Intelligent Systems with Applications 21 (2024) 200336

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

Claude 2.0 large language model: Tackling a real-world classification problem with a new iterative prompt engineering approach

Loredana Caruccio [a], Stefano Cirillo [a,*], Giuseppe Polese [a], Giandomenico Solimando [a], Shanmugam Sundaramurthy [b], Genoveffa Tortora [a]

[a] Department of Computer Science, University of Salerno, Fisciano, Salerno, Italy
[b] Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, Chennai, India

ARTICLE INFO

Keywords:
Claude 2.0
Large language model
Online learning
Machine learning
Massive online analytics
Forest cover-type

ABSTRACT

In the last year, Large Language Models (LLMs) have transformed the way of tackling problems, opening up new perspectives in various works and research fields, due to their ability to generate and understand human languages. In this regard, the recent release of Claude 2.0 has contributed to the processing of more complex prompts. In this scenario, the goal of this paper is to evaluate the effectiveness of Claude 2.0 in a specific classification task. In particular, we considered the Forest cover-type problem, concerning the prediction of a cover-type value according to the geospatial characterization of target worldwide areas. To this end, we propose a novel iterative prompt template engineering approach, which integrates files by exploiting prompts and evaluates the quality of responses provided by the LLM. Moreover, we conducted several comparative analyses to evaluate the effectiveness of Claude 2.0 with respect to online and batch learning models. The results demonstrated that, although some online and batch models performed better than Claude 2.0, the new iterative prompt engineering approach improved the quality of responses, leading to better performance with increases ranging from 14% to 32% in terms of accuracy, precision, recall, and F1-score.

1. Introduction

In the area of Natural Language Processing (NLP), Large Language Models (LLMs) represent advanced models based on deep neural networks, which are trained on huge amounts of text and are able to generate high-quality outputs. They represent a significant evolution in language processing thanks to their understanding of context, semantics, and grammar rules, by also transforming the way to approach problems related to human language. The LLMs have found applications in a wide range of research areas. For example, they have been employed for the analysis of large volumes of textual data (Yu et al., 2023), extracting knowledge from data (Haensch et al., 2023), assisting in the creation of high-quality contents (Xiao & Chen, 2023), and supporting clinicians for diagnoses identification (Caruccio et al., 2024). They also opened new perspectives for the automation of challenging tasks, such as improving human-machine interaction, by offering immediate and precise answers to user questions and/or assisting in writing and understanding tasks. Despite their powerful capabilities, communicating with LLMs is challenging. In fact, the correct definition of the prompts for a model, i.e., the inputs, is crucial since the choice of words, the structure of sentences, and the specificity of the provided information can affect the quality of the generated answers. Therefore, it is necessary to define focused prompting methodologies to interact with LLMs and achieve results that are as relevant as possible to submitted requests.

Among the best-known LLMs, we have GPT-3, GPT-4, and Google Bard, which have spread due to their advanced text generation capabilities and faster learning ability. Recently, a new LLM has been proposed, namely Claude 2.0, which is able to process not only textual prompts but also more complex prompts, namely multimodal prompts, which can include small- and medium-sized files. Nevertheless, designing multimodal prompts is more complex and it requires the definition of new advanced prompt engineering strategies, that can directly consider input files.

According to this scenario, in this paper, we propose a new iterative prompt template engineering strategy that integrates the use of files

* Corresponding author.
E-mail addresses: lcaruccio@unisa.it (L. Caruccio), scirillo@unisa.it (S. Cirillo), gpolese@unisa.it (G. Polese), gsolimando@unisa.it (G. Solimando), shanmugam.network13@gmail.com (S. Sundaramurthy), tortora@unisa.it (G. Tortora).

https://doi.org/10.1016/j.iswa.2024.200336
Received 2 August 2023; Received in revised form 26 January 2024; Accepted 1 February 2024
Available online 6 February 2024

within the prompt composition process. Moreover, multiple interactions with LLMs in the same context can progressively improve responses during a conversation since they allow the model to better understand the specific context and to refine outputs based on the information exchanged during the conversation, dynamically adapting them to the multiple requests (Feng et al., 2023). More specifically, in this paper, we delve into how such an iterative prompt engineering approach can influence the capabilities of LLM in tackling classification tasks, by focusing on the problem of forest cover-type identification. The achieved results have also been compared with those obtained by online and batch learning models in order to highlight the advantages and limitations of Claude 2.0 in such a classification process. In particular, the considered case study treats the attribution of a specific category of forest cover, such as firs, pines, and/or oaks, to a given area in order to identify the best forest planning and preserve biodiversity conservation. This is an extremely relevant problem, especially in the years in which we are witnessing an increase in environmental concerns, highlighting the importance of the sustainable management of forest resources.

Therefore, by considering the cover-type classification problem, we try to answer the following research questions (RQs):

RQ1: What are the benefits and limitations of using Claude 2.0 in classification tasks?
RQ2: How does Claude 2.0 compare with ChatGPT in classification tasks?
RQ3: How do batch and online learning models compare to Claude 2.0 in classification tasks?

The overall contributions of this research can be summarized as follows:

- A new processing pipeline for interacting with LLMs and using them in the forest cover-type classification problem;
- A new iterative prompt engineering approach specifically tailored for interacting with LLMs and achieving classification outcomes;
- A comparative analysis of the performances of Claude 2.0 and ChatGPT focused on the forest cover-type classification problem;
- A comparative analysis among batch and online machine learning models and Claude 2.0 on the forest cover-type classification problem, by measuring their performances in terms of accuracy, precision, recall, and F1-score.

The remainder of the paper is organized as follows. In Section 2 we describe relevant studies concerning the applications of LLMs in classification tasks. In Section 3 we introduce the problem of forest cover-type classification and preliminary notions on Claude 2.0. Section 4 discusses the new process pipeline underlying our study to address the problem of forest cover-type classification through predictive models and LLMs. Section 5 shows the prompt template engineering approach designed for interacting with LLMs. Section 6 provides an overview of the considered dataset, experimental settings, and evaluation metrics used in our analysis. Section 7 shows the results of Claude 2.0 in the forest cover-type classification tasks. Instead, in Section 8 we discuss the results of the comparative evaluation between Claude 2.0 and ChatGPT considering different test configurations and in Section 9 the ones of the comparative evaluation among Claude 2.0 and online and batch learning models. Finally, conclusions and future directions are discussed in Section 10.

2. Related work

This section discusses recent works on the forest cover-type prediction problem and how machine learning and LLMs can help in making predictions.

Forest cover-type prediction problem. In terms of agricultural and bioclimatic events, forests are crucial natural resources. The spatial distribution of forest cover is important because it provides a variety of ecosystem services that benefit humans. In (Kumar & Sinha, 2020), the authors make use of a publicly available forest cover-type dataset and the random forest machine learning technique, to classify cover-type values, by reaching an accuracy of 70.8% and performing better than any comparable approach, proposed so far. Instead, in (Kumar et al., 2022), an ensemble approach for classifying cover-types has been presented. It relies on machine learning-based classifiers, namely Decision Trees (DT), Random Forest (RF), and K-Nearest Neighbour (KNN). The approach outperforms the literature with an accuracy of 97.10%. An experimental analysis of the application of machine learning algorithms for predicting future shifts in forest cover and identifying their causes, has been provided in (Liu et al., 2020). In particular, nonparametric models such as Support Vector Regression (SVR), Artificial Neural Networks (ANN), Random Forest (RF), and Gradient-Boosted Regression Trees (GBRT) were evaluated in several case studies located in Tasmania, Australia. According to the results, RF performed better than SVR and ANN in terms of fitting and projection accuracy, whereas GBRT outperforms all other methods.

LLMs and its applications. In the last few months, researchers have extensively investigated the development and advancements of LLMs, such as ChatGPT, GPT-4, and Claude 2.0, and their ever increasing impact on applications, e.g., Natural Language Processing (NLP) (Schick & Schütze, 2020), question-answering (Caruccio et al., 2024), and machine translation and content generation (Martins et al., 2023, Rizou et al., 2023).

As LLMs continue to evolve, they have the potential to transform human-computer interactions. They are capable of analyzing huge volumes of text input, recognizing complicated patterns, and producing insightful outputs. On the other hand, they put themselves in danger by spreading inaccurate information, being politically biased, and not having access to secret information (Gao, 2023). This section provides an overview of recent LLM techniques and the use cases, in which LLMs have been applied.

To classify sentiment without training examples, in (Zhong et al., 2021), a prompting approach has been used with a zero-shot strategy. More specifically, the authors provided the Language Model (LM) with a review and a label description, such as “Does the user like this movie?” and asked if the next word should be “Yes” or “No”. Moreover, to remove the word misalignment still obtained by applying the zero-shot strategy, the authors also propose a meta-fine-tuning approach on a pre-trained LM by using a collection of datasets. Instead, in (Mu et al., 2023), the authors examine various prompting approaches by assessing ChatGPT and OpenAssistant with a zero-shot strategy in Computational Social Science classification tasks. In particular, the goal of this analysis was to investigate how the complexity of prompts, the defined labels, synonyms, and prior memories impact the prediction of the models. The evaluation results highlight that different prompting approaches have a substantial impact on the prediction performances independently from the specificity of the considered models. Several prompting LLM-based techniques have also been proposed in the medical domain. The latter represents one of the most critical environments in which LLM’s drawbacks can have a wide impact due to both the possible presence of personal information over clinical corpora and the necessity of providing accurate explanations in model-generated texts.

A groundbreaking clinical NLP framework based on prompts, namely HealthPrompt, has been proposed in (Sivarajkumar & Wang, 2022). It uses a prompt-based learning approach to construct task descriptions for clinical texts, revealing that prompts effectively capture the context of clinical texts and have good performances even in the absence of training data. Furthermore, in this domain, most of the studies focused on the analysis of prompting techniques applied to a specific LLM, i.e., ChatGPT (Eysenbach et al., 2023). As an example, to analyze the user interpretability of the model outputs, an evaluation of ChatGPT’s performances has been presented in (Gilson et al., 2023),

by considering the United States Medical Licensing Examination Step 1 and Step 2 exams. In particular, ChatGPT obtained high accuracy in the different configurations of the AMBOSS-Step1 dataset, reaching 64.4% on the best one. By comparing ChatGPT's results with the ones obtained by GPT-3 and InstructGPT, the authors highlighted that ChatGPT outperforms InstructGPT by 8.15% on average across all National Board of Medical Examiners (NBME) datasets. This highlights ChatGPT's potential as an interactive medical education tool for supporting learning.

Another interesting study in the medical domain is the zero-shot medical picture categorization framework presented in (Liu & Hu, et al., 2023). It makes use of CLIP and ChatGPT for explaining medical diagnosis. In particular, the authors used both private and public datasets to build an evaluation pipeline to assess the effectiveness and explainability capabilities of the proposed framework. Results demonstrated that the usage of the category names in queries, provided to LLMs, can improve the explainability performances of models underlying the proposed framework. Finally, the "DeID-GPT" de-identification framework based on GPT-4 has been proposed in (Liu & Hu, et al., 2023), aiming at hiding personally identifiable information in unstructured medical literature. By specifically focusing on Claude 2.0, its performances are analyzed in several domains, such as clinical, and financial, demonstrating its great understanding and technical capabilities. In particular, an interesting analysis of the performances of Claude 2.0 is presented in (Laban et al., 2023). The authors propose a FlipFlop experiment by considering six LLMs, namely LLaMA, Cmd, Mistral, GPT, PaLM, and Claude, on seven classification tasks. The experiment was split into two phases, the first related to the classification task, followed by an evaluation step to analyze the results and understand possible errors made by the LLMs, in which a new prompt with the correct answers associated with the data is provided to them. The results demonstrated the great technical and context understanding capabilities of GPT-4 and Claude 2.0, by achieving better performances than the other LLMs in terms of accuracy score.

All the above-described approaches highlight the potential of the usage of LLMs for solving problems in different critical contexts, but also the necessity to further investigate and refine LLM-based solutions. Moreover, the ever-increasing introduction of novel LLM models and versions, which can potentially introduce novel features and capabilities, demands new approaches and methodologies for their proper application in different use cases. With this paper, we addressed the forest cover-type classification problem by evaluating the potential of using one of the most recent LLM model versions, i.e., Claude 2.0, by also exploiting its new capability of managing file-based prompt strategies. This required the definition of a novel iterative prompt template engineering approach through which it has been possible to properly exploit Claude 2.0, evaluate its results on the considered problem, and compare Claude 2.0 performances with respect to the ones obtained with batch and online machine learning approaches.

3. Background

Forests cover about 30% of the Earth's land area and contain over 80% of terrestrial biodiversity. They provide vital ecosystem services such as carbon sequestration, water filtration, soil conservation, and habitat for wildlife. However, deforestation and forest degradation continue to be major threats to forest biodiversity worldwide. Advanced technologies like machine learning and artificial intelligence can assist in monitoring forests and analyzing threats to biodiversity. In this section, we will briefly provide an overview of the Forest cover-type Classification problem and the LLM we used in our study, i.e., Claude 2.0.

3.1. Problem overview

Over the past three decades, the world has lost 420 million hectares of forest, an area larger than India. The main factors that have led to deforestation include agricultural expansion, infrastructure development, logging, and wildfires. Deforestation is higher in tropical regions like the Amazon and Southeast Asia. According to the Food and Agriculture Organization (FAO), forests comprise one-third of the world's land area, but this proportion has decreased since the 1990s. As we well know, forests regulate carbon dioxide, water cycles, and land-atmospheric feedback. If the forests are cleared, the changes in land cover worsen the emission of greenhouse gases by around 15% in many regions. This greenhouse gas emission contributes to changes in patterns of rainfall, increasing temperatures, fluctuations in patterns of water, and an increased frequency of extreme weather conditions. Consequently, the significance of monitoring above-ground biomass (AGB) in deforested regions is also demonstrated.

The possibility of collecting and using data from sensors through advanced artificial intelligence models has led ecologists, biologists, and environmental scientists to involve these techniques in their studies, in order to improve the understanding and prediction of forest cover changes and to enable the conservation of forest biodiversity. The data currently available are synthetic aperture radar data, cartographic data, and textual data. This data is, mainly gathered through satellite sensors from two agencies, namely US Forest Service and US Geological Survey.

In this paper, we have chosen to study the forest cover-type problem, since, due to its multiple implications, which are becoming even more evident in the last years. To this end, we investigate how LLMs, which represent one of the most recent artificial intelligence techniques, can support experts to address this problem.

3.2. Claude 2.0 language model

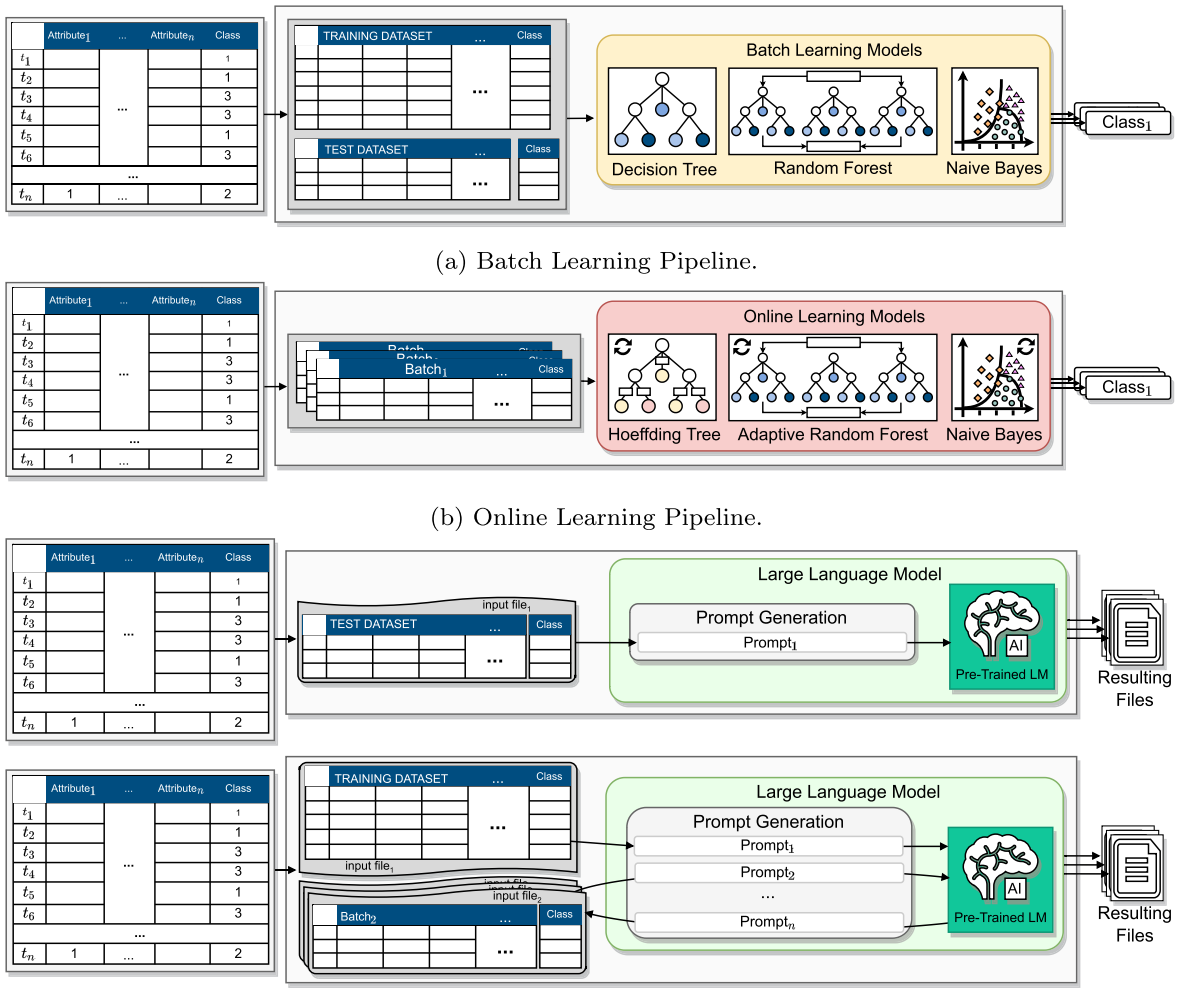
Anthropic Claude 2 AI, a.k.a., Claude 2.0,[1] is a second-generation artificial intelligence technique.

Compared to the previous version, Claude 2.0 significantly increased its processing capabilities, leading to managing up to 100,000 tokens, or approximately 75,000 words, in a single request, and enabling it to provide more context-related and refined responses.

Among the relevant features behind Claude 2.0, we have:

- Enhanced Performances: Claude 2.0 exceeds its predecessor in terms of performance by providing responses that are both longer and more thorough. It is also capable of processing large blocks of texts, providing users with more informative responses to their questions.
- API Access: Anthropic has made Claude 2.0 accessible to developers through its Application Programming Interface (API), allowing them to include this LLM in their own applications and services. This access to the API enables a wide variety of possibilities for the development of customized solutions and the improvement of user experiences.
- Website with a Public-Facing Beta Version: Anthropic has introduced a new website with a public-facing beta version named claude.ai, on which visitors can interact directly with Claude 2.0 through a direct experience with the platform.
- Multimodal Prompts: Claude 2.0 is able to process and interpret different types of input, such as text, code, and structured files of data, in order to generate comprehensive responses. Unlike other LLMs, which are often restricted to text-only prompts, Claude 2.0 offers enhanced flexibility in the definition of the prompts.
- Open Beta Testing: Anthropic has released Claude 2.0 for open beta testing, and the users have been invited to test out the LLM and

[1] www.claude.ai.



(a) Batch Learning Pipeline.

(b) Online Learning Pipeline.

(c) Large Language Model Pipelines.

Fig. 1. Pipelines of the processes for identifying cover-type using ML and LLMs.

provide insightful feedback, yielding the corporation to improve the functionalities of Claude 2.0.

Claude 2.0 is highly competitive with respect to other well-known LLMs, such as GPT and Google Bard, exhibiting high capabilities in text generation and data interpretation.

In our study, we analyzed the capabilities of Claude 2.0 for predicting forest cover-type values and compared its performances with the ones provided by batch and online learning models, such as Decision Tree, Random Forest, Adaptive Random Forest, Hoeffding Tree, and Naive Bayes, and Chat-GPT.

4. Communication pipeline between LLMs and predictive models

As introduced above, LLMs, such as Google Bard, GPT-3, and Claude 2.0, are powerful AI models that are capable of generating high-quality text, answering questions, completing sentences, and performing a wide range of natural language processing tasks. To interact with LLMs, it is necessary to define a prompt engineering methodology to drive the model in generating relevant responses, to be consistent with the requests of the users. Thanks to the new capabilities of Claude 2.0, it is possible to specify both textual and multimodal prompts, thus greatly extending the tasks in which these LLMs can be used. In fact, while textual prompts are simple instructions or questions that are given to the model, multimodal prompts combine text with other types of inputs, such as images or datasets, to get more complex and contextually rich answers. However, although defining textual prompt engineering methodologies is a challenging task, the definition of multimodal prompts is even more complex, since it is also necessary to keep track of how files need to be structured to maximize the understanding of the LLM. To this end, we have defined a new iterative prompt engineering methodology for interacting with Claude 2.0. Moreover, to perform a comparative analysis we also used several batch and online learning models leading to the necessity of designing specific pipelines for interacting with such models. Fig. 1 shows the pipelines for identifying forest cover-types using batch and online learning models and Claude 2.0. More specifically, Fig. 1a shows the pipeline for interacting batch learning models. As we can see, starting from a dataset in which a label is associated with each occurrence, to communicate with the batch learning models, we split the dataset into train and test sets, making sure that there is a correct distribution of the classes in the dataset. On the other hand, for online learning models, data is considered as a stream of information, leading the dataset to be split into multiple batches that are continuously read over time (Fig. 1b). In fact, these types of models generally process the data in multiple iterations, updating model parameters when a new batch of data arrives.

For interacting with the LLM through a multimodal prompt, we consider two different pipelines in order to compare the results of the LLM with those achieved by the models of both types of learning methodologies. Fig. 1c shows the two different pipelines for interacting with

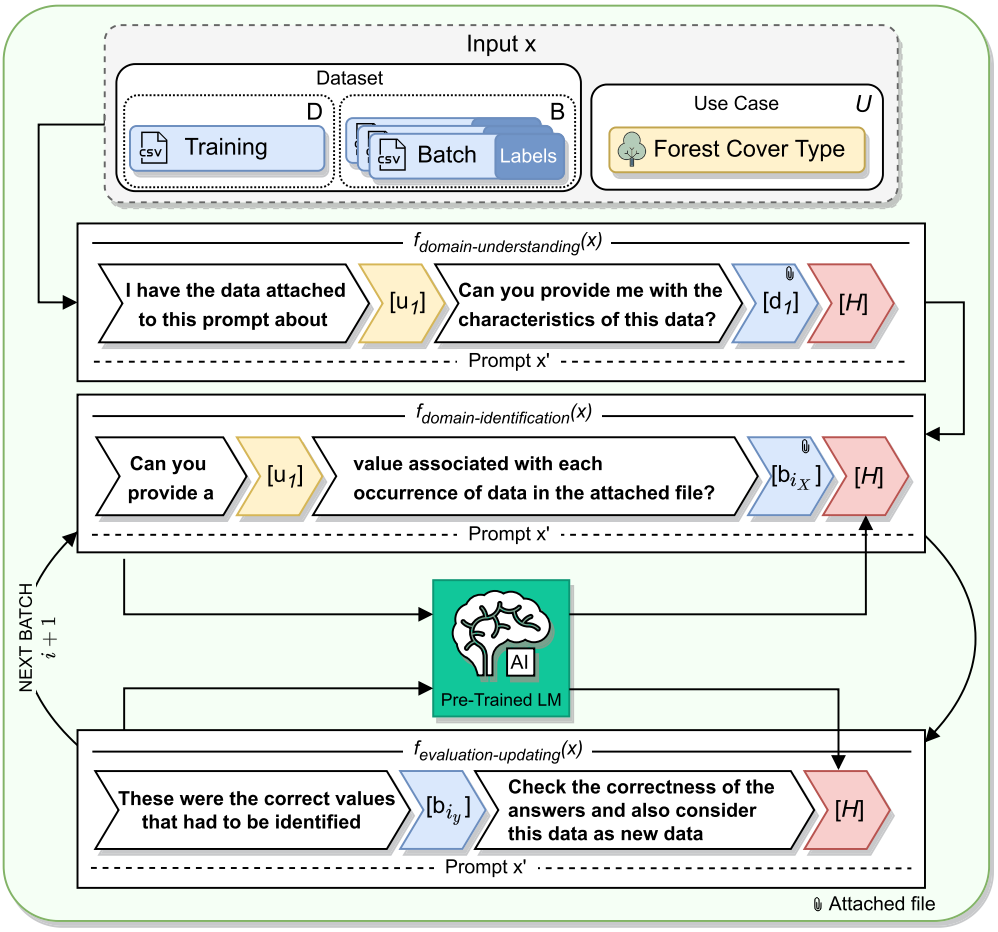


Fig. 2. Overview of the iterative prompt engineering methodology.

an LLM. The first pipeline simulates the interaction with batch learning models and considers a prompt with a batch of tuples representing the test data. In the prompt, the model is asked to predict the class associated to each occurrence in the test batch. In this way, it is possible to directly exploit the LLM to process inputs and analyze data with the aim of providing an answer to a classification task based on the LLM's knowledge (Fig. 1c, top). The second pipeline simulates the interaction with online learning models by considering an initial batch of tuples and several subsequent batches on which making predictions (Fig. 1c, bottom). The first prompt aims to provide the LLM with a sample of data to give them initial information on the problem at hand. In particular, the information attached to this prompt contains a snippet of the data with the corresponding expected responses, whereas the next prompts contain batches of tuples to classify. After a test batch response is provided by the LLM, to validate the results and understand possible errors, a new prompt with the correct answers associated with the data is provided to it.

Differently from the previous pipeline (i.e., Fig. 1c, top), in this pipeline (i.e., Fig. 1c, bottom), we exploit the fact that an LLM can refine answers during the continuous interaction and processing of batches of data, especially when the LLM can validate the outputs through specific prompts reporting any errors made.

In our study, we applied this pipeline to the problem of forest cover-type identification, considering as data to be processed the Forest cover-type dataset and using different batch and online predictive models. Furthermore, as said above, we have considered Claude 2.0, which is currently one of the few LLMs capable of processing small- and medium-sized datasets as input.

In the following sections, we will discuss in detail the prompt engineering strategy to communicate with Claude 2.0 and show the results obtained from the different experimental evaluations.

5. Iterative prompt engineering approach

The process of interacting with LLMs is a challenging problem, since the definition of appropriate prompts to interact with LLMs can affect the accuracy and the consistency of responses provided to users.

As introduced in the previous sections, there are mainly two categories of prompts, i.e., textual prompts and multimodal prompts. The textual prompts are composed of only textual content and their definition requires avoiding composing vague or verbose sentences. The multimodal prompts are composed of textual content and files and require combining the textual prompt with the attached files in order to achieve answers that also consider the file content. The definition of prompts for a classification task can be particularly complex. In fact, it is necessary to define a prompt to ask the LLM to assign correct labels to different data samples, by considering either the only knowledge on which the LLM has been pre-trained (zero-shot) or a few examples of classified data (few-shot). In both cases, it is important to balance prompt and data specificity without overloading the model with not useful information. In the literature, several prompt engineering methodologies have been defined (Chen et al., 2022, Liu & Yuan, et al., 2023), which have been applied in different areas, such as translation, text synthesis, and the classification tasks (Hegselmann et al., 2023). Nevertheless, it is necessary to define new prompt engineering strategies in order to interact with recent LLMs that also process multimodal prompts. To this end, in this study, we have proposed a new iterative prompt template engineering approach that aims to define textual and multi-

modal prompts for a classification task. In particular, we have designed three different ad-hoc prompting functions, i.e., $f_{\text{domain-understanding}}(x)$, $f_{\text{domain-identification}}(x)$, and $f_{\text{evaluation-updating}}(x)$, where x is the original input. Fig. 2 provides an overview of the iterative process for creating prompts. All the prompts have been created through the Manual Template Engineering approach, considered the most intuitive method for creating templates based on human insights (Liu & Yuan, et al., 2023).

Prompt 1. The first prompt was created with the aim of enabling the LLM to identify the context in which the classification task is performed, considering a small sample of data provided as input. For this prompt, we have defined a prompting function $f_{\text{domain-understanding}}(x)$ that aims to complete the sentence x in order to achieve the prompt sentence $x' = f_{\text{domain-understanding}}(x)$. The template of this sentence was defined as follows:

> **I have the data attached to this prompt about** [U]. **Can you provide me with the characteristics of this data?** [D] [H]

where [U] is the slot of the use cases in which the LLM must focus, [D] is the slot containing files with the sample data provided as input, and [H] is the slot related to the response of the LLM that will be filled with information about the data. More formally, the slot [U] is the set $U = \{u_1, u_2, \ldots, u_n\}$ of the use cases in which the LLM should perform classification tasks, and [D] is the set of input files $D = \{d_1, d_2, \ldots, d_m\}$ such that each file d_i is a pair $d_i = [d_{i_X}, d_{i_y}]$, in which d_{i_X} consists of independent variables of the dataset, and d_{i_y} are the target values.

It is important to notice that, in our study, we only consider a single use case, that is the problem of the forest cover-type identification. An example of the first prompt is:

> **I have the data attached to this prompt about forest cover-types. Can you provide me with the characteristics of this data?**

Prompt 2. The second prompt was created with the aim of asking the LLM to provide a classification on a batch of data in the analyzed use case. For this prompt, we have defined a prompting function $f_{\text{domain-identification}}(x)$ to achieve a prompt sentence $x' = f_{\text{domain-identification}}(x)$ using the following template:

> **Can you provide a** [U] **associated with each occurrence of data in the attached file?** [B] [H]

where [B] is the slot containing files of the samples of data to be classified. In particular, similar to slot [D] of the prompt 1, slot [B] is the set of batch files to be classified $B = \{b_1, b_2, \ldots, b_k\}$, where each batch file b_i is a pair $b_i = [b_{i_X}, b_{i_y}]$. For this prompt, we consider only the set b_{i_X} without target values, which will be used in the next prompt for evaluating the classification results achieved by the LLM. An example of the second prompt is:

> **Can you provide a forest cover-type value associated with each occurrence of data in the attached file?**

Prompt 3. The third prompt was created with the aim of making the LLM understand any mistakes made and of having an evaluation of the predictions made as output. Moreover, through this prompt, we guide the LLM to use the new data with the correct target values to update the initial sample of data and enrich their knowledge in the ongoing conversation. Similar to the other two prompts, for this prompt, we have defined a prompting function $f_{\text{evaluating-updating}}(x)$ that is designed according to the following template:

> **These were the correct values that had to be identified** [B] **Check the correctness of the answers and also consider this data as new data** [B] [H]

In this case, we consider only the set b_{i_y} of target values associated with the set b_{i_X} specified in the *Prompt 2* in order to evaluate the results of the LLM considering the expected values. An example of the third prompt is:

> **These were the correct values that had to be identified: Lodgepole Pine, Spruce/Fir, Spruce/Fir, Krummholz. Check the correctness of the answers and also consider this data as new data.**

By considering the above-defined three prompts, the iterative process starts by providing the LLM a sample of the occurrences to be classified, following Prompt 1. In the case of Claude 2.0, we consider the sample attached to a file in a prompt, but this can also be used with textual prompts including the sample of the data in the prompt. This step aims to provide the LLM with targeted examples of the classification problem which, added to its basic knowledge, should support it in solving the classification task. Starting from this, the composition of Prompt 2 and Prompt 3 allows us to create an iterative process able to communicate several times with the LLM and make predictions on different data samples. In fact, after completing the processing of Prompt 3, it is possible to compose a new classification request of another batch through Prompt 2, and evaluate the results through Prompt 3. In this way, this approach is able to exploit the ability of the LLM to progressively improve responses during a conversation with multiple interactions, also known as Iterative Refinement (Feng et al., 2023). Notice that we considered several batches of data since existing LLMs have an upper limit on the size of files that can be attached to prompts. Nonetheless, through a prompt engineering approach relying on batches, it is possible to apply the above-defined strategy considering the only limit to the maximum size of files that can be attached to prompts.

6. Experimental evaluation

In this Section, we first introduce the dataset involved in our evaluation and the preprocessing steps to evaluate their structure and to make it suitable for our evaluations. Then, we introduce the experimental settings and the evaluation metrics adopted for evaluating performances achieved by online and batch learning models and Claude 2.0.

6.1. Forest cover-type dataset

For our experimental evaluation, we adopt the *Forest CoverType* dataset publicly available on the UCI Machine Learning Repository.[2] This dataset represents one of the largest datasets and merges three observations from four different areas of the Roosevelt National Forest in Colorado. The dataset contains both numerical and binary features, with a total of 580,000 rows and 54 attributes, including 10 quantitative features and 44 binary features representing various soil types and wilderness areas. The quantitative attributes include elevation, aspect, slope, horizontal distance to water sources, horizontal distance to roadways, and others. Each instance represents a 30×30-meter cell of forest land. The actual forest cover-type for a given observation was determined from US Forest Service (USFS) Region 2 Resource Information System (RIS) data. Independent variables were derived from data originally obtained from US Geological Survey (USGS) and USFS data. Starting from the recent literature focused on the problem of forest cover-type, which considered the same dataset, it has been demonstrated that some attributes are not relevant in the classification of forest cover-type values (Gupta et al., 2015, Mohammed Al Sameer et

[2] www.kaggle.com/datasets/uciml/forest-cover-type-dataset.

Table 1
Attributes of the observations of the Forest cover-type dataset.

Data Type	Attributes Name	Description	#Features	Types
Quantitative	Elevation	Elevation in meters	1	Number
Quantitative	Aspect	Aspect in degrees azimuth	1	Number
Quantitative	Slope	Slope in degrees	1	Number
Quantitative	Horizontal Distance To Hydrology	Horizontal distance to nearest surface water	1	Number
Quantitative	Vertical Distance To Hydrology	Vertical distance to nearest surface water	1	Number
Quantitative	Horizontal Distance To Roadways	Horizontal distance to nearest roadway	1	Number
Quantitative	Hillshade 9 am	Hill shade index at 9 am, summer solstice	1	In range [0 to 255[
Quantitative	Hillshade Noon	Hill shade index at noon, summer solstice	1	In range [0 to 255[
Quantitative	Hillshade 3 pm	Hill shade index at 3 am, summer solstice	1	In range [0 to 255[
Quantitative	Horizontal Distance To Fire Points	Horizontal distance to nearest wildfire ignition points	1	Number

Table 2
Distribution of the 7 cover-type classes of the Forest cover-type dataset.

Target Class	Distribution (%)
Lodgepole Pine	48.76
Spruce/Fir	36.46
Ponderosa Pine	6.15
Krummholz	3.53
Douglas-fir	2.99
Aspen	1.63
Cottonwood/Willow	0.47

al., 2021). In particular, results revealed that the quantitative feature exhibited high importance, while other binary features showed poor correlation. To this end, in our study, we only consider the quantitative features in the *Forest CoverType* dataset. Table 1 provides an overview of the considered features in the dataset by also detailing their meaning and data types. Each occurrence in the dataset is associated with a coverage type value ranging from 1 to 7, each corresponding to a specific forest vegetation category, i.e., Spruce/Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-Fir, and Krummholz, respectively. However, as shown in Table 2, the balance between the classes is not optimal, since the Lodgepole Pine and Spruce/Fir classes have a higher number of occurrences than the other classes, while the other classes have only a few occurrences. This produces an imbalance in the dataset that can affect the results of predictive models.

As we can see, the dataset contains a strong data imbalance, and the difference between the number of occurrences of the major cover-types, i.e., Lodgepole Pine and Spruce/Fir, and all the others is very large. To this end, we performed a data rebalancing to achieve an equal distribution of data occurrences of the different cover-types. Through an undersampling approach to the majority classes, it is possible to provide a more balanced snapshot of the data to the LLM within the token boundary, allowing the model to get adequate exposure to all classes during training. Furthermore, it is important to notice that LLMs often have constraints on how many input tokens they can process in a single pass.

6.2. Experimental settings and evaluation metrics

The experimental evaluation has been conducted using the dataset shown in Section 6.1 downstream of the pre-processing steps to balance the data occurrences for each type of cover-type. In order to make the Forest cover-type dataset also suitable for online scenarios and for the iterative prompt engineering approach, the occurrences of the dataset have been split into multiple batches. More specifically, we performed different experimental sessions with the aim of evaluating the behavior of Claude 2.0 in different scenarios and comparing it with machine learning models and other LLMs. In order to investigate the capabilities of the LLM, we first conduct an ablation study to discern how the input features affect the classification of Claude 2.0. Than of this evaluation, we also investigated how the complexity of the classification task affects the performance of Claude 2.0, by considering different cover-types at a time, i.e., 2, 3, 4, 5, 6, and 7. The cover-types have been chosen according to the distribution of occurrences in the original dataset, and for each evaluation, we apply an undersampling approach to rebalance the number of occurrences for the different cover-types. In this scenario, the outputs produced by LLMs can exhibit considerable variability across different executions, even when the prompt and model configuration remain unchanged (Chang et al., 2023).

This variability probably stems from multiple factors related to the prompt and dataset, as well as the statistical nature of the models. For this reason, we performed multiple executions considering different cover-types at a time, in order to determine confidence intervals around the evaluation metrics, providing a robust estimate of the average performance of Claude 2.0. For all evaluations, we adapted the dataset size according to the models and/or LLMs involved in the experimental session. However, only in the case of online learning models, these batches have been parsed in a stream of data by means of generators provided by Massive Online Analysis (MOA) and Scikit Multiflow frameworks. Instead, for the other batch learning models, we have split the dataset into 80% and 20%, by balancing the number of different classes in the training set.

To make Claude 2.0 suitable for the comparison with batch learning models, we consider a single prompt as training, in which the LLM tries to understand data and perform a descriptive analysis of them. Then, the LLM classifies the instances in the test set, giving a cover-type value for each observation. On the other hand, concerning the comparison with online learning models, we first provide a smaller sample of data to the LLM and then we split the test set into several batches to perform an evaluation similar to the online learning approach. For this experimental evaluation, we reduced the dataset size due to the maximum length limit of prompts for Claude 2.0. Furthermore, we limited the choice to three predictive models, since these models have been developed in both batch and online learning versions, i.e., Random Forest, Decision Tree, and Naive Bayes.

The results achieved by Claude 2.0 have also been compared with those of ChatGPT, highlighting the strengths and weaknesses of both LLMs in the classification tasks. We use Claude 2.0 and ChatGPT in their latest versions available online at the time of the composition of this work. Both the LLMs have been chosen due to the large number of parameters and data on which they have been trained. However, different from ChatGPT,[3] Claude 2.0 enables us to directly interact using multimodal prompts.

[3] The free version of ChatGPT only allows the use of textual prompts.

In order to evaluate the performances achieved by the models involved in our study, we considered Accuracy, Kappa score, Precision, Recall, and F1-Score evaluation metrics. They are defined in terms of the number of True Positives (TP), i.e., when a forest cover-type was correctly predicted; False Positives (FP), i.e., where a not forest cover-type was wrongly predicted, True Negatives (TN), i.e., when a forest cover-type was wrongly predicted, and False Negatives (FN), i.e., when a not forest cover-type was accurately predicted. In what follows, we provide details about these metrics:

- **Accuracy**: Percentage of occurrences successfully classified by the model so far: $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$.
- **Kappa score**: A statistic that is used to measure inter-rater reliability, a score of how much homogeneity exists in the classification.
- **Precision**: The ratio of correctly predicted positive observations to all positive observations in the positive class: $Precision = \frac{TP}{TP+FP}$.
- **Recall**: The ratio of correctly predicted positive observations to all observations in the positive class: $Recall = \frac{TP}{TP+FN}$.
- **F1-score**: A weighted average of precision and recall, leading to take into account both FP and FN: $F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$.

In the following sections we try to answer the RQs discussed in Section 1, according to the experimental settings presented in this section.

7. RQ1: what are the benefits and limitations of using Claude 2.0 in classification tasks?

As introduced above, Claude 2.0 is a new LLM model developed in 2023 by Anthropic PBC, AI safety and research company based in San Francisco.[4] This model seems extremely hopeful for automating a wide variety of text classification applications. The pre-trained model provides a powerful starting point, preventing users from building solutions entirely from scratch for every new use case. This transfer learning capability is quite valuable given the time and data requirements of developing robust classifiers. The pre-trained capabilities can be transferred to new classification problems through fine-tuning, allowing for quicker and more accurate learning on limited data. In order to study the effectiveness of Claude 2.0 in the classification of forest cover-types, we first conduct an ablation study to identify relevant features for the considered problem, which are then used in the experimental evaluations. Thereafter, we analyze the effectiveness of Claude 2.0 by considering different types of forest cover to be classified.

Ablation study. To select the most representative features on the forest cover-type classification task with the considered dataset, we have performed a comparison between the performances of Claude 2.0 by using the features elected as meaningful by the literature (see Section 6.1) and the performances of Claude 2.0 by using the features obtained through the application of 3 feature selection approaches, i.e., Recursive Feature Elimination (RFE), Recursive Feature Elimination with Cross-Validation (RFECV),[5] and SHAP-Selection (SHAP) (Marcílio & Eler, 2020).

The RFE approach is a feature selection technique that iteratively prunes the least useful feature and selects a subset of informative features for a prediction problem. It wraps a machine learning model that is capable of assigning importance scores to the input features. RFE starts by training the model on the full set of candidate features and then the model computes an importance score for each feature, reflecting how useful it is for the model's predictions. The feature with the lowest importance score is then removed from the feature set.

Concerning RFECV, it wraps predictive models to perform recursive feature elimination with cross-validation to select the optimal number of features to infer the optimal dimensionality of the feature space.

[4] www.anthropic.com.
[5] www.scikit-learn.org.

Table 3
Ablation study on feature selection.

Features Name	RFE	RFECV	SHAP
Elevation	√	√	√
Aspect	√	✗	✗
Slope	✗	√	✗
Horizontal Distance To Hydrology	✗	✗	✗
Vertical Distance To Hydrology	✗	√	√
Horizontal Distance To Roadways	✗	✗	✗
Hillshade 9 am	√	√	√
Hillshade Noon	√	√	✗
Hillshade 3 pm	✗	√	√
Horizontal Distance To Fire Points	✗	✗	✗

Table 4
Results of Claude 2.0 model with different feature sets.

	Set of Features	Accuracy	Precision	Recall	F1-Score
Claude 2.0	RFE	0.14	0.11	0.12	0.11
	RFECV	0.16	0.12	0.14	0.12
	SHAP	0.20	0.18	0.20	0.19
	RFE + RFECV	0.23	0.19	0.21	0.22
	RFE + SHAP	0.25	0.23	0.23	0.22
	RFECV + SHAP	0.24	0.22	0.21	0.22
	RFE + RFECV + SHAP	0.25	0.23	0.22	0.22
	ALL FEATURES	**0.42**	**0.49**	**0.46**	**0.45**

By evaluating feature importance and model performances using cross-validation at each iteration, RFECV should reduce overfitting compared to RFE (Nie et al., 2023). It selects the optimal number of features that maximize the model's prediction accuracy.

The last approach, i.e., SHAP, is a model interpretation method that explains individual predictions by determining the contribution of each feature to the model's output. In particular, for each feature, the SHAP value indicates the degree to which that feature has contributed to the predicted output and how it affected the deviation from the expected value.

Table 3 shows the selected features resulting from the application of each previously presented feature selection approach, by considering all the 10 quantitative attributes, i.e., Elevation, Aspect, Slope, Horizontal Distance To Hydrology, Vertical Distance To Hydrology, Horizontal Distance To Roadways, Hillshade 9 am, Hillshade Noon, Hillshade 3 pm, Horizontal Distance To Fire Points, as we motivated in Section 6.1.

Concerning the RFE approach, it assigned 4 out of 10 features to the highest weights, i.e., considering them the most representative and important features. Instead, for the second approach, i.e., RFECV, 6 out of 10 features achieved high values of weights. Finally, for the SHAP approach, only 4 features out of 10 resulted in actively contributing to the prediction of the model output.

By considering the set of features selected by each approach, their combinations, and the total set of the considered features, we show the results of a comparative evaluation of the Claude 2.0 performances in Table 4. More specifically, we performed a cross-evaluation of RFE, RFECV, and SHAP by merging feature sets identified by each approach in all possible combinations of them. As we can see, the results show that, by individually considering each approach, the accuracy, precision, recall, and F1-score values never exceed 0.25, 0.23, 0.22, and 0.22, respectively. In all three cases, the limited number of features selected by each approach was likely insufficient for identifying the correct forest cover-type.

Moreover, the evaluation performed by combining the feature sets resulting from the different approaches also showed that the metric

Table 5
Results of Claude 2.0 model with a different number of samples.

	Forest Cover Types [#]	Accuracy	Precision	Recall	F1-Score
Claude 2.0	**2**	± 0.54	± 0.42	± 0.41	± 0.46
	3	± 0.29	± 0.22	± 0.22	± 0.22
	4	± 0.13	± 0.02	± 0.13	± 0.06
	5	± 0.07	± 0.04	± 0.13	± 0.21
	6	± 0.28	± 0.21	± 0.21	± 0.23
	7	± 0.43	± 0.35	± 0.34	± 0.32

Table 6
Results of ChatGPT model with a different number of samples.

	Forest Cover Types [#]	Accuracy	Precision	Recall	F1-Score
ChatGPT	**2**	± 0.21	± 0.14	± 0.13	± 0.13
	3	± 0.10	± 0.07	± 0.07	± 0.08
	4	± 0.26	± 0.20	± 0.22	± 0.22
	5	± 0.18	± 0.13	± 0.12	± 0.13
	6	± 0.13	± 0.10	± 0.12	± 0.14
	7	± 0.16	± 0.13	± 0.14	± 0.13

scores still remain low. Among these, the combination of RFECV and SHAP approaches (i.e., RFECV + SHAP) achieved the best performances, reaching values of 0.24, 0.22, 0.21, and 0.22 for the accuracy, precision, recall, and F1-score, respectively.

Table 4 also shows that Claude 2.0 presented the best performances for all four metrics with the configuration in which all quantitative features are considered meaningful for the prediction task. In fact, in this configuration, Claude 2.0 achieved 0.42, 0.49, 0.46, and 0.45, for accuracy, precision, recall, and F1-score, respectively. This is probably due to the fact that the LLM is more capable of discerning the cover-type value in the presence of more precise information in the prompt.

Evaluation by varying the number of cover-types. Table 5 shows the average results achieved by Claude 2.0 considering different numbers of forest cover-types, i.e., 2, 3, 4, 5, 6, and 7 target classes, by performing three executions for each cover-type. As we can see, Claude 2.0 achieves better performances when considering two target classes, i.e., 0.54, 0.42, 0.41, and 0.46 for the accuracy, precision, recall, and F1-score, respectively. In general, as expected, Claude 2.0 has shown good performance in the identification of a few forest cover-types, and it can make predictions with more difficulty when the number of classes increases. Nevertheless, results achieved in the evaluation with the largest number of cover-types, highlight an improvement in the performance of Claude 2.0. Moreover, by deepening the achieved results, we noticed that most of the occurrences correctly classified were to the Ponderosa Pine and Douglas-fir cover-types. In fact, the data related to these two types of cover values show an *Elevation* value that ranges from 1800 to 2800 meters for Ponderosa Pine, and from 2400 to 3000 meters for Douglas-fir. These values are mainly related to these cover-types, enabling the LLM to correctly identify most of the Ponderosa Pine and Douglas-fir types. By summarizing, the results demonstrate that Claude 2.0 is capable of capturing the characteristics of the data allowing it to discriminate the correct cover-type classes to associate with new instances of data. Moreover, we have noticed that the classes from which the model achieved incorrect results are *Spruce/Fir*, *Cottonwood/Willow*, *Krummholz* and *Aspen*. In fact, these cover-types share many attribute values, such as *Hillshade 9 am*, *Hillshade Noon*, and *Hillshade 3 pm* attributes, that can be easily confused with other cover-types.

8. RQ2: how does Claude 2.0 compare with ChatGPT in classification tasks?

Recently, the introduction of advanced generative LLMs, such as ChatGPT and Claude 2.0, has led to the possibility of tackling a large number of tasks of different natures, by only defining a prompt to interact with these models. As we have introduced above, some peculiarities that distinguish Claude 2.0 from other LLMs now in circulation are related to the ability to receive textual endings as attachments, and the possibility of providing detailed analyses and considerations of the file structure on these. ChatGPT is an AI-powered language model developed by OpenAI, capable of generating human-like text based on context and past conversations. Differing from Claude 2.0, the open version of ChatGPT offers limited context-handling capabilities compared to Claude 2.0. In fact, it does not offer additional features and is much more limited than the paid version. On the other hand, Claude 2.0 seems to offer a larger set of features in different fields, including the possibility to directly analyze data from files. Claude 2.0 and ChatGPT have different strengths: the first focuses on greater information capacity and more human writing, whereas the latter offers advantages in technical scope and capacity.

Table 6 shows the results achieved by ChatGPT considering the same experimental settings used for Claude 2.0 in Section 7. As we can see, ChatGPT proves to have underperforming and less robust common sense reasoning on the cover-type classification task with respect to Claude 2.0 (see Table 5). In fact, ChatGPT has achieved the best scores for the classification considering two and four target classes, with a value that does not exceed 0.26 for accuracy, precision, recall, and F1-score, respectively. From the results, we have noticed that ChatGPT distinguishes cover-types more difficult than Claude 2.0, misclassifying many of the target classes with easily identifiable values, such as Ponderosa Pine and Douglas-fir. As discussed in the previous section, Claude 2.0 also suffered the same difficulties due to an insufficiently discriminating set of attributes for certain cover-types. However, we can see that the performances of ChatGPT compared to Claude are more affected by these problems, probably due to the analytical capability of the data included in Claude 2.0.

In summary, the results show that the best-performing model is Claude 2.0 compared to ChatGPT. The performance resulting from this analysis showed the better performance of Claude 2.0 for our classification task.

9. RQ3: how do batch and online learning models compare to Claude 2.0 in classification tasks?

Motivated by the idea of understanding the performance of Claude 2.0 in the cover-types classification task, we discuss the result achieved by machine learning models, i.e., batch and online models, compared with Claude 2.0.

Traditional machine-learning approaches typically involve creating a model based on a dataset of pre-defined features and labels. The model is then trained on this data to predict labels for new instances. These approaches require significant effort to manually design and select appropriate features, and require large amounts of high-quality labeled data for the training step. When we consider the problem of cover-types classification, traditional machine learning approaches are limited by the quality and quantity of data, as well as the ability to manually identify and select relevant features. Instead, an LLM is able to learn from a wide range of text data and can generate responses based on natural language input. This can make LLMs, such as Claude 2.0, more flexible and powerful models for classification tasks, especially in cases where there is limited or noisy data available.

To compare the performances of traditional machine learning approaches with Claude 2.0 for cover-types classification using the dataset shown in Section 6.1, we have performed an experimental evaluation adopting three classifiers and their version for online learning tasks, i.e., Random Forest, Decision Tree, and Naive Bayes.

Table 7
Results of batch and online learning models and Claude 2.0.

		Train Batch [#]	Test Batch [#]	Accuracy	Kappa	Precision	Recall	F1-score
Batch Learning	**Random Forest**	1	1	**0.87**	-	**0.87**	**0.87**	**0.87**
	Decision Tree			0.80	-	0.80	**0.87**	0.80
	Naive Bayes			0.60	-	0.64	0.60	0.56
Online Learning	**Adaptive RF**	1	7	0.41	0.31	0.36	0.36	0.36
	Hoeffding Tree			**0.52**	**0.45**	**0.51**	**0.46**	**0.48**
	Naive Bayes			0.42	0.35	0.41	0.36	0.38
Claude 2.0	**Batch**	1	1	0.14	-	0.16	0.14	0.14
	Online	1	7	**0.46**	**0.23**	**0.34**	**0.28**	**0.31**

Table 7 shows the results achieved by the classification models on the two different types of machine learning techniques, i.e., batch and online learning. As we can see, for batch learning, the best performing model is Random Forest with values of about 0.87, for accuracy, precision, recall, and F1-score. The other batch models achieve higher performance with an accuracy value greater than 0.60, i.e., Decision Tree and Naive Bayes.

Concerning the online learning models, as we can see from Table 7, Hoeffding Tree and Naive Bayes models achieved the highest accuracy value, achieving values of 0.523, 0.451, 0.512, 0.461, and 0.482, for accuracy, kappa, precision, recall, and F1-score, respectively. Moreover, comparing the results achieved by the other classifiers, we can see accuracy values of 0.523 and 0.423 for Hoeffding Tree and Naive Bayes, respectively.

Regarding the evaluation of Claude 2.0 simulating a batch learning approach, as we can notice, it achieves a value of 0.14 for accuracy, recall, and F1-score value, and a precision of 0.16. In this case, the aim of this experiment was to understand the capabilities of Claude 2.0 when performing a similar batch-learning task as an ML model. The results shown in the table regarding Claude 2.0 performing the classification task on seven cover-type values, in one execution. In this case, the LLM receives a training dataset on which it makes an analysis for understanding the data, simulating the training step of an ML model. In the second step, i.e., the classification step, the test set was given to the LLM to predict the target classes. This strategy proved to be unsuccessful for our classification task, probably due to the fact that Claude 2.0 is not able to identify a large set of patterns from data, only providing a detailed analysis at a high level.

On the other hand, for online learning evaluation, we can see how the performances of Claude 2.0 outperformed the classification task compared to the previous evaluation. As we discussed in Section 7, the recent LLMs have the capabilities to have the ability to refine their capabilities, trying to acquire knowledge from the conversations, in order to perform iterative learning. In this evaluation, we try to simulate an online learning task considering a small batch as a pre-train of the model and passing different other batches according to the Iterative Prompt Engineering approach defined in Section 5. As we can see from Table 7, Claude 2.0 achieves an accuracy of 0.46, a kappa of 0.23, a precision of 0.49, a recall of 0.46, and an F1-score of 0.45. The results shown in the table regarding Claude 2.0 performing the classification task on seven cover-type values, in one execution. Although the values achieved simulating online learning tasks are not higher than those achieved by traditional ML models, they prove that an iterative engineering approach can enable the LLM to achieve higher performances with respect to the evaluation in which we consider one training dataset and one test set. Moreover, it is necessary to consider that the LLMs, such as Claude 2.0, continue to be limited in the number of interactions and prompt sizes. However, considering a small set of batches, we were able to demonstrate that this approach has the potential to improve classification tasks performed with LLMs, by leveraging the iterative learning methodology underlying LLMs.

10. Conclusion and future directions

This work highlighted the versatility and some of the capabilities of one of the most recent LLMs, i.e., Claude 2.0, in addressing classification tasks, focusing on the forest cover-type classification problem. The characteristics of Claude 2.0 make it able to process multimodal prompts and integrate small or medium-sized files, also opening new horizons in advanced data analytics and processing studies. In this scenario, we have proposed a new iterative prompt engineering approach that aims to provide the model with a continuous evaluation of the answers given during the classification process. This enabled us to both continuously analyze Claude 2.0's errors and refine its subsequent answers. Furthermore, we performed several experimental sessions to evaluate the behavior of Claude 2.0 in different scenarios using the new iterative prompt engineering approach, with the aim of evaluating the effectiveness of the new approach and the performance of Claude 2.0 compared to those obtained by other models. To this end, we conducted an ablation study using three feature selection approaches to discern how various input features affect the classification performances of Claude 2.0. The results highlighted that Claude 2.0 achieved the best performance considering all the quantitative attributes. This is probably due to the fact that the LLM is able to understand the cover-type value to predict in the presence of more information, regardless of the intrinsic knowledge on which it has been trained. Moreover, we investigated how the complexity of the classification task affects the performance of Claude 2.0, by considering a different number of cover-types to be classified at a time. The results achieved in these evaluations have been also compared with those obtained by ChatGPT, in order to highlight the strengths and the weaknesses of both LLMs. The results suggest that ChatGPT can distinguish cover-types with greater difficulty than Claude 2.0, misclassifying many of the target classes. This is probably due to the fact that the open version of ChatGPT offers limited context-handling capabilities and minor technical performances compared to the ones of Claude 2.0. Moreover, we conducted a comparative evaluation of the performances of machine learning models, i.e. batch and online models, against the ones of Claude 2.0 in the task of classifying cover-types. The results highlighted that traditional machine-learning models tend to outperform LLMs in classification tasks. This suggests, that despite the significant advances in the field of LLMs, there is still room for improvement and optimization of such models when performing specific classification tasks. As an insight, all the experimental sessions proved that the new Iterative Prompt Engineering approach is effective, resulting in Claude 2.0 increasing its performance in the classification task. This is probably due to the iterative refinement process enabling the prompts to effectively guide Claude 2.0 in extracting and utilizing relevant features from the input data. Consequently, this approach facilitates the understanding of the underlying patterns and relationships

within the considered data, leading to enhanced identification of the different forest cover-types.

It is important to notice that, although the new prompt engineering approach demonstrated to improve the capabilities of Claude 2.0 in the zero-shot classification task in single conversations, the main advantage lies in its capability to foster increasingly coherent and contextually relevant conversations. However, due to the nature of Claude 2.0, i.e., a non-open source LLM, it is difficult to interpret how the new iterative prompt engineering approach affects the components of its architecture. Additionally, the results discussed in this study are focused on the forest cover-type problem and its related dataset, which leads to all the comparative considerations limited to them. In the future, it would be interesting to further refine the iterative prompt engineering approach, trying to maximize the potential of other LLMs. Furthermore, to better investigate the behavior of Claude 2.0 in classification tasks in new domains, it would be interesting to consider the possibility of equipping the prompts with larger files and documents related to the considered domain, such as technical documents or domain-specific literature. Moreover, for the freely released LLMs, it would be interesting to investigate how tuning the model for a specific problem by directly extending the model's capabilities. Finally, another perspective could be the integration of LLMs and ML models, in order to provide effective tools that combine the strengths of both types of models.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Caruccio, L., Cirillo, S., Polese, G., Solimando, G., Sundaramurthy, S., & Tortora, G. (2024). Can chatgpt provide intelligent diagnoses? A comparative study between predictive models and chatgpt to define a new medical diagnostic bot. *Expert Systems with Applications*, *235*, Article 121186.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2023). A survey on evaluation of large language models. arXiv:2307.03109.

Chen, X., Zhang, N., Xie, X., Deng, S., Yao, Y., Tan, C., Huang, F., Si, L., & Chen, H. (2022). Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web conference 2022* (pp. 2778–2788).

Eysenbach, G., et al. (2023). The role of chatgpt, generative language models, and artificial intelligence in medical education: A conversation with chatgpt and a call for papers. *JMIR Medical Education*, *9*, Article e46885.

Feng, J., Tao, C., Geng, X., Shen, T., Xu, C., Long, G., Zhao, D., & Jiang, D. (2023). Knowledge refinement via interaction between search engines and large language models. arXiv preprint arXiv:2305.07402.

Gao, A. (2023). *Implications of chatgpt and large language models for environmental policymaking*. Available at SSRN 4499643.

Gilson, A., Safranek, C. W., Huang, T., Socrates, V., Chi, L., Taylor, R. A., Chartash, D., et al. (2023). How does chatgpt perform on the United States medical licensing examination? The implications of large language models for medical education and knowledge assessment. *JMIR Medical Education*, *9*, Article e45312.

Gupta, A., Jagadeesh, R., Sahwney, H., & Zalani, Z. (2015). Classifying forest categories using cartographic variables. Technical Report, Indian Institute of Technology.

Haensch, A. C., Ball, S., Herklotz, M., & Kreuter, F. (2023). Seeing chatgpt through students' eyes: An analysis of tiktok data. arXiv preprint arXiv:2303.05349.

Hegselmann, S., Buendia, A., Lang, H., Agrawal, M., Jiang, X., & Sontag, D. (2023). Tabllm: Few-shot classification of tabular data with large language models. In *International conference on artificial intelligence and statistics* (pp. 5549–5581). PMLR.

Kumar, A., Maurya, P., Tiwari, S. M., Ali, A., Vasisht, H., & Baghel, A. S. (2022). Classification of forest cover-type using ensemble of decision tree, random forest and k nearest neighbor. *JIMS8I International Journal of Information Communication and Computing Technology*, *10*, 615–619.

Kumar, A., & Sinha, N. (2020). Classification of forest cover type using random forests algorithm. In *Advances in data and information sciences: Proceedings of ICDIS 2019* (pp. 395–402). Springer.

Laban, P., Murakhovs'ka, L., Xiong, C., & Wu, C. S. (2023). Are you sure? Challenging llms leads to performance drops in the flipflop experiment. arXiv:2311.08596.

Liu, B., Gao, L., Li, B., Marcos-Martinez, R., & Bryan, B. A. (2020). Nonparametric machine learning for mapping forest cover and exploring influential factors. *Landscape Ecology*, *35*, 1683–1699.

Liu, J., Hu, T., Zhang, Y., Gai, X., Feng, Y., & Liu, Z. (2023). A chatgpt aided explainable framework for zero-shot medical image diagnosis. arXiv preprint arXiv:2307.01981.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, *55*, 1–35.

Marcílio, W. E., & Eler, D. M. (2020). From explanations to feature selection: Assessing shap values as feature selection mechanism. In *2020 33rd SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)* (pp. 340–347).

Martins, J., Branco, F., & Mamede, H. (2023). Combining low-code development with chatgpt to novel no-code approaches: A focus-group study. In *Intelligent systems with applications* (p. 200289).

Mohammed Al Sameer, M., Prasanth, T., & Anuradha, R. (2021). Rapid forest cover detection using ensemble learning. In *International virtual conference on industry 4.0: Select proceedings of IVCI4. 0 2020* (pp. 181–190). Springer.

Mu, Y., Wu, B. P., Thorne, W., Robinson, A., Aletras, N., Scarton, C., Bontcheva, K., & Song, X. (2023). Navigating prompt complexity for zero-shot classification: A study of large language models in computational social science. arXiv preprint arXiv:2305.14310.

Nie, L., Wu, R., Ren, Y., & Tan, M. (2023). Research on fault diagnosis of hvac systems based on the relieff-rfecv-svm combined model. *Actuators*, *12*. https://doi.org/10.3390/act12060242. https://www.mdpi.com/2076-0825/12/6/242.

Rizou, S., Theofilatos, A., Paflioti, A., Pissari, E., Varlamis, I., Sarigiannidis, G., & Chatzisavvas, K. C. (2023). Efficient intent classification and entity recognition for university administrative services employing deep learning models. In *Intelligent systems with applications* (p. 200247).

Schick, T., & Schütze, H. (2020). Exploiting cloze questions for few shot text classification and natural language inference. arXiv preprint arXiv:2001.07676.

Sivarajkumar, S., & Wang, Y. (2022). Healthprompt: A zero-shot learning paradigm for clinical natural language processing. In *AMIA annual symposium proceedings, American medical informatics association* (p. 972).

Xiao, L., & Chen, X. (2023). Enhancing llm with evolutionary fine tuning for news summary generation. arXiv preprint arXiv:2307.02839.

Yu, X., Chen, Z., Ling, Y., Dong, S., Liu, Z., & Lu, Y. (2023). Temporal data meets llm–explainable financial time series forecasting. arXiv preprint arXiv:2306.11025.

Zhong, R., Lee, K., Zhang, Z., & Klein, D. (2021). Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. arXiv preprint arXiv:2104.04670.