Selected code due to failure of Coursera .py saves for modules

# use requests.get() method with the provided static\_url

# assign the response to a object

r = requests.get(url = static\_url)

# Use BeautifulSoup() to create a BeautifulSoup object from a response text content

soup = BeautifulSoup(r.content, "html.parser")

# Use soup.title attribute

print(soup.title)

# Use the find\_all function in the BeautifulSoup object, with element type `table`

# Assign the result to a list called `html\_tables`

html\_tables = soup.find\_all("table")

# Let's print the third table and check its content

first\_launch\_table = html\_tables[2]

print(first\_launch\_table)

th = first\_launch\_table.find\_all('th')

for th in th:

c = extract\_column\_from\_header(th)

if c is not None and len(c) > 0:

column\_names.append(c)

else:

column\_names = column\_names

Week 2

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset\_part\_1.csv

r = df.value\_counts('LaunchSite')

print(r)

# Apply value\_counts() on column LaunchSite

r = df.value\_counts('LaunchSite')

print(r)

# Apply value\_counts on Orbit column

r = df.value\_counts('Orbit')

print(r)

# landing\_outcomes = values on Outcome column

landing\_outcomes = df.value\_counts('Outcome')

landing\_outcomes

for i,outcome in enumerate(landing\_outcomes.keys()):

print(i,outcome)

# landing\_class = 0 if bad\_outcome

# landing\_class = 1 otherwise

good\_outcomes=set(landing\_outcomes.keys()[[0,2,4]])

landing\_class = []

landing\_clas = df['Outcome'].isin(good\_outcomes)

landing\_class = landing\_clas.astype(int)

landing\_class

dcc.Dropdown(id='site-dropdown',

options=[

{'label': 'All Sites', 'value': 'ALL'},

{'label': 'CCAF LC', 'value': 'Site 1'},

{'label': 'CCAF SLC', 'value': 'Site 2'}

{'label': 'KSC', 'value': 'Site 3'}

{'label': 'VAFB', 'value': 'Site 4'}

],

**value**='ALL',

placeholder="Select a Launch Site Here",

searchable=True

),

SELECT DISTINCT "Launch\_Sites" FROM SPACEXTBL;

#cur.execute('''

#SELECT DISTINCT("Launch\_Sites") FROM SPACEXTBL

# ''')

#for row in cur.fetchall():

# print (row)

SELECT \* FROM SPACEXTBL WHERE ("Launch\_Sites" LIKE '%CCA%')

SELECT SUM(Payload\_Mass) as sum\_pmass

FROM SPACEXTBL;

SELECT SUM(Payload\_Mass) as sum\_pmass

FROM SPACEXTBL;

SELECT MIN("Date") FROM SPACEXTBLE;

SELECT "Booster\_Name" FROM SPACEXTBL AS B\_NAME

WHERE B\_NAME ("Mission\_Outcome" Like '%Success&' AND "Payload\_Mass" BETWEEN 4000 AND 6000);

SELECT SUM("Mission\_Outcome" Like '&success&')

SELECT SUM("Mission\_Outcome" Like '&failure&')

SELECT "Booster\_Version"

FROM SPACEXTBL

WHERE Payload\_Mass = (

SELECT MAX("Payload\_Mass")

FROM SPACEXTBL

);