Model Selection

Get API Key

In [1]:
```
from helper import load_mistral_api_key
api_key, dlai_endpoint = load_mistral_api_key(ret_key=True)
```

- Note: in the classroom, if you print out this `api_key` variable, it is not a real API key (for security reasons).
- If you wish to run this code on your own machine, outside of the classroom, you can still reuse the code that you see in `helper.py`.
- It uses python-dotenv (https://pypi.org/project/python-dotenv/) library to securely save and load sensitive information such as API keys.

In [2]:
```
import os
from mistralai.client import MistralClient
from mistralai.models.chat_completion import ChatMessage

def mistral(user_message, model="mistral-small-latest", is_json=
    client = MistralClient(api_key=api_key, endpoint=dlai_endpoi
    messages = [ChatMessage(role="user", content=user_message)]

    if is_json:
        chat_response = client.chat(
            model=model, messages=messages, response_format={"ty
        )
    else:
        chat_response = client.chat(model=model, messages=messag

    return chat_response.choices[0].message.content
```

Mistral Small

Good for simple tasks, fast inference, lower cost.

- classification

In [3]:
```
prompt = """
Classify the following email to determine if it is spam or not.
Only respond with the exact text "Spam" or "Not Spam".

# Email:
🎉 Urgent! You've Won a $1,000,000 Cash Prize!
💰 To claim your prize, please click on the link below:
https://bit.ly/claim-your-prize
"""
```

In [4]:
```
mistral(prompt, model="mistral-small-latest")
```

```
'Spam'
```

Mistral Medium

Good for intermediate tasks such as language transformation.

- Composing text based on provided context (e.g. writing a customer service email based on purchas information).

In [5]:
```
prompt = """
Compose a welcome email for new customers who have just made
their first purchase with your product.
Start by expressing your gratitude for their business,
and then convey your excitement for having them as a customer.
Include relevant details about their recent order.
Sign the email with "The Fun Shop Team".

Order details:
- Customer name: Anna
- Product: hat
- Estimate date of delivery: Feb. 25, 2024
- Return policy: 30 days
"""
```

In [6]:
```
response_medium = mistral(prompt, model="mistral-medium-latest")
```

In [7]:
```python
print(response_medium)
```

```
Subject: Welcome to The Fun Shop, Anna! Your Hat is on Its Way 🎩

Dear Anna,

We are absolutely thrilled to have you as a new customer at The Fu
n Shop! First and foremost, thank you so much for choosing our hat
for your recent purchase. Your support means the world to us.

Your order is now being processed, and we're excited to let you kn
ow that your stylish new hat will be on its way soon. According to
our current estimates, your order should arrive at your doorstep b
y February 25, 2024. We'll send you an email with tracking informa
tion once your package has shipped, so you can keep an eye on its
progress.

In the meantime, we want you to know that your satisfaction is our
top priority. That's why we offer a 30-day return policy, allowing
you to exchange or return your hat for any reason within that time
frame. If you have any questions or concerns about your order, ple
ase don't hesitate to reach out to our friendly customer service t
eam at [support@thefunshop.com](mailto:support@thefunshop.com).

Again, thank you for your business, Anna! We're so glad to have yo
u as a part of our Fun Shop family, and we can't wait for you to e
njoy your new hat. Stay tuned for more fun products and updates fr
om our team!

Best regards,

The Fun Shop Team 😊

www.thefunshop.com

P.S. Don't forget to follow us on social media for exclusive deal
s, giveaways, and a daily dose of fun!

Facebook: @TheFunShop
Instagram: @TheFunShopOfficial
Twitter: @TheFunShopNow
```

Mistral Large:

Good for complex tasks that require advanced reasoning.

- Math and reasoning with numbers.

In [8]:
```
prompt = """
Calculate the difference in payment dates between the two \
customers whose payment amounts are closest to each other \
in the following dataset. Do not write code.

# dataset:
'{
  "transaction_id":{"0":"T1001","1":"T1002","2":"T1003","3":"T10
    "customer_id":{"0":"C001","1":"C002","2":"C003","3":"C002","
    "payment_amount":{"0":125.5,"1":89.99,"2":120.0,"3":54.3,"4"
"payment_date":{"0":"2021-10-05","1":"2021-10-06","2":"2021-10-0
    "payment_status":{"0":"Paid","1":"Unpaid","2":"Paid","3":"Pa
}'
"""
```

In [9]:
```
response_small = mistral(prompt, model="mistral-small-latest")
```

In [10]:
```
print(response_small)
```

```
To find the difference in payment dates between the two customers
whose payment amounts are closest to each other, let's first ident
ify the customers with the closest payment amounts.

From the dataset, the payment amounts for each customer are as fol
lows:
- Customer C001: $125.5 and $210.2
- Customer C002: $89.99 and $54.3
- Customer C003: $120.0

The closest payment amounts between two customers belong to C001 a
nd C002. For C001, the payment amount is $125.5, and for C002, it
is $89.99.

Now, let's find the payment dates for these customers:
- Customer C001: $125.5 paid on "2021-10-05"
- Customer C002: $89.99 paid on "2021-10-06"

Since the payment amounts for these two customers are closest to e
ach other, we can now calculate the difference in payment dates.

The payment date for C001 is "2021-10-05", and the payment date fo
r C002 is "2021-10-06". The difference between these dates is 1 da
y.

So, the difference in payment dates between the two customers whos
e payment amounts are closest to each other is 1 day.
```

In [11]:
```
response_large = mistral(prompt, model="mistral-large-latest")
```

In [12]:
```
print(response_large)
```

```
To solve this problem without writing code, we first need to ident
ify the two customers whose payment amounts are closest to each ot
her.

The payment amounts are as follows:
- C001: 125.5, 210.2
- C002: 89.99, 54.3
- C003: 120.0

The closest payments are 125.5 (C001) and 120.0 (C003) with a diff
erence of 5.5.

Next, we need to calculate the difference in payment dates. The pa
yment dates are:
- C001: 2021-10-05, 2021-10-08
- C002: 2021-10-06, 2021-10-05
- C003: 2021-10-07

The payments we're interested in were made on 2021-10-05 (C001) an
d 2021-10-07 (C003). The difference in payment dates is 2 days.
```

Expense reporting task

In [13]:
```
transactions = """
McDonald's: 8.40
Safeway: 10.30
Carrefour: 15.00
Toys R Us: 20.50
Panda Express: 10.20
Beanie Baby Outlet: 25.60
World Food Wraps: 22.70
Stuffed Animals Shop: 45.10
Sanrio Store: 85.70
"""

prompt = f"""
Given the purchase details, how much did I spend on each categor
1) restaurants
2) groceries
3) stuffed animals and props
{transactions}
"""
```

In [14]:
```
response_small = mistral(prompt, model="mistral-small-latest")
print(response_small)
```

```
To find out how much you spent on each category, we need to group
your purchases accordingly. Here's the breakdown:

1) restaurants:
   - McDonald's: $8.40
   - Panda Express: $10.20
Total spent on restaurants: $18.60

2) groceries:
   - Safeway: $10.30
   - Carrefour: $15.00
   - World Food Wraps: $22.70
Total spent on groceries: $48.00

3) stuffed animals and props:
   - Toys R Us: $20.50
   - Beanie Baby Outlet: $25.60
   - Stuffed Animals Shop: $45.10
   - Sanrio Store: $85.70
Total spent on stuffed animals and props: $176.90

So, you spent $18.60 on restaurants, $48.00 on groceries, and $17
6.90 on stuffed animals and props.
```

In [15]:
```
response_large = mistral(prompt, model="mistral-large-latest")
print(response_large)
```

```
Sure, I'd be happy to help you categorize your spending. Here's th
e breakdown:

1) Restaurants:
   - McDonald's: $8.40
   - Panda Express: $10.20
   - World Food Wraps: $22.70
   Total spent on restaurants: $41.30

2) Groceries:
   - Safeway: $10.30
   - Carrefour: $15.00
   Total spent on groceries: $25.30

3) Stuffed animals and props:
   - Toys R Us: $20.50
   - Beanie Baby Outlet: $25.60
   - Stuffed Animals Shop: $45.10
   - Sanrio Store: $85.70
   Total spent on stuffed animals and props: $176.90
```

Writing and checking code

In [16]:
```
user_message = """
Given an array of integers nums and an integer target, return in

You may assume that each input would have exactly one solution,

You can return the answer in any order.

Your code should pass these tests:

assert twoSum([2,7,11,15], 9) == [0,1]
assert twoSum([3,2,4], 6) == [1,2]
assert twoSum([3,3], 6) == [0,1]
"""
```

In [17]:
```
print(mistral(user_message, model="mistral-large-latest"))
```

~~~
Here is a Python solution for the problem:

```python
def twoSum(nums, target):
    if len(nums) <= 1:
        return False

    num_map = {}
    for i, num in enumerate(nums):
        num_map[num] = i

    for i, num in enumerate(nums):
        complement = target - num
        if complement in num_map and num_map[complement] != i:
            return [i, num_map[complement]]

    return False
```

This function first checks if the input list is too small to have
a solution. Then it creates a dictionary (num_map) to store the nu
mbers in the list as keys and their indices as values.

Then it iterates over the list again, and for each number, it calc
ulates the complement (the number that needs to be added to the cu
rrent number to get the target). If the complement is in the num_m
ap and its index is not the same as the current index (to avoid us
ing the same element twice), it returns the current index and the
index of the complement.

If it finishes iterating over the list without finding a solution,
it returns False.

This function should pass the tests you provided.
~~~

Try out the code that the model provided

- Copy the code that the model provided and try running it!

Here is the code that was output at the time of filming:

```
def twoSum(nums, target):
    seen = {}
    for i, num in enumerate(nums):
        complement = target - num
        if complement in seen:
            return [seen[complement], i]
        seen[num] = i
```

- Also try running the assert statements in the original prompt

```
assert twoSum([2,7,11,15], 9) == [0,1]
assert twoSum([3,2,4], 6) == [1,2]
assert twoSum([3,3], 6) == [0,1]
```

In [18]:
```
def twoSum(nums, target):
    seen = {}
    for i, num in enumerate(nums):
        complement = target - num
        if complement in seen:
            return [seen[complement], i]
        seen[num] = i
```

In [19]:
```
assert twoSum([2,7,11,15], 9) == [0,1]
assert twoSum([3,2,4], 6) == [1,2]
assert twoSum([3,3], 6) == [0,1]
```

Natively Fluent in English, French, Spanish, German, and Italian

- This means that you can use Mistral models for more than translating from one language to another.
- If you are a native Spanish speaker, for instance, you can communicate with Mistral models in Spanish for any of your tasks.

In [20]:
```
user_message = """
Lequel est le plus lourd une livre de fer ou un kilogramme de pl
"""
```

In [21]:
```
print(mistral(user_message, model="mistral-large-latest"))
```

```
Une livre de fer et un kilogramme de plumes ont des poids différen
ts.

Une livre est une unité de mesure utilisée notamment aux États-Uni
s, et elle est égale à environ 0,453592 kilogramme. Donc, une livr
e de fer est plus légère qu'un kilogramme de plumes.

Cependant, il est important de noter que cette comparaison est un
peu inhabituelle, car nous avons tendance à penser en termes de po
ids similaires lorsque nous comparons des matériaux différents. Le
s plumes sont connues pour être extrêmement légères, donc un kilog
ramme de plumes représenterait un très grand nombre de plumes.
```

Try it out for yourself

- Try communicating with the Mistral Large model in Spanish
  - (If you need help, you can first translate a prompt from English to Spanish, and then prompt the model in Spanish).

In [22]:
```
user_message = """
Puedes entender espanol?
"""
```

In [23]:
```
print(mistral(user_message, model="mistral-large-latest"))
```

```
Sí, puedo entender y responder en español. ¿En qué puedo ayudarte?
```

List of Mistral models that you can call:

You can also call the two open source mistral models via API calls. Here is the list of models that you car
try:

```
open-mistral-7b
open-mixtral-8x7b
open-mixtral-8x22b
mistral-small-latest
mistral-medium-latest
mistral-large-latest
```

For example:

```
mistral(prompt, model="open-mixtral-8x22b")
```

Note that we just released the `open-mixtral-8x22b` model. Check out our release blog (https://mistral.ai/news/mixtral-8x22b/) for details.

In [ ]:
```
```