scheduler

July 4, 2024

0.1 AI Scheduling assistant

- hours will always be in 24 hour format
- minutes will be between 0 - 60

[32]:
```
import pandas as pd
import datetime
import os
```

[ ]:
```

```

[33]:
```
days = {
    0: 'Monday',
    1: 'Tuesday',
    2: 'Wednesday',
    3: 'Thursday',
    4: 'Friday',
    5: 'Saturday',
    6: 'Sunday'
}

weekdays = [0,1,2,3,4]
weekends = [5,6]

units = {
    'h': 'hour',
    'm': 'minutes',
    'd': 'day',
    'w': 'week',
    'm': 'month',
    'y': 'year',
    's': 'seconds'
}


default_rules = [{
    'event': 'lunch',
    'hour': 12,
    'span': 1,
    'span_unit': 'h',
    'repeat': 1,
    'repeat_unit': 'd',
    'start_date': datetime.date.today(),
    'end_date': None
},{
    'event': 'Meeting with doctor',
    'hour': 13,
    'span': 1,
    'span_unit': 'h',
    'repeat': 0,
    'repeat_unit': 'd',
    'start_date': datetime.date.today(),
    'end_date': datetime.date.today()
},{
    'event': 'dinner',
    'hour': 19,
    'span': 1,
    'span_unit': 'h',
    'repeat': 1,
    'repeat_unit': 'd',
    'start_date': datetime.date.today(),
    'end_date': None

},{
    'event': 'sleep',
    'hour': 23,
    'span': 7,
    'span_unit': 'h',
    'repeat': 1,
    'repeat_unit': 'd',
    'start_date': datetime.date.today(),
    'end_date': None
}]
```

Displaying a base calendar from the nearest monday

[34]:
```
def generate_week_dates(start_date):
    # Generate a list of 7 dates starting from the given date
    week_dates = [start_date + datetime.timedelta(days=i) for i in range(7)]
    return week_dates

def get_nearest_monday(date):
    # Find the weekday number of the given date (Monday=0, Sunday=6)
    weekday = date.weekday()

    # Calculate the difference to the nearest Monday
```

```python
    if weekday <= 3:  # If the date is closer to the previous Monday or same day
        delta = -weekday
    else:  # If the date is closer to the next Monday
        delta = 7 - weekday

    # Calculate the nearest Monday
    nearest_monday = date + datetime.timedelta(days=delta)

    return nearest_monday


def calendar_creator(start_date):
    hours = [i/10 for i in range(0,245,5)]
    week_dates = generate_week_dates(get_nearest_monday(start_date))
    days_arr = [f'{day} - {wk_dt}' for day, wk_dt in zip(days.values(),␣
 ↪week_dates)]


    base_cal_dict = {
        'hour': hours
    }

    for day in days_arr:
        base_cal_dict[day] = ['' for hour in hours]

    base_cal = pd.DataFrame(base_cal_dict)
    base_cal.set_index('hour', inplace=True)
    return base_cal

base_cal = calendar_creator(datetime.date.today())
base_cal
```

[34]:
```
     Monday - 2024-07-01 Tuesday - 2024-07-02 Wednesday - 2024-07-03  \
hour
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
```

```
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0
19.5
20.0
20.5
21.0
21.5
22.0
22.5
23.0
23.5
24.0

     Thursday - 2024-07-04 Friday - 2024-07-05 Saturday - 2024-07-06  \
hour
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
```

```
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0
19.5
20.0
20.5
21.0
21.5
22.0
22.5
23.0
23.5
24.0

     Sunday - 2024-07-07
hour
0.0
0.5
1.0
```

```
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0
19.5
20.0
20.5
21.0
21.5
22.0
22.5
23.0
23.5
24.0
```

[35]:
```
def get_hours_to_block(rule):

    rule_hr_day = []
    hour = rule['hour']
    span = rule['span']
    span_unit = rule['span_unit']
    repeat = rule['repeat']
    repeat_unit = rule['repeat_unit']
    event = rule['event']
    start_date = rule['start_date']
    end_date = rule['end_date'] if rule['end_date'] is not None else␣
 ↪rule['start_date'] + datetime.timedelta(days=7)

    looper = start_date

    # For simplicity we are only going to use hour and day as repeaters

    if (start_date == end_date) and (span_unit == 'h'):
        rule_hr_day.append({
            'col': f'{days[looper.weekday()]} - {looper}',
            'hour': [rule['hour'] + i/10 for i in range(0,(span*10)+5,5)],
            'event': event

        })
    else:
        while looper != end_date:
            if repeat_unit == 'd':
                rule_hr_day.append({
                    'col': f'{days[looper.weekday()]} - {looper}',
                    'hour': [rule['hour'] + i/10 for i in range(0,(span*10)+␣
 ↪5,5)],
                    'event': event

                })
            looper += datetime.timedelta(days=1)

    return rule_hr_day
```

[36]:
```
# rules_with_hrs = []
# rules_with_hrs.extend(get_hours_to_block(
#     default_rules[i]
# ) for i in range(len(default_rules)))
# rules_with_hrs
```

[37]:
```
def fill_up_calendar(cal, rule_list):

    for rl in rule_list:
        colm = rl['col']
        event = rl['event']
        if colm in cal.columns:
            # for hr in rl['hour']:
            hr = [i - 24 if  i > 23.5 else i for i in rl['hour']]
            cal.loc[hr, colm] = event
        else:
            pass

    return cal

for i in range(len(default_rules)):
    rules_with_hrs = get_hours_to_block(default_rules[i])
    fill_up_calendar(base_cal, rule_list=rules_with_hrs)
```

[38]:
```
base_cal
```

[38]:
```
     Monday - 2024-07-01 Tuesday - 2024-07-02 Wednesday - 2024-07-03  \
hour
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
```

```
14.0
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0
19.5
20.0
20.5
21.0
21.5
22.0
22.5
23.0
23.5
24.0

     Thursday - 2024-07-04 Friday - 2024-07-05 Saturday - 2024-07-06  \
hour
0.0                  sleep               sleep                 sleep
0.5                  sleep               sleep                 sleep
1.0                  sleep               sleep                 sleep
1.5                  sleep               sleep                 sleep
2.0                  sleep               sleep                 sleep
2.5                  sleep               sleep                 sleep
3.0                  sleep               sleep                 sleep
3.5                  sleep               sleep                 sleep
4.0                  sleep               sleep                 sleep
4.5                  sleep               sleep                 sleep
5.0                  sleep               sleep                 sleep
5.5                  sleep               sleep                 sleep
6.0                  sleep               sleep                 sleep
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
```

```
11.5
12.0                 lunch               lunch                 lunch
12.5                 lunch               lunch                 lunch
13.0   Meeting with doctor               lunch                 lunch
13.5   Meeting with doctor
14.0   Meeting with doctor
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0                dinner              dinner                dinner
19.5                dinner              dinner                dinner
20.0                dinner              dinner                dinner
20.5
21.0
21.5
22.0
22.5
23.0                 sleep               sleep                 sleep
23.5                 sleep               sleep                 sleep
24.0

     Sunday - 2024-07-07
hour
0.0                sleep
0.5                sleep
1.0                sleep
1.5                sleep
2.0                sleep
2.5                sleep
3.0                sleep
3.5                sleep
4.0                sleep
4.5                sleep
5.0                sleep
5.5                sleep
6.0                sleep
6.5
7.0
7.5
8.0
8.5
```

```
9.0
9.5
10.0
10.5
11.0
11.5
12.0                 lunch
12.5                 lunch
13.0                 lunch
13.5
14.0
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0                dinner
19.5                dinner
20.0                dinner
20.5
21.0
21.5
22.0
22.5
23.0                 sleep
23.5                 sleep
24.0
```

- we can see the base calendar with this is working fine, we will now use LLMS to generate rule json for shifting

[39]:
```python
base_rule_json = {
    'event': None,
    'hour': None,
    'span': None,
    'span_unit': None,
    'repeat': None,
    'repeat_unit': None,
    'start_date': None,
    'end_date': None
}

with open('openai_key.txt', 'r') as f:
    openai_key = f.readline()

os.environ['OPENAI_API_KEY'] = openai_key
```

```
[49]: import requests
import json
def query_gpt(base_rule_json, question):

    # Define the endpoint
    url = 'https://api.openai.com/v1/chat/completions'

    # Set up the headers
    augment_prompt = f"""
        Please follow the following instructions and always respond in JSON only.
 ↪ I do not need any explainations. Your only job is to reply a filled json. I␣
 ↪am creating an AI scheduling assistant where the event is described using the␣
 ↪following dictionary:

        {str(base_rule_json)}
        \n
        units are described using the following dictionary
        {str(units)}
        \n
        This is the current schedule of the user
        {str(default_rules)}
        Wherever the end date is null, the end date is 7 days from the start␣
 ↪date.
        \n
        The current schedule has a repeatation of events between start date and␣
 ↪end date, if a customer is only asking for a change on one day please pay␣
 ↪close attention that the end_date should be of that particular day.
        \n
        Now, using the information above and the following user request
        {question}
        \n
        Give me a json which tells me the event type to alter, also values to␣
 ↪alter, and the dates to alter them on.
        \n
        Give me the json in the following format

            'event':'value',
            all the values to alter
            'start_date': '',
            'end_date': ''

    """
    headers = {
        'Content-Type': 'application/json',
        'Authorization': f'Bearer {openai_key}',
    }

    # Define the data payload
    data = {
        "model": "gpt-4",
        "messages": [
            {"role": "system", "content": "You are a helpful assistant who fills␣
 ↪up and responds in json"},
            {"role": "user", "content": augment_prompt}
        ]
    }

    # Send the request
    response = requests.post(url, headers=headers, data=json.dumps(data))

    # Check if the request was successful
    if response.status_code == 200:
        response_data = response.json()
        print("Response from ChatGPT API:")
        print(json.dumps(response_data, indent=2))
        return json.loads(response_data['choices'][0]['message']['content'])
    else:
        print(f"Error: {response.status_code}")
        print(response.text)


altering_json = query_gpt(
    base_rule_json=base_rule_json,
    question='Can you shift my lunch by half an hour tomorrow?'
)

altering_json
```

```
Response from ChatGPT API:
{
  "id": "chatcmpl-9hUKpQEqwNSVPXNKZjUlroI3CUjlG",
  "object": "chat.completion",
  "created": 1720151739,
  "model": "gpt-4-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "{\n    \"event\": \"lunch\",\n    \"hour\": 12.5,\n
\"start_date\": \"2024-07-05\",\n    \"end_date\": \"2024-07-05\"\n}"
```

```
      },
      "logprobs": null,
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 588,
    "completion_tokens": 45,
    "total_tokens": 633
  },
  "system_fingerprint": null
}
```

[49]:
```
{'event': 'lunch',
 'hour': 12.5,
 'start_date': '2024-07-05',
 'end_date': '2024-07-05'}
```

[50]:
```python
def fill_missing_keys(all_rules, llm_response):

    selected_rule = None
    for rule in all_rules:
        if rule['event'] == llm_response['event']:
            selected_rule = rule
            break

    if selected_rule is None:
        print('No matching rule found')
        return None

    for key in selected_rule.keys():
        if key not in llm_response:
            llm_response[key] = selected_rule[key]

    date_format = "%Y-%m-%d"
    if isinstance(llm_response['start_date'], str):
        llm_response['start_date'] = datetime.datetime.
 ↪strptime(llm_response['start_date'], date_format).date()

    if isinstance(llm_response['end_date'], str):
        llm_response['end_date'] = datetime.datetime.
 ↪strptime(llm_response['end_date'], date_format).date()

    return llm_response


llm_response = fill_missing_keys(default_rules, altering_json)
```

```
llm_response
```

```
[50]: {'event': 'lunch',
       'hour': 12.5,
       'start_date': datetime.date(2024, 7, 5),
       'end_date': datetime.date(2024, 7, 5),
       'span': 1,
       'span_unit': 'h',
       'repeat': 1,
       'repeat_unit': 'd'}
```

```
[51]: def clear_blocks(rule_with_hrs, cal):

          for rl in rule_with_hrs:
              print(rl['col'])
              if rl['col'] in cal.columns:
                  print('in here')
                  cal[rl['col']] = cal[rl['col']].replace({
                      rl['event']: ''
                  })

          return cal

      rules_with_hrs = get_hours_to_block(llm_response)
      cal = clear_blocks(rule_with_hrs=rules_with_hrs, cal=base_cal)

      rules_with_hrs
```

```
Friday - 2024-07-05
in here
```

```
[51]: [{'col': 'Friday - 2024-07-05', 'hour': [12.5, 13.0, 13.5], 'event': 'lunch'}]
```

```
[52]: cal
```

```
[52]:      Monday - 2024-07-01 Tuesday - 2024-07-02 Wednesday - 2024-07-03  \
      hour
      0.0
      0.5
      1.0
      1.5
      2.0
      2.5
      3.0
      3.5
      4.0
      4.5
      5.0
```

```
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0
19.5
20.0
20.5
21.0
21.5
22.0
22.5
23.0
23.5
24.0

     Thursday - 2024-07-04 Friday - 2024-07-05 Saturday - 2024-07-06  \
hour
0.0                  sleep               sleep                 sleep
0.5                  sleep               sleep                 sleep
1.0                  sleep               sleep                 sleep
1.5                  sleep               sleep                 sleep
2.0                  sleep               sleep                 sleep
2.5                  sleep               sleep                 sleep
```

```
3.0                  sleep               sleep                  sleep
3.5                  sleep               sleep                  sleep
4.0                  sleep               sleep                  sleep
4.5                  sleep               sleep                  sleep
5.0                  sleep               sleep                  sleep
5.5                  sleep               sleep                  sleep
6.0                  sleep               sleep                  sleep
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0                 lunch                                      lunch
12.5                 lunch                                      lunch
13.0   Meeting with doctor                                      lunch
13.5   Meeting with doctor
14.0   Meeting with doctor
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0                dinner              dinner                 dinner
19.5                dinner              dinner                 dinner
20.0                dinner              dinner                 dinner
20.5
21.0
21.5
22.0
22.5
23.0                 sleep               sleep                  sleep
23.5                 sleep               sleep                  sleep
24.0

     Sunday - 2024-07-07
hour
0.0                sleep
```

```
0.5                sleep
1.0                sleep
1.5                sleep
2.0                sleep
2.5                sleep
3.0                sleep
3.5                sleep
4.0                sleep
4.5                sleep
5.0                sleep
5.5                sleep
6.0                sleep
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0               lunch
12.5               lunch
13.0               lunch
13.5
14.0
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0              dinner
19.5              dinner
20.0              dinner
20.5
21.0
21.5
22.0
22.5
23.0               sleep
23.5               sleep
```

```
24.0
```

[53]:
```
fill_up_calendar(base_cal, rule_list=rules_with_hrs)
```

[53]:
```
     Monday - 2024-07-01 Tuesday - 2024-07-02 Wednesday - 2024-07-03  \
hour
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0
19.5
20.0
```

```
20.5
21.0
21.5
22.0
22.5
23.0
23.5
24.0

     Thursday - 2024-07-04 Friday - 2024-07-05 Saturday - 2024-07-06  \
hour
0.0                  sleep               sleep                  sleep
0.5                  sleep               sleep                  sleep
1.0                  sleep               sleep                  sleep
1.5                  sleep               sleep                  sleep
2.0                  sleep               sleep                  sleep
2.5                  sleep               sleep                  sleep
3.0                  sleep               sleep                  sleep
3.5                  sleep               sleep                  sleep
4.0                  sleep               sleep                  sleep
4.5                  sleep               sleep                  sleep
5.0                  sleep               sleep                  sleep
5.5                  sleep               sleep                  sleep
6.0                  sleep               sleep                  sleep
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0                 lunch                                      lunch
12.5                 lunch               lunch                  lunch
13.0   Meeting with doctor               lunch                  lunch
13.5   Meeting with doctor               lunch
14.0   Meeting with doctor
14.5
15.0
15.5
16.0
16.5
17.0
17.5
```

```
18.0
18.5
19.0                dinner              dinner                dinner
19.5                dinner              dinner                dinner
20.0                dinner              dinner                dinner
20.5
21.0
21.5
22.0
22.5
23.0                 sleep               sleep                 sleep
23.5                 sleep               sleep                 sleep
24.0

     Sunday - 2024-07-07
hour
0.0                sleep
0.5                sleep
1.0                sleep
1.5                sleep
2.0                sleep
2.5                sleep
3.0                sleep
3.5                sleep
4.0                sleep
4.5                sleep
5.0                sleep
5.5                sleep
6.0                sleep
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0               lunch
12.5               lunch
13.0               lunch
13.5
14.0
14.5
15.0
```

```
15.5
16.0
16.5
17.0
17.5
18.0
18.5
19.0              dinner
19.5              dinner
20.0              dinner
20.5
21.0
21.5
22.0
22.5
23.0               sleep
23.5               sleep
24.0
```

0.1.1 First proof that it can read from LLMs response and alter our calendar

Next Steps:

1. For conflicting schedules it should perform a search on the calendar to get the next best date and attach it there

```
[ ]:
```