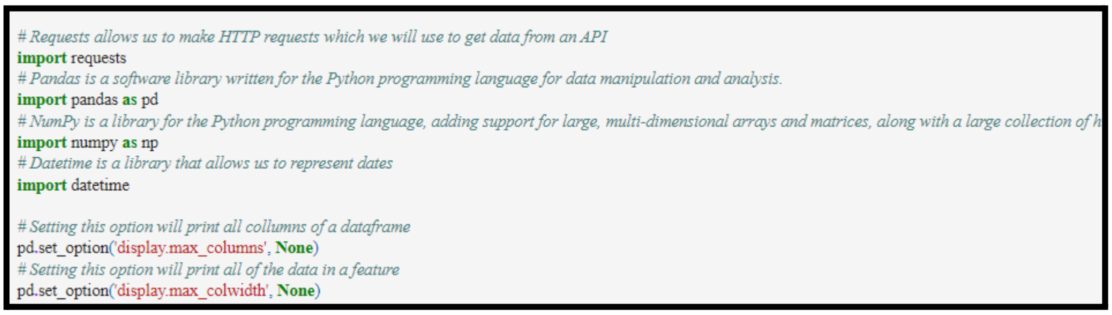
SpaceX Falcon 9 first stage Landing Prediction

This capstone extends points to anticipate the effective landing of the Bird of prey 9 to begin with organize. SpaceX unmistakably advances Hawk 9 rocket dispatches on its site, advertising them at a toll of $62 million, which is essentially lower than the costs charged by other suppliers, frequently surpassing $165 million per dispatch. The key figure contributing to these investment funds is SpaceX's capacity to reuse the primary structure of the rocket. Thus, in order to precisely decide whether the primary arrangement will arrive effectively, we will appraise the cost of a dispatch. This data would be important for an interchange company looking to compete with SpaceX for a rocket dispatch contract. In this lab, you may accumulate information from an API and guarantee that it is legitimately designed. The illustration given underneath illustrates a effective dispatch.

Targets

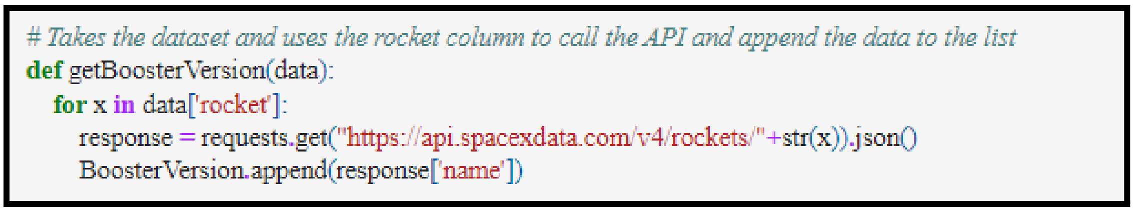
- Ask to the SpaceX API
- Clean the asked information

Import Libraries



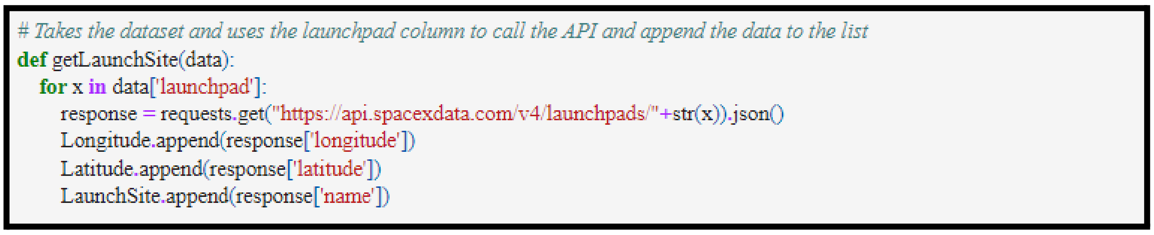
```
# Requests allows us to make HTTP requests which we will use to get data from an API
import requests
# Pandas is a software library written for the Python programming language for data manipulation and analysis.
import pandas as pd
# NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of h
import numpy as np
# Datetime is a library that allows us to represent dates
import datetime

# Setting this option will print all collumns of a dataframe
pd.set_option('display.max_columns', None)
# Setting this option will print all of the data in a feature
pd.set_option('display.max_colwidth', None)
```

The booster name



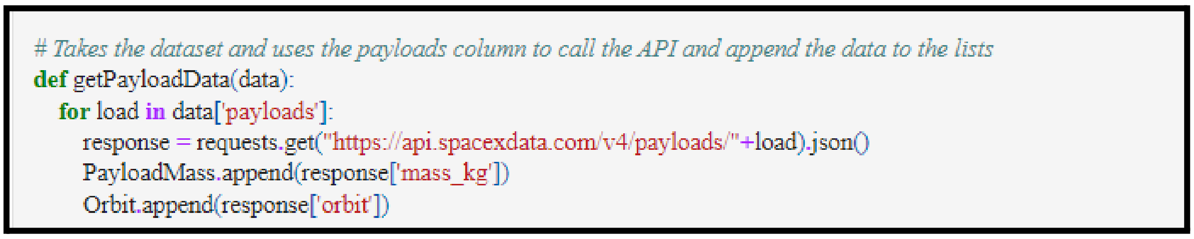
```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

The launchpad



```
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```

The payload



```
# Takes the dataset and uses the payloads column to call the API and append the data to the lists
def getPayloadData(data):
    for load in data['payloads']:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

The study aims to record the landing results, type, number of flights, gridfin usage, reuse, legs, landing cushion, core number, reuse frequency, and center serial.

```python
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

SpaceX API with the following URL:

```python
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

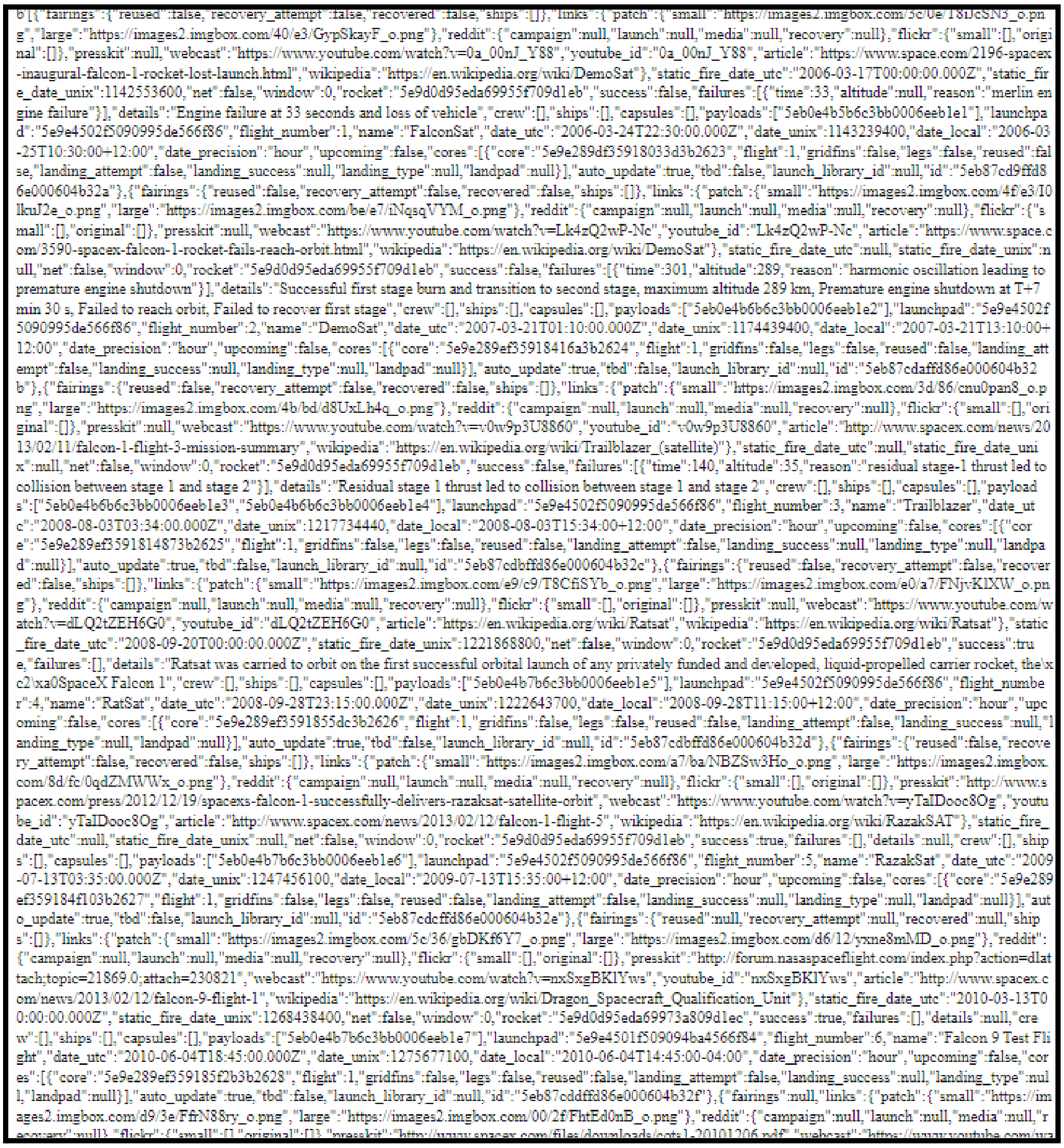
Response:

```python
response = requests.get(spacex_url)
```

Print:

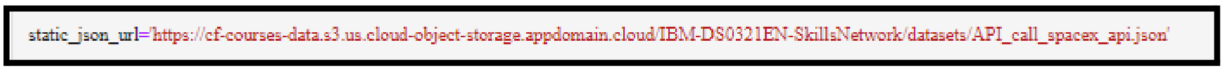
```python
print(response.content)
```

Output:



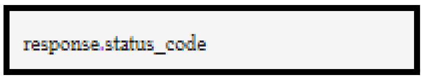
```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/3c/0e/T8iJcSN3_o.pn
g","large":"https://images2.imgbox.com/40/e3/GypSkayF_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"origi
nal":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.space.com/2196-spacex
-inaugural-falcon-1-rocket-lost-launch.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T00:00:00.000Z","static_fir
e_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":false,"failures":[{"time":33,"altitude":null,"reason":"merlin en
gine failure"}],"details":"Engine failure at 33 seconds and loss of vehicle","crew":[],"ships":[],"capsules":[],"payloads":["5eb0e4b5b6c3bb0006eeb1e1"],"launchpa
d":"5e9e4502f5090995de566f86","flight_number":1,"name":"FalconSat","date_utc":"2006-03-24T22:30:00.000Z","date_unix":1143239400,"date_local":"2006-03
-25T10:30:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"core":"5e9e289df35918033d3b2623","flight":1,"gridfins":false,"legs":false,"reused":fal
se,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cd9ffd8
6e000604b32a"},{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/4f/e3/I0
lkuJ2e_o.png","large":"https://images2.imgbox.com/be/e7/iNqsqVYM_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"s
mall":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=Lk4zQ2wP-Nc","youtube_id":"Lk4zQ2wP-Nc","article":"https://www.space.c
om/3590-spacex-falcon-1-rocket-fails-reach-orbit.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":null,"static_fire_date_unix":n
ull,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":false,"failures":[{"time":301,"altitude":289,"reason":"harmonic oscillation leading to
premature engine shutdown"}],"details":"Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7
min 30 s, Failed to reach orbit, Failed to recover first stage","crew":[],"ships":[],"capsules":[],"payloads":["5eb0e4b6b6c3bb0006eeb1e2"],"launchpad":"5e9e4502f
5090995de566f86","flight_number":2,"name":"DemoSat","date_utc":"2007-03-21T01:10:00.000Z","date_unix":1174439400,"date_local":"2007-03-21T13:10:00+
12:00","date_precision":"hour","upcoming":false,"cores":[{"core":"5e9e289ef35918416a3b2624","flight":1,"gridfins":false,"legs":false,"reused":false,"landing_att
empt":false,"landing_success":null,"landing_type":null,"landpad":null}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cdaffd86e000604b32
b"},{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/3d/86/cnu0pan8_o.p
ng","large":"https://images2.imgbox.com/4b/bd/d8UxLh4q_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"ori
ginal":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=v0w9p3U8860","youtube_id":"v0w9p3U8860","article":"http://www.spacex.com/news/20
13/02/11/falcon-1-flight-3-mission-summary","wikipedia":"https://en.wikipedia.org/wiki/Trailblazer_(satellite)"},"static_fire_date_utc":null,"static_fire_date_uni
x":null,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":false,"failures":[{"time":140,"altitude":35,"reason":"residual stage-1 thrust led to
collision between stage 1 and stage 2"}],"details":"Residual stage 1 thrust led to collision between stage 1 and stage 2","crew":[],"ships":[],"capsules":[],"payload
s":["5eb0e4b6b6c3bb0006eeb1e3","5eb0e4b6b6c3bb0006eeb1e4"],"launchpad":"5e9e4502f5090995de566f86","flight_number":3,"name":"Trailblazer","date_ut
c":"2008-08-03T03:34:00.000Z","date_unix":1217734440,"date_local":"2008-08-03T15:34:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"cor
e":"5e9e289ef3591814873b2625","flight":1,"gridfins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpa
d":null}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cdbffd86e000604b32c"},{"fairings":{"reused":false,"recovery_attempt":false,"recover
ed":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/e9/c9/T8CfiSYb_o.png","large":"https://images2.imgbox.com/e0/a7/FNjvKlXW_o.pn
g"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/w
atch?v=dLQ2tZEH6G0","youtube_id":"dLQ2tZEH6G0","article":"https://en.wikipedia.org/wiki/Ratsat","wikipedia":"https://en.wikipedia.org/wiki/Ratsat"},"static
_fire_date_utc":"2008-09-20T00:00:00.000Z","static_fire_date_unix":1221868800,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":tru
e,"failures":[],"details":"Ratsat was carried to orbit on the first successful orbital launch of any privately funded and developed, liquid-propelled carrier rocket, the\x
c2\xa0SpaceX Falcon 1","crew":[],"ships":[],"capsules":[],"payloads":["5eb0e4b7b6c3bb0006eeb1e5"],"launchpad":"5e9e4502f5090995de566f86","flight_numbe
r":4,"name":"RatSat","date_utc":"2008-09-28T23:15:00.000Z","date_unix":1222643700,"date_local":"2008-09-28T11:15:00+12:00","date_precision":"hour","upc
oming":false,"cores":[{"core":"5e9e289ef3591855dc3b2626","flight":1,"gridfins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"l
anding_type":null,"landpad":null}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cdbffd86e000604b32d"},{"fairings":{"reused":false,"recove
ry_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/a7/ba/NBZSw3Ho_o.png","large":"https://images2.imgbox.
com/8d/fc/0qdZMWWx_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":"http://www.s
pacex.com/press/2012/12/19/spacexs-falcon-1-successfully-delivers-razaksat-satellite-orbit","webcast":"https://www.youtube.com/watch?v=yTaIDooc8Og","youtu
be_id":"yTaIDooc8Og","article":"http://www.spacex.com/news/2013/02/12/falcon-1-flight-5","wikipedia":"https://en.wikipedia.org/wiki/RazakSAT"},"static_fire_
date_utc":null,"static_fire_date_unix":null,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":true,"failures":[],"details":null,"crew":[],"ship
s":[],"capsules":[],"payloads":["5eb0e4b7b6c3bb0006eeb1e6"],"launchpad":"5e9e4502f5090995de566f86","flight_number":5,"name":"RazakSat","date_utc":"2009
-07-13T03:35:00.000Z","date_unix":1247456100,"date_local":"2009-07-13T15:35:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"core":"5e9e289
ef359184f103b2627","flight":1,"gridfins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null}],"aut
o_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cdcffd86e000604b32e"},{"fairings":{"reused":null,"recovery_attempt":null,"recovered":null,"ship
s":[]},"links":{"patch":{"small":"https://images2.imgbox.com/5c/36/gbDKf6Y7_o.png","large":"https://images2.imgbox.com/d6/12/yxne8mMD_o.png"},"reddit":
{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":"http://forum.nasaspaceflight.com/index.php?action=dlat
tach;topic=21869.0;attach=230821","webcast":"https://www.youtube.com/watch?v=nxSxgBKlYws","youtube_id":"nxSxgBKlYws","article":"http://www.spacex.c
om/news/2013/02/12/falcon-9-flight-1","wikipedia":"https://en.wikipedia.org/wiki/Dragon_Spacecraft_Qualification_Unit"},"static_fire_date_utc":"2010-03-13T0
0:00:00.000Z","static_fire_date_unix":1268438400,"net":false,"window":0,"rocket":"5e9d0d95eda69973a809d1ec","success":true,"failures":[],"details":null,"cre
w":[],"ships":[],"capsules":[],"payloads":["5eb0e4b7b6c3bb0006eeb1e7"],"launchpad":"5e9e4501f509094ba4566f84","flight_number":6,"name":"Falcon 9 Test Fli
ght","date_utc":"2010-06-04T18:45:00.000Z","date_unix":1275677100,"date_local":"2010-06-04T14:45:00-04:00","date_precision":"hour","upcoming":false,"cor
es":[{"core":"5e9e289ef359185f2b3b2628","flight":1,"gridfins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":nul
l,"landpad":null}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cddffd86e000604b32f"},{"fairings":null,"links":{"patch":{"small":"https://im
ages2.imgbox.com/d9/3e/FfrN88ry_o.png","large":"https://images2.imgbox.com/00/2f/FhtEd0nB_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"r
ecovery":null},"flickr":{"small":[],"original":[]},"presskit":"http://www.spacex.com/files/downloads/cots1-20101206.pdf","webcast":"https://www.youtube.com/wa
```

Task 1: Request and parse the SpaceX launch data using the GET request



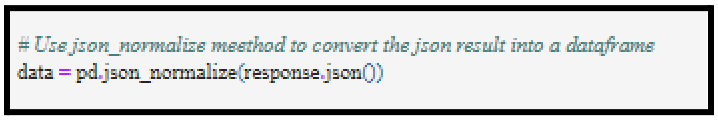
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

The request was successfully processed with the 200 status response code:



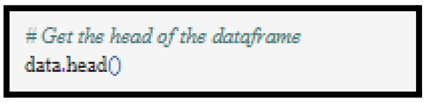
```
response.status_code
```

The response content is decoded as a JSON using.json() and converted into a Pandas dataframe using.json_normalize()

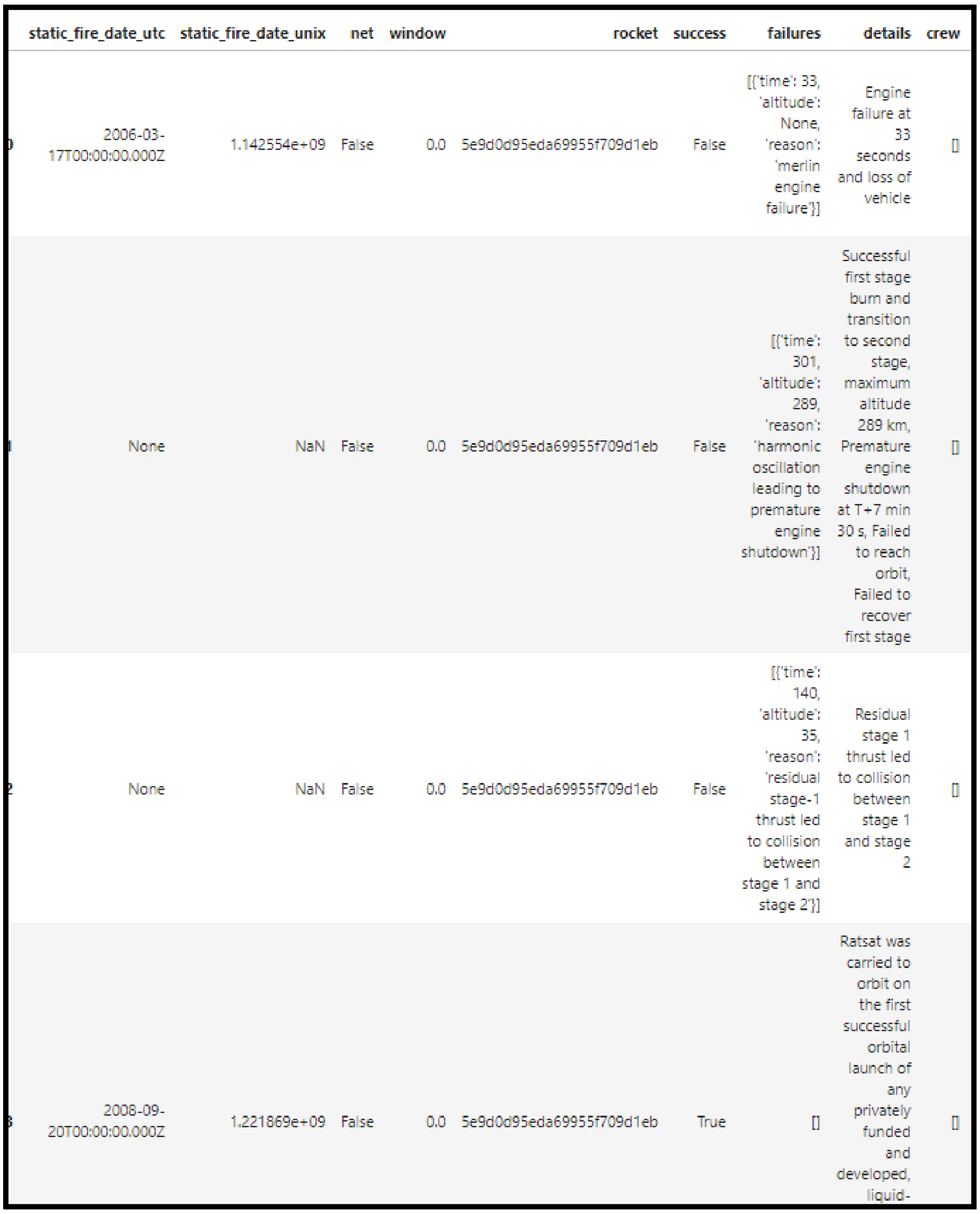


```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Using the dataframe

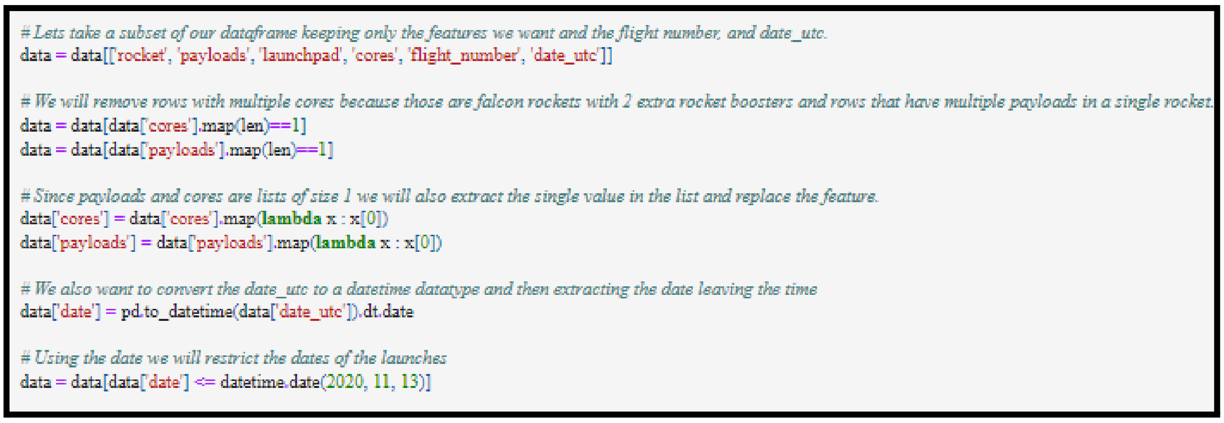


```
# Get the head of the dataframe
data.head()
```



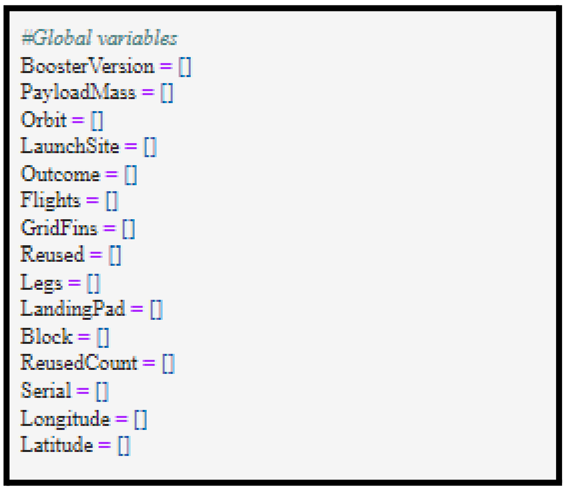
	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]
1	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 301, 'altitude': 289, 'reason': 'harmonic oscillation leading to premature engine shutdown'}]	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first stage	[]
2	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 140, 'altitude': 35, 'reason': 'residual stage-1 thrust led to collision between stage 1 and stage 2'}]	Residual stage 1 thrust led to collision between stage 1 and stage 2	[]
3	2008-09-20T00:00:00.000Z	1.221869e+09	False	0.0	5e9d0d95eda69955f709d1eb	True	[]	Ratsat was carried to orbit on the first successful orbital launch of any privately funded and developed, liquid-	[]

The API will be utilized to retrieve information about launches using the IDs provided, specifically focusing on rocket, payloads, launchpad, and cores.



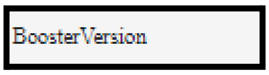
```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

The data collected from these requests will be stored in lists and utilized to create a new dataframe.



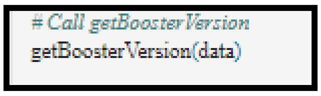
```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

The functions apply global outputs to variables, such as the BoosterVersion variable, which is empty before applying the getBoosterVersion function.



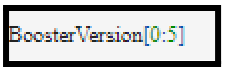
```
BoosterVersion
```

GetBoosterVersion



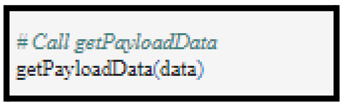
```
# Call getBoosterVersion
getBoosterVersion(data)
```

update



```
BoosterVersion[0:5]
```

['Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 9']



```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getCoreData
getCoreData(data)
```

The dataset will be constructed by combining the obtained data into a dictionary.

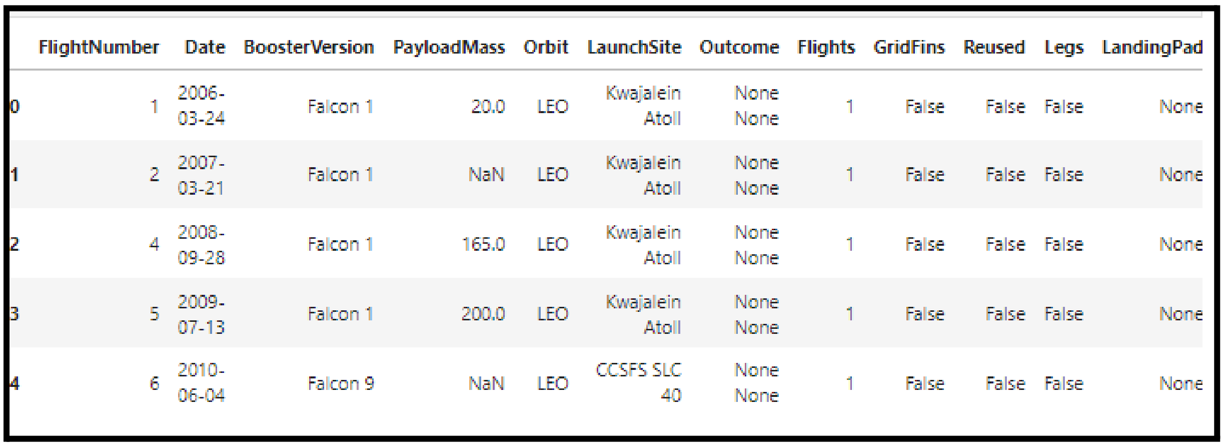
```
launch_dict = {
'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Next, we need to create a Pandas data frame from the dictionary launch_dict.

```
# Create a data from launch_dict
df = pd.DataFrame.from_dict(launch_dict)
```

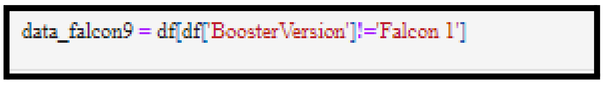
Show the summary of the dataframe

```
# Show the head of the dataframe
df.head()
```

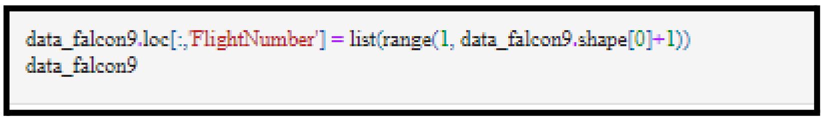


	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None

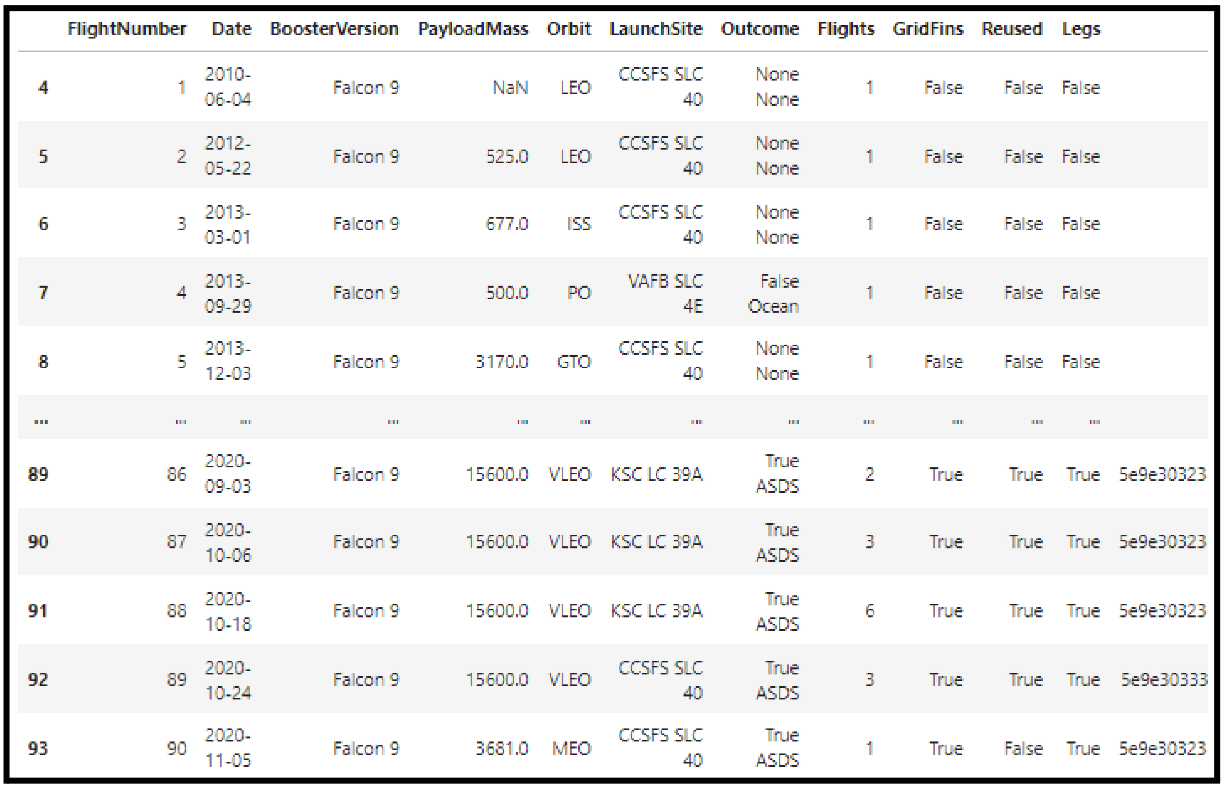
Task 2: Filter the dataframe to only include Falcon 9 launches



```
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

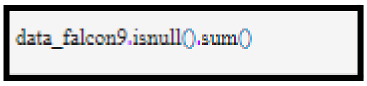


```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

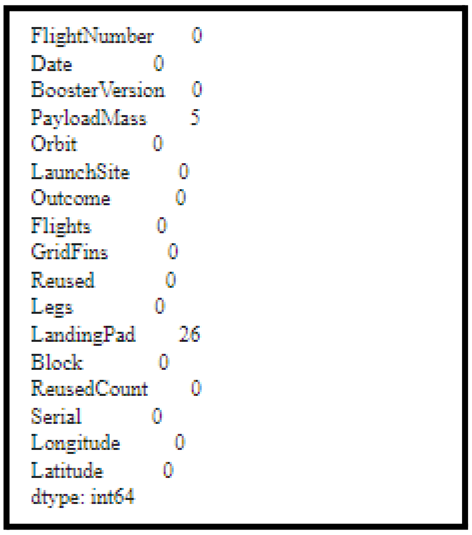


	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e30323
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e30323
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e30323
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e30333
93	90	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e30323

Data Wrangling

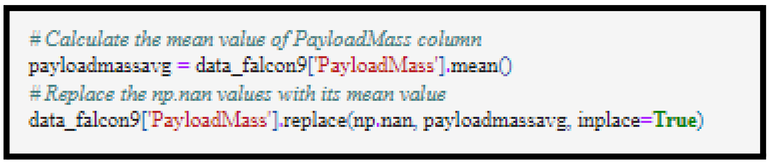


```
data_falcon9.isnull().sum()
```



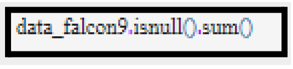
```
FlightNumber      0
Date              0
BoosterVersion    0
PayloadMass       5
Orbit             0
LaunchSite        0
Outcome           0
Flights           0
GridFins          0
Reused            0
Legs              0
LandingPad       26
Block             0
ReusedCount       0
Serial            0
Longitude         0
Latitude          0
dtype: int64
```

Task 3: Dealing with Missing Values

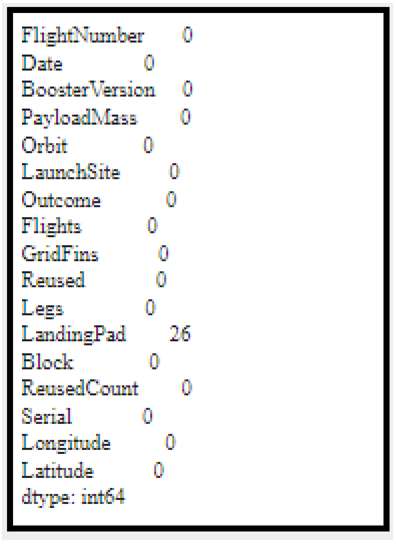


```python
# Calculate the mean value of PayloadMass column
payloadmassavg = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, payloadmassavg, inplace=True)
```

```python
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```



```python
data_falcon9.isnull().sum()
```



```
FlightNumber      0
Date              0
BoosterVersion    0
PayloadMass       0
Orbit             0
LaunchSite        0
Outcome           0
Flights           0
GridFins          0
Reused            0
Legs              0
LandingPad       26
Block             0
ReusedCount       0
Serial            0
Longitude         0
Latitude          0
dtype: int64
```