Untitled

April 28, 2024

[2]:
```
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 high-level mathematical functions to operate on these arrays
import numpy as np
```

[3]:
```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.
 ↪cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")
df.head(10)
```

[3]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	\
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	
5	6	2014-01-06	Falcon 9	3325.000000	GTO	CCAFS SLC 40	
6	7	2014-04-18	Falcon 9	2296.000000	ISS	CCAFS SLC 40	
7	8	2014-07-14	Falcon 9	1316.000000	LEO	CCAFS SLC 40	
8	9	2014-08-05	Falcon 9	4535.000000	GTO	CCAFS SLC 40	
9	10	2014-09-07	Falcon 9	4428.000000	GTO	CCAFS SLC 40	

	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	\
0	None None	1	False	False	False	NaN	1.0	
1	None None	1	False	False	False	NaN	1.0	
2	None None	1	False	False	False	NaN	1.0	
3	False Ocean	1	False	False	False	NaN	1.0	
4	None None	1	False	False	False	NaN	1.0	
5	None None	1	False	False	False	NaN	1.0	
6	True Ocean	1	False	False	True	NaN	1.0	
7	True Ocean	1	False	False	True	NaN	1.0	
8	None None	1	False	False	False	NaN	1.0	
9	None None	1	False	False	False	NaN	1.0	

	ReusedCount	Serial	Longitude	Latitude
0	0	B0003	-80.577366	28.561857
1	0	B0005	-80.577366	28.561857

```
2            0  B0007  -80.577366  28.561857
3            0  B1003 -120.610829  34.632093
4            0  B1004  -80.577366  28.561857
5            0  B1005  -80.577366  28.561857
6            0  B1006  -80.577366  28.561857
7            0  B1007  -80.577366  28.561857
8            0  B1008  -80.577366  28.561857
9            0  B1011  -80.577366  28.561857
```

[4]:
```
df.dtypes
```

[4]:
```
FlightNumber        int64
Date               object
BoosterVersion     object
PayloadMass       float64
Orbit              object
LaunchSite         object
Outcome            object
Flights             int64
GridFins             bool
Reused               bool
Legs                 bool
LandingPad         object
Block             float64
ReusedCount         int64
Serial             object
Longitude         float64
Latitude          float64
dtype: object
```

[5]:
```
launch_counts=df['LaunchSite'].value_counts()
print(launch_counts)
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

[6]:
```
orbit=df['Orbit'].value_counts()
print(orbit)
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
```

```
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

[10]:
```
landing_outcomes=df['Outcome'].value_counts()
print(landing_outcomes)
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

[11]:
```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

[12]:
```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

[12]:
```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

[13]:
```
bad_outcome={'destroyed','uncontrolled','failure'}
```

[15]:
```
df['landing_class']=df['Outcome'].apply(lambda x: 0 if x in bad_outcome else 1)
```

[16]:
```
print(df[['Outcome','landing_class']].head())
```

```
       Outcome  landing_class
0    None None              1
1    None None              1
2    None None              1
3  False Ocean              1
4    None None              1
```

[ ]: