cognitiveclass.ai logo

Space X Falcon 9 First Stage Landing Prediction

Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia

Estimated time needed: **40** minutes

In this lab, you will be performing web scraping to collect Falcon 9 historical launch records from a

Wikipedia page titled `List of Falcon 9 and Falcon Heavy launches`

https://en.wikipedia.org/wiki/List_of_Fal

Falcon 9 first stage will land successfully

Several examples of an unsuccessful landing are shown here:

More specifically, the launch records are stored in a HTML table shown below:

Objectives

Web scrap Falcon 9 launch records with `BeautifulSoup`:

- Extract a Falcon 9 launch records HTML table from Wikipedia

- Parse the table and convert it into a Pandas data frame

First let's import required packages for this lab

In [1]:

```
!pip3 install beautifulsoup4
!pip3 install requests
```

```
Requirement already satisfied: be
autifulsoup4 in /home/jupyterlab/
conda/envs/python/lib/python3.7/s
ite-packages (4.11.1)
Requirement already satisfied: so
upsieve>1.2 in /home/jupyterlab/c
onda/envs/python/lib/python3.7/si
te-packages (from beautifulsoup4)
(2.3.2.post1)
Requirement already satisfied: re
quests in /home/jupyterlab/conda/
envs/python/lib/python3.7/site-pa
ckages (2.27.1)
Requirement already satisfied: ce
rtifi>=2017.4.17 in /home/jupyter
lab/conda/envs/python/lib/python
3.7/site-packages (from requests)
(2022.5.18.1)
Requirement already satisfied: ur
llib3<1.27,>=1.21.1 in /home/jupy
terlab/conda/envs/python/lib/pyth
on3.7/site-packages (from request
s) (1.26.9)
Requirement already satisfied: id
na<4,>=2.5 in /home/jupyterlab/co
nda/envs/python/lib/python3.7/sit
e-packages (from requests) (3.3)
Requirement already satisfied: ch
arset-normalizer~=2.0.0 in /home/
jupyterlab/conda/envs/python/lib/
```

```
python3.7/site-packages (from req
uests) (2.0.12)
```

In [16]:

```
import sys

import requests
from bs4 import BeautifulSoup
import re
import unicodedata
import pandas as pd
```

and we will provide some helper functions for you to process web scraped HTML table

In [17]:

```
def date_time(table_cells):
    """
    This function returns the data
    Input: the  element of a table
    """
    return [data_time.strip() for

def booster_version(table_cells):
    """
    This function returns the boos
    Input: the  element of a table
    """
    out=''.join([booster_version f
    return out

def landing_status(table_cells):
    """
    This function returns the land
    Input: the  element of a table
    """
    out=[i for i in table_cells.st
    return out


def get_mass(table_cells):
    mass=unicodedata.normalize("NF
    if mass:
        mass.find("kg")
        new_mass=mass[0:mass.find(
    else:
        new_mass=0
    return new_mass


def extract_column_from_header(row
    """
    This function returns the land
    Input: the  element of a table
    """
    if (row.br):
        row.br.extract()
    if row.a:
        row.a.extract()
    if row.sup:
        row.sup.extract()

    colunm_name = ' '.join(row.cor

    # Filter the digit and empty r
    if not(colunm_name.strip().isc
        colunm_name = colunm_name.
        return colunm_name
```

To keep the lab tasks consistent, you will be asked to scrape the data from a snapshot of the `List of Falcon 9 and Falcon Heavy launches` Wikipage updated on `9th June 2021`

In [18]:
```
static_url = "https://en.wikipedia
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the

Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [23]:

```
# use requests.get() method with t
data  = requests.get(static_url).t
print(data)
```

```
<!DOCTYPE html>
<html class="client-nojs" lang="e
n" dir="ltr">
<head>
<meta charset="UTF-8"/>
<title>List of Falcon 9 and Falco
n Heavy launches - Wikipedia</tit
le>
<script>document.documentElement.
className="client-js";RLCONF={"wg
BreakFrames":false,"wgSeparatorTr
ansformTable":["",""],"wgDigitTra
nsformTable":["",""],"wgDefaultDa
teFormat":"dmy","wgMonthNames":
["","January","February","Marc
h","April","May","June","July","A
ugust","September","October","Nov
ember","December"],"wgRequestI
d":"c1da8356-b523-42fa-a53f-d34f2
65164aa","wgCSPNonce":false,"wgCa
nonicalNamespace":"","wgCanonical
SpecialPageName":false,"wgNamespa
ceNumber":0,"wgPageName":"List_of
_Falcon_9_and_Falcon_Heavy_launch
es","wgTitle":"List of Falcon 9 a
nd Falcon Heavy launches","wgCurR
evisionId":1089567688,"wgRevision
Id":1027686922,"wgArticleId":3757
4004,"wgIsArticle":true,"wgIsRedi
rect":false,"wgAction":"view","wg
```

```
UserName":null,"wgUserGroups":
["*"],"wgCategories":["Source att
ribution","All articles with dead
external links","Articles with de
ad external links from February 2
021","Articles with permanently d
ead external links","CS1 Spanish-
language sources (es)","CS1 Indon
esian-language sources (id)"
,"CS1 errors: external links","CS
1 maint: url-status","CS1 German-
language sources (de)","CS1 Korea
n-language sources (ko)","Article
s with short description","Short
description is different from Wik
idata","Use American English from
January 2021","All Wikipedia arti
cles written in American Englis
h","Use dmy dates from January 20
21","Articles containing potentia
lly dated statements from March 2
018","All articles containing pot
entially dated statements","All a
rticles with failed verificatio
n","Articles with failed verifica
tion from May 2021","Articles con
taining potentially dated stateme
nts from April 2020","Pages using
multiple image with auto scaled i
mages","Featured lists","Articles
containing video clips","Falcon 9
and Falcon Heavy launches","Space
X launch vehicles","Lists of rock
et launches"],"wgPageContentLangu
age":"en","wgPageContentModel":"w
ikitext","wgRelevantPageName":"Li
st_of_Falcon_9_and_Falcon_Heavy_l
aunches","wgRelevantArticleId":37
574004,"wgIsProbablyEditable":tru
e,
"wgRelevantPageIsProbablyEditabl
e":true,"wgRestrictionEdit":[],"w
gRestrictionMove":[],"wgFlaggedRe
vsParams":{"tags":{"status":{"lev
els":1}}},"wgVisualEditor":{"page
LanguageCode":"en","pageLanguageD
ir":"ltr","pageVariantFallback
s":"en"},"wgMFDisplayWikibaseDesc
riptions":{"search":true,"nearb
y":true,"watchlist":true,"taglin
e":false},"wgWMESchemaEditAttempt
StepOversample":false,"wgWMEPageL
ength":500000,"wgNoticeProjec
t":"wikipedia","wgMediaViewerOnCl
ick":true,"wgMediaViewerEnabledBy
Default":true,"wgPopupsFlags":1
0,"wgULSCurrentAutonym":"Englis
h","wgEditSubmitButtonLabelPublis
h":true,"wgCentralAuthMobileDomai
n":false,"wgULSPosition":"interla
```

```
lign: middle; text-align: cente
r;">Success
</td>
<td class="table-noAttempt" style
="background: #EEE; vertical-alig
n: middle; white-space: nowrap; t
ext-align: center;">No attempt
</td></tr>
<tr>
<td colspan="9">Last launch of th
e original Falcon 9 v1.0 <a href
="/wiki/Launch_vehicle" title="La
unch vehicle">launch vehicle</a>,
first use of the unpressurized tr
unk section of Dragon.<sup class
="reference" id="cite_ref-sxf9_20
110321_35-0"><a href="#cite_note-
sxf9_20110321-35">[29]</a></sup>
</td></tr>
<tr>
<th rowspan="2" scope="row" style
="text-align:center;">6
</th>
<td>29 September 2013,<br/>16:00<
sup class="reference" id="cite_re
f-pa20130930_36-0"><a href="#cite
_note-pa20130930-36">[30]</a></su
p>
</td>
<td><a href="/wiki/Falcon_9_v1.1"
```

```
title="Falcon 9 v1.1">F9 v1.1</a>
<sup class="reference" id="cite_r
ef-MuskMay2012_13-5"><a href="#ci
te_note-MuskMay2012-13">[7]</a></
sup><br/>B1003<sup class="referen
ce" id="cite_ref-block_numbers_14
-5"><a href="#cite_note-block_num
bers-14">[8]</a></sup>
</td>
<td><a class="mw-redirect" href
="/wiki/Vandenberg_Air_Force_Bas
e" title="Vandenberg Air Force Ba
se">VAFB</a>,<br/><a href="/wiki/
Vandenberg_Space_Launch_Complex_
4" title="Vandenberg Space Launch
Complex 4">SLC-4E</a>
</td>
<td><a href="/wiki/CASSIOPE" titl
e="CASSIOPE">CASSIOPE</a><sup cla
ss="reference" id="cite_ref-sxMan
ifest20120925_28-2"><a href="#cit
e_note-sxManifest20120925-28">[2
2]</a></sup><sup class="referenc
e" id="cite_ref-CASSIOPE_MDA_37-
0"><a href="#cite_note-CASSIOPE_M
DA-37">[31]</a></sup>
</td>
<td>500 kg (1,100 lb)
</td>
<td><a href="/wiki/Polar_orbit" t
```

```
itle="Polar orbit">Polar orbit</a
> <a href="/wiki/Low_Earth_orbit"
title="Low Earth orbit">LEO</a>
</td>
<td><a href="/wiki/Maxar_Technolo
gies" title="Maxar Technologies">
MDA</a>
</td>
<td class="table-success" style
="background: #9EFF9E; vertical-a
lign: middle; text-align: cente
r;">Success<sup class="reference"
id="cite_ref-pa20130930_36-1"><a
href="#cite_note-pa20130930-36">
[30]</a></sup>
</td>
<td class="table-no2" style="back
ground: #FFE3E3; color: black; ve
rtical-align: middle; text-align:
center;">Uncontrolled<br/><small>
(ocean)</small><sup class="refere
nce" id="cite_ref-ocean_landing_3
8-0"><a href="#cite_note-ocean_la
nding-38">[d]</a></sup>
</td></tr>
<tr>
<td colspan="9">First commercial
mission with a private customer,
first launch from Vandenberg, and
demonstration flight of Falcon 9
```

```
v1.1 with an improved 13-tonne to
LEO capacity.<sup class="referenc
e" id="cite_ref-sxf9_20110321_35-
1"><a href="#cite_note-sxf9_20110
321-35">[29]</a></sup> After sepa
ration from the second stage carr
ying Canadian commercial and scie
ntific satellites, the first stag
e booster performed a controlled
reentry,<sup class="reference" id
="cite_ref-39"><a href="#cite_not
e-39">[32]</a></sup> and an <a hr
ef="/wiki/Falcon_9_first-stage_la
nding_tests" title="Falcon 9 firs
t-stage landing tests">ocean touc
hdown test</a> for the first tim
e. This provided good test data,
even though the booster started r
olling as it neared the ocean, le
ading to the shutdown of the cent
ral engine as the roll depleted i
t of fuel, resulting in a hard im
pact with the ocean.<sup class="r
eference" id="cite_ref-pa20130930
_36-2"><a href="#cite_note-pa2013
0930-36">[30]</a></sup> This was
the first known attempt of a rock
et engine being lit to perform a
supersonic retro propulsion, and
allowed SpaceX to enter a public-
```

```
private partnership with <a href
="/wiki/NASA" title="NASA">NASA</
a> and its Mars entry, descent, a
nd landing technologies research
projects.<sup class="reference" i
d="cite_ref-40"><a href="#cite_no
te-40">[33]</a></sup> <small>(<a
href="#Maiden_flight_of_v1.1">mor
e details below</a>)</small>
</td></tr>
<tr>
<th rowspan="2" scope="row" style
="text-align:center;">7
</th>
<td>3 December 2013,<br/>22:41<su
p class="reference" id="cite_ref-
sfn_wwls20130624_41-0"><a href="#
cite_note-sfn_wwls20130624-41">[3
4]</a></sup>
</td>
<td><a href="/wiki/Falcon_9_v1.1"
title="Falcon 9 v1.1">F9 v1.1</a>
<br/>B1004
</td>
<td><a href="/wiki/Cape_Canaveral
_Space_Force_Station" title="Cape
Canaveral Space Force Station">CC
AFS</a>,<br/><a href="/wiki/Cape_
Canaveral_Space_Launch_Complex_4
0" title="Cape Canaveral Space La
```

```
unch Complex 40">SLC-40</a>
</td>
<td><a href="/wiki/SES-8" title
="SES-8">SES-8</a><sup class="ref
erence" id="cite_ref-sxManifest20
120925_28-3"><a href="#cite_note-
sxManifest20120925-28">[22]</a></
sup><sup class="reference" id="ci
te_ref-spx-pr_42-0"><a href="#cit
e_note-spx-pr-42">[35]</a></sup><
sup class="reference" id="cite_re
f-aw20110323_43-0"><a href="#cite
_note-aw20110323-43">[36]</a></su
p>
</td>
<td>3,170 kg (6,990 lb)
</td>
<td><a href="/wiki/Geostationary_
transfer_orbit" title="Geostation
ary transfer orbit">GTO</a>
</td>
<td><a href="/wiki/SES_S.A." titl
e="SES S.A.">SES</a>
</td>
<td class="table-success" style
="background: #9EFF9E; vertical-a
lign: middle; text-align: cente
r;">Success<sup class="reference"
id="cite_ref-SNMissionStatus7_44-
0"><a href="#cite_note-SNMissionS
```

```
tatus7-44">[37]</a></sup>
</td>
<td class="table-noAttempt" style
="background: #EEE; vertical-alig
n: middle; white-space: nowrap; t
ext-align: center;">No attempt<b
r/><sup class="reference" id="cit
e_ref-sf10120131203_45-0"><a href
="#cite_note-sf10120131203-45">[3
8]</a></sup>
</td></tr>
<tr>
<td colspan="9">First <a href="/w
iki/Geostationary_transfer_orbit"
title="Geostationary transfer orb
it">Geostationary transfer orbit
</a> (GTO) launch for Falcon 9,<s
up class="reference" id="cite_ref
-spx-pr_42-1"><a href="#cite_note
-spx-pr-42">[35]</a></sup> and fi
rst successful reignition of the
second stage.<sup class="referenc
e" id="cite_ref-46"><a href="#cit
e_note-46">[39]</a></sup> SES-8 w
as inserted into a <a href="/wik
i/Geostationary_transfer_orbit" t
itle="Geostationary transfer orbi
t">Super-Synchronous Transfer Orb
it</a> of 79,341 km (49,300 mi) i
n apogee with an <a href="/wiki/O
```

```
rbital_inclination" title="Orbita
l inclination">inclination</a> of
20.55° to the <a href="/wiki/Equa
tor" title="Equator">equator</a>.
</td></tr></tbody></table>
```

You should able to see the columns names embedded in the table header elements `<th>` as follows:

```
<tr>
<th
scope="col">Flight
No.
</th>
<th scope="col">Date
and<br/>time (<a
href="/wiki/Coordinated_Univ
 title="Coordinated
Universal
Time">UTC</a>)
</th>
<th scope="col"><a
href="/wiki/List_of_Falcon_9
stage_boosters"
```

```
title="List of Falcon
9 first-stage
boosters">Version,
<br/>Booster</a> <sup
class="reference"
id="cite_ref-
booster_11-0"><a
href="#cite_note-
booster-11">[b]</a>
</sup>
</th>
<th
scope="col">Launch
site
</th>
<th
scope="col">Payload<sup
 class="reference"
id="cite_ref-
Dragon_12-0"><a
href="#cite_note-
Dragon-12">[c]</a>
</sup>
</th>
<th
scope="col">Payload
```

```
mass
</th>
<th scope="col">Orbit
</th>
<th
scope="col">Customer
</th>
<th
scope="col">Launch<br/>outco

</th>
<th scope="col"><a
href="/wiki/Falcon_9_first-
stage_landing_tests"
title="Falcon 9
first-stage landing
tests">Booster<br/>landing</

</th></tr>
```

Next, we just need to iterate through the `<th>` elements and apply the provided `extract_column_from_header()`

to extract column name one by one

In [30]:

```
column_names = []

# Apply find_all() function with `
# Iterate each th element and appl
# Append the Non-empty column name
for row in first_launch_table.finc
    name = extract_column_from_hea
    if (name != None and len(name)
        column_names.append(name)
```

Check the extracted column names

In [31]:

```
print(column_names)
```

```
['Flight No.', 'Date and time (
)', 'Launch site', 'Payload', 'Pa
yload mass', 'Orbit', 'Customer',
'Launch outcome']
```

TASK 3: Create a data frame by

parsing the launch HTML tables

We will create an empty dictionary with keys from the extracted column names in the previous task. Later, this dictionary will be converted into a Pandas dataframe

In [32]:

```
launch_dict= dict.fromkeys(column_

# Remove an irrelvant column
del launch_dict['Date and time ( )

# Let's initial the launch_dict wi
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
```

```
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Next, we just need to fill up the `launch_dict` with launch records extracted from table rows.

Usually, HTML tables in Wiki pages are likely to contain unexpected annotations and other types of noises, such as reference links `B0004.1[8]`, missing values `N/A [e]`, inconsistent formatting, etc.

To simplify the parsing process, we have provided an incomplete code snippet below to help you to fill up the `launch_dict`. Please complete the following code snippet with TODOs or you can choose to write your own logic to parse all launch tables:

In [33]:

```
extracted_row = 0
#Extract each table
for table_number,table in enumerat
   # get table row
    for rows in table.find_all("tr
        #check to see if first tab
        if rows.th:
            if rows.th.string:
                flight_number=rows
                flag=flight_number
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cell
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the fli
            #print(flight_number)
            datatimelist=date_time

            # Date value
            # TODO: Append the dat
            date = datatimelist[0]
            #print(date)

            # Time value
            # TODO: Append the tim
            time = datatimelist[1]
            #print(time)

            # Booster version
            # TODO: Append the bv
            bv=booster_version(row
            if not(bv):
                bv=row[1].a.string
            print(bv)

            # Launch Site
            # TODO: Append the bv
            launch_site = row[2].a
            #print(launch_site)

            # Payload
            # TODO: Append the pay
            payload = row[3].a.str
            #print(payload)

            # Payload Mass
            # TODO: Append the pay
            payload_mass = get_mas
            #print(payload)

            # Orbit
            # TODO: Append the orb
            orbit = row[5].a.strin
            #print(orbit)

            # Customer
            # TODO: Append the cus
            customer = row[6].a.st
            #print(customer)

            # Launch outcome
            # TODO: Append the lau
            launch_outcome = list(
            #print(launch_outcome)

            # Booster landing
            # TODO: Append the lau
            booster_landing = land
            #print(booster_landing
F9 v1.0B0003.1
F9 v1.0B0004.1
F9 v1.0B0005.1
F9 v1.0B0006.1
F9 v1.0B0007.1
F9 v1.1B1003
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 FT
F9 v1.1
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
F9 FT
```

```
F9 FT
F9 FT♲
F9 FT
F9 FT
F9 FT
F9 FTB1029.2
F9 FT
F9 FT
F9 B4
F9 FT
F9 B4
F9 B4
F9 FTB1031.2
F9 B4
F9 FTB1035.2
F9 FTB1036.2
F9 B4
F9 FTB1032.2
F9 FTB1038.2
F9 B4
F9 B4B1041.2
F9 B4B1039.2
F9 B4
F9 B5B1046.1
F9 B4B1043.2
F9 B4B1040.2
F9 B4B1045.2
F9 B5
F9 B5B1048
F9 B5B1046.2
```

```
F9 B5
F9 B5B1048.2
F9 B5B1047.2
F9 B5B1046.3
F9 B5
F9 B5
F9 B5B1049.2
F9 B5B1048.3
F9 B5[268]
F9 B5
F9 B5B1049.3
F9 B5B1051.2
F9 B5B1056.2
F9 B5B1047.3
F9 B5
F9 B5
F9 B5B1056.3
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5B1058.2
F9 B5
```

```
F9 B5B1049.6
F9 B5
F9 B5B1060.2
F9 B5B1058.3
F9 B5B1051.6
F9 B5
F9 B5
F9 B5
F9 B5
F9 B5 ♺
F9 B5 ♺
F9 B5 ♺
F9 B5 ♺
F9 B5
F9 B5B1051.8
F9 B5B1058.5
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
/tmp/ipykernel_703/354354048.py in <module>
     60             # Customer
     61             # TODO: Append the customer into launch_dict with key `Customer`
---> 62             customer = row[6].a.string
     63             #print(customer)
     64 

AttributeError: 'NoneType' object has no attribute 'string'
```

After you have fill in the parsed launch record values into `launch_dict`, you can create a dataframe from it.

In [35]:
```
df=pd.DataFrame(launch_dict)
df.head()
```

Out[35]:

Flight No.	Launch site	Payload	Payload mass	Orl

We can now export it to a **CSV** for the next section, but to make the answers consistent and in case you have difficulties finishing this lab.

Following labs will be using a provided dataset to make each lab independent.

```
df.to_csv('spacex_web_scraped.c
 index=False)
```

Authors

Yan Luo

Nayef Abou Tayoun

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2021-06-09	1.0	Yan Luo	Tasks updates
2020-11-10	1.0	Nayef	Created the initial version