IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Travis Gubbe
February 14, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis
  - Interactive Visual Analytics with Folium
  - Predictive Modeling
- Summary of all results
  - Exploratory Data Analysis results
  - Predictive Modeling results (Best Model)

Introduction

- Project background and context
- SpaceX is able to save millions of dollars because they can reuse the first stage of rocket launches.
- SpaceY wants to outbid SpaceX for a rocket launch
- *Machine Learning Pipeline can be used to predict if the first stage will land successfully.*
- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - What conditions must be met to ensure highest probability of success?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected via web scraping (Wikipedia) and API queries.
- Perform data wrangling
  - Determine how often a launch was successful.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Grid Search Cross Validation performed to determine best model.

Data Collection

- Data collected using various methods
  - Data was collected using get request to SpaceX API
  - Data was cleaned and checked for missing values
  - In addition, web scraping completed of SpaceX Wikipedia for Falcon 9 launch records

Data Collection – SpaceX API

- Get request used to collect data from SpaceX API.
- Response converted to JSON file.

In [18]:
```
# 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'])
```

Now let's start requesting rocket launch data from SpaceX API with the following URL:

In [19]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [20]:
```
response = requests.get(spacex_url)
```

Check the content of the response

In [21]:
```
print(response.content)
```

```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/94/f2/NN6Ph
45r_o.png","large":"https://images2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":
{"small":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.sp
ace.com/2196-spacex-inaugural-falcon-1-rocket-lost-launch.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17
```

Data Collection – SpaceX API

- Data saved in data frame to perform data wrangling and data cleaning.
- Data filtered to only include Falcon 9 launches.

In [38]:
```
# Create a data from launch_dict
df = pd.DataFrame.from_dict(launch_dict)
```

Show the summary of the dataframe

In [39]:
```
# Show the head of the dataframe
df.head()
```

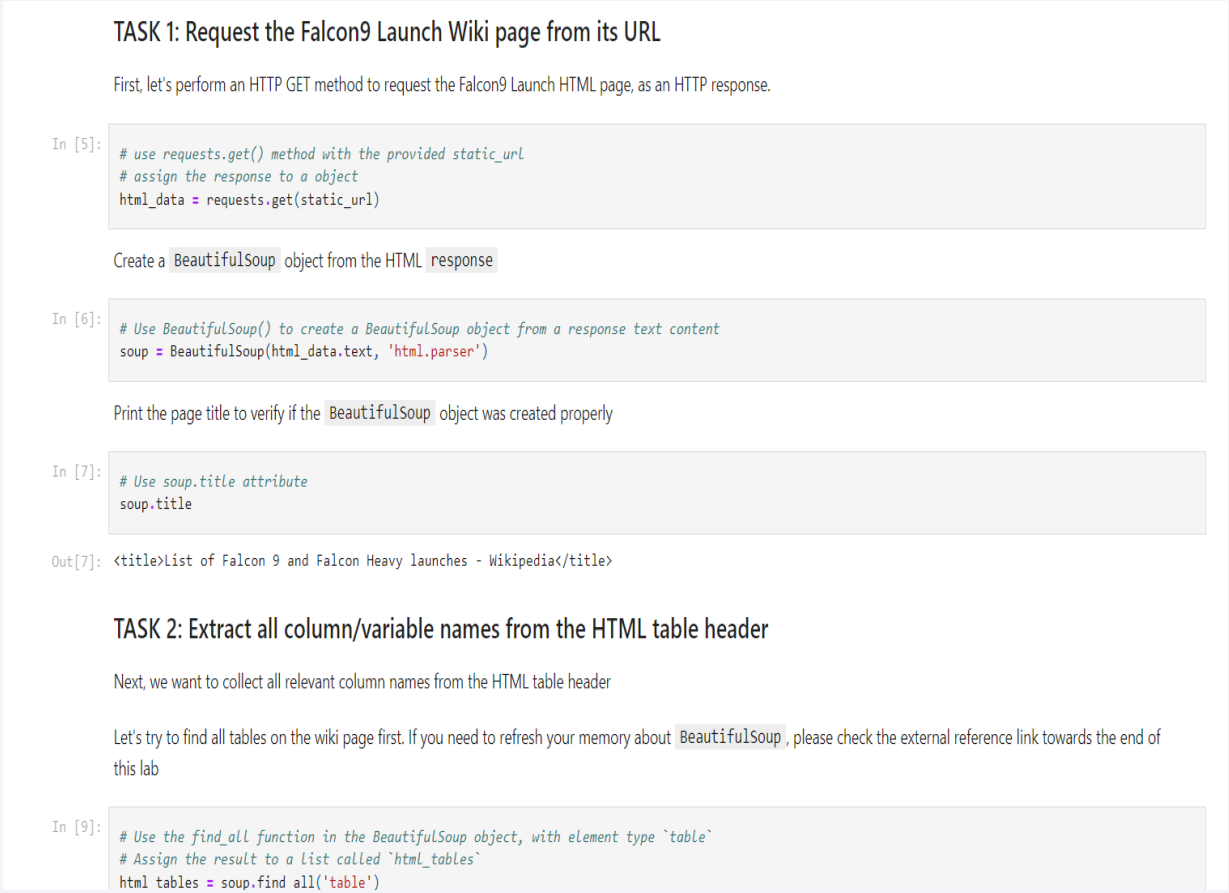
Out[39]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Lat
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.743129	9.04
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.743129	9.04
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.743129	9.04
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin3C	167.743129	9.04
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.56

Source code:
https://github.com/tgubbe/IBM_DataScience_Project/blob/main/Data%20Collection%20for%20Final%20Project.ipynb

Data Collection - Scraping

- Extract Falcon 9 launch records table from Wikipedia.
  - BeautifulSoup
- Table converted to Pandas dataframe.
- Table setup exported to .csv



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 [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

In [9]:
```
# 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')
```

Source code:
https://github.com/tgubbe/IBM_DataScience_Project/blob/main/Data%20Collection%20Web%20Scraping.ipynb

Data Wrangling

- Perform Exploratory Data Analysis to understand the data.
- Create set of outcomes of launches at each site, both successful and failures.
- Add created set list to data frame in new column “Class”.

TASK 2: Calculate the number and occurrence of each orbit

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

In [6]:
```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

Out[6]:
```
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
```

TASK 3: Calculate the number and occurence of mission outcome per orbit type

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes`.Then assign it to a variable landing_outcomes.

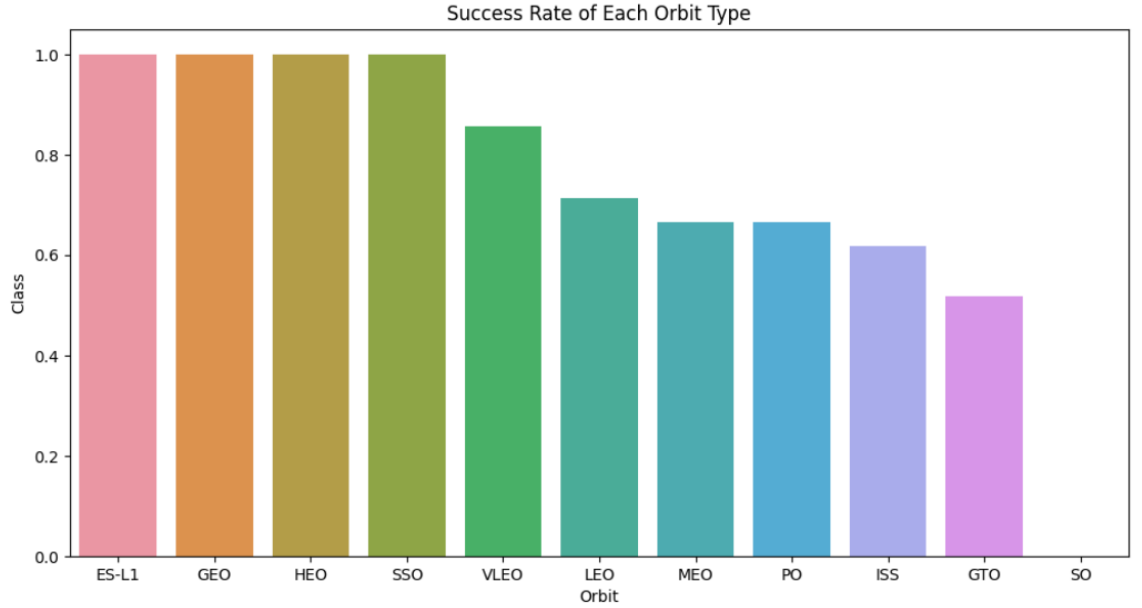
In [8]:
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()

landing_outcomes
```

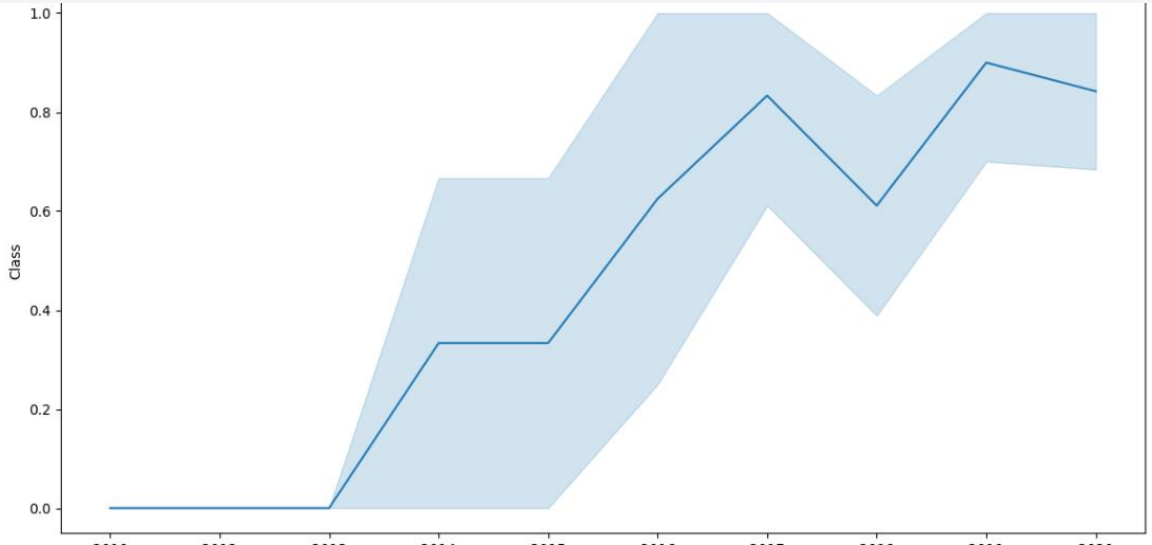
Out[8]:
```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

Source code:
https://github.com/tgubbe/IBM_DataScience_Project/blob/main/Data%20Wrangling.ipynb

EDA with Data Visualization

- Data explored by visualizing the relationship between the variables.
  - Ex: Flight number vs. launch site, payload vs. launch site
- Bar Graph displays success rate of each orbit type.
- Line Graph displays launch success yearly trend.





Source code:
https://github.com/tgubbe/IBM_DataScience_Project/blob/main/EDA%20with%20Visualizations.ipynb

EDA with SQL

- SQL queries include:
  - Display names of unique launch sites
  - Display total payload mass carried by boosters launched by NASA (CRS)
  - List date of first successful landing outcome when grounding pad was achieved.
  - List boosters which have success in drone ship with payload mass between 4000-6000 kg.
  - List total number of successful and failed outcomes.
  - Rank the count of successful landing outcomes between 6/4/2010 – 3/20/2017.

Source code:
https://github.com/tgubbe/IBM_DataScience_Project/blob/main/EDA%20Using%20SQL.ipynb

Build an Interactive Map with Folium

- All launch sites marked, map objects (markers, circles, lines) added to mark successful or failed launches for each site.
- Assigned launch outcome (Failure/Success) to class 0 and 1 respectively.
- Calculated distances between launch site and proximities (highways, railroads, coastlines, cities).
    - Used to determine the proximity of launch sites to highways, railroads, coastlines, and cities.

Source code:
https://github.com/tgubbe/IBM_DataScience_Project/blob/main/Interactive%20Visuals%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Interactive dashboard built with Plotly.
- Pie charts created to display the total launches for each site.
- Scatter graph created to show relationship between outcome and payload mass for each booster version.

Source code:
https://github.com/tgubbe/IBM_DataScience_Project/blob/main/spacex_dash_app.py

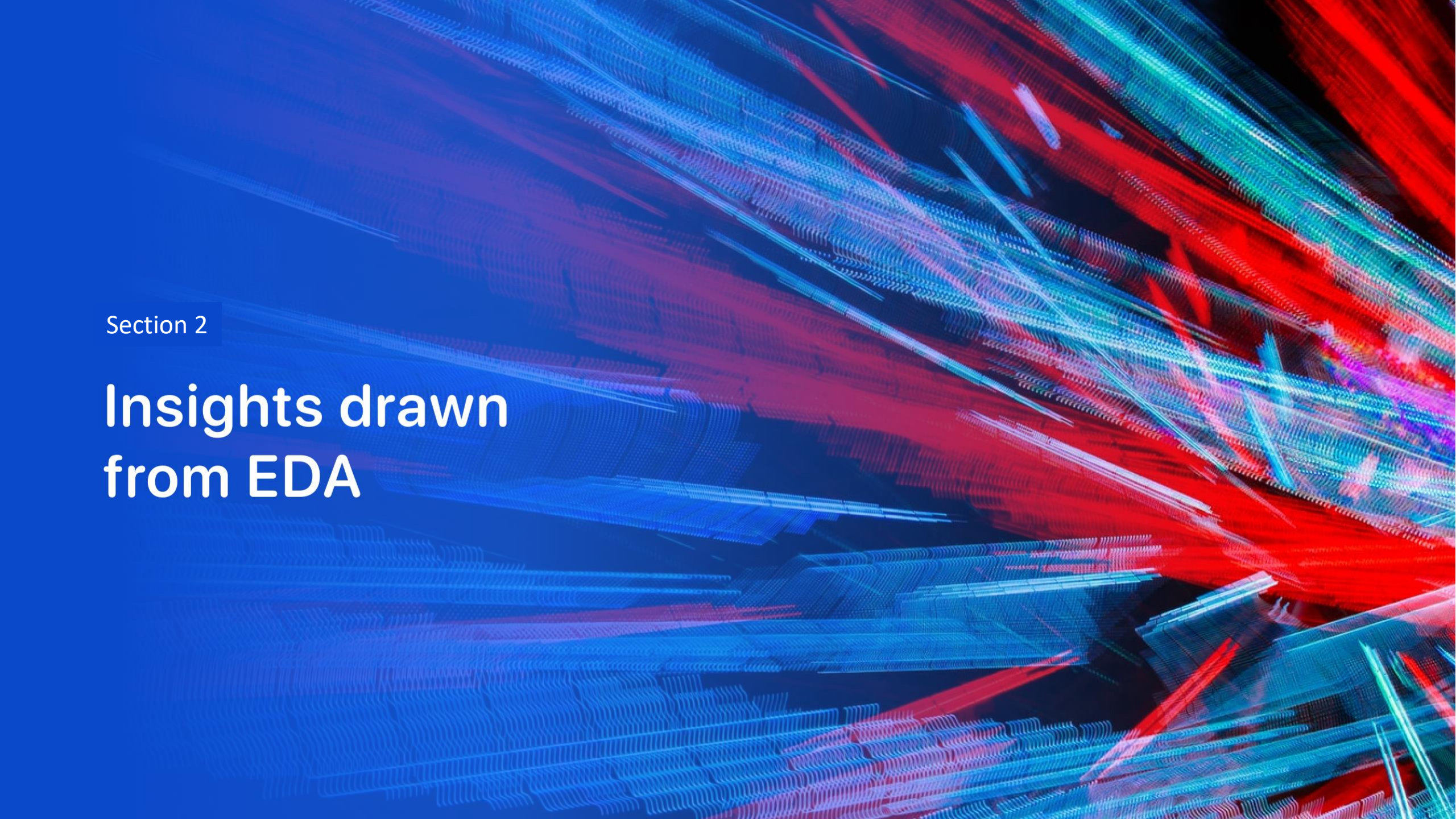
Predictive Analysis (Classification)

- Data was standardized, then split into training and test data sets.
- Various machine learning models were built, then tuned with hyperparameters.
  - SVM
  - Logistic Regression
  - K Nearest Neighbors
  - Decision Tree
- Best performing model determined by the accuracy of each model.

Source code:
https://github.com/tgubbe/IBM_DataScience_Project/blob/main/Machine%20Learning%20Predictions.ipynb

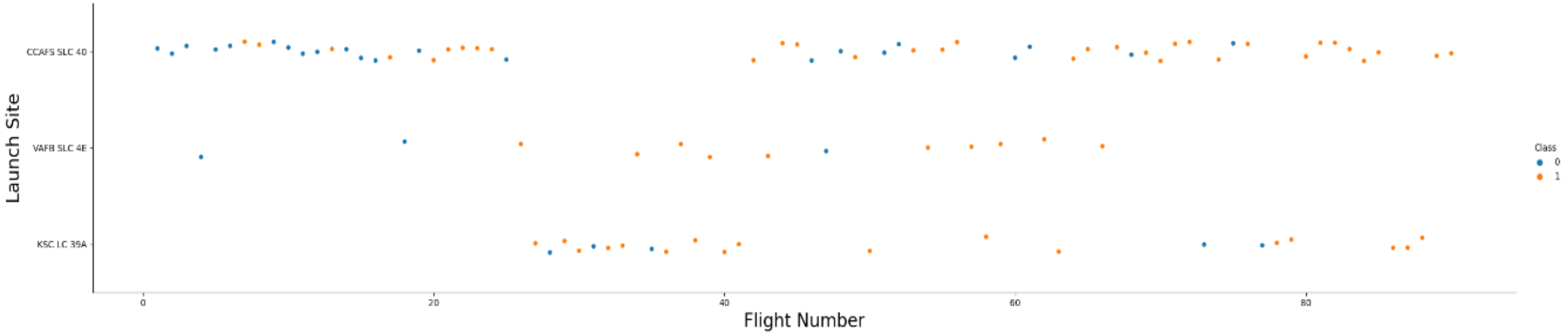
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

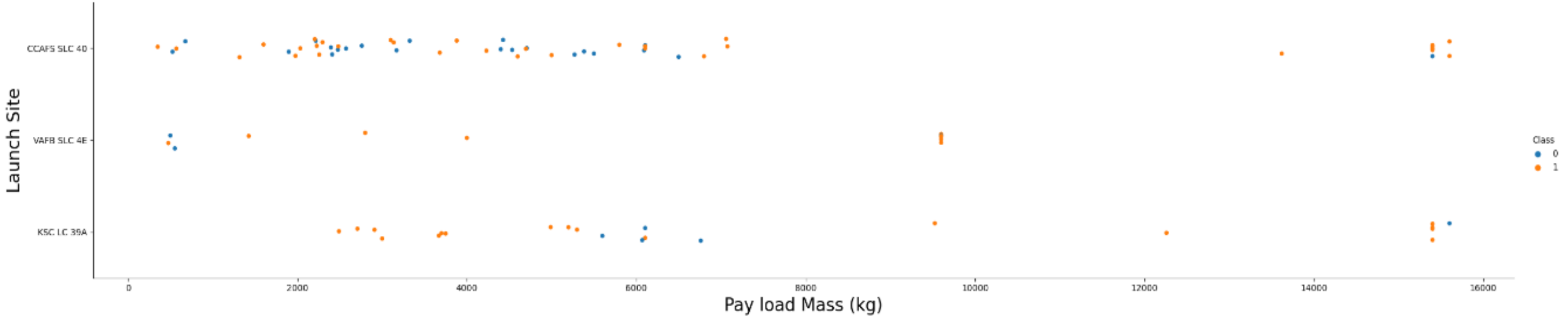
Flight Number vs. Launch Site

- From the scatter plot, the larger number of flights at each launch site, the greater amount of success rate.



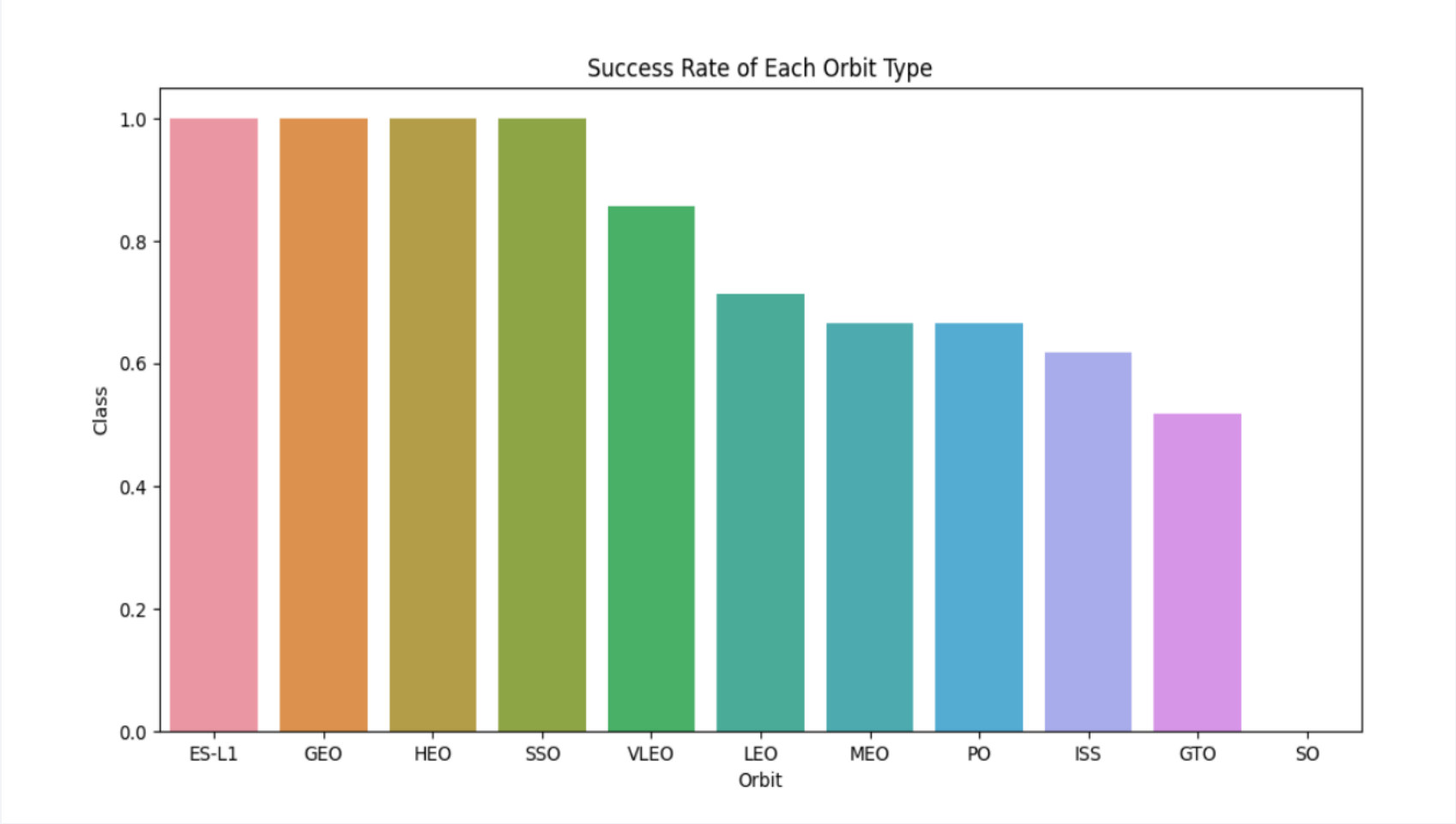
Payload vs. Launch Site

- From the scatter plot, it appears the CCAFS launch site had more success with larger payload masses.



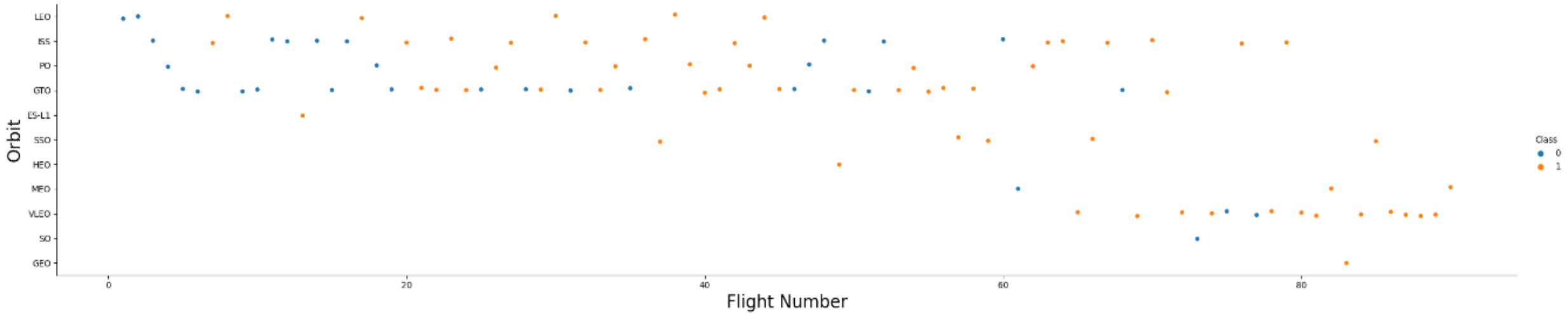
Success Rate vs. Orbit Type

- From the bar plot, the following orbits had the most success:
  - ES-L1
  - GEO
  - HEO
  - SSO



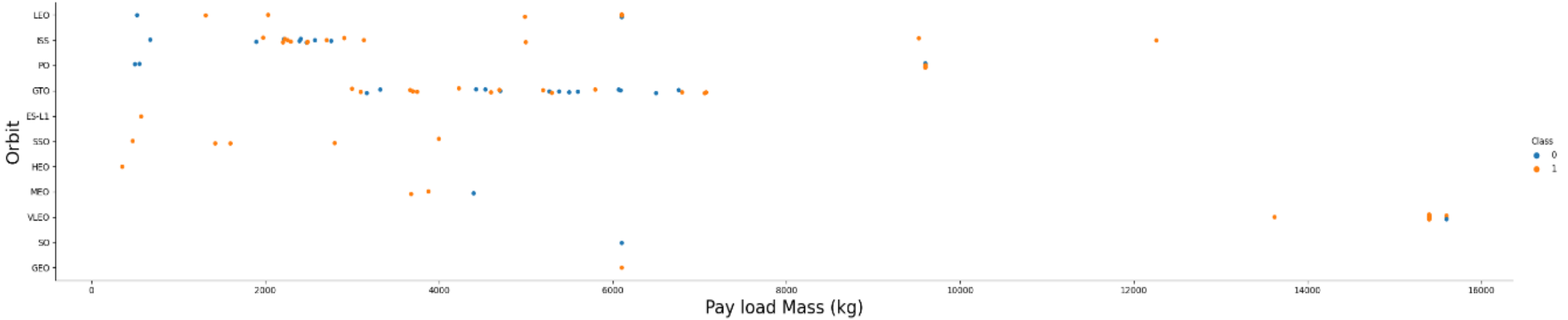
Flight Number vs. Orbit Type

- Scatter plot of Flight number vs. Orbit type
  - It appears when flight number increases, the chance of success increases.



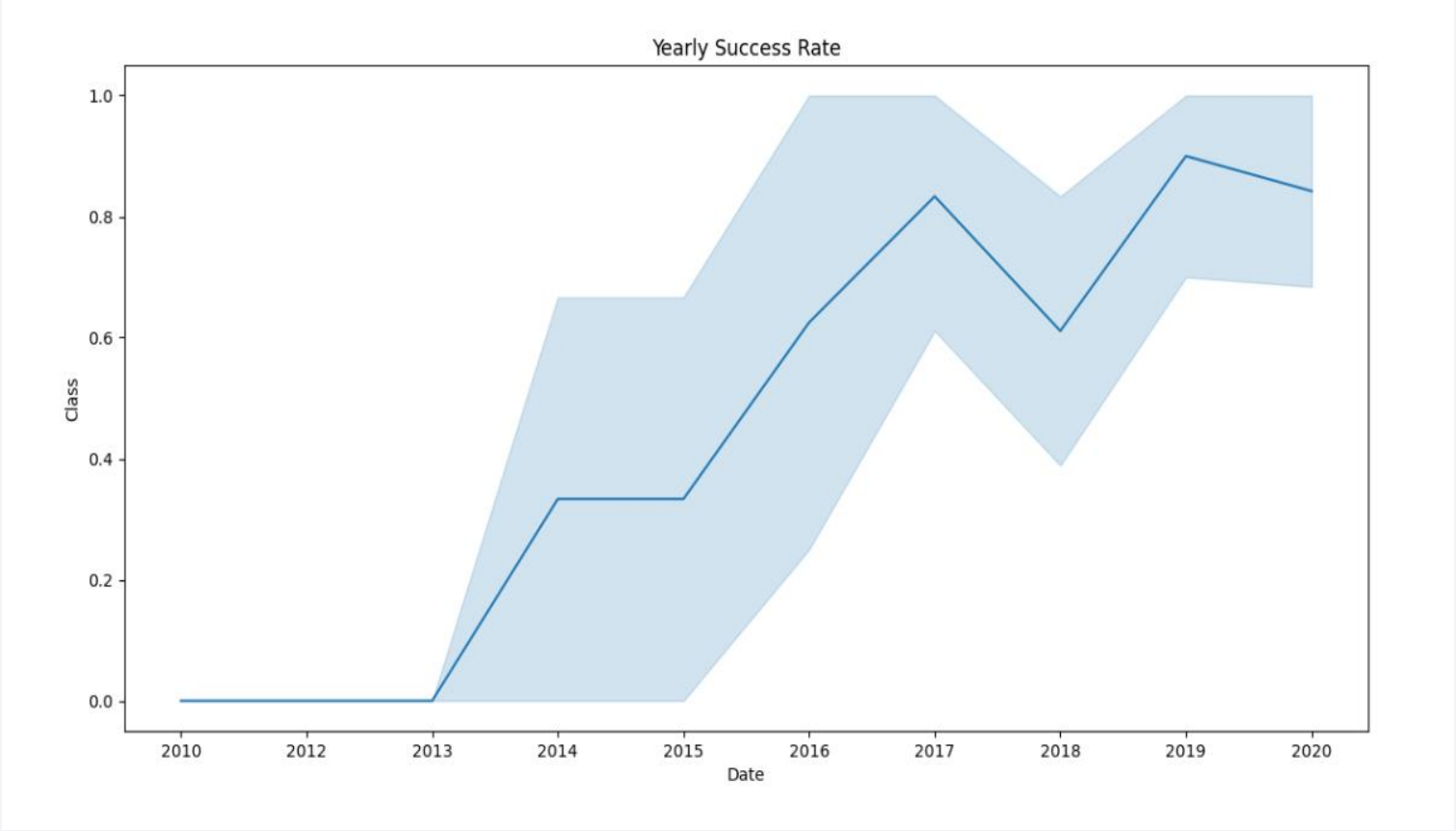
Payload vs. Orbit Type

- Scatter plot of payload vs. orbit type.
  - It appears heavier payloads are ISS, PO, and VLEO.



Launch Success Yearly Trend

- Line chart of yearly average success rate
- The success rate had its largest increase from 2015 – 2017.
- There has not been a large increase in success rate since 2017.



All Launch Site Names

- 5 Unique Launch Site names using 'DISTINCT' SQL command.
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E

Display the names of the unique launch sites in the space mission

In [7]:
```
%sql SELECT DISTINCT Launch_Site FROM SPACEX
```

```
 * ibm_db_sa://tpw78977:***@125f9f61-9715-46f9-9399-c8177b21803b.
Done.
```

Out[7]:

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

Launch Site Names Begin with 'CCA'

- Display 5 records where launch sites begin with `CCA`

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT * FROM SPACEX WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://tpw78977:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
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

- Calculated the total payload carried by boosters from NASA (45,596 kg) using the query below

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

```
%sql SELECT SUM(Payload_Mass__KG_) AS Total_Payload_Mass_KG FROM SPACEX WHERE Customer LIKE 'NASA (CRS)'
```

```
 * ibm_db_sa://tpw78977:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.
Done.
```

total_payload_mass_kg
45596

Average Payload Mass by F9 v1.1

- Calculated the average payload mass carried by booster version F9 v1.1 (2,928 kg) using the query below.

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

```
%sql SELECT AVG(Payload_Mass__KG_) AS AVG_Payload_Mass_KG FROM SPACEX WHERE Booster_Version = 'F9 v1.1'
```

```
 * ibm_db_sa://tpw78977:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.
Done.
```

avg_payload_mass_kg
2928

First Successful Ground Landing Date

- Found date of the first successful landing outcome on ground pad (December 22, 2015) using the query below.

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql SELECT Min(Date) AS First_Landing FROM SPACEX WHERE Landing__Outcome LIKE 'Success (ground pad)'
```

```
 * ibm_db_sa://tpw78977:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.c
Done.
```

first_landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Found the boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

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

```
%%sql SELECT Booster_Version FROM SPACEX WHERE Landing__Outcome = 'Success (drone ship)'
AND Payload_Mass__KG_ > 4000
AND Payload_Mass__KG_ < 6000
```

```
 * ibm_db_sa://tpw78977:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/
Done.
```

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Calculated the total number of successful and failure mission outcomes
  - 99 Success
  - 1 Success (payload status unclear)
  - 1 Failure

List the total number of successful and failure mission outcomes

```
%sql SELECT DISTINCT Mission_Outcome, COUNT(*) AS Total FROM SPACEX GROUP BY Mission_Outcome
```

```
 * ibm_db_sa://tpw78977:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.a
Done.
```

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

Boosters Carried Maximum Payload

- Listed the names of the booster which have carried the maximum payload mass.
  - 12 in total

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql SELECT DISTINCT Booster_Version FROM SPACEX
WHERE Payload_Mass__KG_ = (SELECT MAX(Payload_Mass__KG_) FROM SPACEX)
ORDER BY Booster_Version
```

```
 * ibm_db_sa://tpw78977:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.
Done.
```

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

- List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 2 in total, both at launch site CCAFS LC-40

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%%sql SELECT Booster_Version, Launch_Site FROM SPACEX WHERE Landing__Outcome = 'Failure (drone ship)'
AND DATE_PART('YEAR', Date) = 2015
```

```
 * ibm_db_sa://tpw78977:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

- Ranked the count of successful landing outcomes from June 4, 2010 and March 20, 2017.
  - 8 landing outcomes recorded.

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%%sql SELECT Landing__Outcome, COUNT(*) AS Total FROM SPACEX
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing__Outcome
ORDER BY Total DESC
```

```
 * ibm_db_sa://tpw78977:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain
Done.
```

landing__outcome	total
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

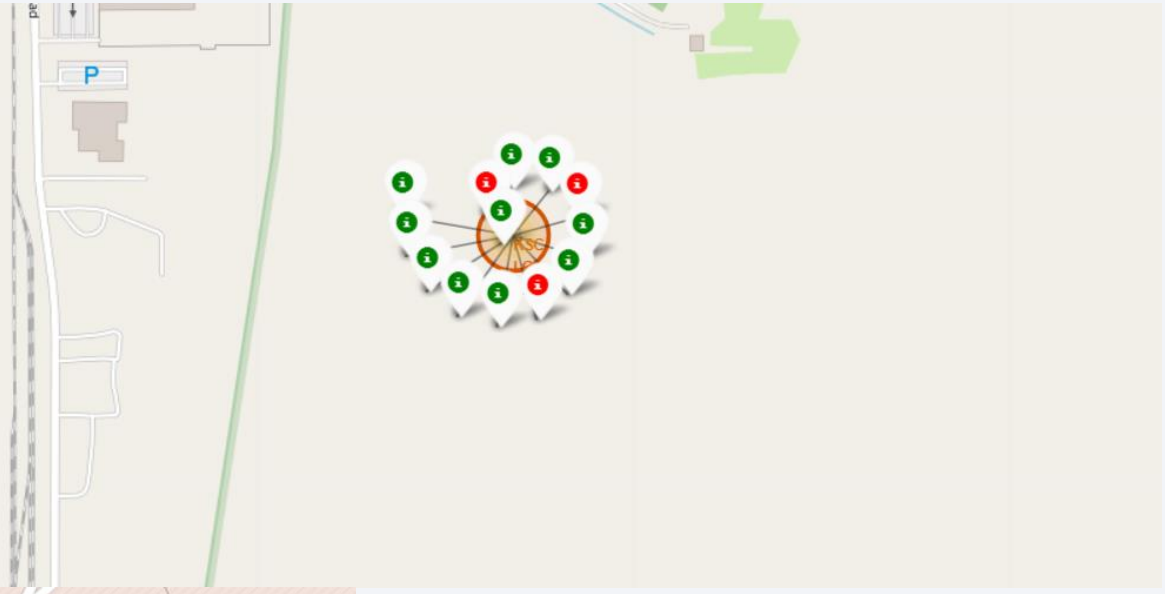
Section 3

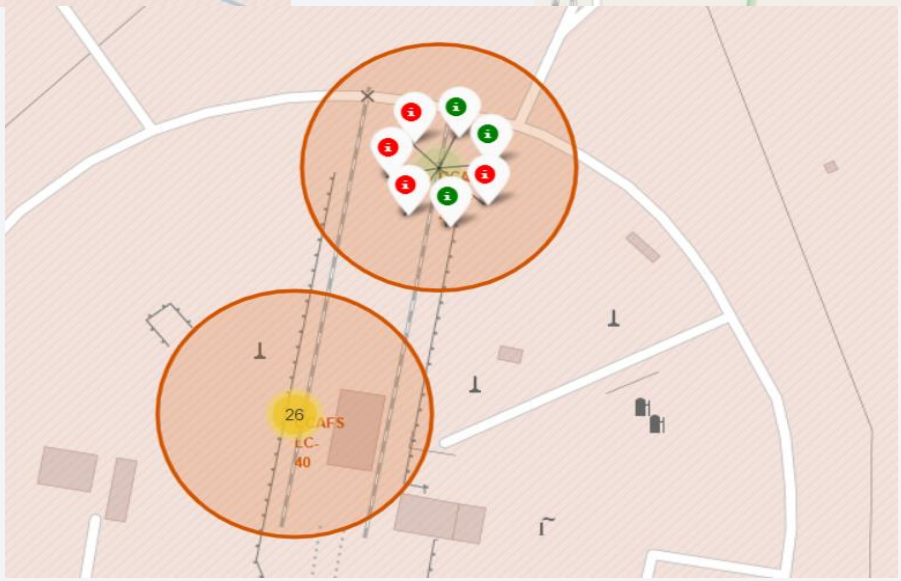
Launch Sites Proximities Analysis

SpaceX Launch Sites

- Launch Sites located on both West Coast and East Coast.
  - California and Florida

Launch Outcomes at Each Site



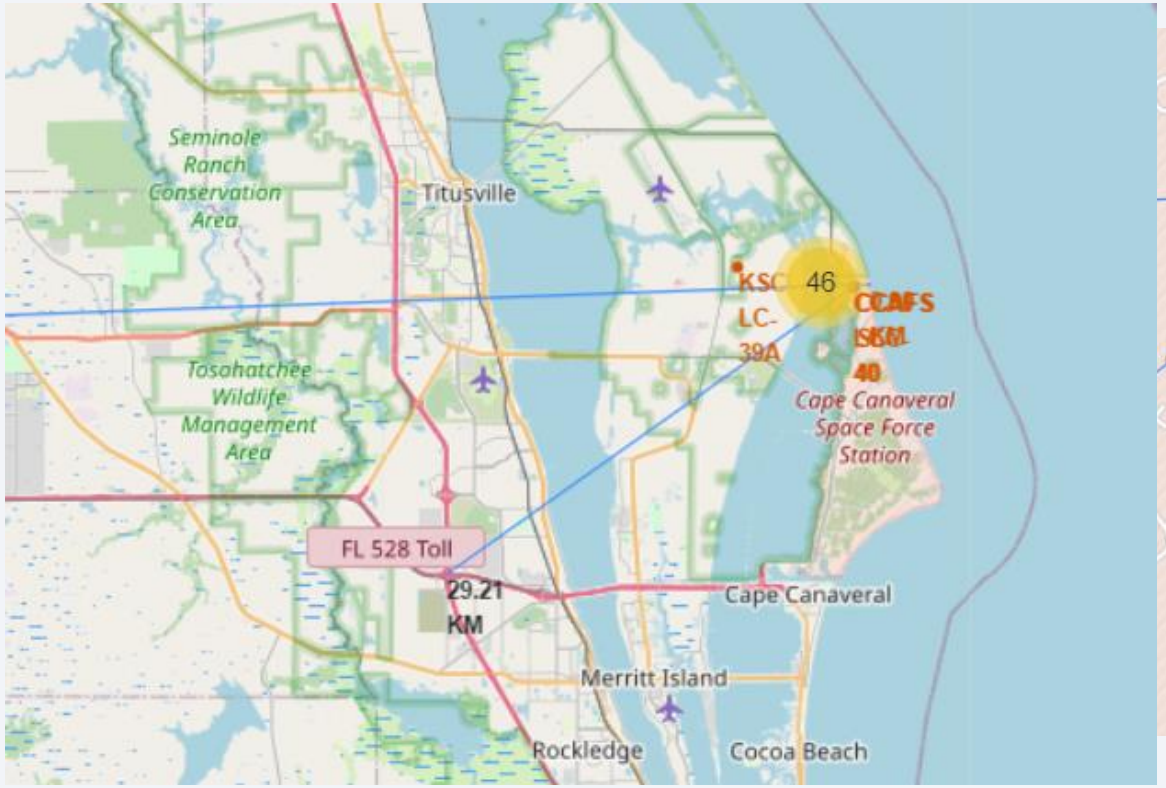


Displays launch outcomes for each site

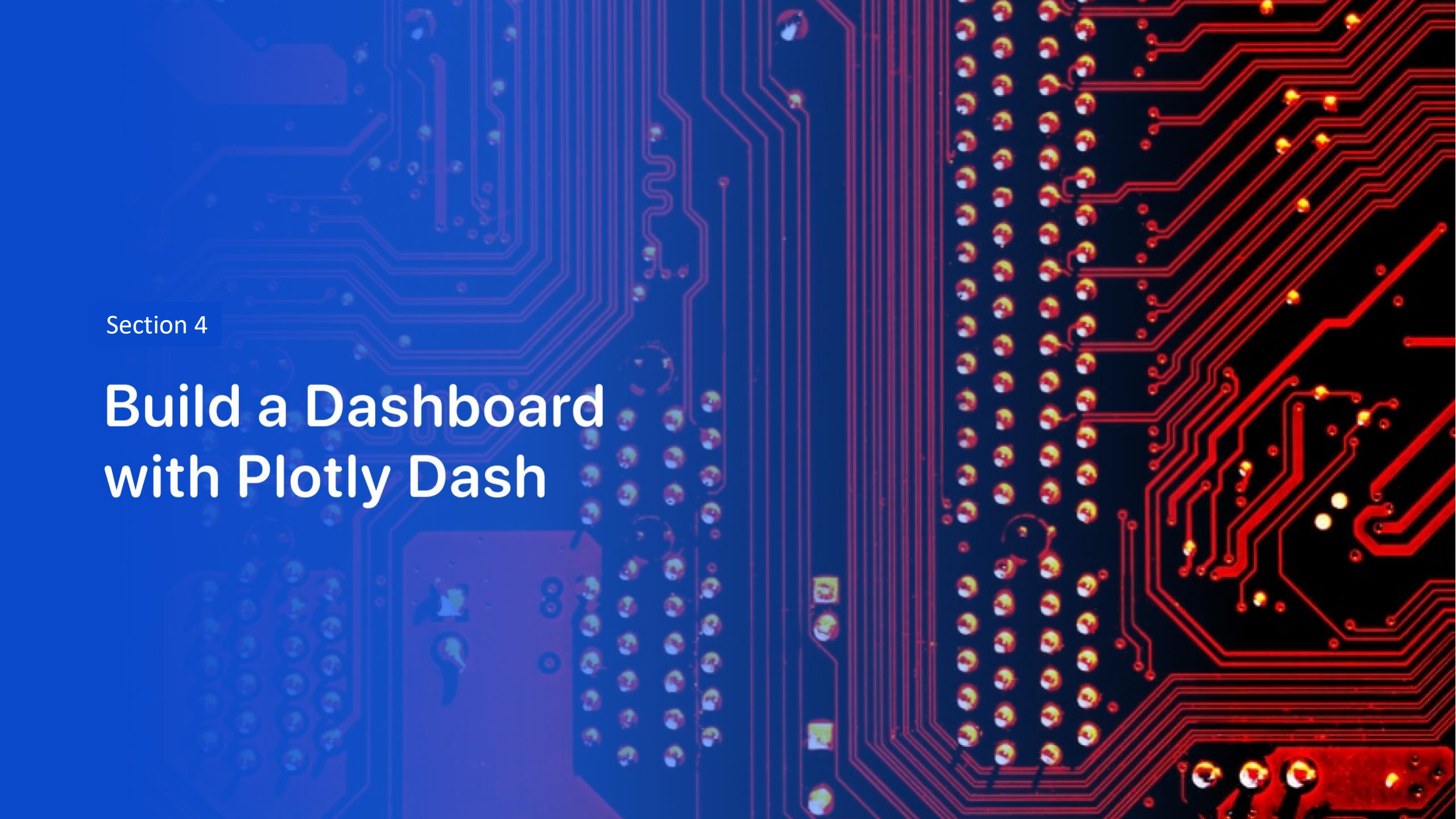
- Green = Success
- Red = Failure

Launch Site Distance from Landmarks

Displays proximity of landmarks (highways, coast, railroads) from launch sites.

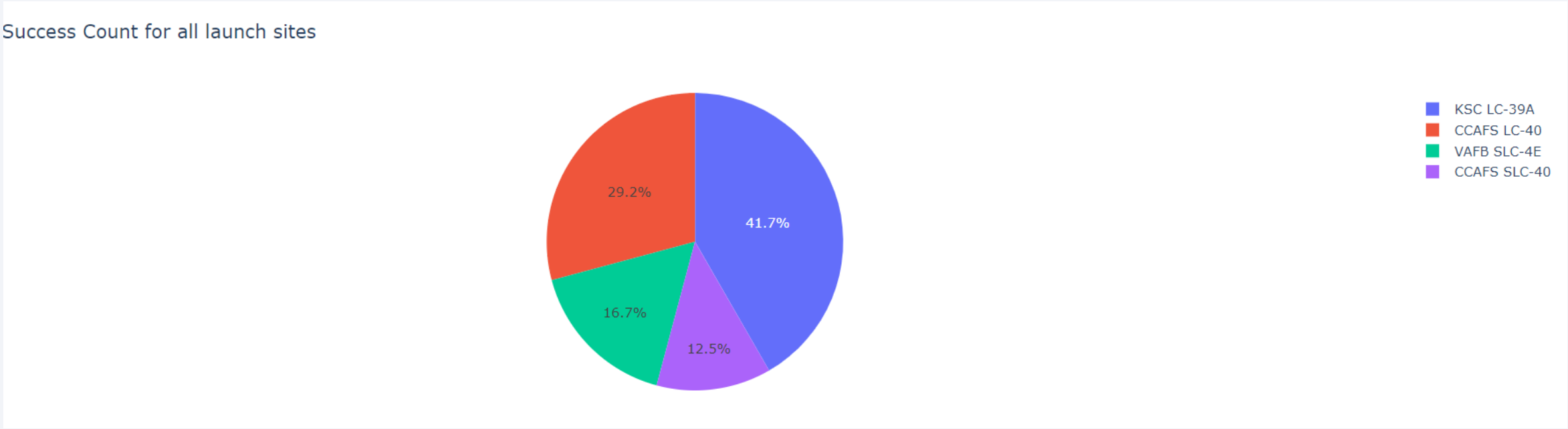


Launch Sites are close to coastlines, far from transportation (highways, railroads)


Section 4
Build a Dashboard with Plotly Dash

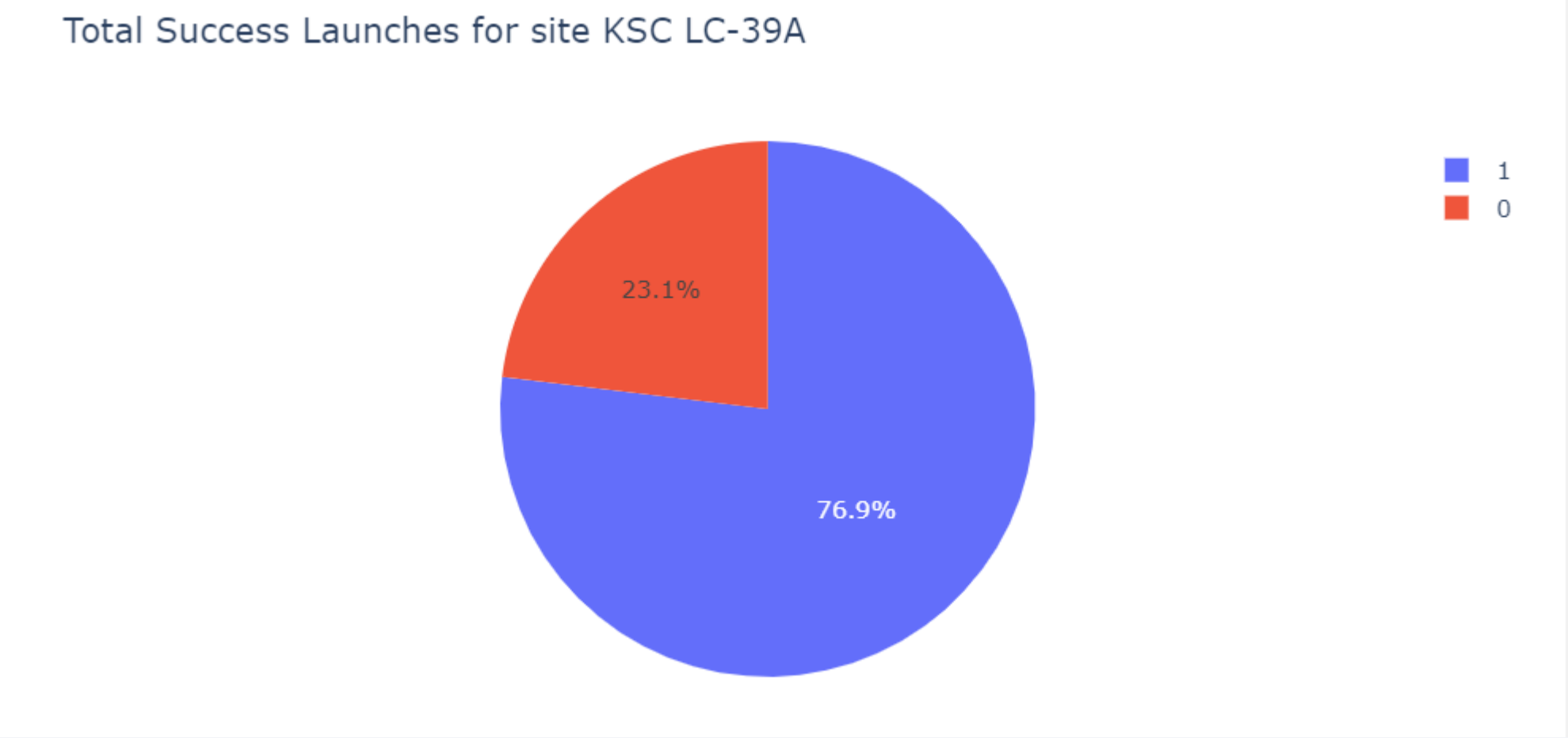
Pie Chart of Launch Success for All Sites

- KSC LC-39A had the most successful launch sites with 41.7%
- CCAFS SLC-40 had the least successful launch sites with 12.5%



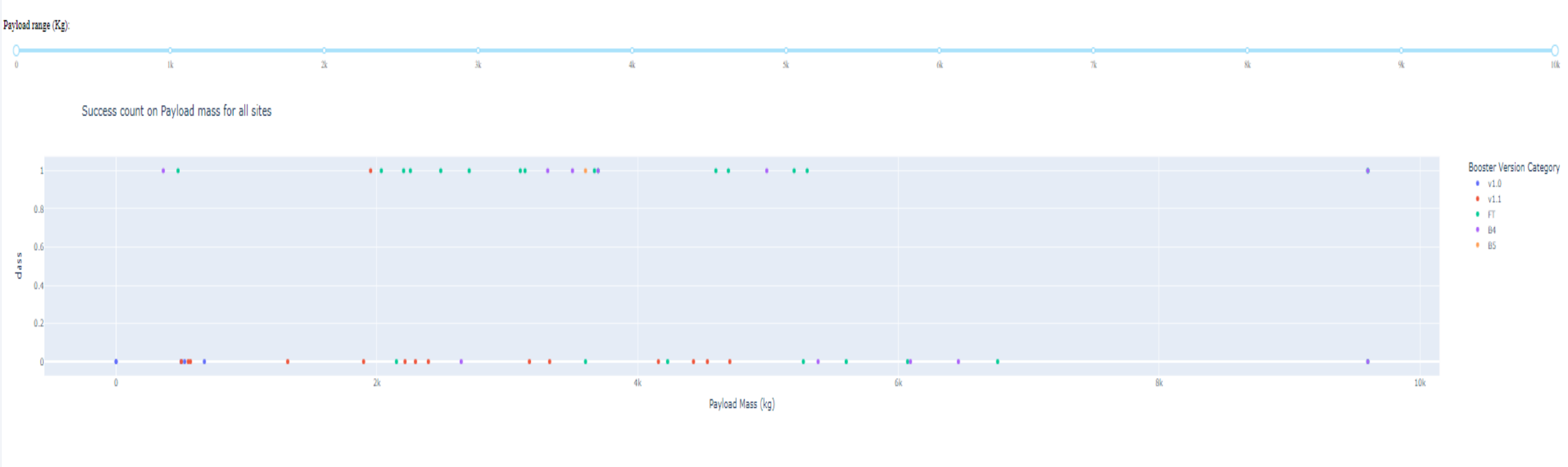
Launch Site with Highest Success Ratio

- Launch site KSC LC-39A had highest launch success rate (76.9%)



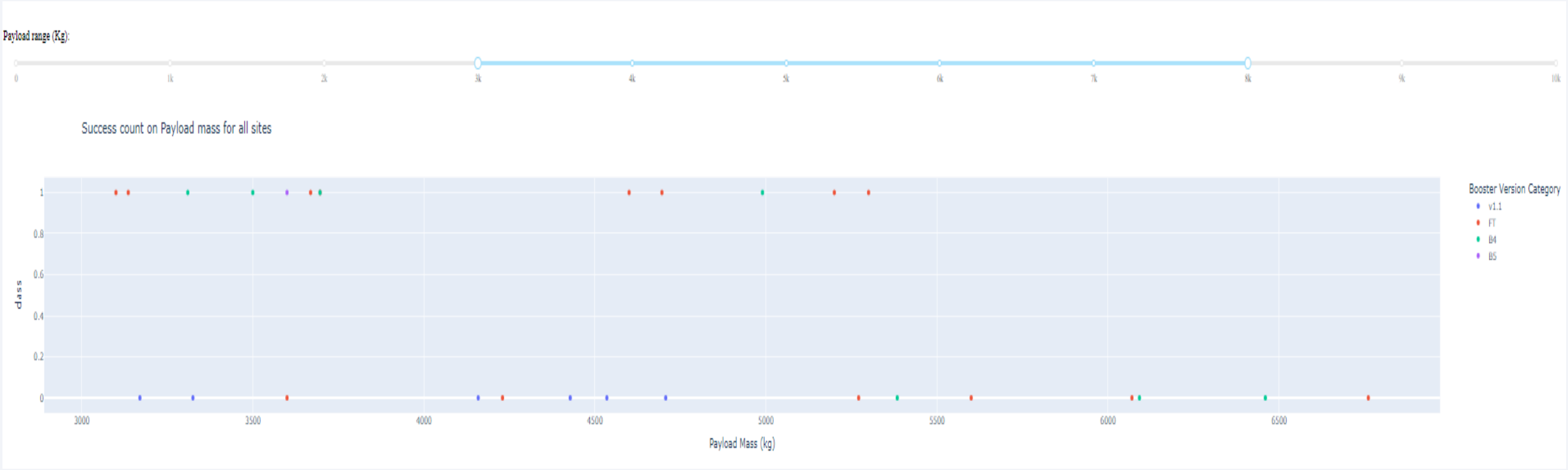
Payload vs. Launch Outcome

- Payload range from 0 kg – 10,000 kg



Payload vs. Launch Outcome

- Payload range from 3,000 kg – 10,000 kg



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Best model was Decision Tree
- Total Accuracy = 87.5%

Find the method performs best:

```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)

print('The best model is: ', bestalgorithm)
```

```
The best model is:  DecisionTree
```

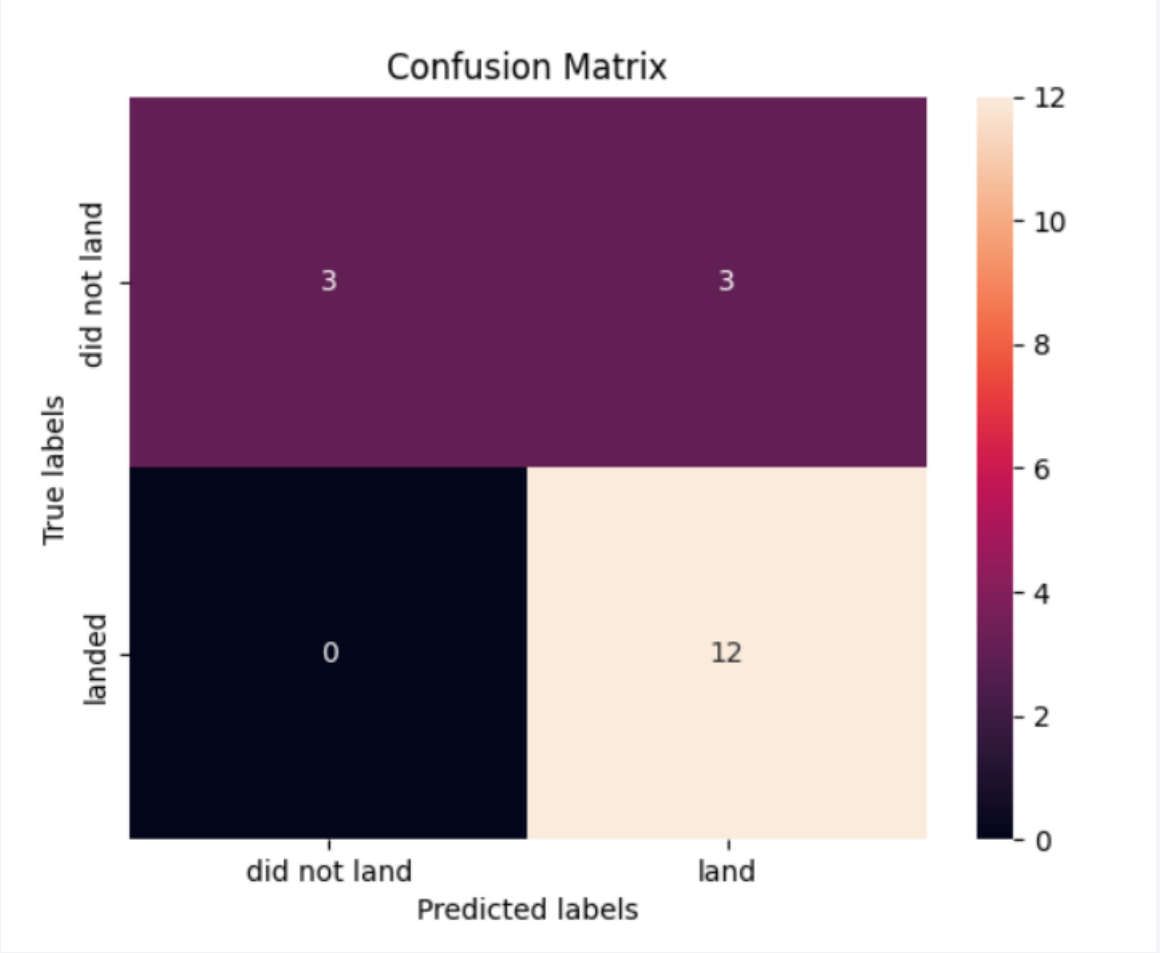
```
tree_gridsearch = GridSearchCV(
    estimator = tree,
    param_grid = parameters,
    cv = 10
)

tree_cv = tree_gridsearch.fit(X_train, Y_train)
```

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

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 8, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split':
10, 'splitter': 'best'}
accuracy : 0.875
```

Confusion Matrix

- Confusion matrix for Decision Tree classifier.
- Model able to predict successful outcomes very well.
- Model struggled predicting failed outcomes.



Conclusions

- The more flights, the higher the success rate.
- Launch success rate increased between 2013 – 2017.
  - Success rate has not significantly increased since 2017.
- Most successful orbits were ES-L1, GEO, HEO, and SSO.
- KSC LC-39A had highest success rate.
- Decision Tree classifier was best machine learning model.

Thank you!