IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Evan Meikleham
05 December 2021

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- We want to enter the Commercial Space Sector, but there is a large barrier to entry
- If we can guess whether a rocket is reusable, we can better predict our costs
- We will be training a predictive model that takes in the data about a launch and predicts whether the Falcon 9 first stage rocket will or will not be able to land.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX has a publicly available API (api.spacexdata.com) where physical and engineering data about each launch can be retrieved. Additional data was retrieved by web-scraping from Wikipedia tables.
- Perform data wrangling
  - Python (NumPy and Pandas) were used to format data to be fed into statistical models. Missing data was removed or cleaned, and launches were classified as “success” or “failure”
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - SkiKit-Learn was used to train various classes of statistical classification models, and the best ones was selected.

Data Collection

- Physical data was collected by using SpaceX's public API https://api.spacexdata.com/v4/
  - Rockets/Boosters
  - Launch Sites
  - Payload Data
  - Core Data
- Python packages (Mostly NumPy and PANDAS) were used to drop irrelevant fields and place the data into usable tables
- Missing values were either dropped or else replaced with mean values (among Falcon 9 launches)

Data Collection – SpaceX API

- Data collection via SpaceX REST calls can be seen at

  https://github.com/evanakm/coursera-project/blob/master/Week%201-1.ipynb

Data Collection - Scraping

- Webscraping via BeautifulSoup can be seen at

  https://github.com/evanakm/coursera-project/blob/master/Week%201-2.ipynb

Get Falcon 9 Data from Wikipage (use requests Python library)

Use BeautifulSoup to extract tables from pages

Get table of first launches

Extract Flight number, site, payload, mass, orbit, customer, outcome, booster, landing, date and time

Organize into a dictionary

Export as a PANDAS DataFrame

Data Wrangling

- Making the decision about how to classify launches as successful or not
- Notebook available at https://github.com/evanakm/coursera-project/blob/master/Week%201-3.ipynb

Figure out the number of launches and success rate per launch site

Figure out the outcome and success rate per orbit type

Separate successes (grouped by landing in ocean, ground pad, drone ship) from unsuccessful landings/mission failures.

Encode success as 1 in the 'Class' variable

EDA with Data Visualization

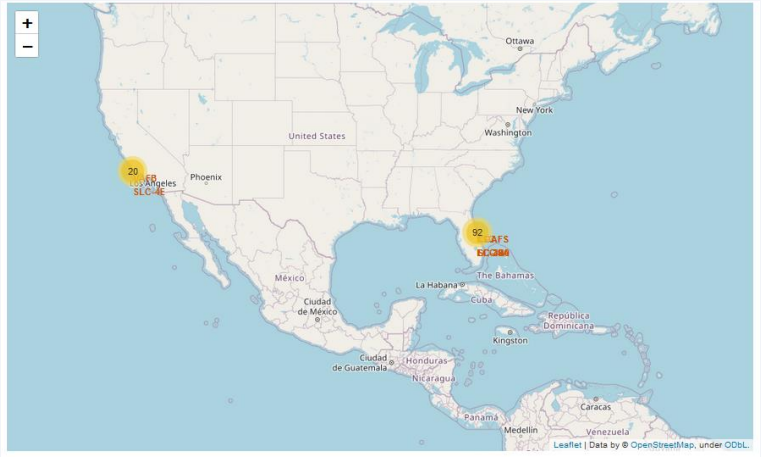
- Plots to analyze
- Payload Mass vs Flight Number (color coded by success/failure) (to see if heavier/lighter rockets had more success as SpaceX became more experienced)
- Launch Site vs Flight Number (see if some sites are more prone to success/failure over time)
- Payload vs Site (see if heavier loads are launched from some locations)
- Success Rates of mission/orbit types (and also see successes over time)
- Success Rate by year (see the rate of technological improvement)
- https://github.com/evanakm/coursera-project/blob/master/Week%202-2.ipynb

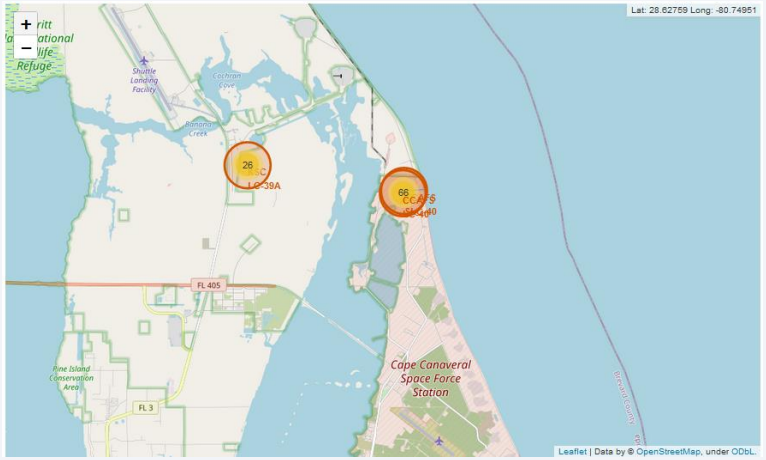
EDA with SQL

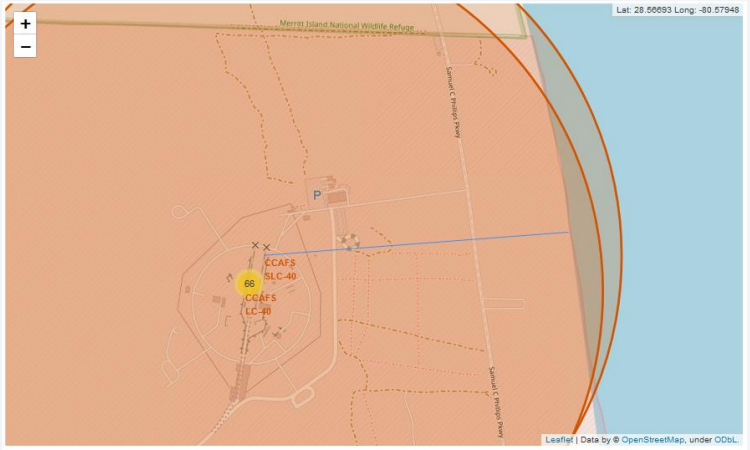
- SQL Queries
- Distinct Launch sites
- Cumulative and average payloads by site/booster version
- Date of first successful launch
- Boosters that have had successes
- Number and rate of success/failures
- https://github.com/evanakm/coursera-project/blob/master/Week%202-1.ipynb

Build an Interactive Map with Folium

- We used the Folium package to create masks with launch sites, with clusters of launch sites so that the maps were not too busy or confusing
- We wanted to see how close the launch sites are to the coast
- https://github.com/evanakm/coursera-project/blob/master/Week%202-3.ipynb

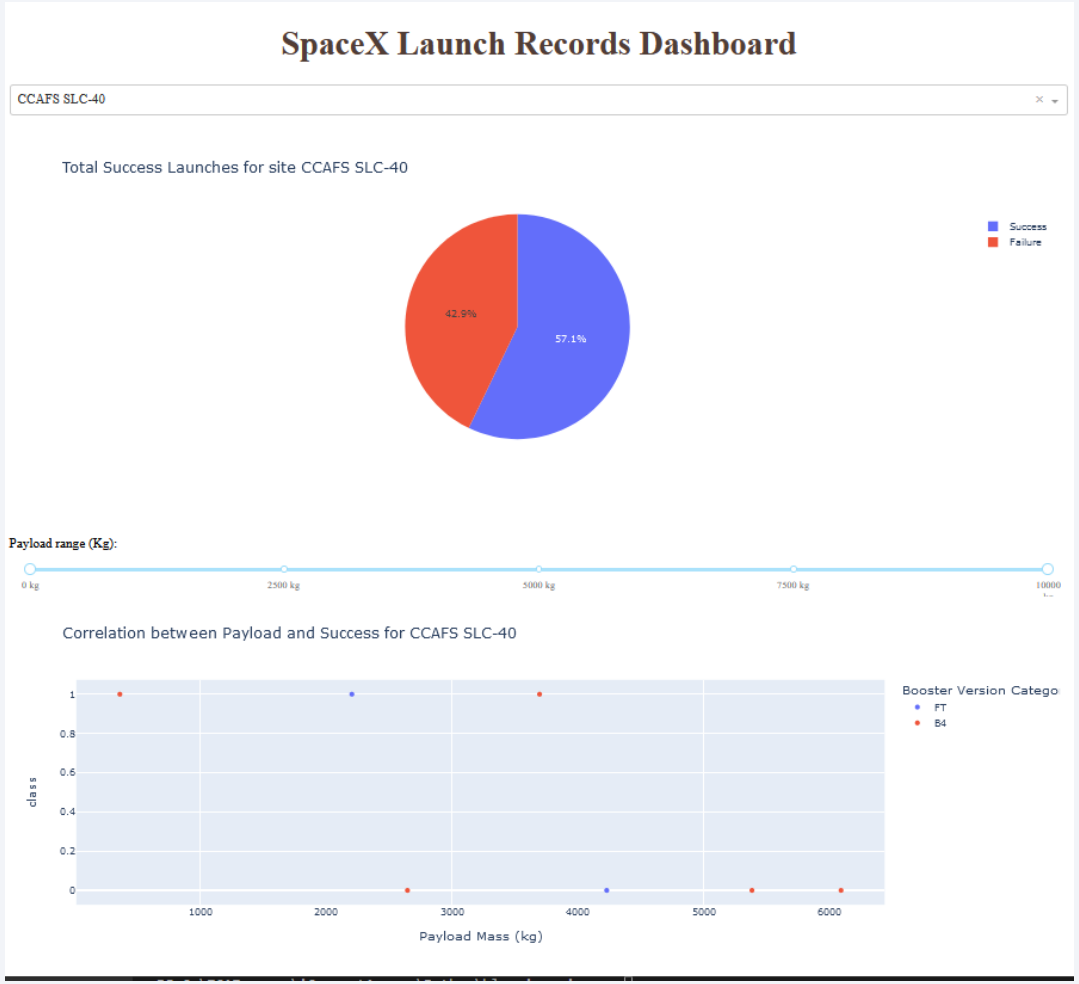






Build a Dashboard with Plotly Dash

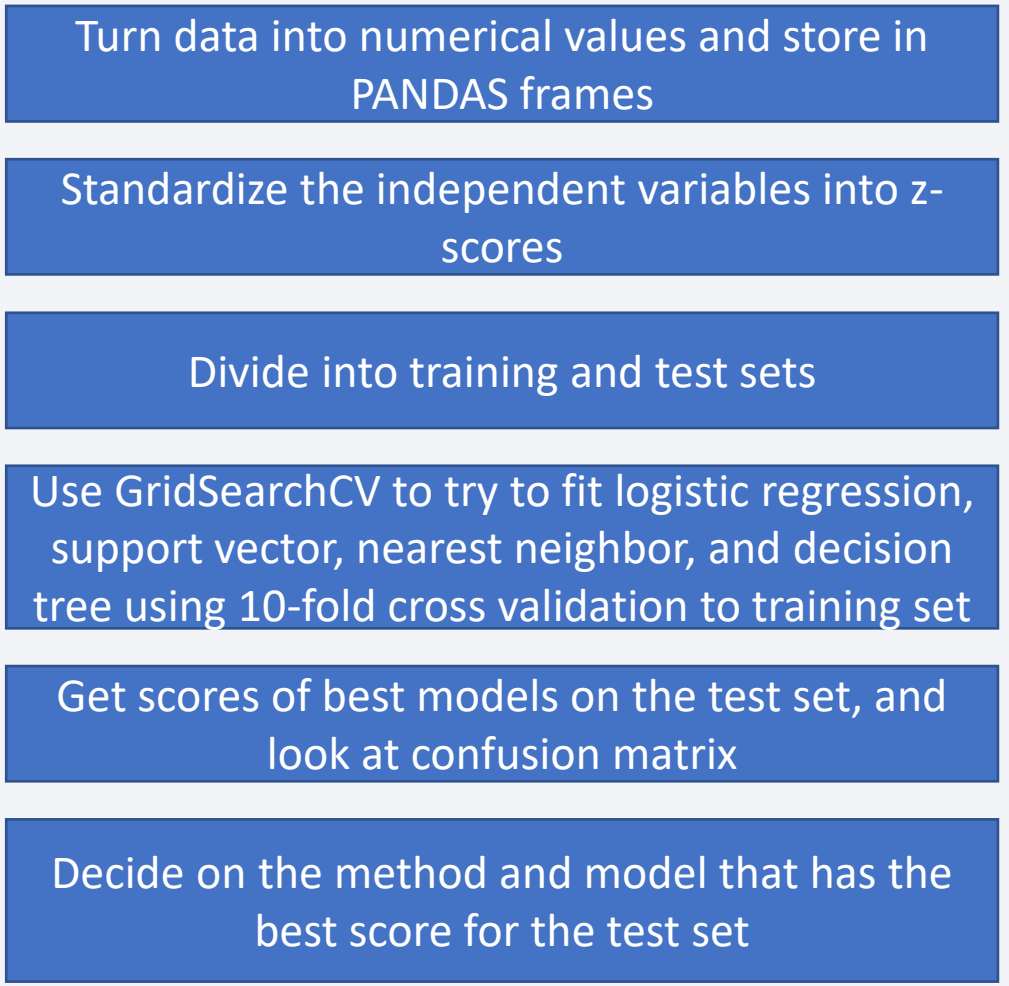
- We can plot success rates by launch site and be able to change a pie chart interactively
- Also plotted successes vs payload mass (and booster category) filtered interactively by Launch Site
- https://github.com/evanakm/coursera-project/blob/master/spacex_dash_app.py



Predictive Analysis (Classification)

- Turn all values into numerical and one-hot categorical variables
- There are two categories to predict, so simple separation models can be trained. (Logistic regression, support vector, nearest neighbor, and decision tree)

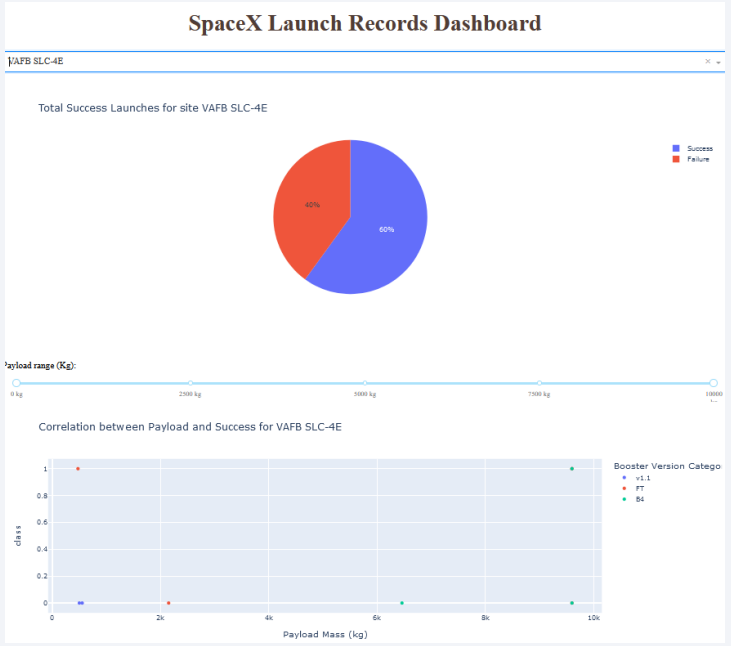
https://github.com/evanakm/coursera-project/blob/master/Week%204-1.ipynb

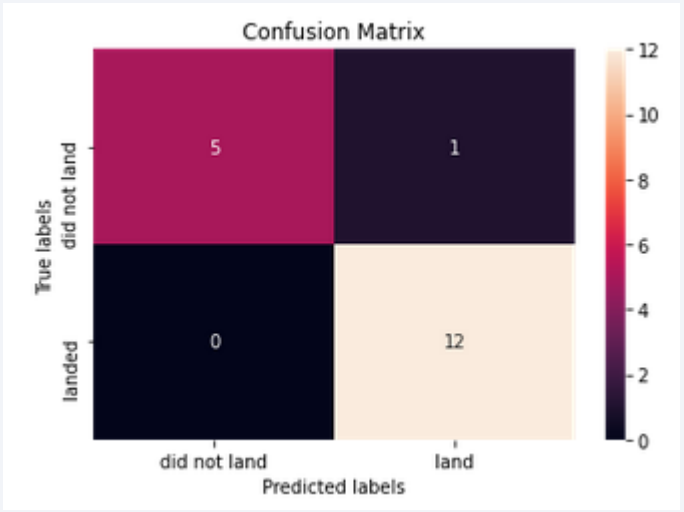


Results

- Exploratory data analysis results: an predictive dataframe was created (topmost image)
- Interactive analytics demo in lower left (different site than the previous slide)
- Predictive analysis. Confusion matrix was the same for all methods. (Lower right)

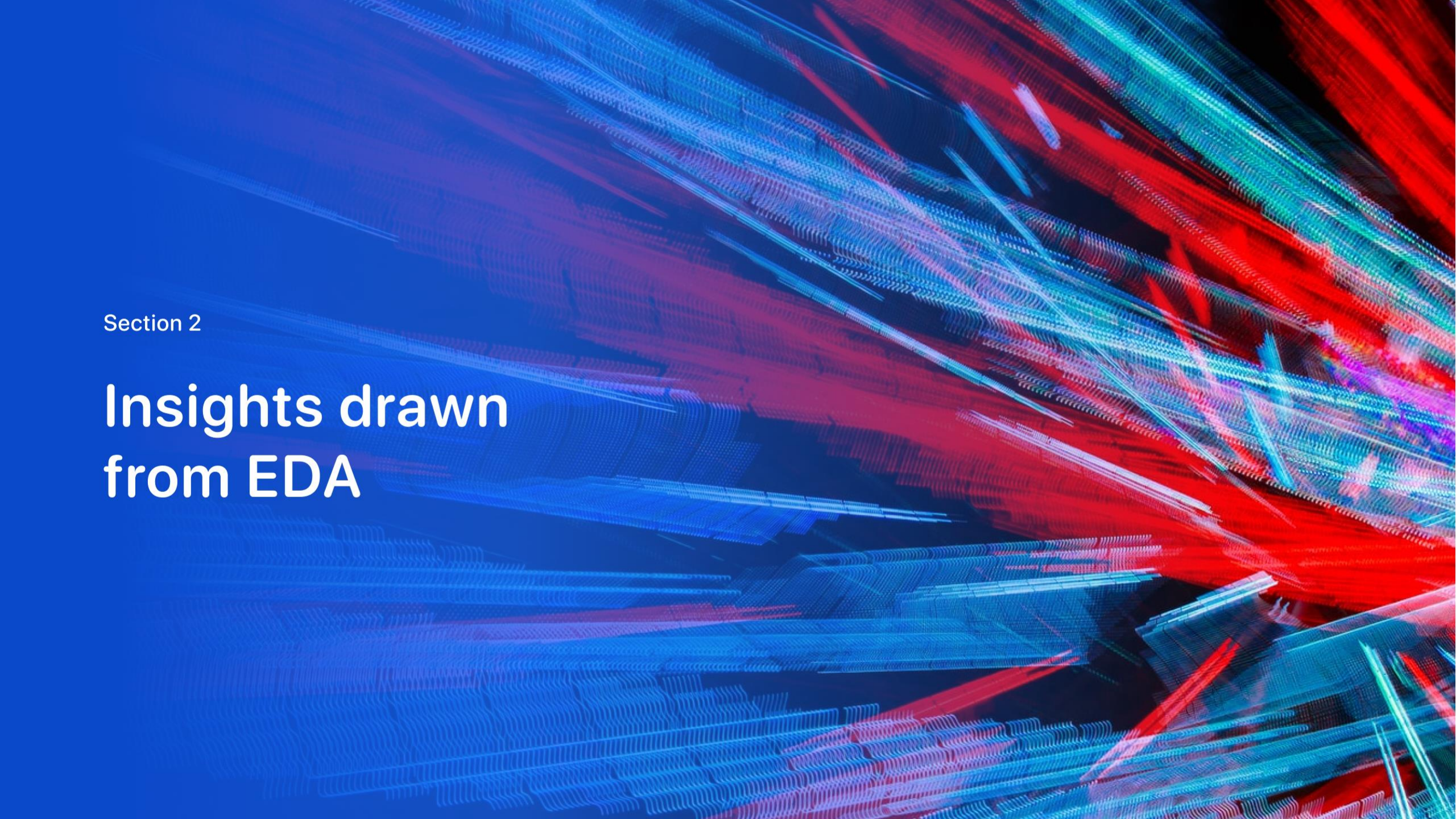
	FlightNumber	PayloadMass	Flights	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	Orbit_GTO	Orbit_HEO	Orbit_ISS	...	Serial_B1058	Serial_B1059	Serial_B1060	Serial_B1(
0	1.0	6104.959412	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
1	2.0	525.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
2	3.0	677.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	
3	4.0	500.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
4	5.0	3170.000000	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
85	86.0	15400.000000	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	
86	87.0	15400.000000	3.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	
87	88.0	15400.000000	6.0	5.0	5.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
88	89.0	15400.000000	3.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	
89	90.0	3681.000000	1.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	





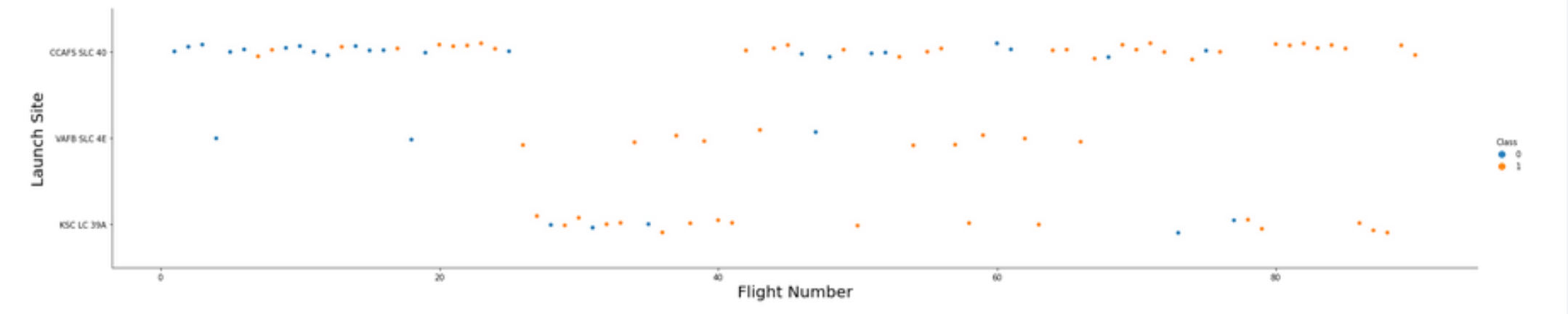
Section 2

Insights drawn from EDA



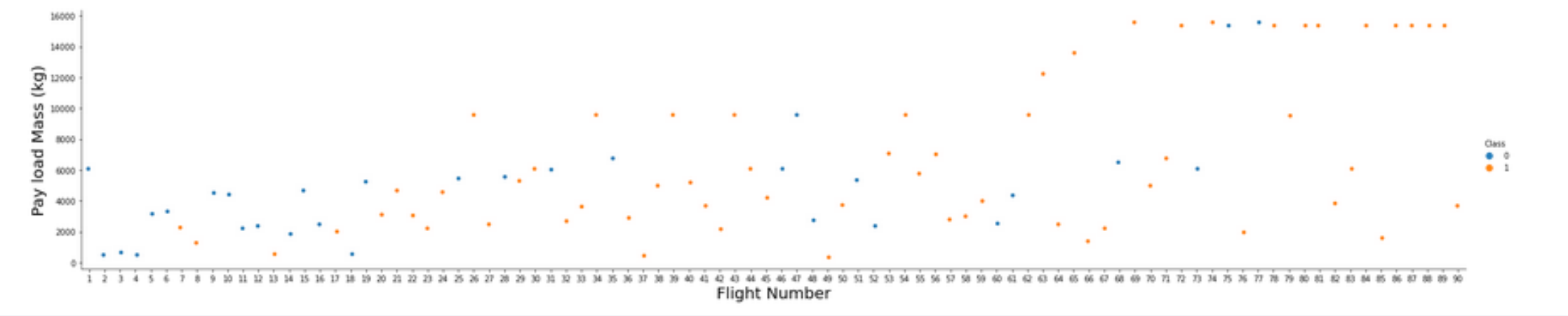
Flight Number vs. Launch Site

- As time goes on, more missions were launched from CCAFS SLC40, and the success rate clearly goes from mostly failing to mostly succeeding
- As time goes on, fewer launches are made from VAFB SLC 4E



Payload vs. Launch Site

- As time goes on, more missions were launched with higher payloads, up to a max of 16000 kg, and the success rate (of rocket reusability) is getting closer to 100%
- It is harder to visually see patterns below the maximum payload, but the success rate is clearly higher as time goes on

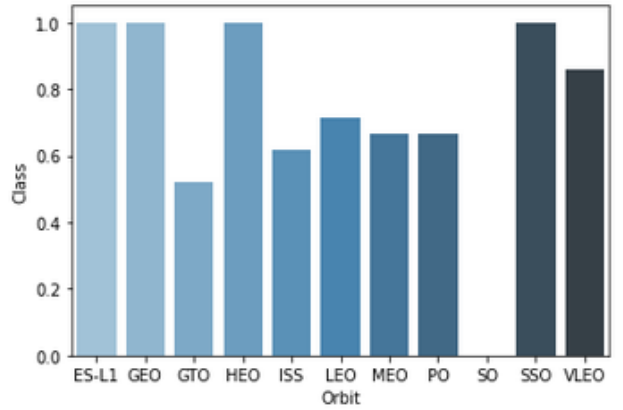


Success Rate vs. Orbit Type

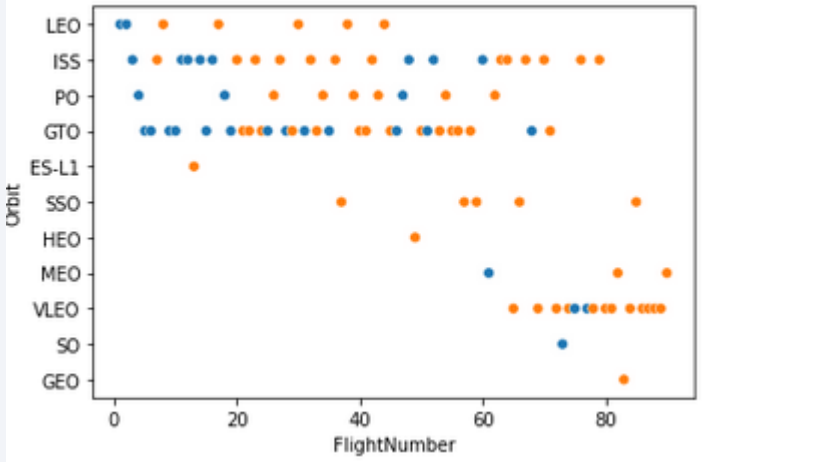
- There are 100% success rates for certain orbits (ES-L1, GEO, HEO, and SSO)
- GTO and ISS missions have a lower success rate
- We can see from the Orbit vs Flight Number plot that there were many more GTO/ISS missions, whereas some of the orbits with high success rates only had one or two missions.

```
# HINT use groupby method on Orbit column and
avg = df.groupby('Orbit').mean().reset_index()
sns.barplot(x="Orbit", y="Class", palette="Blu
```

```
<AxesSubplot:xlabel='Orbit', ylabel='Class'>
```

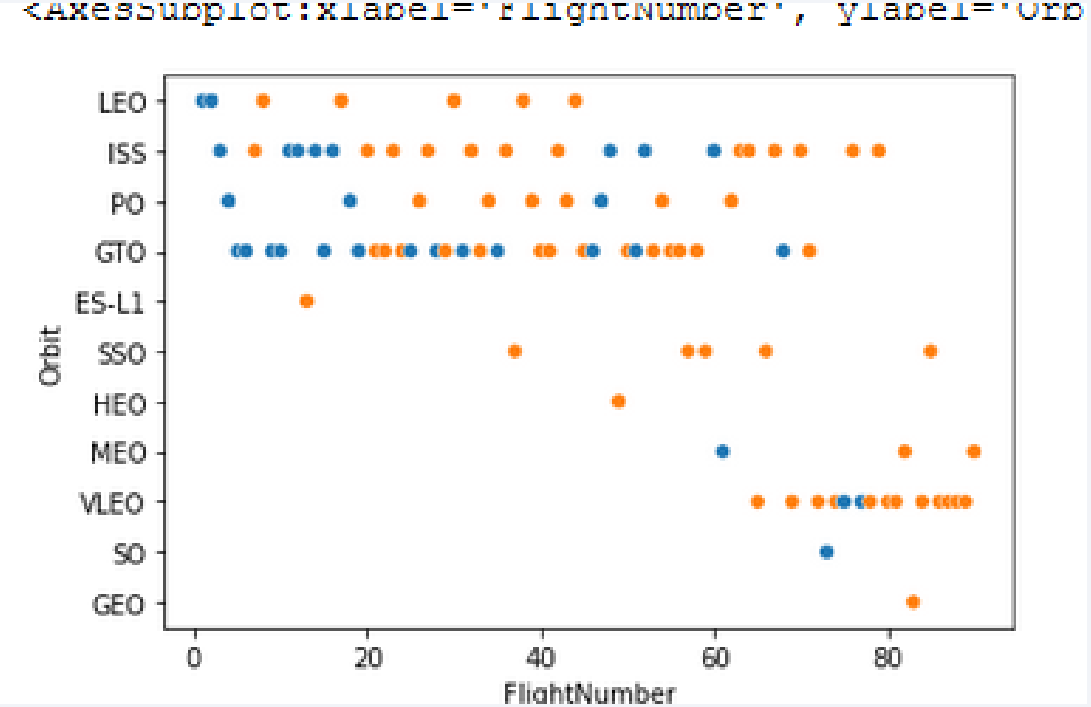



```
AxesSubplot:xlabel='FlightNumber', ylabel='Orbit'>
```



Flight Number vs. Orbit Type

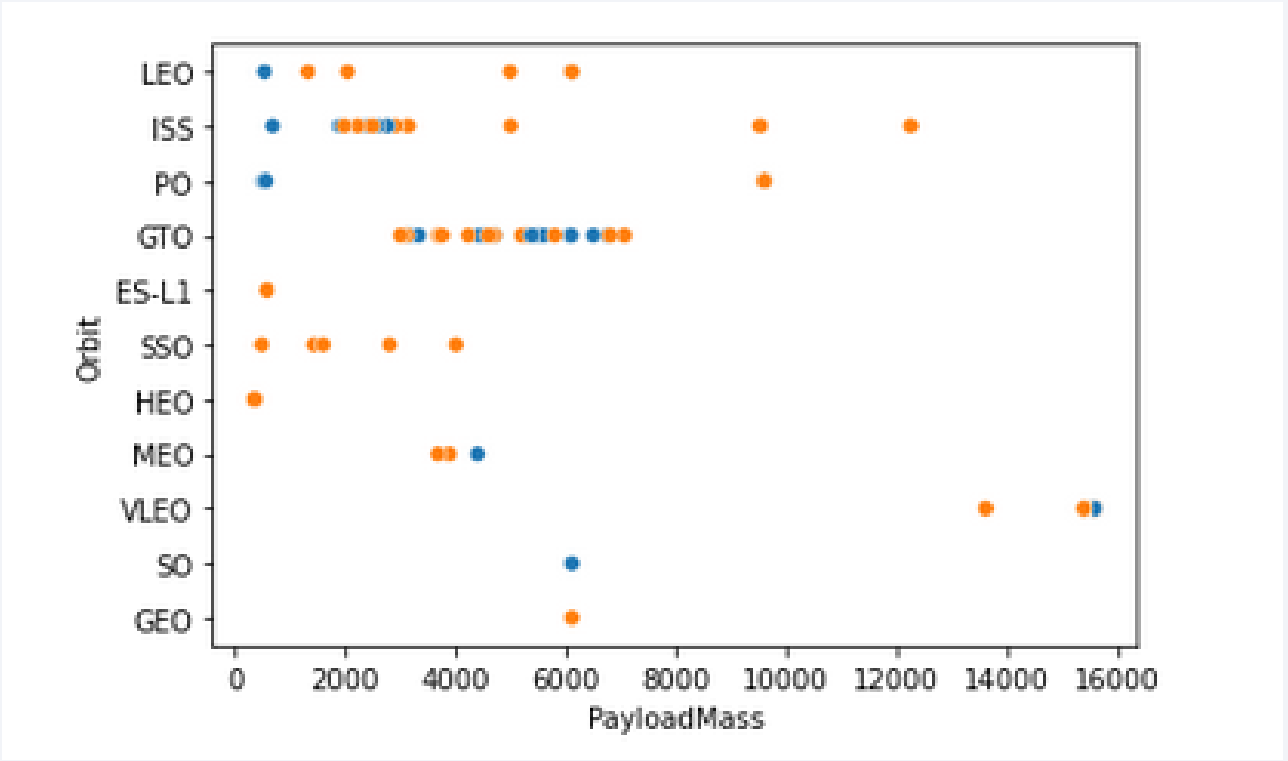
- As time goes on, there are more Very Low missions.
- Most of the missions are to the ISS (International Space Station) or GTO missions.



Payload vs. Orbit Type

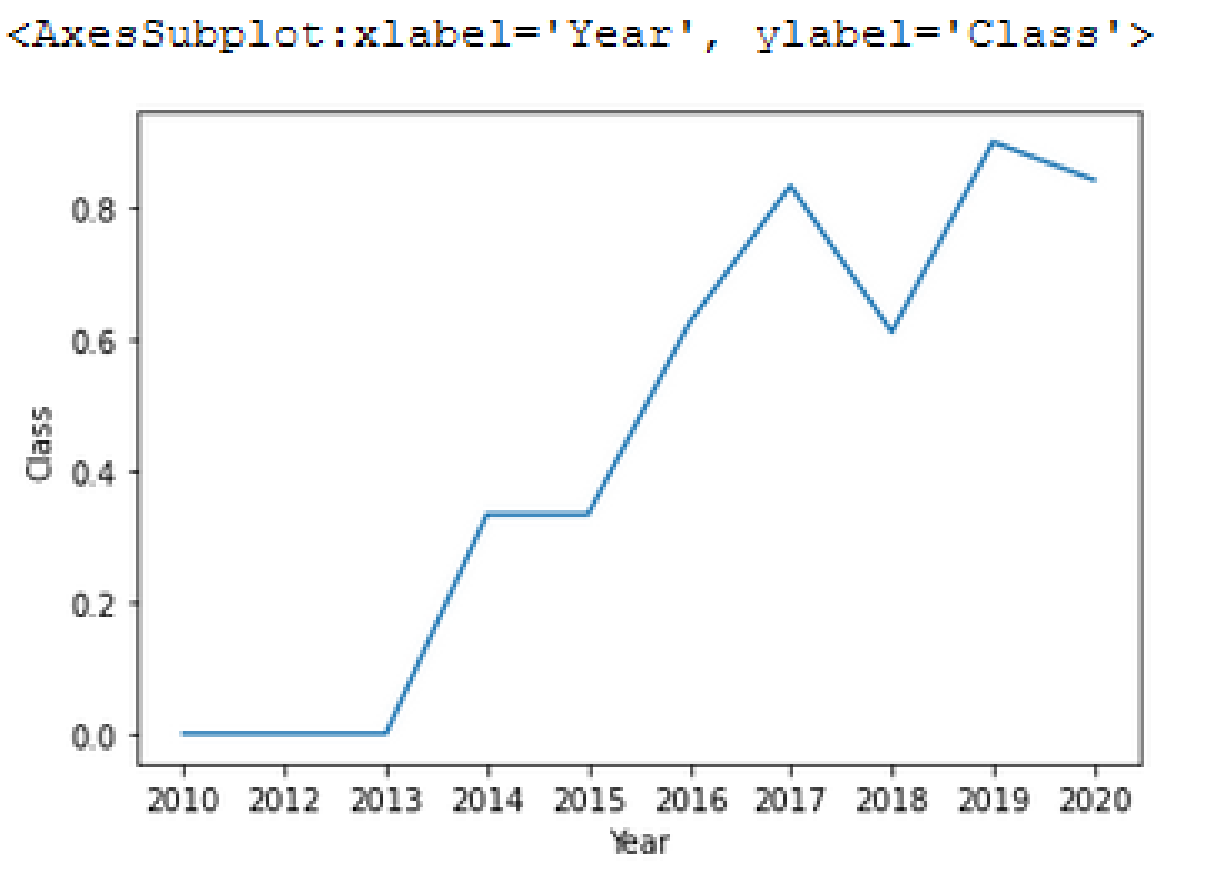
- The highest payloads were for the Very Low (VLEO) missions

- Most GTO missions had payloads between ~3000 and ~8000 kilograms



Launch Success Yearly Trend

- The first success was in 2014
- Since 2019, the success rate has been above 80%
- There have been more successes than failures since 2016.



All Launch Site Names

- The SQL query for the unique launch sites
- There are four unique launch sites

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

```
 * ibm_db_sa://lkv48600:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb?authSource=admin&replicaSet=replset
Done.
```

launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- 5 records of launches from sites starting with CCA
- Missions here are launched by both SpaceX and NASA. There are no successful landings, but the years are early.

```
]: %sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
 * ibm_db_sa://lkv48600:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb?authSource=admin&replicaSet=replset
Done.
```

]:

DATE	time_utc	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- NASA (CRS) rockets carried a cumulative total of 45,596 kilograms

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql select sum(payload_mass__kg_) from SPACEXTBL where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://lkv48600:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb?authSource=admin&replicaSet=replset
Done.
```

1
45596

Average Payload Mass by F9 v1.1

- The average payload of a F9 v1.1 mission is 2928 kg

Display average payload mass carried by booster version F9 v1.1

```
%sql select avg(payload_mass__kg_) from SPACEXTBL where booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://lkv48600:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb?authSource=admin&replicaSet=replset
Done.
```

1
2928

First Successful Ground Landing Date

- First successful ground landing date was 2018-07-22

Hint:Use min function

```
[47]: %sql select landing_outcome, min(DATE) from SPACEXTBL where landing_outcome = 'Success' group by Landing_Outcome
```

```
 * ibm_db_sa://lkv48600:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb?authSource=admin&replicaSet=replset
Done.
```

[47]:

landing_outcome	2
Success	2018-07-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000: (Prefix only, full names in screenshot)
  - F9 v1.1
  - F9 FT
  - F9 B4
  - F9 B5

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [27]:
```
%sql select booster_version, mission_outcome, payload_mass__kg_ from SPACEXTBL where mission_outcome = 'Success'
and payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000
```

```
 * ibm_db_sa://lkv48600:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb?authSource=admin&replicaSet=replset
Done.
```

Out[27]:

booster_version	mission_outcome	payload_mass__kg_
F9 v1.1	Success	4535
F9 v1.1 B1011	Success	4428
F9 v1.1 B1014	Success	4159
F9 v1.1 B1016	Success	4707
F9 FT B1020	Success	5271
F9 FT B1022	Success	4696
F9 FT B1026	Success	4600
F9 FT B1030	Success	5600
F9 FT B1021.2	Success	5300
F9 FT B1032.1	Success	5300
F9 B4 B1040.1	Success	4990
F9 FT B1031.2	Success	5200
F9 FT B1032.2	Success	4230
F9 B4 B1040.2	Success	5384
F9 B5 B1046.2	Success	5800
F9 B5 B1047.2	Success	5300
F9 B5B1054	Success	4400
F9 B5 B1048.3	Success	4850
F9 B5 B1051.2	Success	4200
F9 B5B1060.1	Success	4311
F9 B5 B1058.2	Success	5500
F9 B5B1062.1	Success	4311

Total Number of Successful and Failure Mission Outcomes

- 100 successful missions,
- 1 failure

List the total number of successful and failure mission outcomes

```
8]: %sql select mission_outcome, count(*) from SPACEXTBL group by mission_outcome
```

```
 * ibm_db_sa://lkv48600:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb?authSource=admin&replicaSet=replset
Done.
```

8]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- All boosters that carried the maximum payload were F9 B5

```
In [32]: %sql select distinct(booster_version) from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) as
MAX from SPACEXTBL)
         ##%sql select max(payload_mass__kg_) as MAX from SPACEXTBL
```

```
 * ibm_db_sa://lkv48600:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu01qde00.databases.appdomain.clou
d:31321/bludb?authSource=admin&replicaSet=replset
Done.
```

Out[32]:

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- There were two failed landing outcomes in 2015 (“No attempt” and “Precluded” were not included in failures)
- Both were launched from CCAFS LC-40, and both were LEO ISS missions launched by NASA.

In [37]:
```
%sql select * from SPACEXTBL where year(DATE) = 2015 and lower(landing_outcome) like '%fail%'
```

```
 * ibm_db_sa://lkv48600:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.clou
d:31321/bludb?authSource=admin&replicaSet=replset
Done.
```

Out[37]:

DATE	time_utc	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing_outcome
2015-01-10	09:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Most landings were not attempted, followed by failures on a drone ship and success on a drone ship, in equal quantities.

```
In [45]: %sql select landing_outcome, count(*) as amount from SPACEXTBL where date >= '2010-06-04' and date <= '2017-03-20' group by(landing_outcome) order by amount desc
```

```
 * ibm_db_sa://lkv48600:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb?authSource=admin&replicaSet=replset
Done.
```

Out[45]:

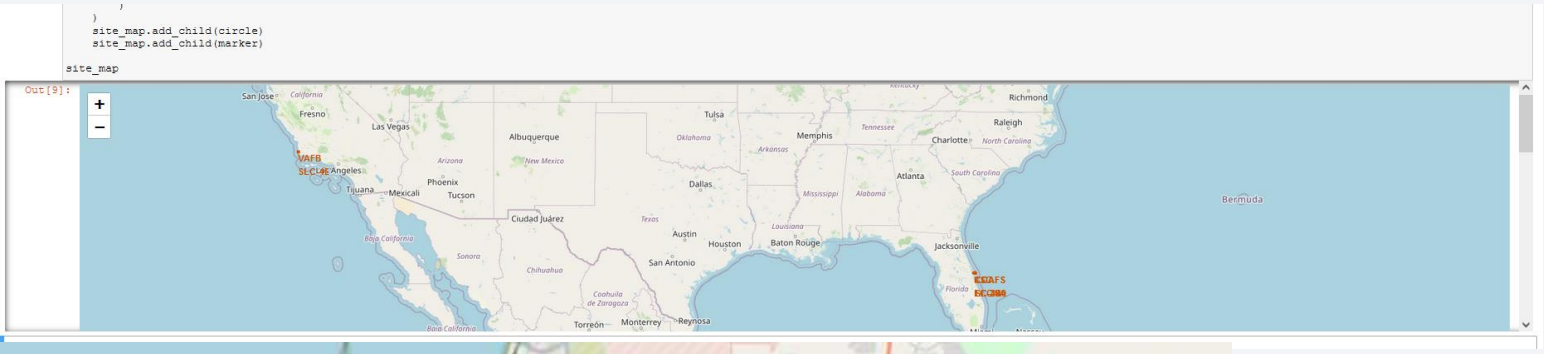
landing_outcome	amount
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

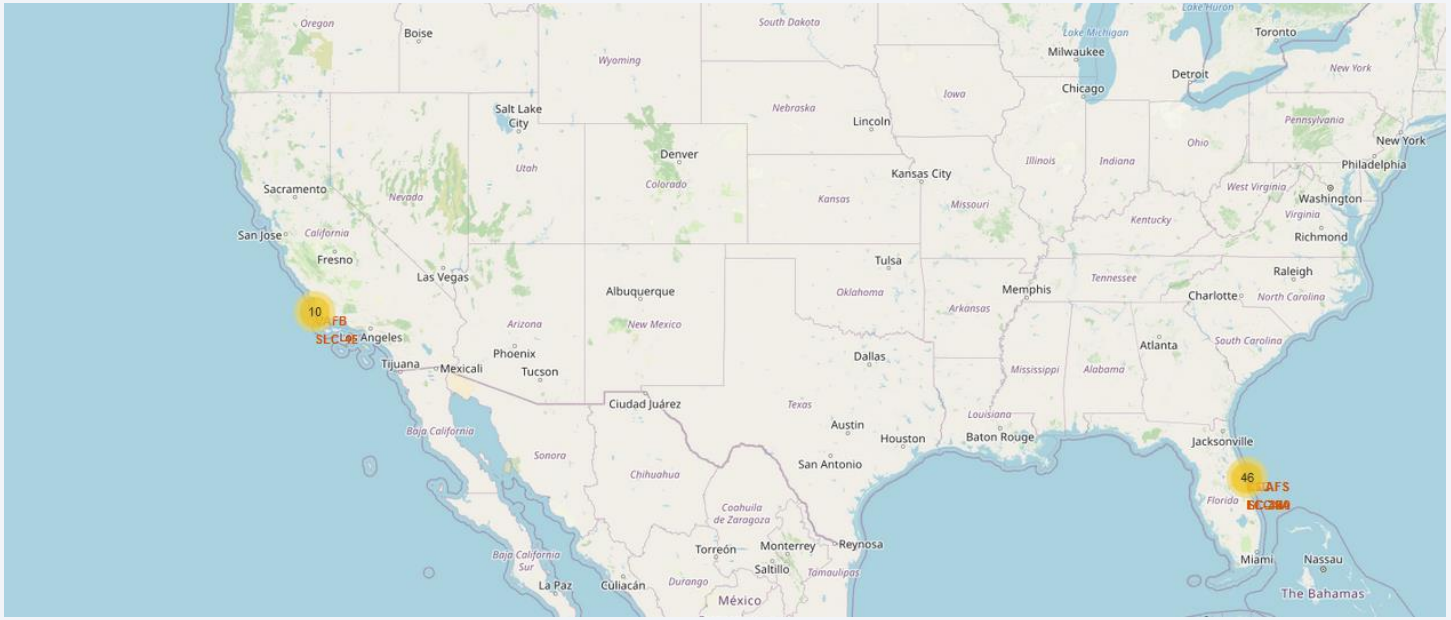
Launch Sites

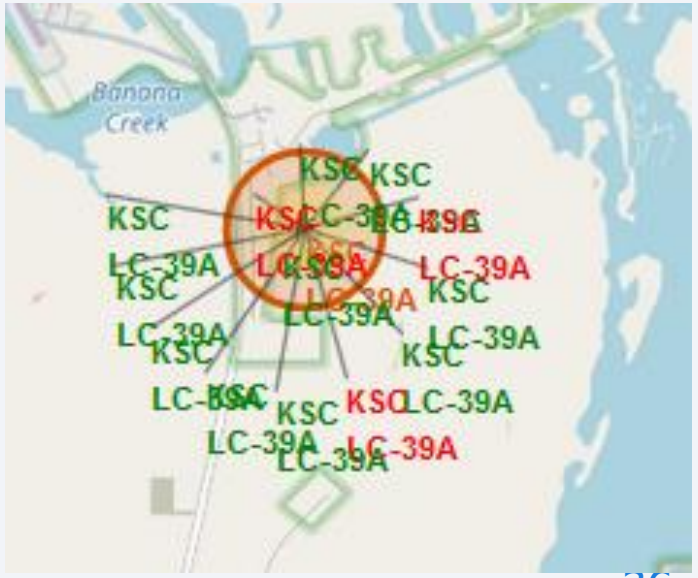
- The Launch sites are in California and Florida, near the coasts



Launches by Launch Site

- Launches are clustered by site, especially when zoomed out.
- Launches with successful landings are indicated by green, otherwise red. (Colored labels only seen when clicked, as in the lower right)





Distance from the coast

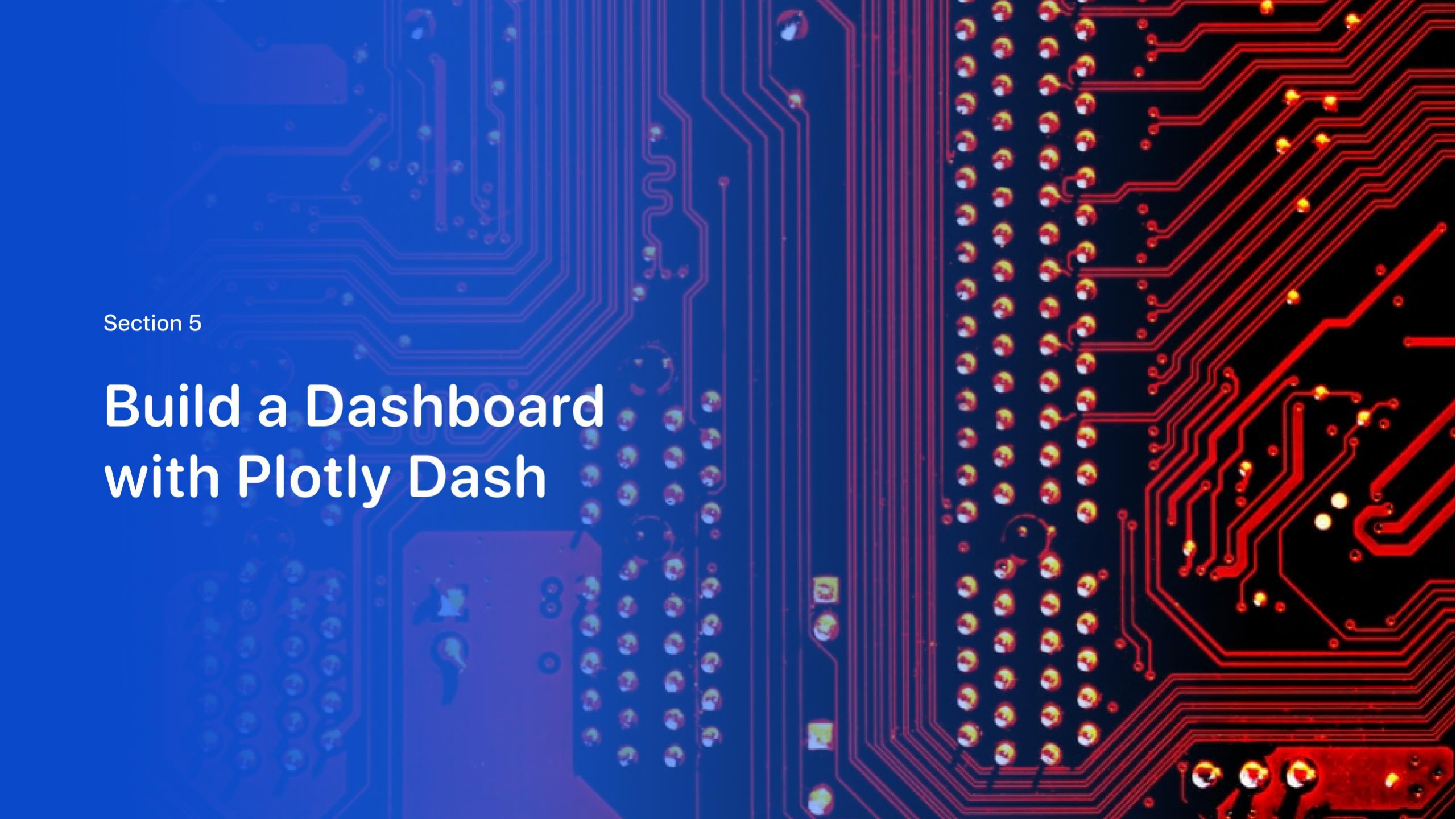
The CCAFS location is 0.8534 km from the coast

```
launch_site_lon = -80.57682
coastline_lat = 28.56379
coastline_lon = -80.56811

distance_coastline = calculate_distance(launch_site_lat, launch_site_lon, coastline_lat, coastline_lon)
distance_coastline
```

[18]: 0.8534225593732755

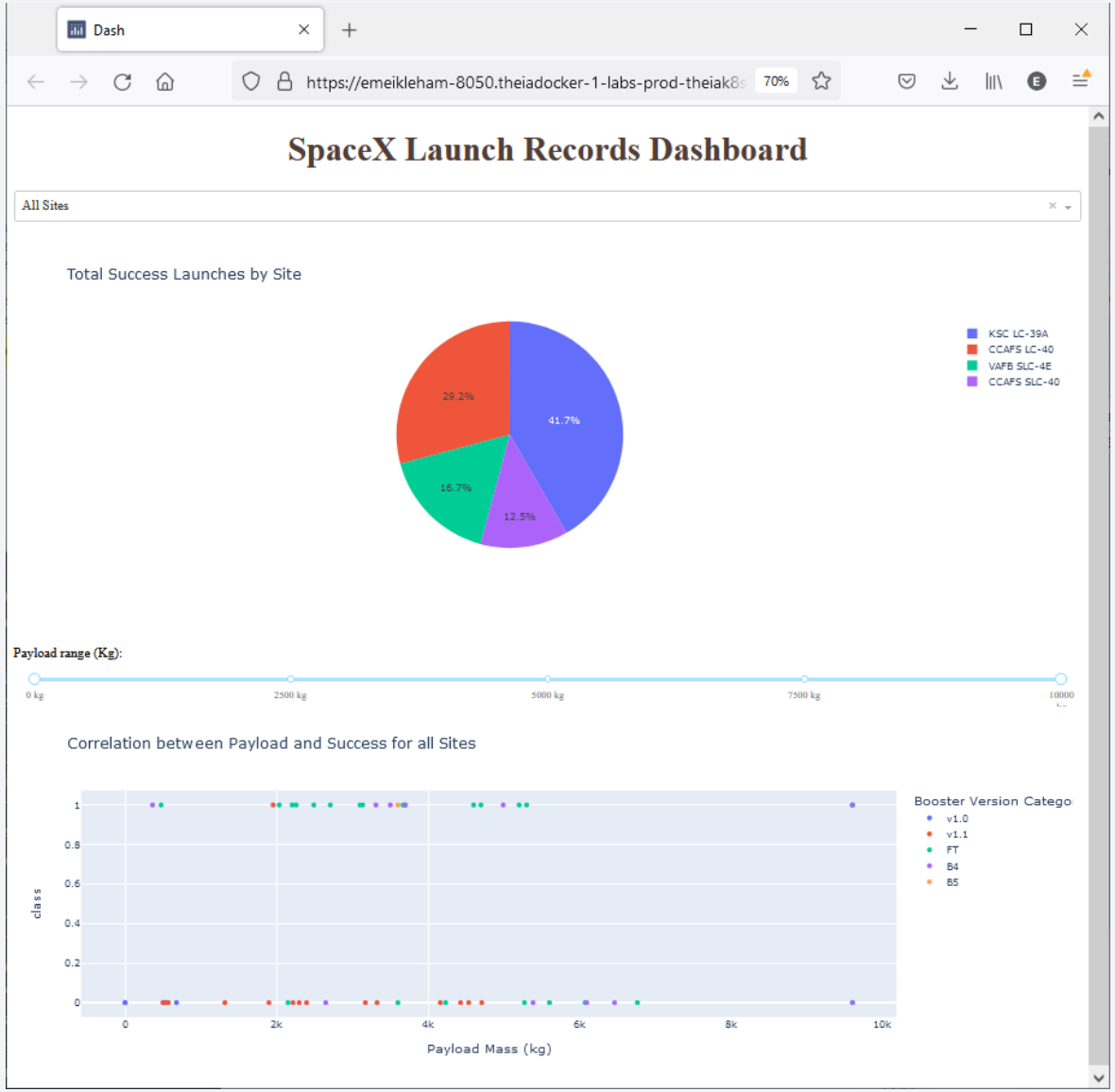
Section 5

Build a Dashboard with Plotly Dash



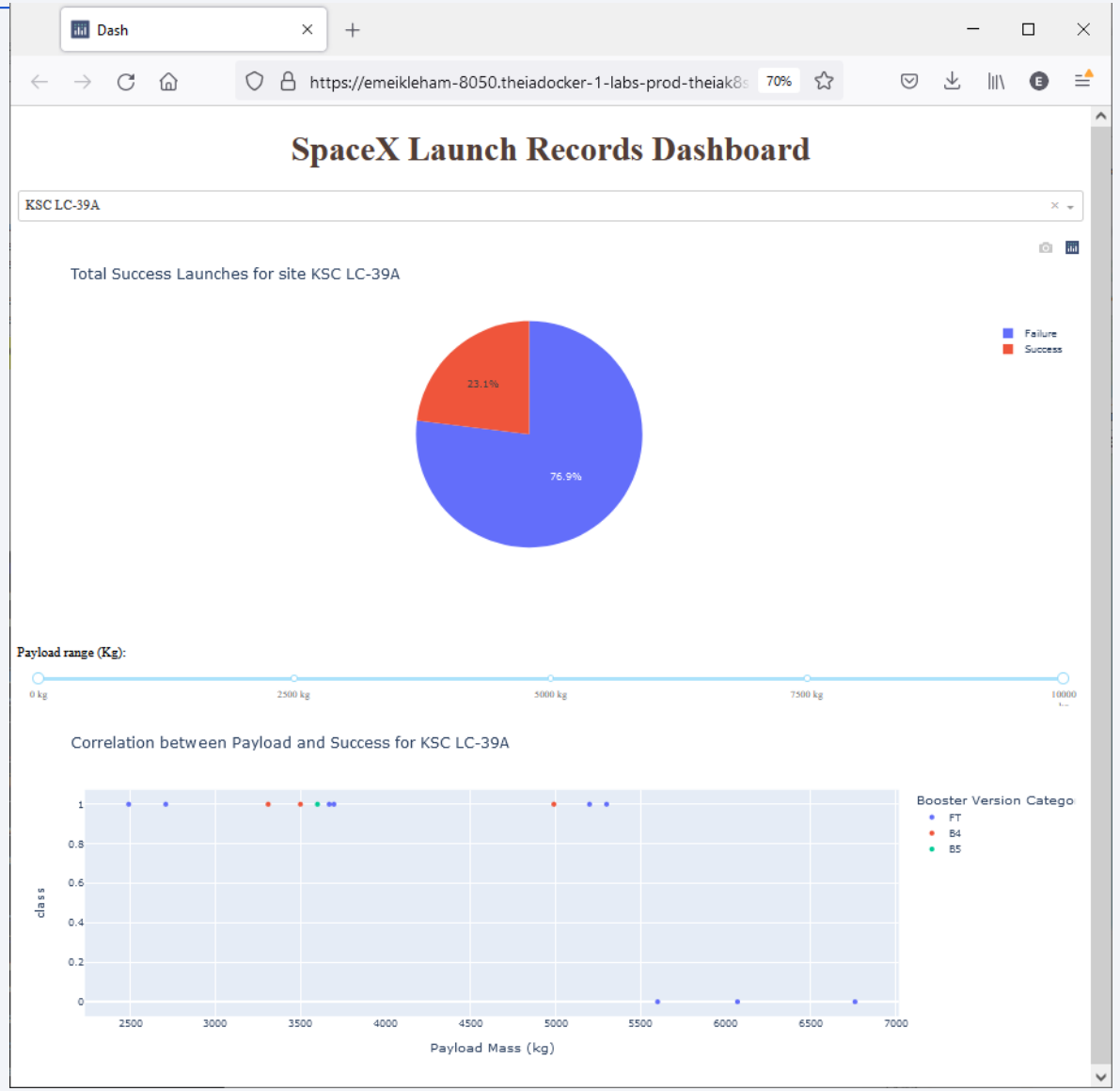
Launch Successes by Type

- 41.7% (the most) successes came from Kissimmee St Cloud (KSC LC-39A)
- Note that “All Sites” is chosen in the dropdown menu.



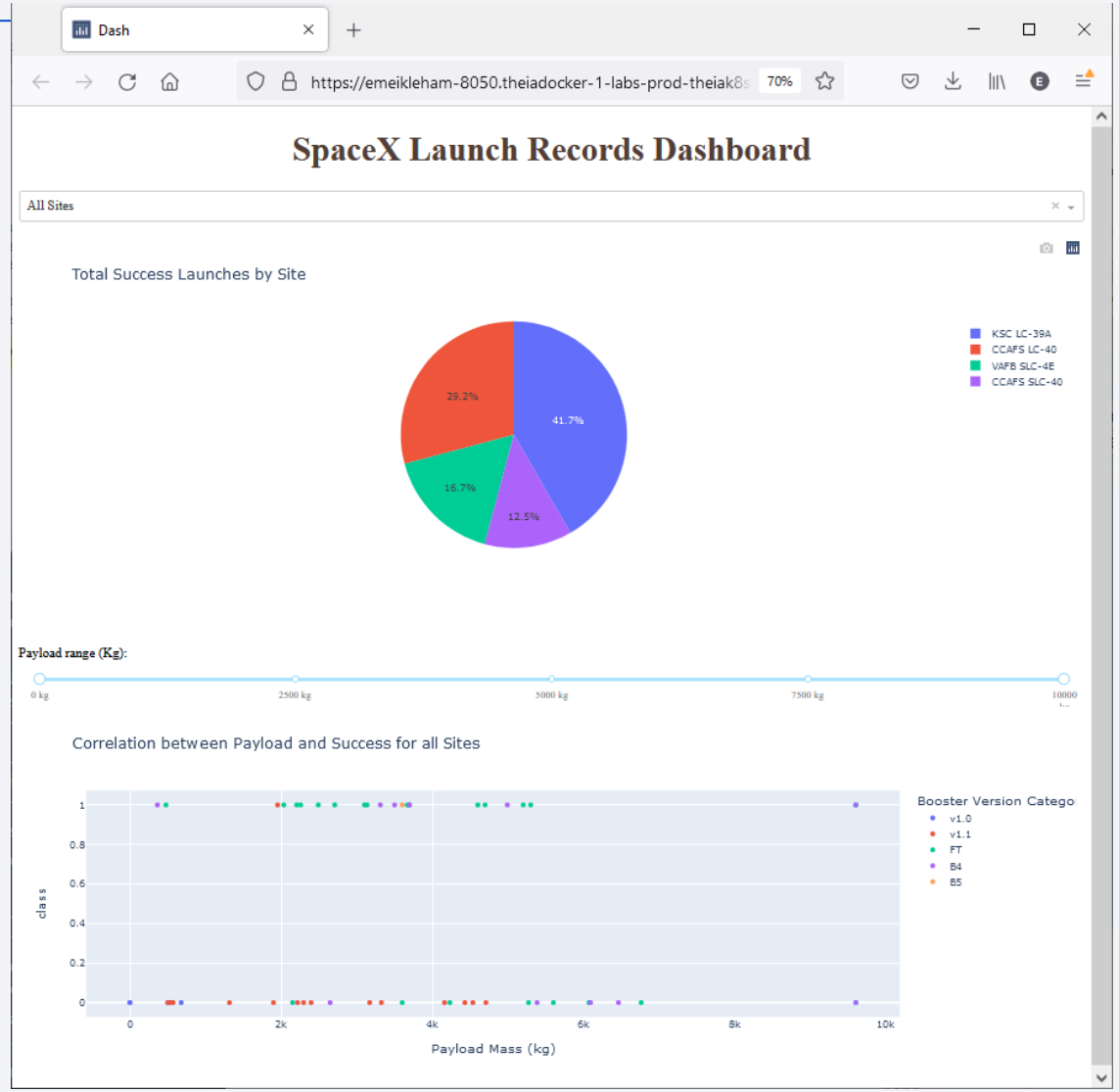
Piechart for Site with Highest Success Rate

- Kissimmee was the site with the highest success rate.
- 76.9% of launches from KSC LC-39A were successful.



Scatter Plot

- All sites is chosen in the dropdown menu
- Payload (X-axis) vs Outcome (i.e. class, 1=success, Y-axis)
- The slider has been chosen for the range 2500kg to 10000 kg

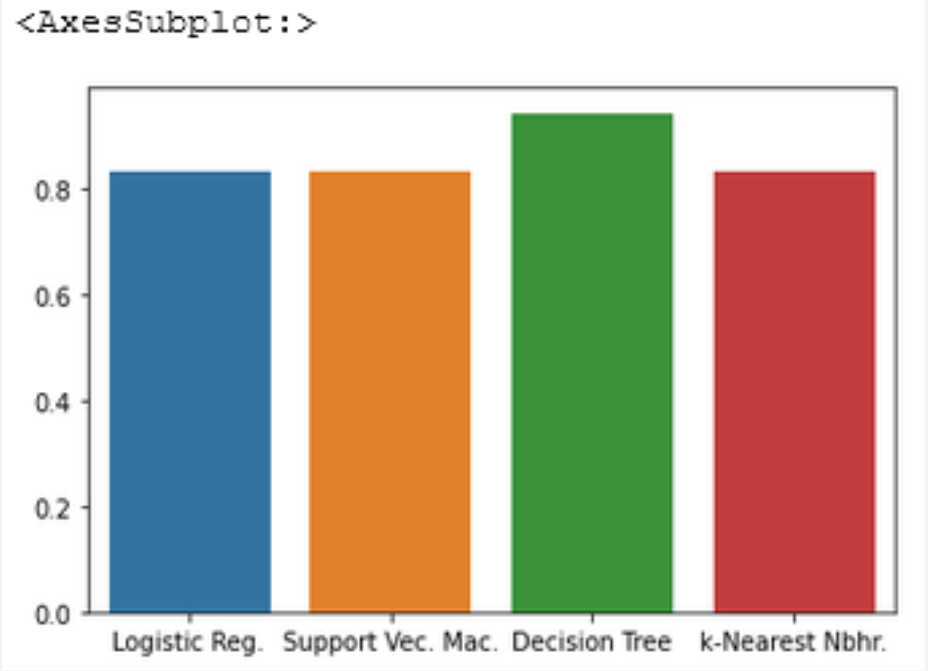


Section 6

Predictive Analysis (Classification)

Classification Accuracy

- After using GridSearchCV to try various parameters for Logistic Regression, Support Vector Machines, Decision Tree, and k-NN models, and getting scores from the test sets, the best model for classification is the Decision Tree.



Confusion Matrix

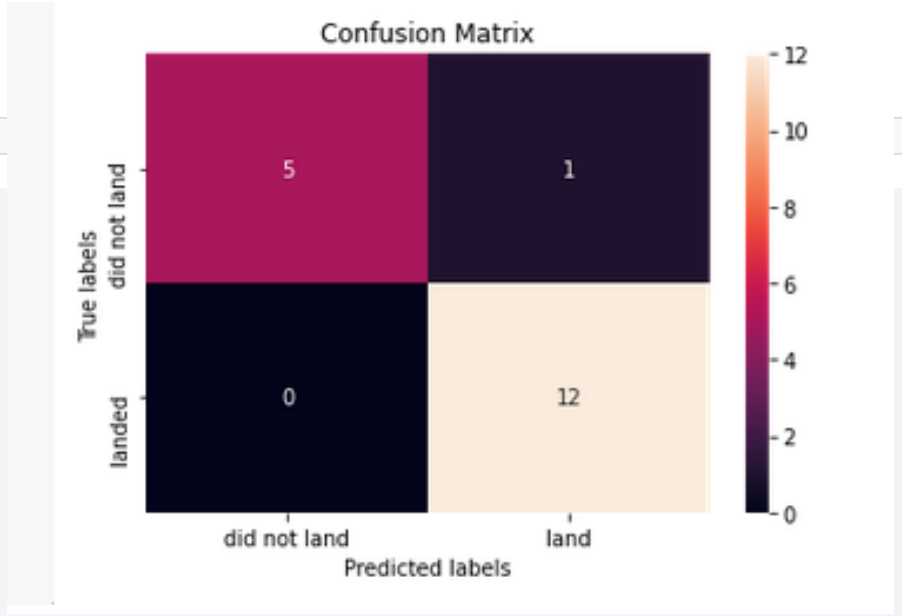
- The best model is a Decision Tree using Gini criterion, a max depth of 14, 2 samples per leaf, 10 samples per split, and the “best” splitter.

```
In [22]: print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
         print("accuracy :",tree_cv.best_score_)

tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 14, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'splitter': 'best'}
accuracy : 0.8875
```

TASK 9

Calculate the accuracy of tree_cv on the test data using the method `score`:

```
In [23]: tree_cv.score(X_test,Y_test)

Out[23]: 0.9444444444444444
```



Conclusions

- It seems as though we have found an effective set of descriptive parameters
- The model seems well fit when used on the test set
- Nevertheless, there are only ~100 datapoints, so there are only ~20 test points
- The test score of the best model is over 90%

Appendix

- The prediction matrix has 83 variables (including one-hot variables)
- There are only 90 data points for F9 missions
- This table is pre-scaling.

	FlightNumber	PayloadMass	Flights	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	Orbit_GTO	Orbit_HEO	Orbit_ISS	...	Serial_B1058	Serial_B1059	Serial_B1060	Serial_B1062	GridFins_False	GridFins_True	Reused_False	Reused_True	Legs_False	Legs_True
0	1.0	6104.959412	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
1	2.0	525.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
2	3.0	677.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
3	4.0	500.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
4	5.0	3170.000000	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
85	86.0	15400.000000	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0
86	87.0	15400.000000	3.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0
87	88.0	15400.000000	6.0	5.0	5.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0
88	89.0	15400.000000	3.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0
89	90.0	3681.000000	1.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	1.0

90 rows × 83 columns

Thank you!