



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Lam Chi Kin  
4<sup>th</sup> Oct 2022



# Outline

---

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

# Executive Summary

---

- Summary of methodologies
- Summary of all results

# Introduction

---

- Space X
  - Advertise Falcon 9 rocket
    - Launch with cost of \$62m
    - Compare with other providers: up to \$165m
    - Mush of saving because of the reusage of first stage
- Determine
  - Successful rate of Falcon 9 first stage will land successfully
    - Find the cost of a launch



Section 1

# Methodology

# Methodology

---

## Executive Summary

- Data collection methodology:
  - Download Falcon 9 and Falcon Heavy Launch data through
    - Space X API
    - Wikipedia
- Perform data wrangling
  - Using BeautifulSoup extract data from HTML
  - Using Pandas
  - Data preprocessing
  - Apply one-hot encoding method to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Apply Scikit-Learn
  - Classification:
    - K-Nearest Neighbors, Decision Tree, Support Vector Machine
    - Use GridSearchCV to find the best parameters
    - Evaluate by train dataset
      - Ratio of train-test size: 8:2

# Data Collection

---

- Request data

- From Space X API

- `requests.get()`
    - -> json file

- From Wikipedia

- BeautifulSoup

- Use of Pandas

- Convert json file from Space X API to dataframe

- `.json_normalize()`

- Convert HTML file from Wiki to dataframe

- Data preprocessing

- Extract necessary columns
    - Check and fill missing values

# Data Collection – SpaceX API

- Space X API
  - requests.get
    - Get json data from api
  - Pandas
    - .json\_normalize()
      - Convert json to Pandas dataframe
- Link:
  - <https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-spacex-data-collection-api.ipynb>

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

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
Check the content of the response
```

```
print(response.content)
```

輸出已擧疊 ...

You should see the response contains massive information about SpaceX launches. Next, let's try to discover some more relevant information for this project.

### Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successful with the 200 status response code

```
response = requests.get(static_json_url)
response.status_code
```

```
200
```

Now we decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json\_normalize()

```
# Use json_normalize meethod to convert the json result into a dataframe
# df = pd.read_json(response.json())
data = pd.json_normalize(requests.get(static_json_url).json())
```



# Data Collection - Scraping

---

- Falcon 9 launch data
  - From Wikipedia
    - BeautifulSoup
    - Parse table & convert to Dataframe
      - With use of Pandas
- Link
  - <https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-webscraping.ipynb>

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)

# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')

# 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)
```

# Data Wrangling

---

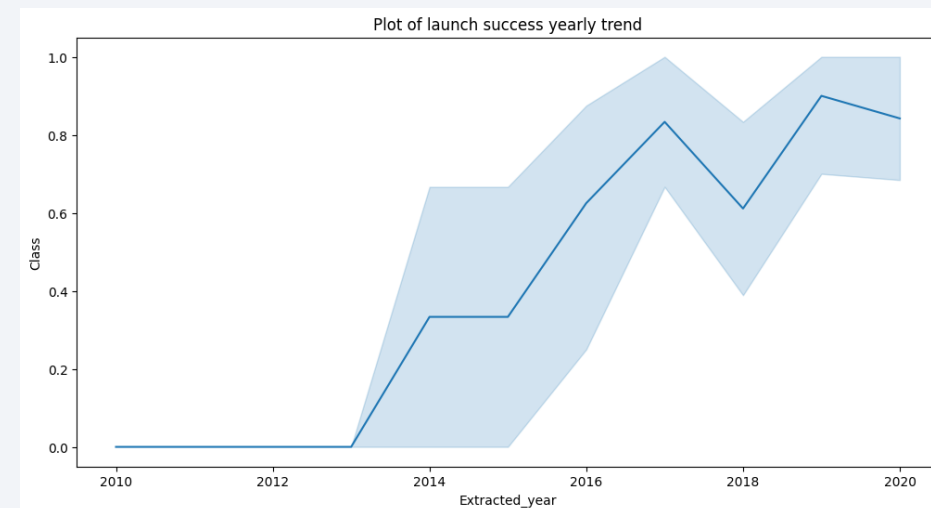
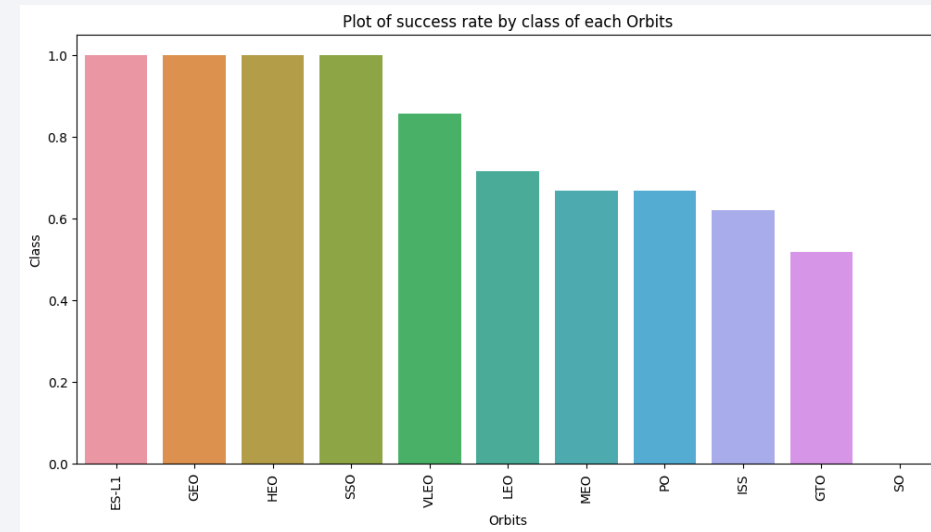
- Exploratory Data Analysis & Determine Training Labels
  - Calculate
    - number of launches on each site
    - number and occurrence of each orbit
    - number and occurrence of mission outcome per orbit type
  - Create landing outcome label
    - From 'Outcome'
- Link:
  - <https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/labs-jupyter-spacex-Data%20wrangling.ipynb>

# EDA with Data Visualization

- Use Matplotlib & Seaborn
  - Visualize
    - Launch sites vs Payload
    - Success rate by class of each Orbits
    - Launch success yearly trend

- Link:

- <https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-dataviz.ipynb>



# EDA with SQL

---

- Load csv file to database
  - SQLite
- Queries
  - Name of Launch site
  - Calculate average payload mass
  - Find 1<sup>st</sup> successful landing date
  - Find number of successful and failure landing outcomes
- Link:
  - [https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera\\_sqlite.ipynb](https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/jupyter-labs-eda-sql-coursera_sqlite.ipynb)

# Build an Interactive Map with Folium

---

- Folium with Map function
  - Circle
    - Mark launch site
  - Marker
    - Recognize success/failed launch for each site
    - Green: success
    - Red: fail
  - Calculate and draw distance
    - Launch site vs its proximities
- Link:
  - [https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/lab\\_jupyter\\_launch\\_site\\_location.ipynb](https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/lab_jupyter_launch_site_location.ipynb)

# Build a Dashboard with Plotly Dash

---

- Pie chart
  - All launch site: successful rate compare to each site
  - Each site: display success and fail rate
- Scatter chart
  - Display payload mass of each booster
  - Range slider
    - For listener/user to limit the range of payload mass
    - Find out the success/fail rate of each booster in specific range
- Link:
  - [https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/spacex\\_dash\\_app.py](https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/spacex_dash_app.py)



# Predictive Analysis (Classification)

---

- Select 'Class' from train data as result label
  - Covert to array with numpy
- Scikit-Learn
  - Standardize: `preprocessing.StandardScaler()`
  - split train/test data: `train_test_split()`
- Search best parameter among classification methods
  - Logistic regression, Support Vector Machine, Decision Tree, kNN
  - With use of GridSearchCV
- Compare different classification methods
  - Find the best method
    - By compare accuracy score
- Link:
  - [https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/SpaceX\\_Machine%20Learning%20Prediction\\_Part\\_5.ipynb](https://github.com/CKLam33/IBM-Data-Science-Professional-Certificate/blob/main/Applied%20Data%20Science%20Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb)

# Results

---

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



The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a complex pattern of diagonal streaks in shades of blue and red on the right. These streaks are layered and have a textured, almost woven appearance. A faint, light blue grid pattern is visible across the entire background, particularly prominent in the blue areas.

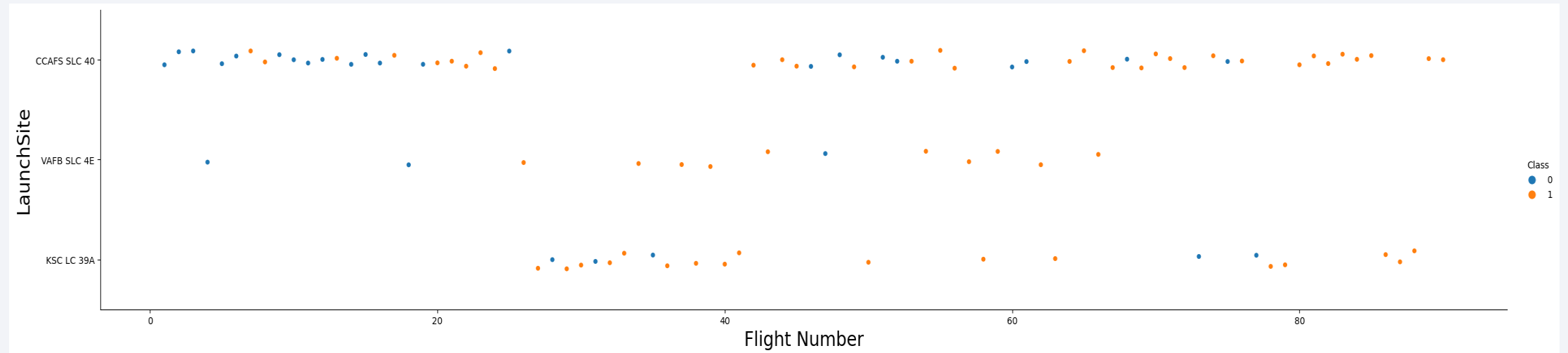
Section 2

# Insights drawn from EDA



# Flight Number vs. Launch Site

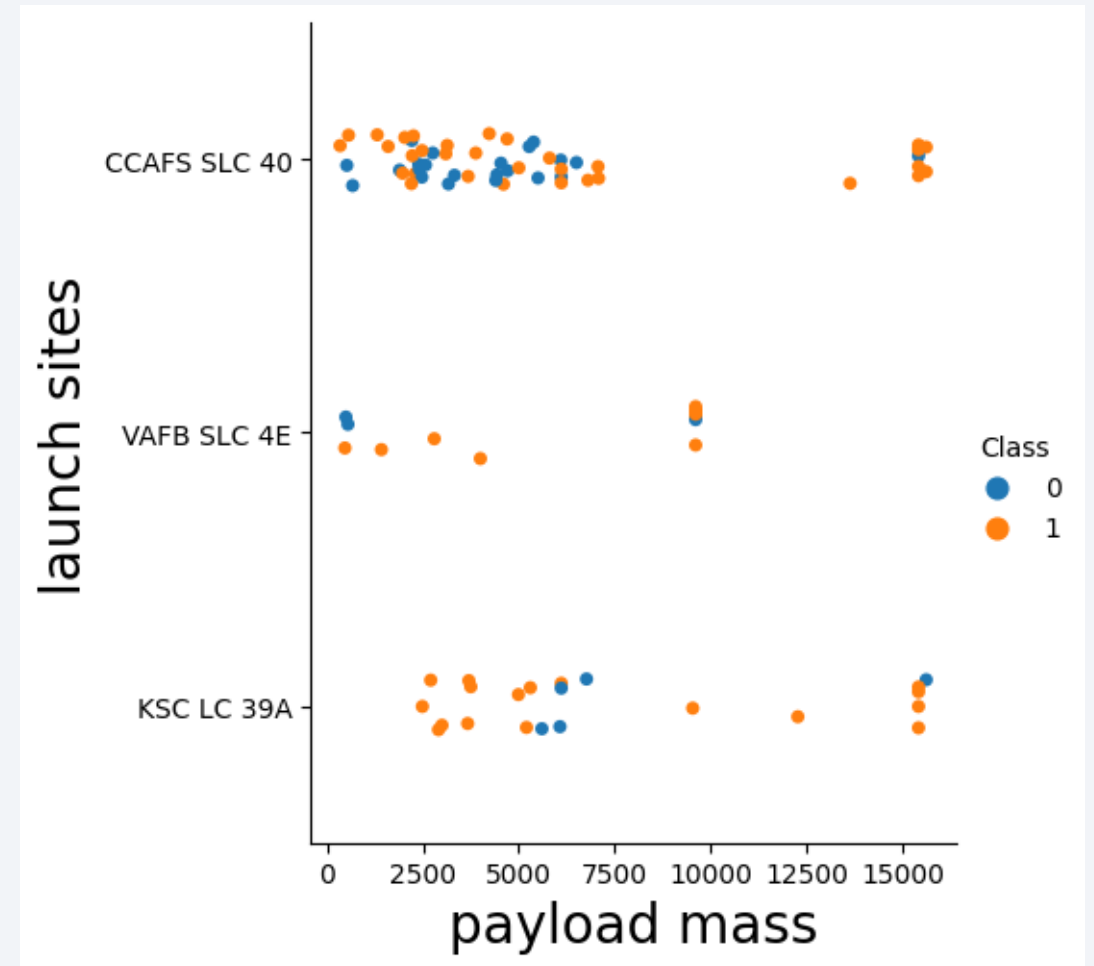
Scatter plot of Flight Number vs. Launch Site



- Most rockets are launched at CCAFS SLC 40
- Number of success outcome increase along the number of flight

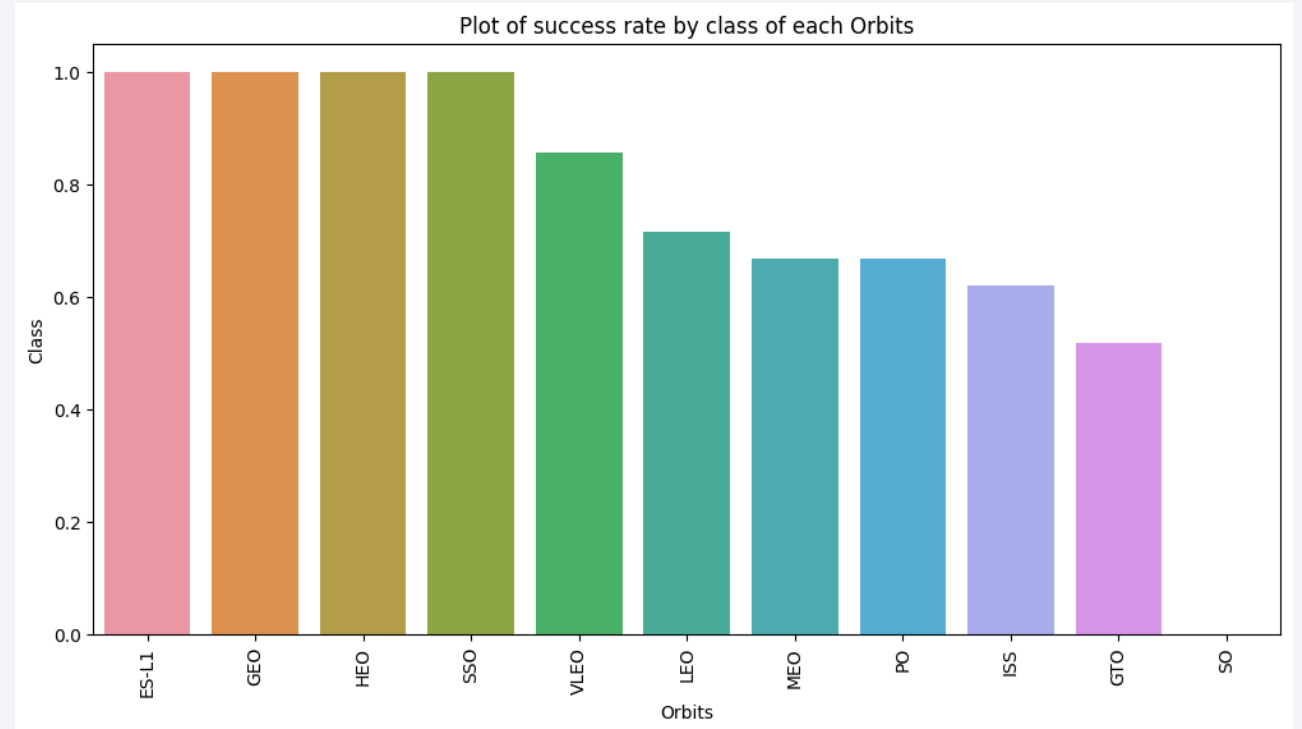
# Payload vs. Launch Site

- Most Payload Mass of flights < 7500
- Few flights with Payload Mass between 7500 and 15000
- Only few flights use VAFB SLC 4E



# Success Rate vs. Orbit Type

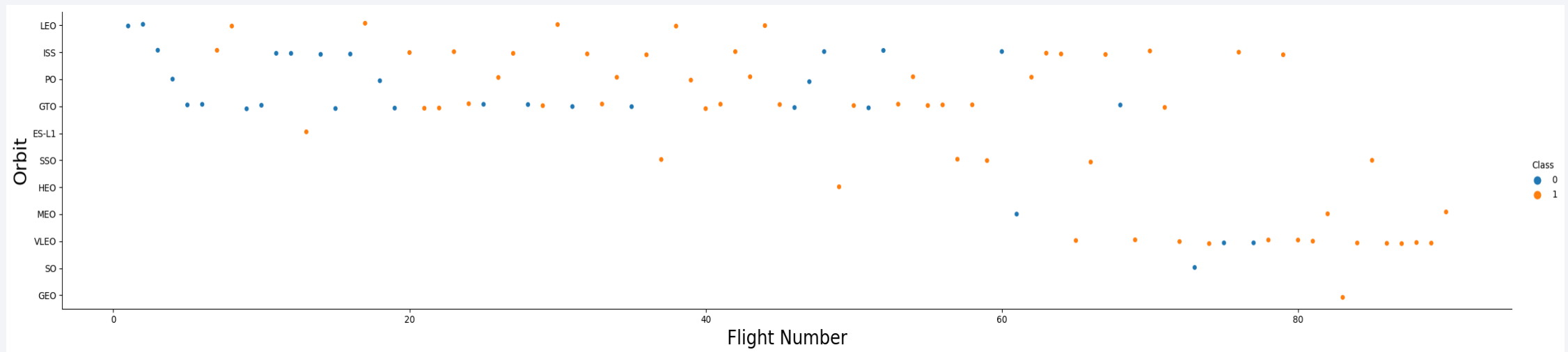
- ES-L1, GEO, HEO, SSO
  - Greatest success rate among orbits
- VLEO
  - > 80% success rate



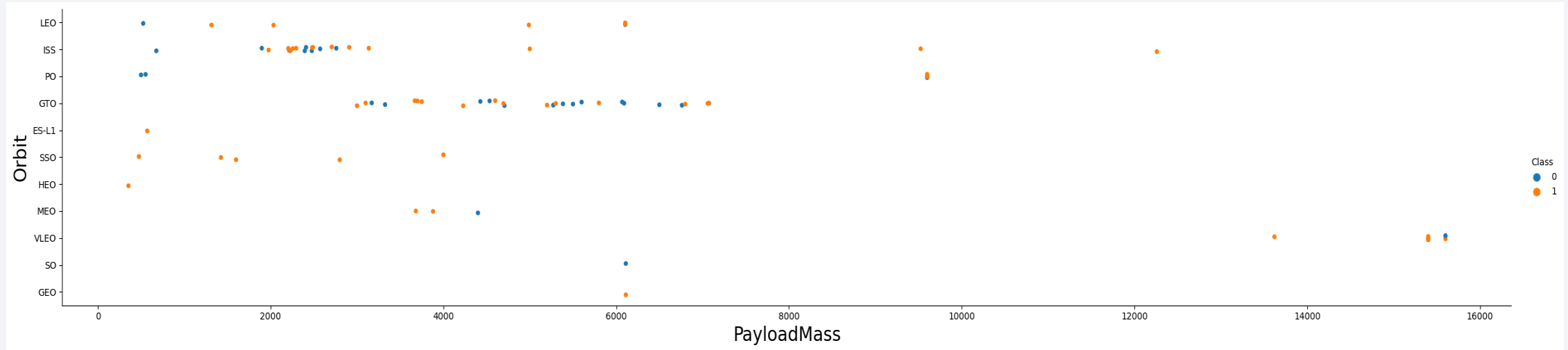


# Flight Number vs. Orbit Type

- LEO, ISS, PO, GTO
  - Most used orbits in early time
- VLEO
  - Start to replace old orbit



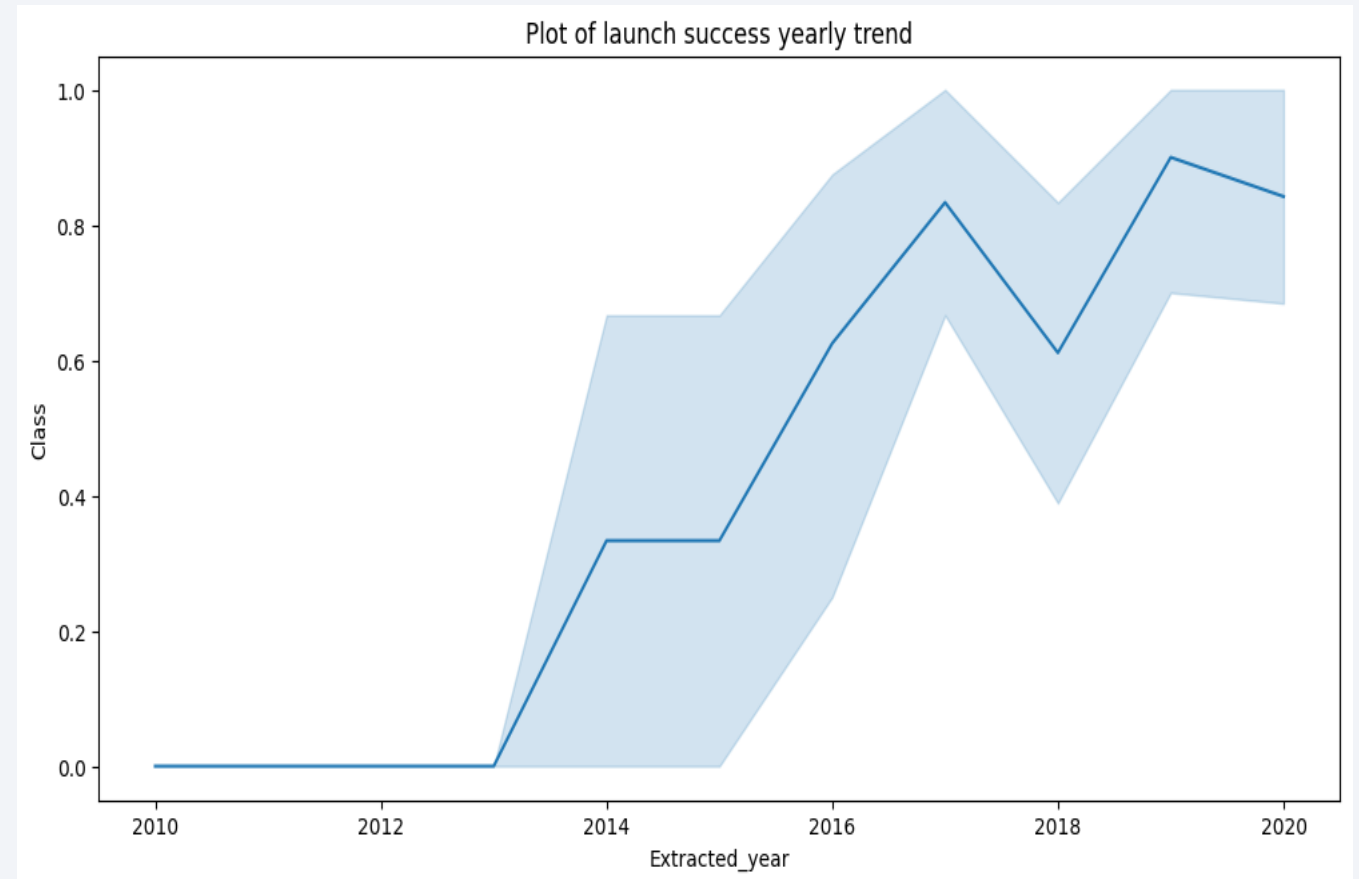
# Payload vs. Orbit Type



- ISS & GTO
  - Mostly for payload mass < 8000
  - GTO -> high fail rate
- VLEO
  - For high payload mass

# Launch Success Yearly Trend

- Overall success rate increase since 2013
  - Significant drop in 2018



# All Launch Site Names

---

Find the names of the unique launch sites

Present your query result with a short explanation here

# Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTBL where Launch_Site Like 'CCA%' limit 5
```

```
* sqlite:///my_data1.db  
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- All mission outcomes were success
- First 2 records were qualification test
  - Fail to land to parachute
- Did not let booster land to any place after first 2 attempt

# Total Payload Mass

---

- Total payload mass from NASA was 45596kg

```
%sql select sum(PAYLOAD_MASS_KG_) from SPACEXTBL where Customer = 'NASA (CRS)'  
  
* sqlite:///my_data1.db  
Done.  
  
sum(PAYLOAD_MASS_KG_)  
45596
```



# Average Payload Mass by F9 v1.1

---

```
%sql select avg(PAYLOAD_MASS_KG) as AvgPayload_Mass from SPACEXTBL where Booster_Version = 'F9 v1.1'

* sqlite:///my_data1.db
Done.

AvgPayload_Mass
2928.4
```

- Average payload mass by F9 v1.1 was 2928.4kg

# First Successful Ground Landing Date

---

```
%sql select MIN(Date) as FirstSuccessfull_landing_date from SPACEXTBL where Landing_Outcome like 'Success (ground pad)'  
  
* sqlite:///my_data1.db  
Done.  
  
FirstSuccessfull_landing_date  
2015-12-22 00:00:00
```

- 1<sup>st</sup> successful landing to ground pad
  - 22<sup>nd</sup> Dec 2015

# Successful Drone Ship Landing with Payload between 4000 and 6000

---

- Boosters that land to drone ship successfully with Payload between 4000 and 6000:

```
%sql select Booster_Version from SPACEXTBL where Landing_Outcome like 'Success (drone ship)' and PAYLOAD_MASS_KG_ > 4000 and PAYLOAD_MASS_KG_ < 6000

* sqlite:///my_data1.db
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

---

- Most missions are success
- Only 1 failed

```
successful = %sql select count(Mission_Outcome) as SuccessfulOutcome from SPACEXTBL where Mission_Outcome like 'Success%'
print(successful)

failure = %sql select count(Mission_Outcome) as FailureOutcome from SPACEXTBL where Mission_Outcome like 'Failure%'
print(failure)
```

```
* sqlite:///my_data1.db
Done.
```

```
+-----+
| SuccessfulOutcome |
+-----+
|          100      |
+-----+
```

```
* sqlite:///my_data1.db
Done.
```

```
+-----+
| FailureOutcome |
+-----+
|           1     |
+-----+
```

# Boosters Carried Maximum Payload

---

- F9 B5 series has the most payload mass with 15600kg

```
tk8 = %sql select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL) order by Booster_Version
tk8
```

```
* sqlite:///my_data1.db
Done.
```

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

# 2015 Launch Records

---

- In year 2015, 2 failure in landing drone ship
  - Both were F9 v1.1 in May

```
tk9 = %sql select Landing_Outcome, Booster_Version, Launch_Site, substr(Date,4,2) as Month from SPACEXTBL where Landing_Outcome Like 'Failure (drone ship)' and Date Like '2015%'
tk9
```

```
* sqlite:///my_data1.db
```

```
Done.
```

Landing_Outcome	Booster_Version	Launch_Site	Month
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	5-
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	5-



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

---

```
tk10 = %sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY COUNT(Landing_Outcome) DESC  
tk10
```

```
* sqlite:///my_data1.db
```

```
Done.
```

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

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a dark blue sky and a view of the Earth's surface, which is covered in a dense network of city lights and clouds. The lights are concentrated in the lower right portion of the image, while the upper left portion shows a clear blue sky.

Section 3

# Launch Sites Proximities Analysis

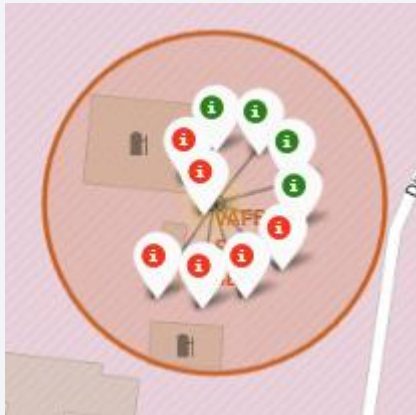
# Launch site of Space X in global map



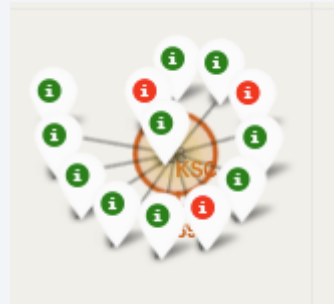
Florida and California in USA are the only  
places Space X to launch their rocket

# Color-labeled launched outcomes on map

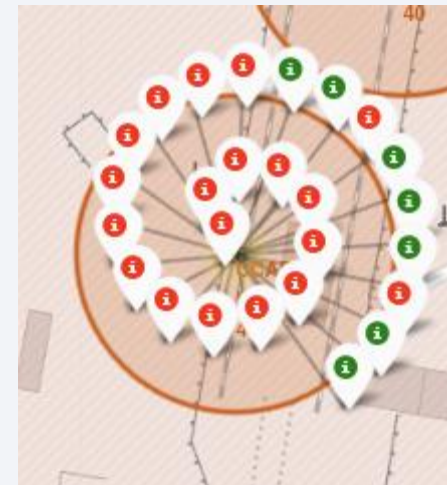
---



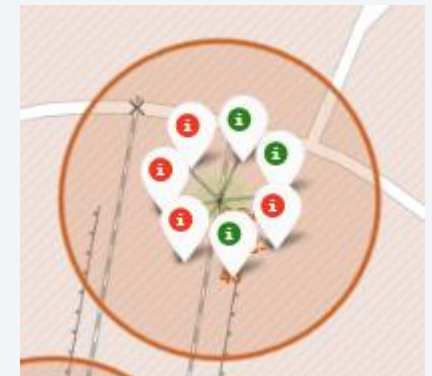
VAFB SLC-4E



KSC LC-39A



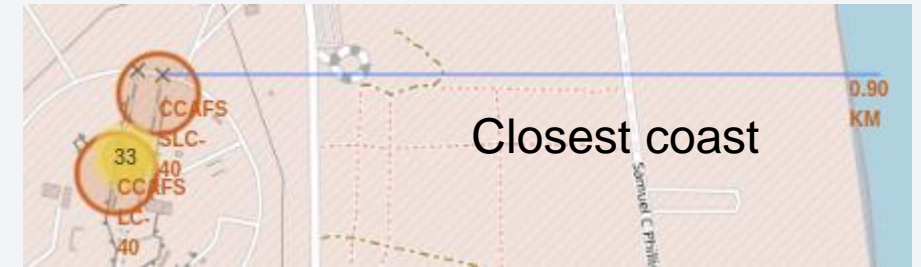
CCAFS LC-40



CCAFS SLC-40



# Launch site distance to landmarks in Florida







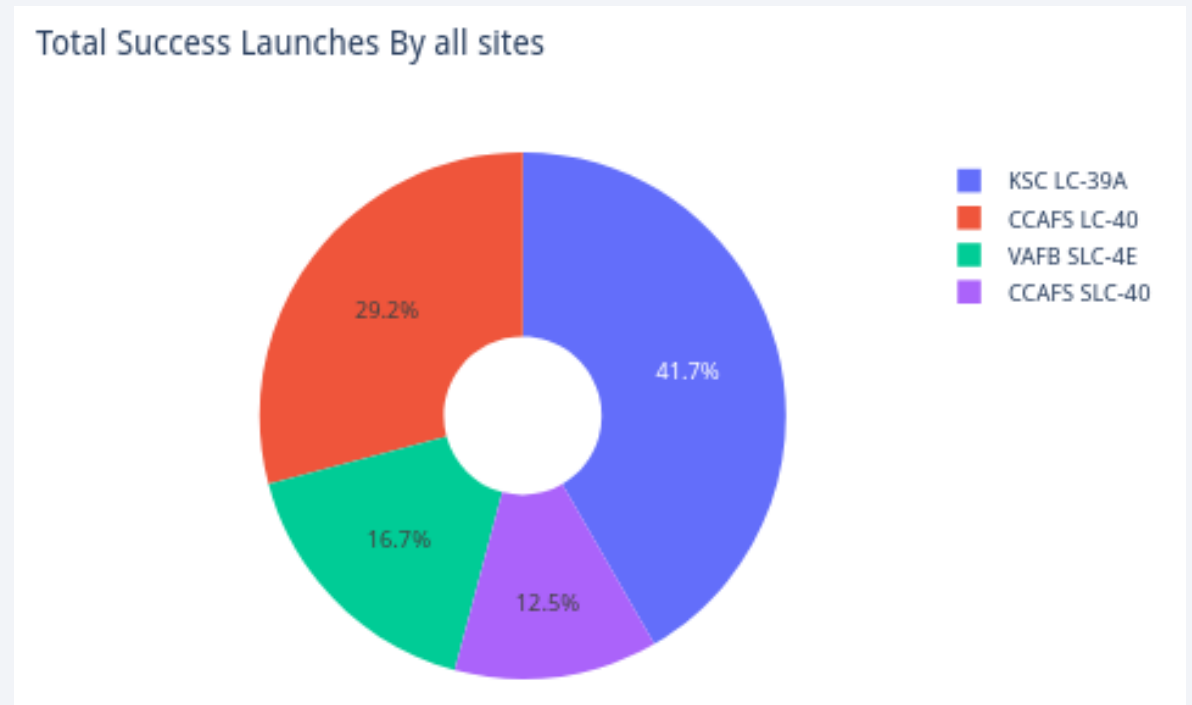
Section 4

# Build a Dashboard with Plotly Dash

# Pie chart of success launches by all sites

---

- KSC LC-39A
  - Has most success launches in all site
- CCAFS SLC-40
  - The least success launches in all site

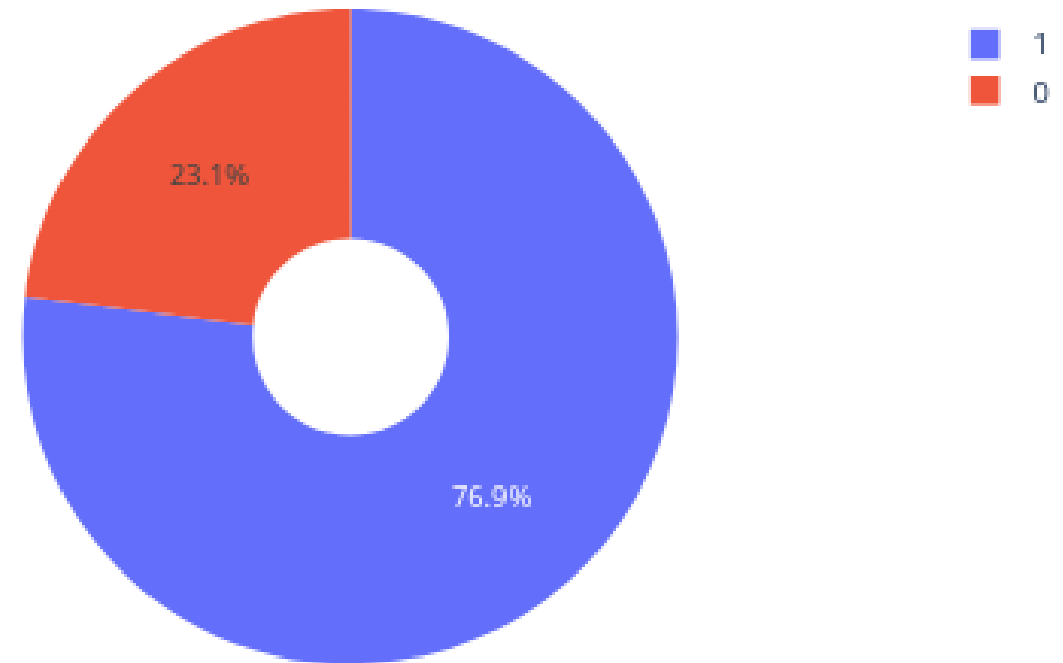


# Pie chart of total success launches in KSC LC-39A

---

- 76.9% of successful rate
- 23.1% of failure rate

Total Success Launches for site KSC LC-39A





# Payload vs Launch Outcome scatter plot for all sites



Payload vs Launch Outcome  
(0 – 5000kg)

Payload vs Launch Outcome  
(5000kg – 10000kg)



Section 5

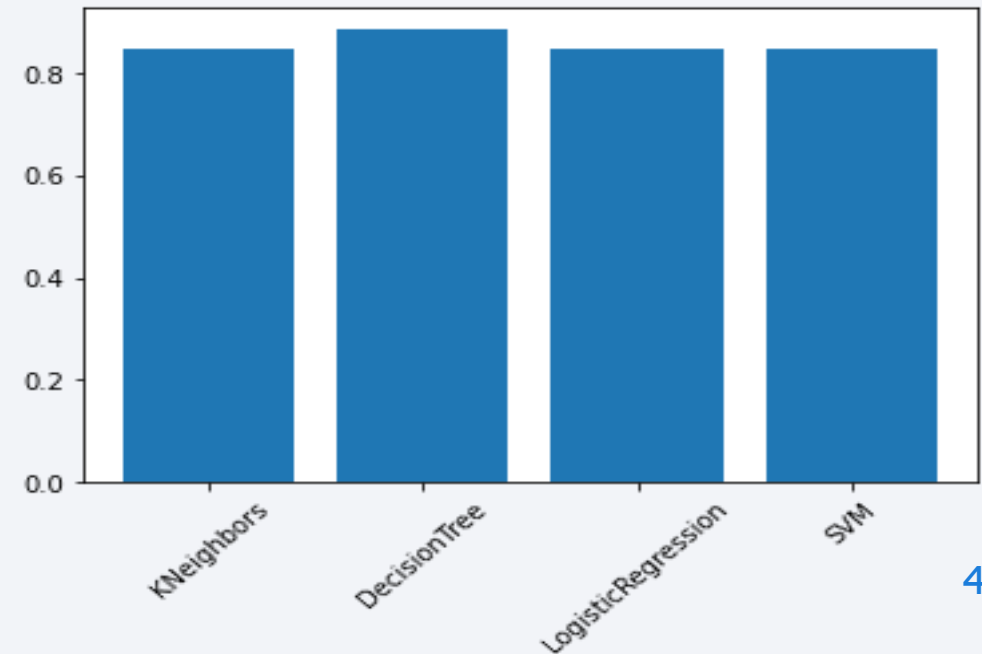
# Predictive Analysis (Classification)

# Classification Accuracy

```
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('Best model is', bestalgorithm, 'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is:', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is:', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is:', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is:', svm_cv.best_params_)

Best model is DecisionTree with a score of 0.9035714285714287
Best params is: {'criterion': 'gini', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random'}
```



# Confusion Matrix

- Confusion Matrix of prediction with Decision Tree
  - Can handle most cases
  - But it has a probability that mark unsuccessful landing as success



# Conclusions

---

- Launch success rate increase since 2013
- CCAFS SLC-40 is the best launch site
- ISS & GTO
  - For payload mass < 8000
- VLEO
  - For higher payload mass
- Decision Tree
  - Best classification method to predict successful launch and landing



Thank you!

