT-4 only had access to information contained in the prompt for that section. However, the separate script sections had to work together when combined. In particular, later sections needed to use variables defined in earlier sections. Therefore, we developed a fully automated process in Python to pass information on earlier sections of the model script into prompts used for later sections [21]. This worked as follows (Fig. 3):

1. The prompts were loaded into Python as strings. A separate prompt was developed for each model section.
2. Alongside each prompt, a 'section tag' was added which indicated what part of the model the prompt referred to. For example, there were six section tags available for prompts for input calculation sections, which covered general categories of input calculation. These were: drug acquisition cost calculation, transition cost calculation, health state cost calculation, other cost calculation, util-

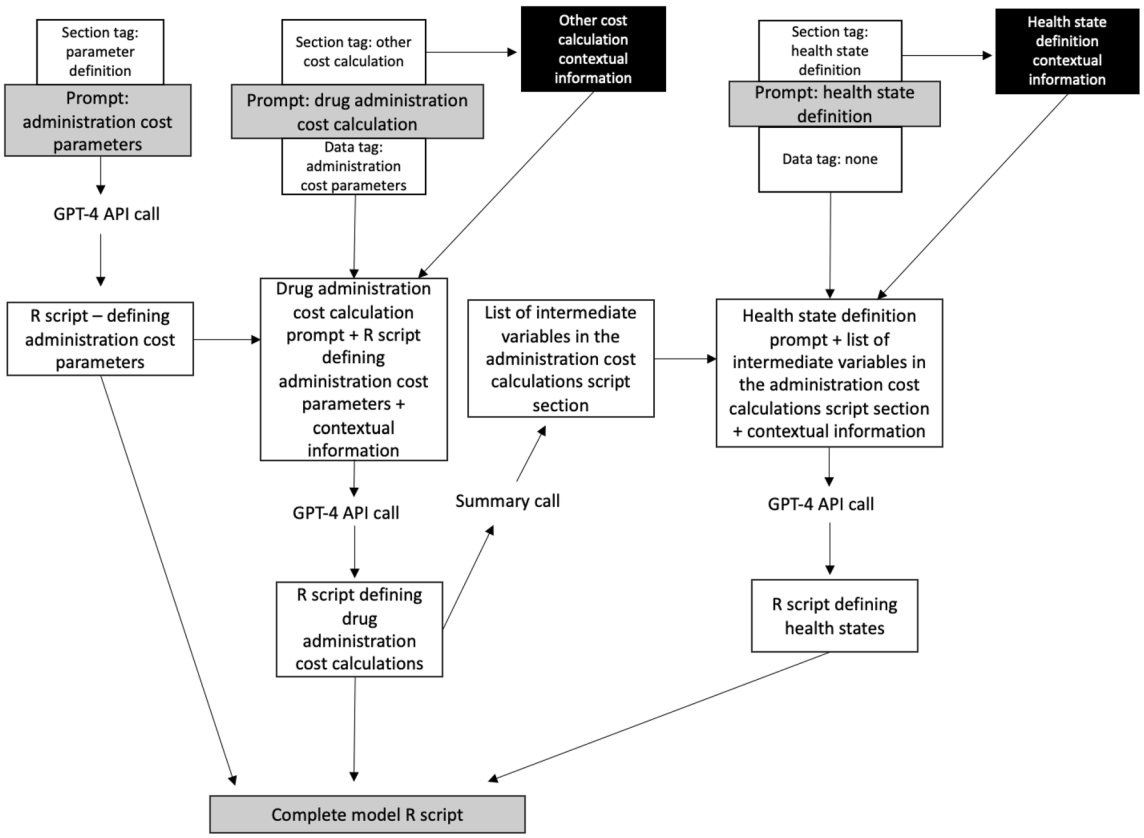

Fig. 3 Diagram showing the structure of the automated process used to construct each replica model in Python. *API* application programming interf*ace, GPT-4* generative pre-trained transformer 4

ity decrement calculation and health state utility calculation. These options were sufficient to construct both the RCC and NSCLC models.

3. For each prompt, the user could provide a further 'data tag'. These tags linked the prompt to one or more of the parameter definition prompts.
4. When the process was initiated, the prompts were passed automatically to GPT-4 using API calls. The order was determined by the section tags.
5. The parameter definition sections of the scripts were automatically appended to the prompts for calculation sections based on the data tags. This ensured that GPT-4 had information on the variable names of model parameters required for the calculation sections.
6. Once all prompts had been passed to GPT-4 and all the script sections had been generated, these were automatically combined into a complete model script through concatenation. Again, the order was determined by the section tags. The final output could be copied into R and run without any human edits.

As well as passing the variable names of model parameters into prompts, it was also necessary to pass some intermediate variable names. An intermediate variable stores the result of a calculation for use in a later section of the model script. For example, models commonly calculate per cycle costs which are applied later in the trace calculations.

This posed a separate problem as the user cannot know in advance what intermediate variables will be generated by GPT-4 and how these variables will be named. Therefore, a solution analogous to the tagging approach was not feasible. Instead, we developed automated 'summary calls' that were API calls prompting GPT-4 to list the intermediate variables defined in a section of the model script. Summary calls were added into specific stages of the automated process to pass intermediate variable names from earlier script sections into the prompts used to generate later script sections (Fig. 3). The automated process was able to handle both model cases and was not changed between constructing the NSCLC and RCC models.

2.3.2 Contextual Information

We observed that GPT-4 made far more errors when using functions from health economic modelling packages than when implementing base functions in R. GPT-4 also

The contextual information initiates with instructions relevant to the model section, and a worked example. This ensures GPT-4 uses the correct function from the Heemod package, and names the variables in a predictable format.

Your task is to write an R script defining the dr_cost and dr_benefit
variables based on the given information, and using the rescale_discount_rate function.
Give the script the following title: ###### Discounting inputs ######

Question: Produce an R script defining the dr_cost and dr_benefit variables
based on the given information.

Information: discount benefits at 5% per year and costs at 2% per year

Answer:###### Discounting inputs ######

dr_cost <- rescale_discount_rate(0.02, 1, cycle_length) # assume cycle_length is defined elsewhere
dr_benefit <- rescale_discount_rate(0.05, 1, cycle_length)

Question: Produce an R script defining the dr_cost and dr_benefit variables
based on the given information.

The prompt specifying the model's discounting rates is automatically inserted below (see underlined text)

Information: Discount costs and benefits at 3% per year.

Generic statements were included in all contextual information to improve performance and ensure a valid R script is produced

Let's work this out in a step-by-step way to be sure we have the right answer.

Please do not print the calculation results. Please do not provide any text before and
after the script, so that the output will run in R. Please do not include ```R before the
script, or ``` after the script. Do not write 'Answer:' before the script.

Fig. 4 Example of contextual information. This contextual information was automatically appended to prompts tagged as 'discounting' prompts. *GPT-4* generative pre-trained transformer 4

incorrectly implemented certain common health economic assumptions, such as vial wastage, when prompted. Further, intermediate variables were stored in an inconsistent manner (scalars, vectors and arrays) which caused errors when these variables were used in later script sections. It therefore became clear that we needed to provide GPT-4 with contextual information on top of information specifying the model assumptions, methods and parameter values. This information needed to describe how to use functions from health economic modelling packages, explain common health economics assumptions, and provide instructions on the desired structure of the model code (for example, specifying how to store intermediate variables). To this end, we drafted contextual information relevant to each model section, and integrated this into the Python process. The information was automatically prepended to the calculation and data prompts, based on the section tags, as shown in Fig. 3.

An example of contextual information is provided in Fig. 4. We included worked examples, as this has been shown to improve the performance of LLMs in multi-step reasoning tasks [16]. The contextual information was developed iteratively in the same manner as the prompts. The final set of contextual information was generic and applicable to both models. It formed part of the back-end structure of the Python process and was not changed when we used the process to construct the RCC and NSCLC models.

2.3.3 Prompt content

Through the process of iterative development, we reached a final prompt set of 33 prompts to specify the NSCLC model. A total of 17 of these prompts contained only parameter values ('data prompts'), with the remaining 16 prompts describing methodology and assumptions ('method prompts'). The final prompt set for the RCC model used 21 data prompts and 16 method prompts. All final prompts are provided in the Online Resource.

The method prompts differed in length depending on the complexity of the methods described. Figure 5a provides an example of a simple method prompt for the RCC model and the data prompts to which it was linked, and Fig. 5b provides an example of a complex method prompt for the NSCLC model.

For sections of the models that required multi-step methodology, performance was generally improved by explicitly setting out the methodological steps in order. We noted that on occasion, the performance of prompts could depend on phrasing and word choice.

To avoid submitting sensitive data to GPT-4, dummy values were used in data prompts, which required human intervention to replace dummy values with the correct values in the output scripts. However, this step could be avoided through the use of a private LLM that ensures the confidentiality of sensitive information (see Discussion).

2.4 Output Generation and Assessment

The final set of prompts for each model were loaded into Python and the automated process was initiated. This produced a text string with AI-generated R code for each model. The string was copied into R and run without human edits. No change was made to the automated process (including the contextual information) between generating the NSCLC and RCC models. The results of the generated scripts were compared with the published values and a health economist performed line-by-line technical quality assurance to identify any errors.

Metrics collected were the base case incremental cost-effectiveness ratio (ICER) result as well as the number and category of errors in the generated models. Errors were categorised into minor, intermediate and major errors. Classification was based on the time it took for a health economist to correct the errors once they had been identified. Minor errors took less than 2 min to rectify, intermediate errors took less than 10 min, and major errors took more than 10 min. As this measure could vary from health economist to health economist, a description of all errors is provided in the appendices.

Despite setting the temperature of GPT-4 to 0, ('temperature' controls the randomness of the text generated by GPT-4) outputs were observed to vary when the same prompt set was used on multiple occasions. Therefore, we generated 15 scripts for each model to capture variation in performance.

3 Results

Example AI-generated scripts for each model are provided in the Online Resource. The accuracy of the NSCLC and RCC models is shown in Fig. 6. The NSCLC model was fully replicated with high accuracy. Overall, 100% (15/15) of the AI-generated NSCLC models were error free or contained only a single minor error, and 93% (14/15) of the AI-generated NSCLC models were completely error free. Only one minor error was observed across the 15 test runs.

The RCC model was also closely replicated. However, human intervention was required to simplify one element of the model design (one of the model's fifteen input calculations). This is because it used too many sequential steps to be implemented in a single prompt. This had only a minor impact on model results. The original calculation used an elaborate approach to calculate weight-based drug dosing. A simplification was applied by providing the proportion of patients in each weight category and the midpoint weights directly, as well as limiting the set of available vial sizes.

This was performed manually at the prompting stage, so that GPT-4 was instructed to build the simplified version of the model. With the simplification, 87% (13/15) of the AI-generated RCC models were error free or contained only a single minor error, while 60% (9/15) of the AI-generated RCC models were completely error free. In total, six minor errors and one intermediate error were observed across the 15 test runs.

All error-free scripts for both models replicated the published ICERs to within 1%. For the NSCLC model, the error-free AI-generated ICERs all evaluated to USD$117,600/quality per quality-adjusted life-year (QALY), compared with the published value of USD$117,739/QALY. For the RCC model, the error-free AI-generated ICERs all evaluated to CHF107,284/QALY versus sunitinib and CHF105,965/QALY versuss pazopanib, compared with the published values of CHF108,326/QALY and CHF106,996/QALY. Deviation was explained by minor differences in the calculation engine of the Heemod R package versus the Excel models. For example, the AI-generated models applied discounting on a per-cycle basis, whilst the Excel models applied this on a year-by-year basis. Similarly, the R models assumed progression-free survival state occupancy was 100% in the first model cycle, whereas half-cycle correction was applied in the first model cycle for one of the Excel models.

Of the 30 AI-generated models, none required more than 10 min of edits to rectify errors following human quality assurance. The average time taken by GPT-4 to generate the NSCLC model was 715 s (standard deviation 29 s) and the average time taken by GPT-4 to generate the RCC model was 956 s (standard deviation 52 s).

4 Discussion

In this case study we aimed to assess whether GPT-4 could be used to automatically construct two health economic analyses based on descriptions of the model's assumptions, methods

Method prompt

"The cost category is 'drug_aq'.[1] The dosing schedule for pazopanib is 800mg every day. The dosing schedule for sunitinib is 50mg every day, using a 4 weeks on, two weeks off repetition. Apply the RDI for sunitinib and pazopanib to each treatment after week 29. Assume vial sharing."

Linked data prompts

1. "pazopanib cost of 60 pills of 400mg = CHF4025.75
sunitinib cost of 28 pills of 50mg = CHF5476.25"

2. "RDI for sunitinib and pazopanib after week 29 = 1"

a

Method prompt

"The cost category is 'sub_therapy_drug_admin'.[1] For patients receiving first line nivolumab, apply the total cost of drug administration for subsequent therapies when a patient leaves progression-free survival. For patients receiving first line docetaxel, apply the total cost of drug administration for subsequent therapies when a patient finishes first-line treatment. To calculate the total cost, first calculate the cost of drug administration for each of the six 2nd line therapies, assuming the specified subsequent therapy duration. To do this, calculate the number of doses in that duration, and multiply by the administration cost. Please note there is not an administration cost for erlotinib which is a pill.
Then, calculate the total cost as a weighted average over the 2nd line therapies, using the proportion of patients receiving each 2nd-line therapy after first-line nivolumab and first-line docetaxel.
The dosing schedule for nivolumab is one dose of 480mg per 4 weeks (two 40 mg vials and four 100mg vials).
The dosing schedule for docetaxel is one dose of 75mg per meter squared of body surface area per 3 weeks.
The dosing schedule for gemcitabine is 2500mg per meter squared of body surface area per 3 weeks.
The dosing schedule for pemetrexed is 500mg per meter squared of body surface area per 3 weeks.
The dosing schedule for carboplatin is 400mg per meter squared of body surface area per 4 weeks.
The dosing schedule for erlotinib is 150mg once per day."

Linked data prompts

1. "Subsequent therapy duration weeks = 10"

2. "Proportion receiving Nivolumab as subsequent therapy after 1st line nivolumab = 0
Proportion receiving Docetaxel as subsequent therapy after 1st line nivolumab = 0
Proportion receiving Gemcitabine as subsequent therapy after 1st line nivolumab = 0
Proportion receiving Pemetrexed as subsequent therapy after 1st line nivolumab = 0
Proportion receiving Carboplatin as subsequent therapy after 1st line nivolumab = 0
Proportion receiving Erlotinib as subsequent therapy after 1st line nivolumab = 0
Proportion receiving Nivolumab as subsequent therapy after 1st line docetaxel = 0
Proportion receiving Docetaxel as subsequent therapy after 1st line docetaxel = 0
Proportion receiving Gemcitabine as subsequent therapy after 1st line docetaxel = 0
Proportion receiving Pemetrexed as subsequent therapy after 1st line docetaxel = 0
Proportion receiving Carboplatin as subsequent therapy after 1st line docetaxel = 0
Proportion receiving Erlotinib as subsequent therapy after 1st line docetaxel = 0"

3. "Administration cost per dose = $143.08"

b

Fig. 5 **A** Example of a specification prompt for a simple model component. Dummy values are underlined. [1]We found that including a definition of the cost category in snake case would lead to shorter and more precise variable names in the resulting R script. This is why "the cost category is 'drug_aq'" was included in the method prompt. *CHF* Swiss franc, *RDI* relative dose intensity. **B** Example of a specification prompt for a more complex model component. Dummy values are underlined. [1]We found that including a definition of the cost category in snake case would lead to shorter and more precise variable names in the resulting R script. This is why "the cost category is 'sub_therapy_drug_admin'" was included in the method prompt

and parameter values. Model construction is the third phase of model development, in which the model is programmed in a software such as R or Excel on the basis of a prior design, and should be distinguished from model conceptualisation, estimation of parameter values and model validation (technical and external) that were not automated during this study.

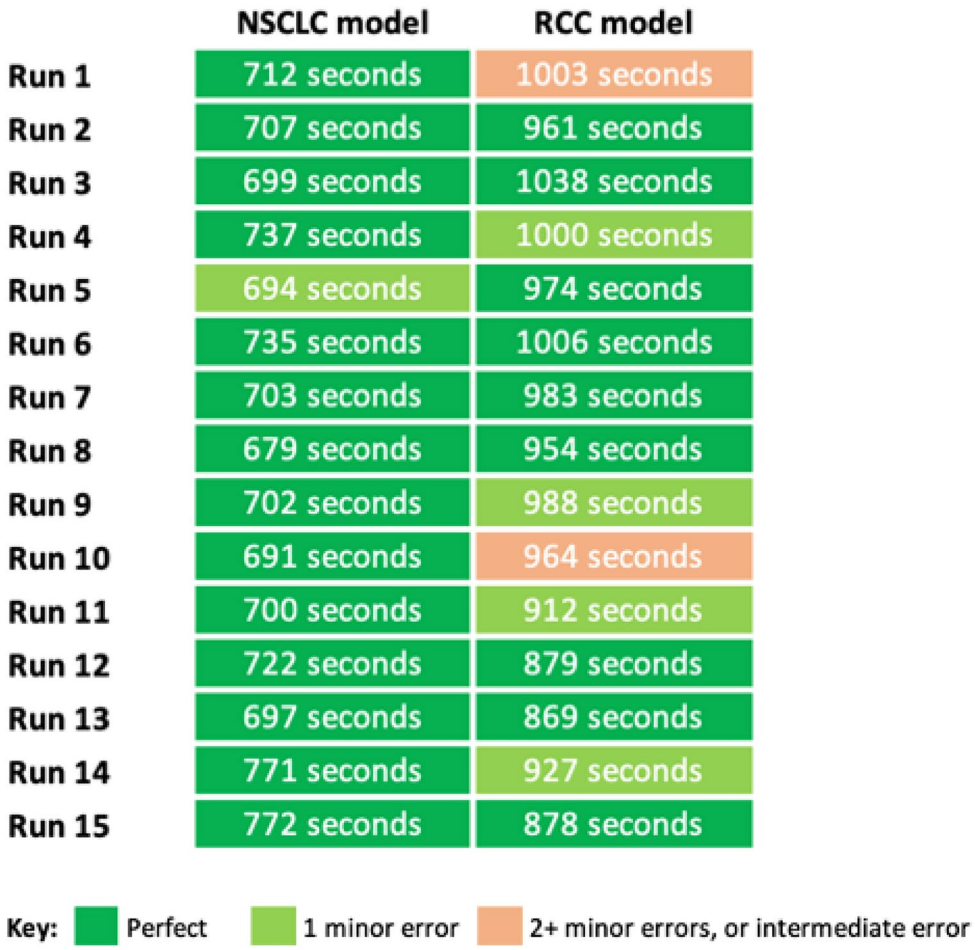

Fig. 6 Accuracy of the AI-generated replica models. *NSCLC* non-small cell lung cancer, *RCC* renal cell carcinoma

In response to this question and through iterative prompt development we reached a novel process for automating health economic model construction in R using an LLM. In addition to prompts describing the model's methods, assumptions and parameter values, the process required contextual information. However, this information was generalisable across the two models we generated and described how to use health economics R packages, how to interpret common health economic assumptions and how to structure code.

Using this novel process, we automatically constructed versions of the two published models. No human intervention was required between writing the prompts describing the model designs and receiving back the fully programmed model R scripts. Across 15 runs for each model, most of the runs were error free or contained only a single minor error. These results are promising given that these are health technology assessment (HTA)-ready models and that virtually all human built health economic models contain technical errors prior to quality assurance [4]. None of the AI-generated models required more than 10 min of human edits to correct errors following full technical quality control, which demonstrates the minor nature of errors observed in our study.

It should be noted that one calculation in the published RCC model had to be simplified for the AI-generated model, as it used too many sequential steps for a single prompt. To fully replicate the published RCC model this section of the AI-generated script would require human editing, indicating that with current generation LLMs human intervention may be required for atypical and complex model sections. However, simplification was required for only one section of the 28 calculation sections across the two models. The need for occasional human intervention does not greatly undermine the potential benefits achievable through LLM-based automation of model construction.

4.1 Study Limitations

There were a number of limitations in our case study. Firstly, sensitive data in the prompts we developed had to be redacted using dummy values and manually added back in to the AI-generated models. This is because prompts submitted to LLMs may be retained by the LLM provider and become vulnerable to data breaches. Also, LLMs may be trained on submitted prompts, which could result in data leaks. Data security is of great importance in HEOR and should not

be jeopardised as we take advantage of the opportunities offered by LLMs. Since this research was performed, several options for the secure use of LLMs have emerged, such as dedicated hosting of private instances of LLMs, downloadable instances of open-source models and API services where prompts are not stored or used to train models. This would enable inclusion of sensitive data in model design prompts.

Secondly, to constrain the scope of our study, we replicated only the base case analyses of the published models. The ability of LLMs to programme sensitivity analyses, which are important components of health economic analyses, was not evaluated and is an area for future research. Additionally, the AI-generated models were both three-state partitioned survival models (PSMs) in late-stage anticancer treatment. It remains to be demonstrated whether LLMs can accurately programme a range of model types with varying levels of sophistication, such as decision-tree analysis, Markov models and individual patient simulation approaches, and whether this can be achieved across a wider range of disease areas.

Thirdly, following technical quality control of the AI-generated scripts, errors were corrected by the same health economist who had developed the prompts. Due to the nature of the iterative development process, the health economist had some familiarity with the type of errors likely to be made, which may have reduced the time taken to correct them. More time may be required to correct errors without the prior knowledge gained through developing prompts using an iterative process.

4.2 Implications for Future Policy and Research

The implications of our research are many fold. We replicated published models in this study to demonstrate the accuracy of the LLM-generated models by comparing results against established values. However, the same processes could be used to automatically construct a de novo model, where model conceptualisation, estimation of parameter values and model validation (technical and external) are performed manually as for human-built models. When developing a de novo model, it is common practice to specify the model in detail prior to starting any programming (for example, in a model specification document). This information could be used to develop model design prompts and perform LLM-based model construction for de novo models.

With this in mind, there are numerous potential applications for LLM-based model construction. As a first use case, AI-generated models could be used to rapidly perform double-programming technical validation of human-built models. This is a method in which the same model is built independently by two health economists, and differences in the results are investigated to reveal technical errors. In this use case, the LLM could take on the role of one of the two health economists to save time and potentially increase accuracy. Secondly, LLM-based model construction could enable rapid production of additional models to perform assessments of structural uncertainty. For example, rapidly constructing a PSM in parallel to a Markov model, which may otherwise not be possible due to time and resource constraints. Thirdly, it may be possible to quickly adapt LLM-generated models through editing of the model design prompts (for example, adding a new comparator) which would be of particular use at an early modelling stage.

In addition to this, many countries have HTA agencies to robustly assess the costs and effectiveness of new technologies [22, 23]. However, the process can be lengthy and thereby delay patients' access to medicines [24–26], which in turn can affect patient outcomes [27, 28]. In the longer term, using LLMs to automate model construction could result in a reduction in the person hours required for model development, which could accelerate timelines for HTA processes and reduce costs. As AI is implemented into other aspects of clinical development and health economics and outcomes research (HEOR) it may increase the complexity as well as demand for HTAs [29, 30]. Therefore, it may be necessary to automate some aspects of the economic modelling to free up time for tasks that cannot be automated. AI is also being assessed in other processes that are relevant to HTAs and HEOR such as conducting systematic literature reviews [36, 37] and the use of large amounts of clinical data (real-world and "big data") [38].

Finally, LLM-based model construction could open the door to deploying economic modelling more widely in healthcare decision making, if significant reductions in costs and resource requirements can be achieved.

The above applications primarily derive from the potential of LLM-based model construction to reduce the time and resource required to construct models, and therefore to accelerate timelines for model construction. As our study was the first (to the authors' knowledge) to investigate using LLMs to produce health economic models, a high upfront time investment was required to experiment with and identify successful prompting strategies through iterative prompt development. However, prompting strategies may prove generalisable across different decision problems, and this assertion is supported by the similarity between the successful prompt sets we developed for the NSCLC and RCC models (particularly the contextual information, which was reused without edits). If this is the case, the process of developing prompts would shift from experimental, iterative development to adapting prompts from published exemplars based on the specifics of the decision problem in question. Such a streamlined process could enable significant reductions in the time and cost required to programme health economic models. Therefore, a key next research step will be to investigate the generalisability of prompting strategies across a

wider pool of models. In particular, further research should be conducted to assess the accuracy that can be achieved through using prompts transferred from one decision problem to another without iterative optimisation.

There are a number of challenges that must be overcome to integrate LLM-based automation into existing model development workflows. Our case study suggests that AI-generated scripts may contain errors. It is important that these errors are placed in the context of human performance in model construction, which is the relevant comparison, and are not used to discount AI-generated models out of hand [4]. It should also be emphasised that full technical quality assurance should be performed for AI-generated scripts as it is for human-built models.

Additionally, an expanded skillset is required to perform LLM-based model construction in comparison with manually developing health economic models. Firstly, knowledge of how to programme health economic models in R is required, both to perform technical quality control of AI-generated scripts, and to perform manual edits of atypical or complex sections. These skills are not ubiquitous amongst health economists. Although, it is worth noting that LLMs can be used to edit Microsoft Excel files (and therefore Excel-based models), which may become an important use-case in the future. Secondly, basic working knowledge of Python is an advantage (although, if prompting strategies prove generalisable the Python components may not require editing in many cases). Finally, users must understand how to develop effective prompts to specify a model. Educating health economists in these areas is likely to require dedicated training. However, if LLM-based model construction is significantly time saving this should not be a barrier to use.

Furthermore, HTA agencies and evidence assessment groups (EAGs) may be reluctant to accept the use of LLM-based processes in generating evidence. This is because the technologies involved are not yet widely understood, and there is not currently a gold standard for applying LLM-based methods in the field of HEOR. However, it should be noted that the output produced by LLM-based model construction (an R script) is scrutable in the same way as a human generated output, since all working is provided in the code. Therefore, an LLM-generated model could be robustly checked, which is a prerequisite of HEOR methods in an HTA document.

Whilst it is important to consider the above challenges, the results of our study should also be placed in the context of the rapid improvements that have recently been made in the field of generative AI. It is highly likely that next-generation LLMs will allow for the methods described in our case study to be adapted and improved. For example, next-generation LLMs may enhance the accuracy of generated code. Furthermore, models with improved token limits have been released since this study was conducted (GPT-4 turbo with a limit of 128,000 tokens, and Claude 2.1 with a limit of 200,000 tokens. The version of GPT-4 used in this study had a token limit of 8192). Increases to token-limits (which restrict the quantity of text that can be included in prompts and outputs) can simplify the processes described in this paper.

5 Conclusion

Using a novel LLM-based process, we constructed the base case analyses of published three-state partitioned survival analyses in R to a high degree of accuracy, demonstrating the feasibility of using GPT-4 to automate health economic model construction. Potential benefits of automating health economic model construction include accelerated timelines and reduced costs for model development, reduction in human error and novel methods for model validation and exploring structural uncertainty. Potential challenges include managing the perception of AI-generated models, the requirement for an expanded skillset in comparison with manual model construction, and barriers to acceptance of LLM-based methods by HTA bodies. Further research should be conducted to explore the generalisability of LLM-based model construction across a wider range of model types and disease areas, the accuracy that could be achieved through prompts that are reusable across multiple decision problems and the potential to construct Excel-based health economic models using LLMs.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s41669-024-00477-8.

Declarations

Funding This study was funded by Bristol Myers Squibb. The study sponsor was involved in several aspects of the research, including the study design, interpretation of data, writing of the manuscript and decision to submit the manuscript for publication.

Conflict of Interest This study was supported by Bristol Myers Squibb. Estima authors (Tim Reason, William Rawlinson, Julia Langham, Andy Gimblett) are consultants and have worked on behalf of Bill Malcolm and Sven Klijn, who are employees and shareholders of Bristol Myers Squibb.

Availability of Data and Material All data generated or analysed during this study are included in this published article (and its supplementary information files).

Ethics Approval No human participants, their data or biological material were used in this study.

Consent for Publication This article does not contain identifiable photos or patient data.

Consent to Participate No human participants took part in this study.

Code Availability All AI-generated R code has been provided in the Online Resource, in addition to the prompts that were used to describe the designs and inputs of each replicated model, and the user-defined R functions that were used by GPT-4.

Author Contributions All authors (TR, WR, JL, AG, BM and SK) contributed to the study conception and design. Material preparation, data collection and analysis were performed by TR and WR. The first draft of the manuscript was written by WR and all authors (TR, WR, JL, AG, BM and SK) commented on previous versions of the manuscript. All authors (TR, WR, JL, AG, BM and SK) read and approved the final manuscript.

References

1. Drummond MF, Sculpher MJ, Torrance GW, O'Brien BJ, Stoddart GL. Methods for the Economic Evaluation of Health Care Programmes [Internet]. Oxford University Press; 2005. https://EconPapers.repec.org/RePEc:oxp:obooks:9780198529453. Accessed on 01 Sep 2023.
2. Caro JJ, Briggs AH, Siebert U, Kuntz KM. Modeling good research practices—overview: a report of the ISPOR-SMDM Modeling Good Research Practices Task Force-1. Value Health. 2012;15:796–803.
3. (M. Eddy) D. Model transparency and validation: a report of the ISPOR-SMDM modeling good research practices task force-7. Value Health. 2012;15.
4. Radeva D, Hopkin G, Mossialos E, Borrill J, Osipenko L, Naci H. Assessment of technical errors and validation processes in economic models submitted by the company for NICE technology appraisals. Int J Technol Assess Health Care. 2020;36:311–6.
5. R Core Team. R: A Language and Environment for Statistical Computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2023. https://www.R-project.org/. Accessed on 01 Sep 2023.
6. OpenAI. GPT-4 Technical Report. 2023.
7. S. R. Bowman. Eight things to know about large language models. ArXiv [Internet]. 2023. https://doi.org/10.48550/arXiv.2304.00612. Accessed on 01 Sep 2023.
8. Poldrack RA, Lu T, Begu\vs G. AI-assisted coding: experiments with GPT-4. ArXiv [Internet]. 2023;abs/2304.13187. https://api.semanticscholar.org/CorpusID:258331866. Accessed on 01 Sep 2023.
9. Chaudhary MA, Lubinga SJ, Smare C, Hertel N, Penrod JR. Cost-effectiveness of nivolumab in patients with NSCLC in the United States. Am J Manag Care. 2021;27:e254–60.
10. Çakar E, Oniangue-Ndza C, Schneider RP, Klijn SL, Vogl UM, Rothermundt C, et al. Cost-effectiveness of nivolumab plus ipilimumab for the first-line treatment of intermediate/poor-risk advanced and/or metastatic renal cell carcinoma in Switzerland. Pharmacoecon Open. 2023;7:567–77.
11. Harrer S. Attention is not all you need: the complicated case of ethically using large language models in healthcare and medicine. EBioMedicine. 2023;90: 104512.
12. Zhou D, Schärli N, Hou L, Wei J, Scales N, Wang X, et al. Least-to-most prompting enables complex reasoning in large language models. ArXiv [Internet]. 2023. https://doi.org/10.48550/arXiv.2205.10625. Accessed on 01 Sep 2023.
13. Creswell A, Shanahan M. Faithful reasoning using large language models. ArXiv [Internet]. 2022. https://doi.org/10.48550/arXiv.2208.14271. Accessed on 01 Sep 2023.
14. Creswell A, Shanahan M, Higgins I. Selection-inference: exploiting large language models for interpretable logical reasoning. ArXiv [Internet]. 2022. https://doi.org/10.48550/arXiv.2205.09712. Accessed on 01 Sep 2023.
15. Wang X, Wei J, Schuurmans D, Le Q, Chi E, Narang S, et al. Self-consistency improves chain of thought reasoning in language models. ArXiv [Internet]. 2023. https://doi.org/10.48550/arXiv.2203.11171. Accessed on 01 Sep 2023.
16. Wei J, Wang X, Schuurmans D, Bosma M, Chi EH, Le Q, et al. Chain of thought prompting elicits reasoning in large language models. CoRR [Internet]. 2022;abs/2201.11903. https://arxiv.org/abs/2201.11903. Accessed on 01 Sep 2023.
17. Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y. Large language models are zero-shot reasoners. ArXiv [Internet]. 2022;abs/2205.11916. https://api.semanticscholar.org/CorpusID:249017743. Accessed on 01 Sep 2023.
18. Yang C, Wang X, Lu Y, Liu H, Le QV, Zhou D, et al. Large language models as optimizers. 2023. Accessed on 01 Sep 2023.
19. ChatGPT (Oct 12 version) [Internet]. L.L.C., San Francisco: OpenAI; 2023. https://beta.openai.com/docs/models.
20. Guyot P, Ades A, Ouwens MJ, Welton NJ. Enhanced secondary analysis of survival data: reconstructing the data from published Kaplan-Meier survival curves. BMC Med Res Methodol. 2012;12:9.
21. Van Rossum G, Drake FL Jr. Python reference manual. Centrum voor Wiskunde en Informatica Amsterdam; 1995.
22. Angelis A, Lange A, Kanavos P. Using health technology assessment to assess the value of new medicines: results of a systematic review and expert consultation across eight European countries. Eur J Health Econ. 2018;19:123–52.
23. Jenei K, Raymakers AJN, Bayle A, Berger-Thürmel K, Cherla A, Honda K, et al. Health technology assessment for cancer medicines across the G7 countries and Oceania: an international, cross-sectional study. Lancet Oncol. 2023;24:624–35.
24. Büssgen M, Stargardt T. Does health technology assessment compromise access to pharmaceuticals? Eur J Health Econ. 2023;24:437–51.
25. Akehurst RL, Abadie E, Renaudin N, Sarkozy F. Variation in health technology assessment and reimbursement processes in Europe. Value Health J Int Soc Pharmacoecon Outcomes Res. 2017;20:67–76.
26. Kamphuis B. Access to medicines in Europe: delays and challenges for access [Internet]. London School of Economics; 2021. https://doi.org/10.21953/0zaz-k994.
27. Incze A, Kaló Z, Espín J, Kiss É, Kessabi S, Garrison LP. Assessing the consequences of external reference pricing for global access to medicines and innovation: economic analysis and policy implications. Front Pharmacol. 2022;13: 815029.
28. Zhu X, Liu B. Launch delay of new drugs in China and effect on patients' health. Clin Ther. 2020;42:1750-1761.e7.
29. Padula WV, Kreif N, Vanness DJ, Adamson B, Rueda J-D, Felizzi F, et al. Machine learning methods in health economics and outcomes research-The PALISADE checklist: A good

practices report of an ISPOR Task Force. Value Health J Int Soc Pharmacoecon Outcomes Res. 2022;25:1063–80.
30. Davenport T, Kalakota R. The potential for artificial intelligence in healthcare. Future Healthc J. 2019;6:94–8.
31. Askin S, Burkhalter D, Calado G, El Dakrouni S. Artificial intelligence applied to clinical trials: opportunities and challenges. Health Technol. 2023;13:203–13.
32. Hendrix N, Veenstra DL, Cheng M, Anderson NC, Verguet S. Assessing the economic value of clinical artificial intelligence: challenges and opportunities. Value Health J Int Soc Pharmacoecon Outcomes Res. 2022;25:331–9.
33. Unsworth H, Wolfram V, Dillon B, Salmon M, Greaves F, Liu X, et al. Building an evidence standards framework for artificial intelligence-enabled digital health technologies. Lancet Digit Health. 2022;4:e216–7.
34. Vervoort D, Tam DY, Wijeysundera HC. Health technology assessment for cardiovascular digital health technologies and artificial intelligence: why is it different? Can J Cardiol. 2022;38:259–66.
35. Bélisle-Pipon J-C, Couture V, Roy M-C, Ganache I, Goetghebeur M, Cohen IG. What makes artificial intelligence exceptional in health technology assessment? Front Artif Intell. 2021;4: 736697.
36. de la Torre-López J, Ramírez A, Romero JR. Artificial intelligence to automate the systematic review of scientific literature. Computing. 2023;105:2171–94.
37. Blaizot A, Veettil SK, Saidoung P, Moreno-Garcia CF, Wiratunga N, Aceves-Martins M, et al. Using artificial intelligence methods for systematic review in health sciences: a systematic review. Res Synth Methods. 2022;13:353–62.
38. Kang J. Real-world data in health technology assessment: do we know it well enough? In: Bremer A, Strand R, editors. Precis oncol cancer biomark issues stake matters concern [Internet]. Cham: Springer International Publishing; 2022. p. 187–203. https://doi.org/10.1007/978-3-030-92612-0_12.
39. Hogervorst MA, Vreman RA, Mantel-Teeuwisse AK, Goettsch WG. Reported challenges in health technology assessment of complex health technologies. Value Health J Int Soc Pharmacoecon Outcomes Res. 2022;25:992–1001.
40. Breeze PR, Squires H, Ennis K, Meier P, Hayes K, Lomax N, et al. Guidance on the use of complex systems models for economic evaluations of public health interventions. Health Econ. 2023;32:1603–25.