Data Science Lab Project
July 2025 Exam session

Are LLMs Good Journalists? A study on Cycling

Giovanni Noè, Francesco Volpi Ghirardini

Abstract
In an increasingly AI-centric world companies are trying to develop innovative solutions to reduce costs and enhance efficiency. Imagining to be hired by a sports journal, the goal of this project is to build an engine that employs Large Language Models (LLMs) to write articles about cycling competitions.
To achieve this goal, various of these models will be tested on a wide range of article-writing tasks and subsequentially evaluated, in order to produce the best and most versatile engine possible.

Table of Contents

Introduction 1
1. Data Acquisition..2
1.1 Giro d'Italia 2
1.2 Gran Piemonte 2
2. Engine Structure...2
2.1 The Request Builder 2
2.2 The Article Writer......................................3
2.3 The Request Builder for Titles...................3
2.4 The Title Creator...3
3. Evaluation System 3
4. Testing the Engine...4
4.1 Basic Test...4
4.2 Capability Test..5
4.3 Targeted Articles Test 5
Conclusions 6
References 6
Appendix A – llm_journalist..............................7
Appendix B – Prompt Example......................11
Appendix C – Article Example........................16

Introduction

As Artificial Intelligence systems rapidly improve in performance and versatility, they are quickly transforming the way we live and work, becoming an integral part of everyday life. From virtual assistants and recommendation systems to autonomous vehicles and advanced data analysis, these technologies are reshaping industries and redefining human-machine interaction.

Companies are increasingly aware of the importance of this shift and are working to integrate AI based software in their business to reduce costs, save time and boost productivity.

The news industry, in particular, is facing declining public interest in traditional newspapers and journals and therefore recognizes the need to produce articles more efficiently, with reduced costs and fewer resources.

The methods developed in this project address this need by exploring the use of Large Language Models to generate sports articles. More specifically cycling competitions will be reported on, and for this purpose two races have been selected:

- "Giro d'Italia" - the most prominent cycling competition in Italy, chosen to test the system with a large volume of data and its ability to evaluate the changes in general rankings of multiple stage races
- "Gran Piemonte" - a smaller-scale race, selected to evaluate the models' performance with limited data and few information on it available on the internet. This competition was chosen after verifying, through a search on the English versions of Google, that it received extremely small coverage in the English-speaking press

After the data acquisition phase, the engine was developed and four LLMs coming from four different companies were tested on identical tasks to identify the best-performing one. The goal was to build an engine that, using primarily structured data as input, automatically generates prompts for the LLMs to produce complete articles and their corresponding titles.

To assess the quality of the generated content, an evaluation system is designed ensuring the presence of a human in the loop, maintaining editorial oversight and content reliability.

1. Data Acquisition

For each of the two races the most amount of data possible is gathered and the decision on what can be of use is left for a later moment, since it's easier to standardize the whole scraping procedure beforehand.

When this data will be made available to the models it will follow a hierarchy, with primary information being that common to any possible race: route details, weather and arrival ranking. Then there is additional multiple stage race information, which is made available to the LLMs only when writing about this specific type of race.

1.1 Giro d'Italia

All the data is scraped directly from the official website of the Giro [1], with information related to the weather coming from the 3BMeteo official Giro widget.

If the stage allows for it, these are the information made available:

- General information:
  - Route details
  - Weather data
- Stage rankings:
  - Order of arrival
  - Total points
  - Super Team rankings
- Intermediate rankings:
  - KOM points
  - Intermediate sprint points
  - Red Bull KM points
- Other data:
  - Withdrawals
  - "Fuga" points

Since the overall rankings updated for each stage is not available online, they are computed from the data just presented, adding more information for each stage:

- Maglia rosa rankings (time-based standing)
- Maglia ciclamino rankings (points standing)
- Maglia azzurra rankings (mountain standing)
- Intermediate sprints rankings (overall)
- Red Bull KM (overall)
- Fuga (overall)

1.2 Gran Piemonte

All of the data for this competition is scraped from the official website [2], apart from the weather data which is manually gathered from the "iLMeteo" [3] archival database.

The information gathered is:

- Route details
- Weather data
- Order of arrival

2. Engine Structure

The core around which this project revolves around is the engine that automatically generates prompts and activates the LLMs to write the articles and their respective titles. This engine is implemented as a Python module containing four distinct functions, which will be examined one by one in this section.

Given that this work is based on the premise of being commissioned by a journal this section could also act as the user manual for individuals unfamiliar with LLMs or programming that might need to handle the system. This last aspect is also considered while designing the engine, to ensure it requires minimal user input, making it as user-friendly as possible. Some specifications about the code are provided below, for more details see Appendix A - llm_journalist.

2.1 The Request Builder

The first function to be analysed is *request_builder*, which generates the prompts for LLMs. This function takes the following arguments as input:

1. *data* - The information to be included in the prompt for the LLMs. The format must be a dictionary of lists containing DataFrames that match the structure of the tables described in the Data Acquisition section.
2. *competition_name* - The name of the competition to be mentioned in the article (provided as a string).
3. *competition_code* - Since multi stage competitions (such as the Giro d'Italia) provide significantly more data to be included in the article, it is necessary to distinguish them. This argument is used for that purpose and accepts only specific values. In this project, only the Giro is treated as a multi stage competition, so the function accepts only "*G*" (standing for "Giro") as a special value that alters the output accordingly.
4. *approx_length* - The desired approximate length of the article to be requested from the LLM.

5. *additional_requests* - Any additional instructions or requests to be included in the prompt (provided as a string).

Given the input, the function proceeds in three steps. First, it builds a generic request including only primary data. Then, if the right competition code is provided, the function also appends the additional data and the specific instructions related to that competition. Finally, if any additional requests are given, they are incorporated into the prompt.

The output of this function is a string containing the prompt for the LLM; two examples of outputs (one for the Grand Piemonte race and one for the Giro's 20th stage) are provided in Appendix B - Prompt Examples.

Also, the decision to separate the *request_builder* from the article-writing function is made to allow the prompt to be generated and inspected independently, without immediately sending it to an LLM. This approach enables users to review or modify the prompt before execution and makes it possible to use the prompt with language models not available on Novita [4], the provider used for this project (ChatGPT for example).

2.2 The Article Writer

The second most important function is *cycling_journalist*, which is responsible for generating the articles.

The input arguments are the same as the previous function, with the addition of two parameters:

- *HF_Token*, a string containing the token used to access to the HuggingFace [5] API, allowing Python to interact with the selected LLMs
- *model*, a string specifying the version of LLM to be used for writing the article.

The models that appear in the calls of this function in the project are:

- Llama by Meta [6], *model = "meta-llama/llama-4-maverick-17b-128e-instruct-fp8"*, a 17-billion-parameter model designed for complex reasoning and high-quality content generation.
- DeepSeek [7], *model = "deepseek-ai/DeepSeek-V3-0324"*, a general-purpose model trained for multilingual and multi-domain tasks, known for its strong performance in text understanding and generation.
- Qwen [8], *model = "qwen/qwen3-235b-a22b"*, a 235-billion-parameter model capable of advanced language tasks and detailed contextual reasoning.

All other models made available by Novita should also work, as long as the version name is correctly specified in the model argument.

To write the article the function follows the following steps:

1. generating the prompt using the *request_builder*
2. connecting to the provider using the *HF_Token*
3. sending the message contained in the prompt to the specified *model*
4. outputting the response as a string

An example of an article produced by this function is shown in Appendix C - Article Example.

2.3 The Request Builder for Titles

To write titles for articles the proceeding is nearly the same: a request prompt is built and then it is asked to a LLM to produce the title from that prompt.

The function *title_request_builder* is very simple, it takes just the article (a string) as input and creates a prompt for the LLM with the aim to give a title to that article.

The title creation process is designed to be independent from article generation in order to give the user maximum flexibility. For example, this allows the use of two different models, one for writing the title and another for the article, or enables generating a title for an article that was previously written.

2.4 The Title Creator

The function *title_creator* works in the same way as its counterpart for articles, *cycling_journalist*. It takes as input the *HF_Token*, the *article* to be titled, and the *model* version to be used. The function then outputs a title for the article as a string. An example of title generated with Llama is in Appendix C - Article Example.

3. Evaluation System

Since multiple models are going to be tested on various tasks, a method for evaluating the articles produced by the LLMs is necessary. To address this, an evaluation system is designed with the goal of defining a numerical scoring index that can assess article quality across multiple dimensions. Particular attention is given to aspects related to adherence to the prompt and the correct use of the provided data.

The quality dimension considered in this evaluation system are the following:

- Correctness: a measure of how factually correct the article is. This includes verifying that race results, statistics, team names, and rider information are accurate, that no false or misleading statements are made about the event, and that there are no errors in calculations. Additionally, it assesses whether

the model correctly understands and contextualizes the data it uses.
- Completeness and Data Relevance: since LLMs are allowed to select which data to include in the article, this criterion evaluates the quality of those choices by answering the question: “Does the article contain all the most important information about the competition?”.
- Adherence to Provided Data and Prompt: this criterion measures how closely the LLM follows the given instructions and relies solely on the provided input data. It includes avoiding the use of external or unrelated information (even if factually correct), preventing hallucinated content and respecting stylistic or structural constraints from the prompt (e.g., word count, tone, additional requests).
- Language and Readability: a measure of how well the article uses an appropriate language (sector specific or aligned to additional requests) and how well-written and engaging the article is.

All the articles that will be produced in the following tests will be evaluated on a scale from one to five on each criterion. The evaluation will be carried out independently by both members of this project group. Then, for each article, an average of the votes will be computed for each criterion and finally a weighted average will be calculated to account for the different importance of each quality dimension. Moreover, the weights assigned to the criteria will differ depending on the category of the test, as different tests are designed to assess different aspects of article quality

4. Testing the Engine

To demonstrate the quality of the engine and identify the best model for deployment, it is necessary to conduct a series of tests and experiments.

Four models will be put through a basic test: the three mentioned in the section on the *cycling_journalist* function, as well as ChatGPT. Then, the two models that obtained the highest average score, will go through more advanced tests.

Regarding ChatGPT, the prompt was manually copied and pasted into a chat on its website, with a request for the model to ignore any information about the user to avoid bias.

It is important to note that this LLM could not be accessed via the *cycling_journalist* function, as the only way to execute prompts with ChatGPT through an API is by using the official OpenAI API, which requires a paid subscription. Given that this project was conducted on a zero budget, manual input was chosen as a substitute method; it would have been a mistake not to test the world’s most widely adopted LLM. Anyway, in the context of a professional news organization, part of the budget would likely be allocated to integrating ChatGPT access directly into the Python script paying for the API subscription.

To conduct these tests three competitions with their respective sets of data were selected:

- Giro d’Italia’s 10th Stage: a time trial stage with a limited amount of additional data
- Giro d’Italia’s 20th Stage: a very complex mountain stage with many intermediate rankings and the most important general ranking switch of the edition
- Grand Piemonte race: the minor competition selected to test the generic prompt

4.1 Basic Test

For the first set of experiments, all LLMs are tested by asking them to generate articles of approximately 500 words, with each model required to write one article per competition.

The key objective of these initial tests is to assess the general quality of the articles, with a particular focus on the accuracy of the information presented, as well as the models’ adherence to both the input data and the prompt.

Therefore, in the context of article evaluation, greater weight is assigned to the correctness and adherence criteria. The weighted average for the overall evaluation was calculated using the following weights:

- Correctness: 40%
- Completeness and Data Relevance: 20%
- Adherence to Provided Data and Prompt: 30%
- Language and Readability: 10%

While all evaluation categories were considered, the criteria listed above were prioritized due to their relevance to the goals of these initial tests. The remaining two, Completeness and Data Relevance and Language and Readability, still played a role but were assigned comparatively less weight.

Evaluation

DeepSeek: It adheres to the given data pretty well, however it makes a lot of interpretations mistakes and, for stage 10, it gives relevance to some irrelevant information. Writing the article it uses an engaging tone but it is sometimes paired with either the wrong terms or with invented expressions.

LLama: It performes overall well on the stages of the Giro, choosing the right data and interpreting it correctly, its main drawdown is focusing too much on weather data. For the Gran Piemonte race it makes a couple of correctness mistakes derived from bad interpretation. Its language is the simplest out of all the models, which makes the articles less engaging but also free of any technical language mistakes.

Qwen: It is quite good at picking data, however it lacks in its interpretation and includes some information taken from elsewhere. Its writing is by far the worst out of the four models, as it failed across every levels: grammatically, engagement-wise and on a technical level. Its biggest weakness was the fact that before outputting the results article it also displays a sort of "thinking process", which is useless and not required, penalizing it on the adherence front.

ChatGPT: It is the best model at selecting and interpreting the data, but it is also the model that came up with the most amount of information, both true and false. It used a really expressful tone, which is really engaging but it also gives space to a vast amount of incorrect expressions and terms.

Metric	DeepSeek	LLama	Qwen	ChatGPT
Correctness	3.7	4.5	3.7	4.7
Completeness	4.7	4.3	4.5	5
Adherence	4	4.5	1	3
Language	3	4.8	2.5	3
Final Score	3.9	4.5	3.6	4.1

***Table 4.1** – Evaluation Metrics for the Test*

The metric scores are reported in table 4.1. By the final score it can be evaluated how the two models that performed the best are Llama and ChatGPT and, how it has been already mentioned, only these two will be further tested.

4.2 Capability Test

The second kind of experiments is designed to challenge LLMs by asking them to write articles whose lengths are not proportional to the amount of data provided.

This test is divided into two trials:

- **200 Words**: The first kind of test in this section serves to test the capability of LLMs to select few important information to insert in the articles, so the request was to write a short article (approximately 200 words) about Stage 20 of the Giro d'Italia.
- **800 Words**: The second type of test in this section has the opposite objective: to evaluate how effectively LLMs can expand the length of an article without introducing false information or relying on external sources beyond the provided data.

Being that the two tests aim to evaluate different aspects of the LLMs abilities, it is also appropriate for the metrics to be weighed differently:

- **200 Words**:
  - Correctness: 20%
  - Completeness and Data Relevance: 50%
  - Adherence to Provided Data and Prompt: 20%
  - Language and Readability: 10%
- **800 Words**:
  - Correctness: 40%
  - Completeness and Data Relevance: 10%
  - Adherence to Provided Data and Prompt: 40%
  - Language and Readability: 10%

Evaluation

Llama: For the 200 words test it manages to select all the relevant information, except for once again giving too much importance to the weather; its simple and concise language which, once again, is technically correct helps it in this aspect. In the 800 words test it fails on achieving the number of words, it only wrote about 300 words, greatly reducing its adherence to the prompt; the writing is seemingly disconnected from one section to the other and a lot of irrelevant information is included.

ChatGPT: In the 200 words test it selects the right data perfectly but also adds information it was not given; the language is once again engaging but lacks any of the technicality required by the topic. For the 800 words test it undershot the word count required by roughly 150 words, in this regard beating Llama, but it manages to do so only by adding external information and with an excessively verbose language.

Metric	Llama		ChatGPT	
	200	800	200	800
Correctness	5	4	4.5	4.5
Completeness	4	3.5	5	4.5
Adherence	4.5	1	4	2.5
Language	4.5	4	2.5	2.5
Final Score	4.4	2.5	4.5	3.5

***Table 4.2** – Evaluation Metrics for the Test*

The metric scores are reported in table 4.2, in the 200 words test the two models had a very close average performance. On the 800 words test ChatGPT, even if with a low overall score, was clearly better.

4.3 Targeted Articles Test

The final type of experiment is aimed at testing the versatility of the models by including in the prompt an additional request regarding the type of language to be used, tailored to specific target audiences.

This test, like the one before, it's divided into two different trials:

- **Specialistic Journal**: The first experiment of this kind is designed to simulate writing articles for a specialized cycling journal, where the target audience consists of experts or enthusiasts with a deep understanding of the sport. This requires the model to adopt a highly technical language, with strong attention to detail and a high degree of accuracy.

- **Kids Magazine**: The final experiments of this project involves writing articles for a children's magazine. Consequently, the LLMs are asked to use very simple language appropriate for a young audience.

The evaluation metrics are weighed to reflect the aspects of the two contexts:

- **Specialistic Journal**:
    - Correctness: 35%
    - Completeness and Data Relevance: 15%
    - Adherence to Provided Data and Prompt: 15%
    - Language and Readability: 35%
- **Kids Magazine**:
    - Correctness: 20%
    - Completeness and Data Relevance: 30%
    - Adherence to Provided Data and Prompt: 10%
    - Language and Readability: 40%

Evaluation

Llama: For the cycling journal it managed to write an article which went into the right details and which had the right terminology, the only serious mistake made was the misinterpretation of how much minutes matters in the second-last stage. In the kids' magazine test it managed to adapt a language and form that is similar to that of children's narrative books, the main drawback is that it included some measures that kids might not understand.

ChatGPT: Overall the right information was picked from the data but some others, although factually correct, were gathered from the internet; from a grammatical point of view the article is very well written but it lacks the seriousness and correct terminology required by the task. In the kids' magazine test its writing style changed only slightly from the other articles and it also included specific measures that kids might not understand.

Metric	Llama		ChatGPT	
	Journal	Kids	Journal	Kids
Correctness	3.5	5	4.5	5
Completeness	4.5	4	5	3.5
Adherence	4.5	4.5	4	4
Language	4.5	5	3	3.5
Final Score	4.2	4.7	4	3.9

***Table 4.3** – Evaluation Metrics for the Test*

The metric scores are reported in table 4.3, in the test concerning the cycling journal the two LLMs had a similar average performance, even if with a slight advantage for Llama. On the kid's magazine test Llama was the clear winner.

Conclusions

In conclusion, the process implemented to write the articles through the engine proved successful, while the evaluation system developed provided a solid basis for judging the quality of the articles produced.

Out of the four Large Language Models that were tested the two that proved to be the overall best performing are LLama and ChatGPT, having a clear advantage on DeepSeek and Qwen. Out of these two better models, however, there is no clear winner since each one has its strengths and weaknesses.

LLama showed a fair understanding of data, even if these elements proved to lack on the longer paragraph, while its simple language made it avoid inappropriate expressions; in the creative task it demonstrated to have an edge over the other model, especially for the kids' magazine.

ChatGPT's greatest strength was its ability to understand the context and the data provided, which was far greater than the other models, and its writing ability was also exceptional; its main drawdowns are that it takes too many information from other sources and that a lot of expressions and terminology used were exceedingly out of context.

Overall LLama proved to be rather unremarkable in its performance but with a consistency that is its greatest advantage, while ChatGPT has some substantial strengths paired with some equally important weaknesses.

References

[1] Giro d'Italia official website: *https://www.giroditalia.it/*

[2] Gran Piemonte official website: *https://www.ilgranpiemonte.it/*

[3] iLMeteo archive database: *https://www.ilmeteo.it/portale/archivio-meteo*

[4] Novita AI: *https://novita.ai/models/llm*

[5] Hugging Face website: https://huggingface.co/

[6] LLama: *https://www.llama.com/*

[7] DeepSeek: *https://www.deepseek.com/*

[8] Qwen: *https://chat.qwen.ai/*

Appendix A – llm_journalist

The code of the engine of this work (*llm_journalist*) is provided below.

```
from huggingface_hub import InferenceClient

def cycling_journalist(HF_Token, data, competition_name, competition_code = None, approx_length = 500, model = "meta-llama/llama-4-maverick-17b-128e-instruct-fp8", additional_requests =  None):

  article = ''

  request = request_builder(data, competition_name, competition_code, approx_length, additional_requests)

  # Performing LLM Request

  ## Specify the inference client
  client = InferenceClient(provider="novita", api_key=HF_Token)
# "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" or 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

  # Send messages to a specific model hosted on the inference client, specifying system-level instructions
  completion = client.chat.completions.create(
    model=model,
    messages=[
      {
        "role": "user",
        "content": request
      }
     ]
    )
  article = completion.choices[0].message.content
  return article


def request_builder(data, competition_name, competition_code = None, approx_length = 500, additional_requests = None):

  # Generic request building
  request = f"You are a sports journalist, write an article about the cycling competition '{competition_name} of about {approx_length} words'. In the output provide only the body of the article without any text formatting, emojis or additional phrases. The article must be enjoyable to read and must contain only the information provided below, do not include invented data or data from any other source. Here is the data to input along with some context, skip where 'None' is provided:\n-----Data-----\n"
  request += "GENERAL INFORMATION\n"
  request += "Track Information:\n"
  request += f" Starting location of the race: {data['General'][0]['Start']}\n Ending location of the race: {data['General'][0]['Finish']}\n Type of race ('Mountain', 'Hill', 'Flat', 'Chrono'): {data['General'][0]['Type']}\n Difficulty: {data['General'][0]['Difficulty']}\n Length: {data['General'][0]['Lenght']}\n Altitude Gain: {data['General'][0]['Altitude Gain']}\n Technical Info: {data['General'][0]['Technical Info']}\n"
  request += "Weather Conditions at start:\n"
  request += f" Temperature in Celsius: {data['General'][1].loc['Start']['Temperature']}\n Conditions (data in italian translate in english if used in the article): {data['General'][1].loc['Start']['Conditions']}\n Precipitations (in mm): {data['General'][1].loc['Start']['Precipitation']}\n Wind Speed (in Km/h): {data['General'][1].loc['Start']['Wind - Speed']}\n Wind Direction: {data['General'][1].loc['Start']['Wind - Direction']}\n"
  request += "Weather Conditions at finish:\n"
  request += f" Temperature in Celsius: {data['General'][1].loc['Finish']['Temperature']}\n Conditions (data in italian translate in english if used in the article): {data['General'][1].loc['Finish']['Conditions']}\n Precipitations (in mm): {data['General'][1].loc['Finish']['Precipitation']}\n Wind Speed (in Km/h): {data['General'][1].loc['Finish']['Wind - Speed']}\n Wind Direction: {data['General'][1].loc['Finish']['Wind - Direction']}\n"
  request += f"Number of Riders that finished the race: {len(data['General'][2])}\n"
  request += "Order of Arrival, top 15:\n"
  for i in range(min(15,len(data['General'][2]))):
    request += f" Position: {data['General'][2].iloc[i]['Position']}, Rider: {data['General'][2].iloc[i]['Rider']}, Team: {data['General'][2].iloc[i]['Team']}, Time: {data['General'][2].iloc[i]['Time']}, Gap to leader: {data['General'][2].iloc[i]['Gap']}\n"

  # Specialization for non generic competitions
  if competition_code:

    ## Giro d'Italia
    if competition_code == 'G':

      ### Maglia Rosa
      request += "Maglia Rosa Standings after this race, top 15:\n"
      for i in range(min(15,len(data['Maglia Rosa Standing'][0]))):
        request += f" Position: {data['Maglia Rosa Standing'][0].iloc[i]['Position']}, Rider: {data['Maglia Rosa Standing'][0].iloc[i]['Rider']}, Team: {data['Maglia Rosa Standing'][0].iloc[i]['Team']}, Time (total sum): {data['Maglia Rosa Standing'][0].iloc[i]['Time']}\n"

      ### King of the Mountain
      request += "\nKING OF THE MOUNTAINS INFORMATION\n"

      #### Maglia Azzurra
      request += "Maglia Azzurra Standings after this race, top 15:\n"
      for i in range(min(15,len(data['King of the Mountains'][0]))):
        request += f" Position: {data['King of the Mountains'][0].iloc[i]['Position']}, Rider: {data['King of the Mountains'][0].iloc[i]['Rider']}, Team: {data['King of the Mountains'][0].iloc[i]['Team']}, Points (total sum): {data['King of the Mountains'][0].iloc[i]['Points']}\n"

      #### KOMs Current Stage
      for j in range(len(data['King of the Mountains'][1:])):
        request += f"KOM climb {j+1} of this stage standings:\n"
        for i in range(min(15,len(data['King of the Mountains'][j+1]))):
          request += f" Position: {data['King of the Mountains'][j+1].iloc[i]['Position']}, Rider: {data['King of the Mountains'][j+1].iloc[i]['Rider']}, Team: {data['King of the Mountains'][j+1].iloc[i]['Team']}, Points: {data['King of the Mountains'][j+1].iloc[i]['Points']}\n"

      ### Points Classification
      request += "\nPOINTS CLASSIFICATION INFORMATION\n"

      #### Maglia Ciclamino
      request += "Maglia Ciclamino Standings after this race, top 15:\n"
      for i in range(min(15,len(data['Intermediate Sprints'][0]))):
        request += f" Position: {data['Intermediate Sprints'][0].iloc[i]['Position']}, Rider: {data['Intermediate Sprints'][0].iloc[i]['Rider']}, Team: {data['Intermediate Sprints'][0].iloc[i]['Team']}, Points (total sum): {data['Intermediate Sprints'][0].iloc[i]['Points']}\n"

      #### KOMs Current Stage
      for j in range(len(data['Intermediate Sprints'][1:])):
        request += f"KOM climb {j+1} of this stage standings:\n"
        for i in range(min(15,len(data['Intermediate Sprints'][j+1]))):
          request += f" Position: {data['Intermediate Sprints'][j+1].iloc[i]['Position']}, Rider: {data['Intermediate Sprints'][j+1].iloc[i]['Rider']}, Team: {data['Intermediate Sprints'][j+1].iloc[i]['Team']}, Points: {data['Intermediate Sprints'][j+1].iloc[i]['Points']}\n"

      ### Official Withdrawals
      request += "\nOFFICIAL WITHDRAWALS INFORMATION\n"
      for i in range(len(data['Official Withdrawals'][0])):
        request += f" Rider: {data['Official Withdrawals'][0].iloc[i]['Rider']}, Team: {data['Official Withdrawals'][0].iloc[i]['Team']}\n"

      ### Previous Standings
      request += "\nPREVIOUS STANDINGS INFORMATION\n"
      for j in range(len(data['Previous Standings'])):
        n = 5
        if j == 0:
          n = 10
          request += "Maglia Rosa Standings before this race, top 10:\n"
        elif j == 1:
          request += "Maglia Azzurra Standings before this race, top 5:\n"
        elif j == 2:
          request += "Maglia Ciclamino Standings before this race, top 5:\n"
        for i in range(min(n,len(data['Previous Standings'][j]))):
          request += f" Position: {data['Previous Standings'][j].iloc[i]['Position']}, Rider: {data['Previous Standings'][j].iloc[i]['Rider']}, Team: {data['Previous Standings'][j].iloc[i]['Team']}\n"
      request += "-----Data End-----\nDo not mention retirements if they are not in top 15 of any standing and consider talking about the variations of the three standings.\n"

  else:
    print('Competition code not specified or recognised, the article will be generic')
    request += "-----Data End-----\n"
  request += "Use only the most relevant data to make the article as interesting as possible.\n"

    # Additional Requests
  if additional_requests:
    request += f"additionally, {additional_requests}"
  return request


def title_request_builder(article):
  request = "Write a title for the following article. In the output provide only the title without any text formatting, emojis or additional phrases.\n\n"
  request += article
  return request


def title_creator(HF_Token, article, model = "meta-llama/llama-4-maverick-17b-128e-instruct-fp8"):

  title = ''

  request = title_request_builder(article)
```

```
# Performing LLM Request

## Specify the inference client
client = InferenceClient(provider="novita", api_key=HF_Token) # or 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

# Send messages to a specific model hosted on the inference client, specifying system-level instructions
completion = client.chat.completions.create(
    model=model,
    messages=[
        {
            "role": "user",
            "content": request
        }
      ]
    )
title = completion.choices[0].message.content
return title
```

Appendix B – Prompt Example

This appendix includes two examples of prompts created by the function *request_builder*.
To view all the prompts used in the tests described in this report, please refer to the *Data Science Lab Project - Application.ipynb* Colab notebook.
The first prompt regards the Grand Piemonte race and is the output of the following call of the function:

```
request_builder(data,"Gran Piemonte",competition_code = None, approx_length = 500, additional_requests=None)
```

The output is the following:

```
You are a sports journalist, write an article about the cycling competition 'Gran Piemonte of about 500 words'. In the
output provide only the body of the article without any text formatting, emojis or additional phrases. The article must
be enjoyable to read and must contain only the information provided below, do not include invented data or data from
any other source. Here is the data to input along with some context, skip where 'None' is provided:
-----Data-----
GENERAL INFORMATION
Track Information:
 Starting location of the race: Valdengo     -
 Ending location of the race: Borgomanero
 Type of race ('Mountain', 'Hill', 'Flat', 'Chrono'): Hill
 Difficulty: nan
 Length: 182km
 Altitude Gain: 1650mt
 Technical Info: The race runs along a mostly flat route in the first part (more than 100 km) and becomes more hilly in
the second, with various altimetrical difficulties up to the final loop. The first half, after crossing Biella and Candelo,
winds along lowland roads among the rice fields bordering river Sesia. Once in Varallo, a sequence of climbs begins,
consisting of the Passo della Colma and the climbs of Cremosina and Traversagna. After the passage on the finish line,
a circuit (1 lap) of about 11 km will be tackled.Last kmThe last 11 km are rather wavy. The roads are medium-wide with
various traffic-management infrastructures (roundabouts, traffic islands, etc.) and street furniture. The last kilometres
are slightly uphill on a false-flat.
Weather Conditions at start:
 Temperature in Celsius: 21°
 Conditions (data in italian translate in english if used in the article): pioggia
 Precipitations (in mm): 4
 Wind Speed (in Km/h): 6.0
 Wind Direction: nan
Weather Conditions at finish:
 Temperature in Celsius: 15°
 Conditions (data in italian translate in english if used in the article): pioggia
 Precipitations (in mm): 19
 Wind Speed (in Km/h): 6.3
 Wind Direction: nan
Number of Riders that finished the race: 155
Order of Arrival, top 15:
 Position: 1.0, Rider: Neilson Powless, Team: Ef Education - Easypost, Time: 3:57:36, Gap to leader: 0:00
 Position: 2.0, Rider: Corbin Strong, Team: Israel - Premier Tech, Time: 3:57:43, Gap to leader: 0:07
 Position: 3.0, Rider: Alexander Aranburu, Team: Movistar Team, Time: 3:57:43, Gap to leader: 0:07
 Position: 4.0, Rider: Filippo Magli, Team: Vf Group-Bardiani Csf- Faizane', Time: 3:57:43, Gap to leader: 0:07
 Position: 5.0, Rider: Xandro Meurisse, Team: Alpecin-Deceuninck, Time: 3:57:43, Gap to leader: 0:07
 Position: 6.0, Rider: Tobias Halland Johannessen, Team: Uno-X Mobility, Time: 3:57:43, Gap to leader: 0:07
 Position: 7.0, Rider: Natnael Tesfazion, Team: Lidl-Trek, Time: 3:57:43, Gap to leader: 0:07
 Position: 8.0, Rider: Francesco Busatto, Team: Intermarche' - Wanty, Time: 3:57:43, Gap to leader: 0:07
 Position: 9.0, Rider: Edoardo Zambanini, Team: Bahrain Victorious, Time: 3:57:43, Gap to leader: 0:07
```

Position: 10.0, Rider: Matej Mohoric, Team: Bahrain Victorious, Time: 3:57:43, Gap to leader: 0:07
Position: 11.0, Rider: Lorenzo Rota, Team: Intermarche' - Wanty, Time: 3:57:43, Gap to leader: 0:07
Position: 12.0, Rider: Anders Foldager, Team: Team Jayco Alula, Time: 3:57:43, Gap to leader: 0:07
Position: 13.0, Rider: Filippo Baroncini, Team: Uae Team Emirates, Time: 3:57:43, Gap to leader: 0:07
Position: 14.0, Rider: Martijn Tusveld, Team: Team Dsm-Firmenich Postnl, Time: 3:57:43, Gap to leader: 0:07
Position: 15.0, Rider: Thomas Pidcock, Team: Ineos Grenadiers, Time: 3:57:43, Gap to leader: 0:07
-----Data End-----
Use only the most relevant data to make the article as interesting as possible.

The second prompt regards the Giro's 20th stage (takes the competition code "*G*") with the additional request of writing the article with a serious and technical tone for a cycling journal.
The call of the function is the following:

print(request_builder(data,"20th Stage of Giro d'Italia",competition_code = 'G', approx_length = 500, additional_requests="THE FOLLOWING REQUESTS ARE VERY IMPORTANT STICK ON THEM: write the article with a serious and technical tone, it is for a cycling specific journal"))

The resulting prompt is:

You are a sports journalist, write an article about the cycling competition '20th Stage of Giro d'Italia of about 500 words'. In the output provide only the body of the article without any text formatting, emojis or additional phrases. The article must be enjoyable to read and must contain only the information provided below, do not include invented data or data from any other source. Here is the data to input along with some context, skip where 'None' is provided:
-----Data-----
GENERAL INFORMATION
Track Information:
Starting location of the race: Verrès -
Ending location of the race: Sestrière
Type of race ('Mountain', 'Hill', 'Flat', 'Chrono'): Mountain
Difficulty: 5/5
Length: 205km
Altitude Gain: 4500m
Technical Info: The stage starts flat until the Canavese region, then a series of increasingly tough climbs lead into the Valli di Lanzo. After Viù, the Colle del Lys climb begins. A descent brings riders into the Dora Riparia Valley to Susa, where the Colle delle Finestre (this year's Cima Coppi) climb begins. The Finestre climb averages a consistent 9.2%, with a 14% ramp in Meana di Susa. The first 9 km are paved; the final 9 km are gravel. There are 29 switchbacks in less than 4 km (45 total). The descent to Pian dell'Alpe is narrow and exposed. The final climb to Sestriere on the SS23 is steady and manageable.   are all on the SS23, a steady climb at about 5% on a wide, well-paved road. Final straight: 400 m on 6.5-metre-wide asphalt.
Weather Conditions at start:
Temperature in Celsius: 24°
Conditions (data in italian translate in english if used in the article): velature lievi
Precipitations (in mm): 0.0
Wind Speed (in Km/h): 4
Wind Direction: E
Weather Conditions at finish:
Temperature in Celsius: 20°
Conditions (data in italian translate in english if used in the article): pioviggine
Precipitations (in mm): 0.3
Wind Speed (in Km/h): 20
Wind Direction: E
Number of Riders that finished the race: 159
Order of Arrival, top 15:

Position: 1, Rider: Christopher Harper, Team: Team Jayco Alula, Time: 5:27:29, Gap to leader: 0:00
Position: 2, Rider: Alessandro Verre, Team: Arkea-B&B Hotels, Time: 5:29:18, Gap to leader: 01:49
Position: 3, Rider: Simon Yates, Team: Team Visma - Lease A Bike, Time: 5:29:26, Gap to leader: 01:57
Position: 4, Rider: Gianmarco Garofoli, Team: Soudal Quick-Step, Time: 5:31:21, Gap to leader: 03:52
Position: 5, Rider: Rémy Rochas, Team: Groupama-Fdj, Time: 5:31:26, Gap to leader: 03:57
Position: 6, Rider: Martin Marcellusi, Team: Vf Group Bardianicsf-Faizane', Time: 5:32:00, Gap to leader: 04:31
Position: 7, Rider: Carlos Verona, Team: Lidl-Trek, Time: 5:32:00, Gap to leader: 04:31
Position: 8, Rider: Max Poole, Team: Team Picnic Postnl, Time: 5:34:14, Gap to leader: 06:45
Position: 9, Rider: Isaac Del Toro, Team: Uae Team Emirates Xrg, Time: 5:34:39, Gap to leader: 07:10
Position: 10, Rider: Giulio Pellizzari, Team: Red Bull - Bora - Hansgrohe, Time: 5:34:39, Gap to leader: 07:10
Position: 11, Rider: Derek Gee, Team: Israel - Premier Tech, Time: 5:34:39, Gap to leader: 07:10
Position: 12, Rider: Damiano Caruso, Team: Bahrain Victorious, Time: 5:34:39, Gap to leader: 07:10
Position: 13, Rider: Richard Carapaz, Team: Ef Education - Easypost, Time: 5:34:43, Gap to leader: 07:14
Position: 14, Rider: Brandon Mcnulty, Team: Uae Team Emirates Xrg, Time: 5:34:46, Gap to leader: 07:17
Position: 15, Rider: Wout Van Aert, Team: Team Visma - Lease A Bike, Time: 5:35:31, Gap to leader: 08:02
Maglia Rosa Standings after this race, top 15:
Position: 1, Rider: Simon Yates, Team: Team Visma - Lease A Bike, Time (total sum): 79:18:42
Position: 2, Rider: Isaac Del Toro, Team: Uae Team Emirates Xrg, Time (total sum): 79:22:38
Position: 3, Rider: Richard Carapaz, Team: Ef Education - Easypost, Time (total sum): 79:23:25
Position: 4, Rider: Derek Gee, Team: Israel - Premier Tech, Time (total sum): 79:25:05
Position: 5, Rider: Damiano Caruso, Team: Bahrain Victorious, Time (total sum): 79:26:14
Position: 6, Rider: Giulio Pellizzari, Team: Red Bull - Bora - Hansgrohe, Time (total sum): 79:28:10
Position: 7, Rider: Egan Bernal, Team: Ineos Grenadiers, Time (total sum): 79:31:24
Position: 8, Rider: Einer Rubio, Team: Movistar Team, Time (total sum): 79:31:47
Position: 9, Rider: Brandon Mcnulty, Team: Uae Team Emirates Xrg, Time (total sum): 79:32:18
Position: 10, Rider: Michael Storer, Team: Tudor Pro Cycling Team, Time (total sum): 79:33:09
Position: 11, Rider: Max Poole, Team: Team Picnic Postnl, Time (total sum): 79:36:57
Position: 12, Rider: Adam Yates, Team: Uae Team Emirates Xrg, Time (total sum): 79:40:25
Position: 13, Rider: Rafal Majka, Team: Uae Team Emirates Xrg, Time (total sum): 79:42:28
Position: 14, Rider: Davide Piganzoli, Team: Team Polti Visitmalta, Time (total sum): 79:46:35
Position: 15, Rider: Nicolas Prodhomme, Team: Decathlon Ag2R La Mondiale Team, Time (total sum): 79:54:07

KING OF THE MOUNTAINS INFORMATION
Maglia Azzurra Standings after this race, top 15:
Position: 1, Rider: Lorenzo Fortunato, Team: Xds Astana Team, Points (total sum): 353.0
Position: 2, Rider: Christian Scaroni, Team: Xds Astana Team, Points (total sum): 200.0
Position: 3, Rider: Nicolas Prodhomme, Team: Decathlon Ag2R La Mondiale Team, Points (total sum): 107.0
Position: 4, Rider: Manuele Tarozzi, Team: Vf Group Bardianicsf-Faizane', Points (total sum): 87.0
Position: 5, Rider: Carlos Verona, Team: Lidl-Trek, Points (total sum): 61.0
Position: 6, Rider: Christopher Harper, Team: Team Jayco Alula, Points (total sum): 60.0
Position: 7, Rider: Romain Bardet, Team: Team Picnic Postnl, Points (total sum): 47.0
Position: 8, Rider: Richard Carapaz, Team: Ef Education - Easypost, Points (total sum): 47.0
Position: 9, Rider: Isaac Del Toro, Team: Uae Team Emirates Xrg, Points (total sum): 45.0
Position: 10, Rider: Alessandro Verre, Team: Arkea-B&B Hotels, Points (total sum): 42.0
Position: 11, Rider: Pello Bilbao, Team: Bahrain Victorious, Points (total sum): 42.0
Position: 12, Rider: Paul Double, Team: Team Jayco Alula, Points (total sum): 36.0
Position: 13, Rider: Diego Ulissi, Team: Xds Astana Team, Points (total sum): 34.0
Position: 14, Rider: Rafal Majka, Team: Uae Team Emirates Xrg, Points (total sum): 33.0
Position: 15, Rider: Sylvain Moniquet, Team: Cofidis, Points (total sum): 32.0
KOM climb 1 of this stage standings:
Position: 1, Rider: Manuele Tarozzi, Team: Vf Group Bardianicsf-Faizane', Points: 3
Position: 2, Rider: Martin Marcellusi, Team: Vf Group Bardianicsf-Faizane', Points: 2
Position: 3, Rider: Kévin Geniets, Team: Groupama-Fdj, Points: 1
KOM climb 2 of this stage standings:

Position: 1, Rider: Manuele Tarozzi, Team: Vf Group Bardianicsf-Faizane', Points: 18
Position: 2, Rider: Jacopo Mosca, Team: Lidl-Trek, Points: 8
Position: 3, Rider: Carlos Verona, Team: Lidl-Trek, Points: 6
Position: 4, Rider: Mads Pedersen, Team: Lidl-Trek, Points: 4
Position: 5, Rider: Enzo Paleni, Team: Groupama-Fdj, Points: 2
Position: 6, Rider: Kevin Colleoni, Team: Intermarché - Wanty, Points: 1
KOM climb 3 of this stage standings:
Position: 1, Rider: Christopher Harper, Team: Team Jayco Alula, Points: 50
Position: 2, Rider: Alessandro Verre, Team: Arkea-B&B Hotels, Points: 30
Position: 3, Rider: Gianmarco Garofoli, Team: Soudal Quick-Step, Points: 20
Position: 4, Rider: Carlos Verona, Team: Lidl-Trek, Points: 14
Position: 5, Rider: Wout Van Aert, Team: Team Visma - Lease A Bike, Points: 10
Position: 6, Rider: Rémy Rochas, Team: Groupama-Fdj, Points: 6
Position: 7, Rider: Martin Marcellusi, Team: Vf Group Bardianicsf-Faizane', Points: 4
Position: 8, Rider: Simon Yates, Team: Team Visma - Lease A Bike, Points: 2
Position: 9, Rider: Christopher Hamilton, Team: Team Picnic Postnl, Points: 1
KOM climb 4 of this stage standings:
Position: 1, Rider: Christopher Harper, Team: Team Jayco Alula, Points: 9
Position: 2, Rider: Alessandro Verre, Team: Arkea-B&B Hotels, Points: 4
Position: 3, Rider: Simon Yates, Team: Team Visma - Lease A Bike, Points: 2
Position: 4, Rider: Gianmarco Garofoli, Team: Soudal Quick-Step, Points: 1

POINTS CLASSIFICATION INFORMATION
Maglia Ciclamino Standings after this race, top 15:
Position: 1, Rider: Mads Pedersen, Team: Lidl-Trek, Points (total sum): 277.0
Position: 2, Rider: Olav Kooij, Team: Team Visma - Lease A Bike, Points (total sum): 135.0
Position: 3, Rider: Wout Van Aert, Team: Team Visma - Lease A Bike, Points (total sum): 127.0
Position: 4, Rider: Dries De Bondt, Team: Decathlon Ag2R La Mondiale Team, Points (total sum): 119.0
Position: 5, Rider: Isaac Del Toro, Team: Uae Team Emirates Xrg, Points (total sum): 109.0
Position: 6, Rider: Casper Van Uden, Team: Team Picnic Postnl, Points (total sum): 88.0
Position: 7, Rider: Richard Carapaz, Team: Ef Education - Easypost, Points (total sum): 77.0
Position: 8, Rider: Alessandro Tonelli, Team: Team Polti Visitmalta, Points (total sum): 76.0
Position: 9, Rider: Mirco Maestri, Team: Team Polti Visitmalta, Points (total sum): 72.0
Position: 10, Rider: Orluis Aular, Team: Movistar Team, Points (total sum): 68.0
Position: 11, Rider: Kaden Groves, Team: Alpecin-Deceuninck, Points (total sum): 63.0
Position: 12, Rider: Kasper Asgreen, Team: Ef Education - Easypost, Points (total sum): 60.0
Position: 13, Rider: Nico Denz, Team: Red Bull - Bora - Hansgrohe, Points (total sum): 50.0
Position: 14, Rider: Thomas Pidcock, Team: Q36.5 Pro Cycling Team, Points (total sum): 47.0
Position: 15, Rider: Lorenzo Fortunato, Team: Xds Astana Team, Points (total sum): 45.0
KOM climb 1 of this stage standings:
Position: 1, Rider: Dries De Bondt, Team: Decathlon Ag2R La Mondiale Team, Points: 12
Position: 2, Rider: Wout Van Aert, Team: Team Visma - Lease A Bike, Points: 8
Position: 3, Rider: Gianmarco Garofoli, Team: Soudal Quick-Step, Points: 5
Position: 4, Rider: Manuele Tarozzi, Team: Vf Group Bardianicsf-Faizane', Points: 3
Position: 5, Rider: Milan Vader, Team: Q36.5 Pro Cycling Team, Points: 1
KOM climb 2 of this stage standings:
Position: 1, Rider: Wout Van Aert, Team: Team Visma - Lease A Bike, Points: 12
Position: 2, Rider: Dries De Bondt, Team: Decathlon Ag2R La Mondiale Team, Points: 8
Position: 3, Rider: Mirco Maestri, Team: Team Polti Visitmalta, Points: 5
Position: 4, Rider: Francesco Busatto, Team: Intermarché - Wanty, Points: 3
Position: 5, Rider: Ethan Hayter, Team: Soudal Quick-Step, Points: 1
KOM climb 3 of this stage standings:
Position: 1, Rider: Wout Van Aert, Team: Team Visma - Lease A Bike, Points: 20
Position: 2, Rider: Dries De Bondt, Team: Decathlon Ag2R La Mondiale Team, Points: 20

Position: 3, Rider: Christopher Harper, Team: Team Jayco Alula, Points: 15
Position: 4, Rider: Alessandro Verre, Team: Arkea-B&B Hotels, Points: 12
Position: 5, Rider: Gianmarco Garofoli, Team: Soudal Quick-Step, Points: 12
Position: 6, Rider: Simon Yates, Team: Team Visma - Lease A Bike, Points: 9
Position: 7, Rider: Rémy Rochas, Team: Groupama-Fdj, Points: 6
Position: 8, Rider: Martin Marcellusi, Team: Vf Group Bardianicsf-Faizane', Points: 5
Position: 9, Rider: Mirco Maestri, Team: Team Polti Visitmalta, Points: 5
Position: 10, Rider: Carlos Verona, Team: Lidl-Trek, Points: 4
Position: 11, Rider: Max Poole, Team: Team Picnic Postnl, Points: 3
Position: 12, Rider: Manuele Tarozzi, Team: Vf Group Bardianicsf-Faizane', Points: 3
Position: 13, Rider: Francesco Busatto, Team: Intermarché - Wanty, Points: 3
Position: 14, Rider: Isaac Del Toro, Team: Uae Team Emirates Xrg, Points: 2
Position: 15, Rider: Giulio Pellizzari, Team: Red Bull - Bora - Hansgrohe, Points: 1

OFFICIAL WITHDRAWALS INFORMATION
Rider: None, Team: None

PREVIOUS STANDINGS INFORMATION
Maglia Rosa Standings before this race, top 10:
Position: 1, Rider: Isaac Del Toro, Team: Uae Team Emirates Xrg
Position: 2, Rider: Richard Carapaz, Team: Ef Education - Easypost
Position: 3, Rider: Simon Yates, Team: Team Visma - Lease A Bike
Position: 4, Rider: Derek Gee, Team: Israel - Premier Tech
Position: 5, Rider: Damiano Caruso, Team: Bahrain Victorious
Position: 6, Rider: Egan Bernal, Team: Ineos Grenadiers
Position: 7, Rider: Giulio Pellizzari, Team: Red Bull - Bora - Hansgrohe
Position: 8, Rider: Einer Rubio, Team: Movistar Team
Position: 9, Rider: Michael Storer, Team: Tudor Pro Cycling Team
Position: 10, Rider: Brandon Mcnulty, Team: Uae Team Emirates Xrg
Maglia Azzurra Standings before this race, top 5:
Position: 1, Rider: Lorenzo Fortunato, Team: Xds Astana Team
Position: 2, Rider: Christian Scaroni, Team: Xds Astana Team
Position: 3, Rider: Nicolas Prodhomme, Team: Decathlon Ag2R La Mondiale Team
Position: 4, Rider: Manuele Tarozzi, Team: Vf Group Bardianicsf-Faizane'
Position: 5, Rider: Romain Bardet, Team: Team Picnic Postnl
Maglia Ciclamino Standings before this race, top 5:
Position: 1, Rider: Mads Pedersen, Team: Lidl-Trek
Position: 2, Rider: Olav Kooij, Team: Team Visma - Lease A Bike
Position: 3, Rider: Wout Van Aert, Team: Team Visma - Lease A Bike
Position: 4, Rider: Isaac Del Toro, Team: Uae Team Emirates Xrg
Position: 5, Rider: Dries De Bondt, Team: Decathlon Ag2R La Mondiale Team
-----Data End-----
Do not mention retirements if they are not in top 15 of any standing and consider talking about the variations of the three standings.
Use only the most relevant data to make the article as interesting as possible.
additionally, THE FOLLOWING REQUESTS ARE VERY IMPORTANT STICK ON THEM: write the article with a serious and technical tone, it is for a cycling specific journal

Appendix C – Article Example

Below is an example of an article generated using Llama by Meta.
The full file containing all the articles produced within this project is available as a separate document attached to this report.
The article is the combination of outputs of the following calls of the functions *cycling_journalist* and *title_creator*.

cycling_journalist(HF_Token, data, "20th Stage of Giro d'Italia", competition_code = 'G', approx_length = 200, model = 'meta-llama/llama-4-maverick-17b-128e-instruct-fp8')

title_creator(HF_Token, Words_200_articles['Llama'], model = 'meta-llama/llama-4-maverick-17b-128e-instruct-fp8')

Therefore the article is the following:

Giro d'Italia Stage 20 Results and Standings

The 20th stage of the Giro d'Italia was a grueling mountain stage that saw Christopher Harper of Team Jayco Alula take the win, crossing the finish line in Sestrière with a time of 5:27:29. Harper dominated the stage, particularly on the Colle delle Finestre climb, where he took the majority of the points, and finished nearly two minutes ahead of the next closest rider, Alessandro Verre of Arkea-B&B Hotels. Simon Yates of Team Visma - Lease A Bike finished third, further solidifying his position in the Maglia Rosa standings, where he took the lead after the stage. The stage was marked by challenging weather conditions, with light clouds and a temperature of 24 degrees Celsius at the start, and drizzle and a temperature of 20 degrees Celsius at the finish. The overall standings saw significant changes, with Isaac Del Toro dropping from first to second place, and Richard Carapaz and Derek Gee rounding out the top five. The Maglia Azzurra standings remained relatively stable, with Lorenzo Fortunato maintaining his lead, while the Maglia Ciclamino standings saw Mads Pedersen continue to hold the top spot. A total of 159 riders finished the stage out of the starting grid.