IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jeff Gordon
2/26/2024

Outline

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

Executive Summary

- Classification Methods Tested:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K-Nearest Neighbor
- Classification Results
  - 4 way tie on test: .8333
    - Decision Tree best training performance

Introduction

- Elon wants to optimize landing predictions
  - How do the relationships of each variable impact each other?
  - What is the current and previous success rates?

Section 1

Methodology

Methodology

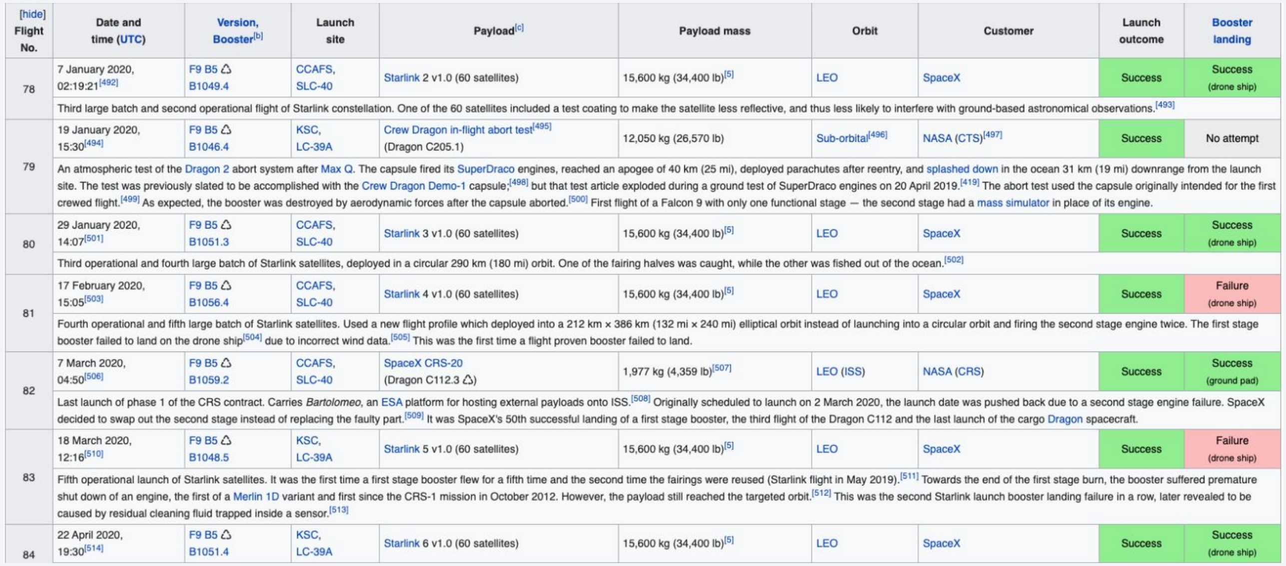
Executive Summary

- Data collection methodology:
  - SPACEX API
  - Wikipedia
- Perform data wrangling
  - Number of Launches at Each Site
  - Number of each Orbit
  - Success Rate by Orbit
  - Create Label for Landing Outcomes
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- SpaceX API

- Wikipedia



[hide] Flight No.	Date and time (UTC)	Version, Booster[b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
78	7 January 2020, 02:19:21[492]	F9 B5 ♺ B1049.4	CCAFS, SLC-40	Starlink 2 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third large batch and second operational flight of Starlink constellation. One of the 60 satellites included a test coating to make the satellite less reflective, and thus less likely to interfere with ground-based astronomical observations.[493]								
79	19 January 2020, 15:30[494]	F9 B5 ♺ B1046.4	KSC, LC-39A	Crew Dragon in-flight abort test[495] (Dragon C205.1)	12,050 kg (26,570 lb)	Sub-orbital[496]	NASA (CTS)[497]	Success	No attempt
	An atmospheric test of the Dragon 2 abort system after Max Q. The capsule fired its SuperDraco engines, reached an apogee of 40 km (25 mi), deployed parachutes after reentry, and splashed down in the ocean 31 km (19 mi) downrange from the launch site. The test was previously slated to be accomplished with the Crew Dragon Demo-1 capsule;[498] but that test article exploded during a ground test of SuperDraco engines on 20 April 2019.[419] The abort test used the capsule originally intended for the first crewed flight.[499] As expected, the booster was destroyed by aerodynamic forces after the capsule aborted.[500] First flight of a Falcon 9 with only one functional stage — the second stage had a mass simulator in place of its engine.								
80	29 January 2020, 14:07[501]	F9 B5 ♺ B1051.3	CCAFS, SLC-40	Starlink 3 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third operational and fourth large batch of Starlink satellites, deployed in a circular 290 km (180 mi) orbit. One of the fairing halves was caught, while the other was fished out of the ocean.[502]								
81	17 February 2020, 15:05[503]	F9 B5 ♺ B1056.4	CCAFS, SLC-40	Starlink 4 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Failure (drone ship)
	Fourth operational and fifth large batch of Starlink satellites. Used a new flight profile which deployed into a 212 km × 386 km (132 mi × 240 mi) elliptical orbit instead of launching into a circular orbit and firing the second stage engine twice. The first stage booster failed to land on the drone ship[504] due to incorrect wind data.[505] This was the first time a flight proven booster failed to land.								
82	7 March 2020, 04:50[506]	F9 B5 ♺ B1059.2	CCAFS, SLC-40	SpaceX CRS-20 (Dragon C112.3 ♺)	1,977 kg (4,359 lb)[507]	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
	Last launch of phase 1 of the CRS contract. Carries *Bartolomeo*, an ESA platform for hosting external payloads onto ISS.[508] Originally scheduled to launch on 2 March 2020, the launch date was pushed back due to a second stage engine failure. SpaceX decided to swap out the second stage instead of replacing the faulty part.[509] It was SpaceX's 50th successful landing of a first stage booster, the third flight of the Dragon C112 and the last launch of the cargo Dragon spacecraft.								
83	18 March 2020, 12:16[510]	F9 B5 ♺ B1048.5	KSC, LC-39A	Starlink 5 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Failure (drone ship)
	Fifth operational launch of Starlink satellites. It was the first time a first stage booster flew for a fifth time and the second time the fairings were reused (Starlink flight in May 2019).[511] Towards the end of the first stage burn, the booster suffered premature shut down of an engine, the first of a Merlin 1D variant and first since the CRS-1 mission in October 2012. However, the payload still reached the targeted orbit.[512] This was the second Starlink launch booster landing failure in a row, later revealed to be caused by residual cleaning fluid trapped inside a sensor.[513]								
84	22 April 2020, 19:30[514]	F9 B5 ♺ B1051.4	KSC, LC-39A	Starlink 6 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)

Data Collection – SpaceX API

- Collect:
  - Booster Version
  - Launch Site
  - Payload Data
  - Core Data
- Github Link:
  - https://github.com/jeffgordonn/gordonrepository/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

```python
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
       if x:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

From the `launchpad` we would like to know the name of the launch site being used, the logitude, and the latitude.

```python
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
       if x:
         response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
         Longitude.append(response['longitude'])
         Latitude.append(response['latitude'])
         LaunchSite.append(response['name'])
```

From the `payload` we would like to learn the mass of the payload and the orbit that it is going to.

```python
# Takes the dataset and uses the payloads column to call the API and append the data to the lists
def getPayloadData(data):
    for load in data['payloads']:
       if load:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

From `cores` we would like to learn the outcome of the landing, the type of the landing, number of flights with that core, whether gridfins were used, wheter the core is reused, wheter legs were used, the landing pad used, the block of the core which is a number used to seperate version of cores, the number of times this specific core has been reused, and the serial of the core.

```python
# 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'])
```

Data Collection - Scraping

- Collect:
  - Date Time
  - Booster Version
  - Landing Status
  - Get the Payload Mass
  - Extract Column Name from Header
- Github Link:
  - https://github.com/jeffgordonn/gordonrepository/blob/main/jupyter-labs-webscraping.ipynb

```python
def date_time(table_cells):
    """
    This function returns the data and time from the HTML  table cell
    Input: the  element of a table data cell extracts extra row
    """
    return [data_time.strip() for data_time in list(table_cells.strings)][0:2]

def booster_version(table_cells):
    """
    This function returns the booster version from the HTML  table cell
    Input: the  element of a table data cell extracts extra row
    """
    out=''.join([booster_version for i,booster_version in enumerate( table_cells.strings) if i%2==0][0:-1])
    return out

def landing_status(table_cells):
    """
    This function returns the landing status from the HTML table cell
    Input: the  element of a table data cell extracts extra row
    """
    out=[i for i in table_cells.strings][0]
    return out


def get_mass(table_cells):
    mass=unicodedata.normalize("NFKD", table_cells.text).strip()
    if mass:
        mass.find("kg")
        new_mass=mass[0:mass.find("kg")+2]
    else:
        new_mass=0
    return new_mass


def extract_column_from_header(row):
    """
    This function returns the landing status from the HTML table cell
    Input: the  element of a table data cell extracts extra row
    """
    if (row.br):
        row.br.extract()
    if row.a:
        row.a.extract()
    if row.sup:
        row.sup.extract()

    colunm_name = ' '.join(row.contents)

    # Filter the digit and empty names
    if not(colunm_name.strip().isdigit()):
        colunm_name = colunm_name.strip()
        return colunm_name
```

Data Wrangling

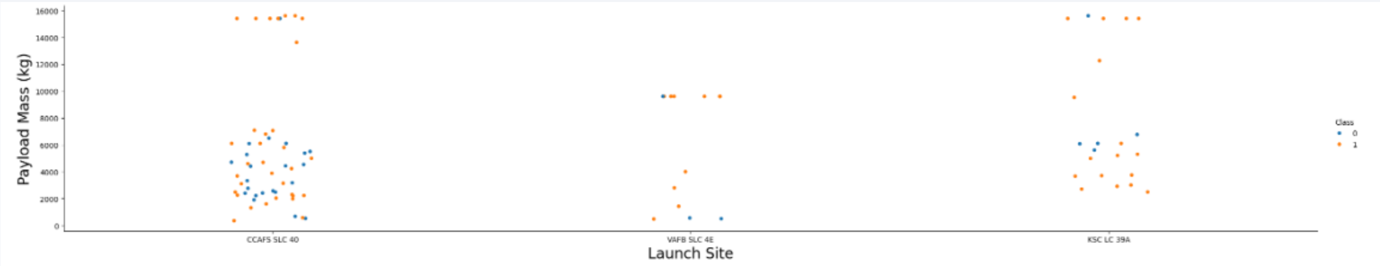
- Data Wrangling Performed:
  - Calculate the Number of Launches by Site
  - Calculate the Number and Occurrence of each Orbit
  - Calculate the Number and Occurrence of Mission Outcomes by Orbit
  - Create a Landing Outcome Label from Outcome column

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0

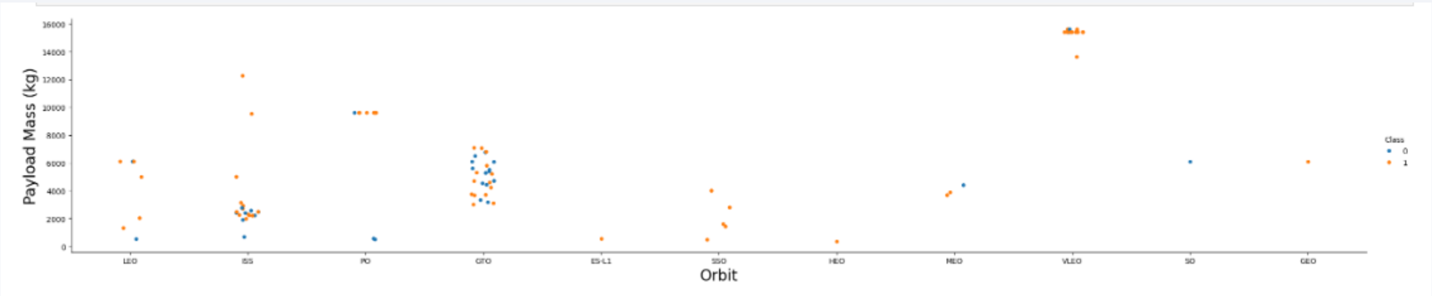
https://github.com/jeffgordonn/gordonrepository/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Payload vs Launch Site

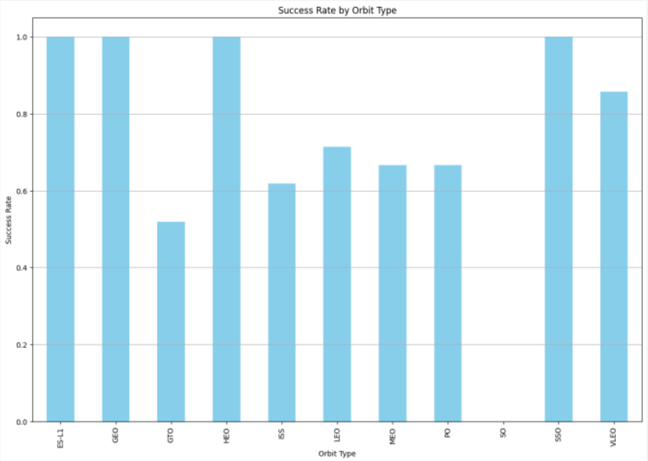


- Payload vs Orbit

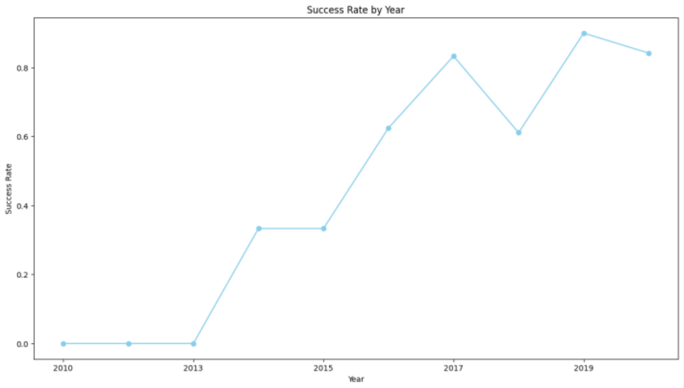


https://github.com/jeffgordonn/gordonrepository/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

- Success Rate by Orbit



- Success Rate by Year



EDA with SQL

- Performed an indepth look into the data e.g.,
  - Outcomes Designations between June 2010 and March 2017
  - Payload Mass of NASA (CRS) flights
  - The names of all launch sites

https://github.com/jeffgordonn/gordonrepository/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

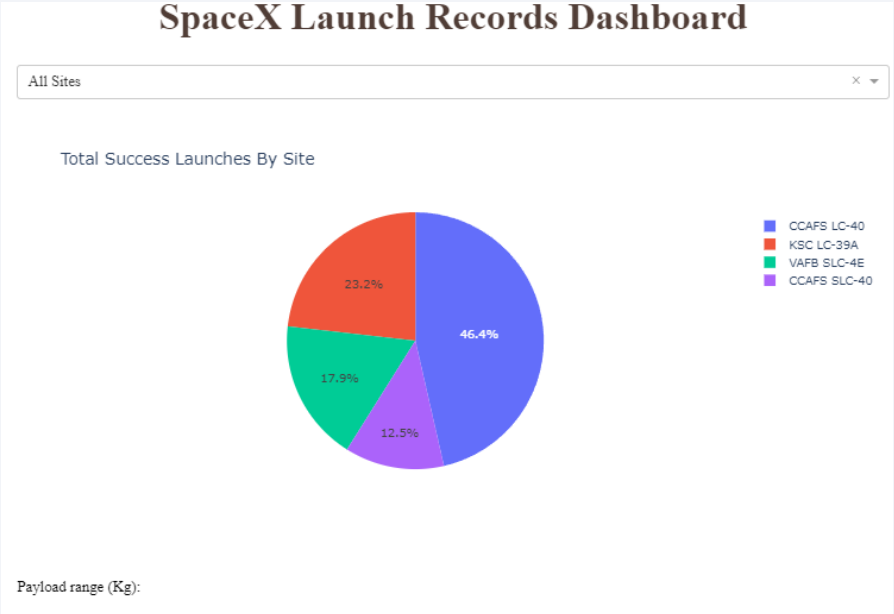
Build an Interactive Map with Folium

- Created Map Markers for:
  - Launch Sites
  - Each Launch and their outcome
  - Distances between objects
- These are critical geographic consideration points

- https://github.com/jeffgordonn/gordonrepository/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Dash
  - Pie Chart
    - Success Rate of Each Site
    - Success Rate Total
  - Slider
    - Payload Range Successes by Site
- Great insight into critical variable selections



- https://github.com/jeffgordonn/gordonrepository/blob/main/jupyter-spacex_dash.ipynb

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results

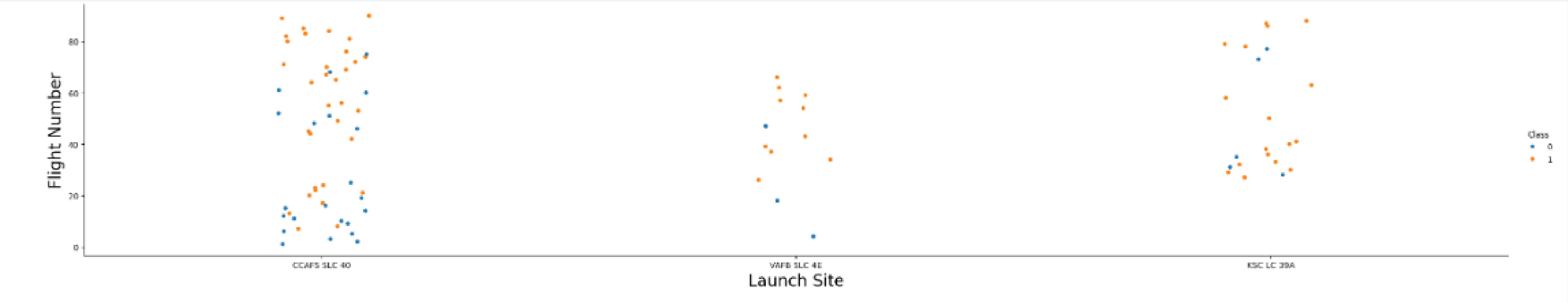
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

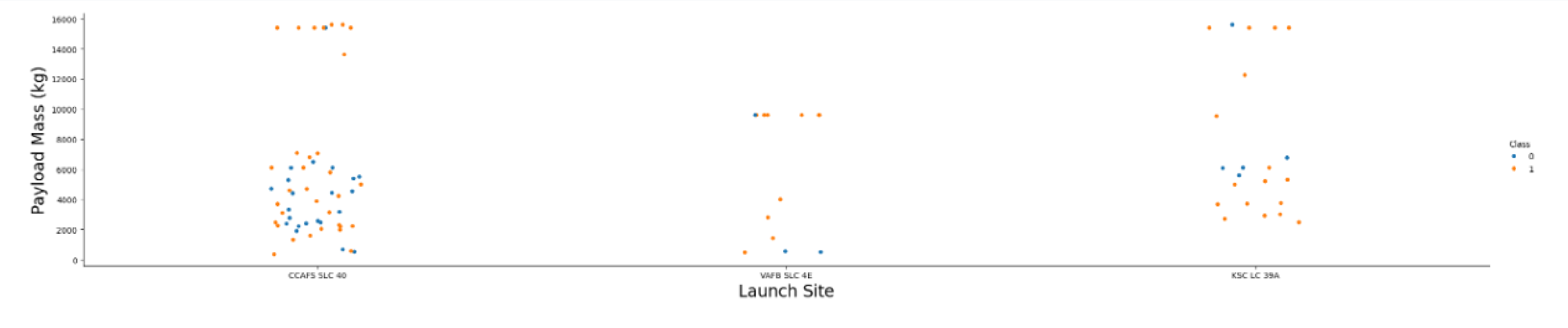
Flight Number vs. Launch Site

- Success has been more prevalent in recent flights
  - Especially CCAFS SLC 40 launches
- VAFB SLC 4E no recent launches



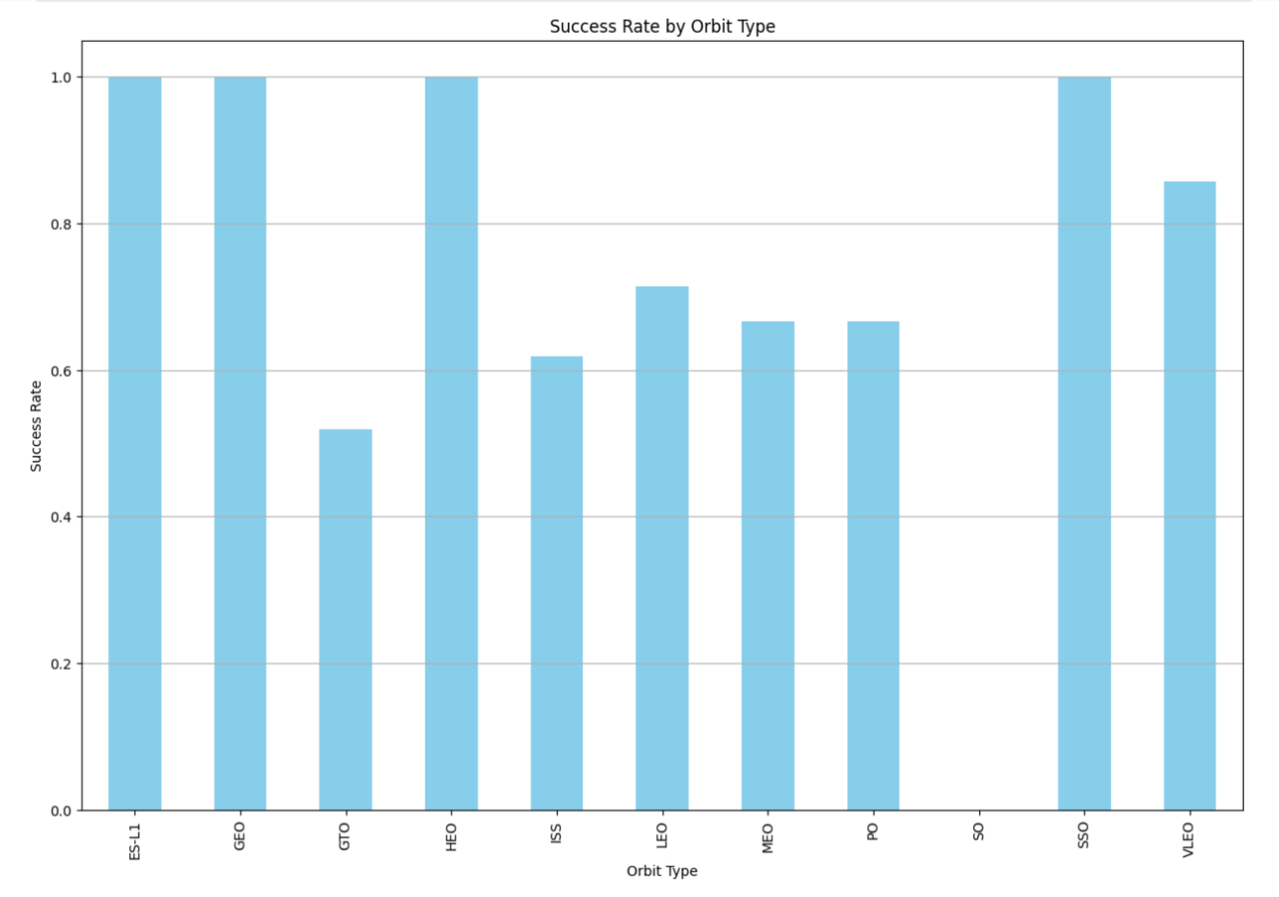
Payload vs. Launch Site

- VAFB SLC 4E does not handle top end Payload Mass
  - Highly successful around 10,000 kg
- KSC LC 39A has best success rates beneath 8000 kg
- CCAFS SLC 40: handles large volume of heavy payloads well



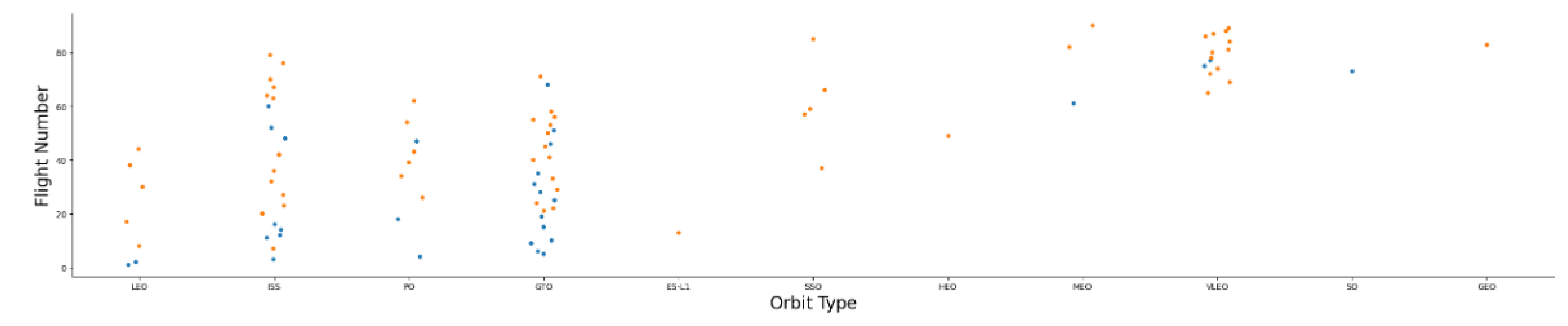
Success Rate vs. Orbit Type

- Perfect Success Rates:
  - ES-L1, GEO, HEO, SSO
- VLEO next best
- SO, GTO worst



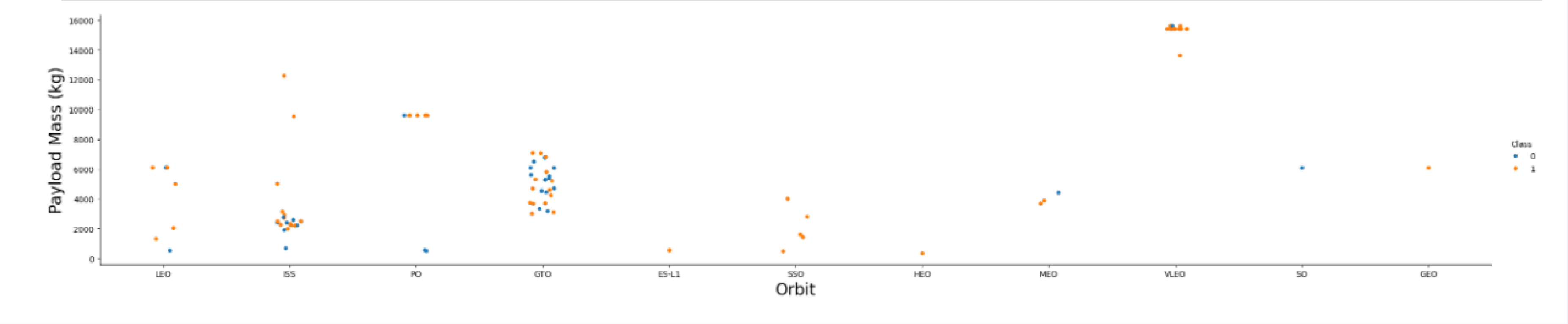
Flight Number vs. Orbit Type

- SO, GEO, ES-L1, HEO low volume of flights
- GTO highest volume, most amount of variance in results
- VLEO highest number of flights recently
  - High success despite large volume
- SSO highest volume with perfect success rate



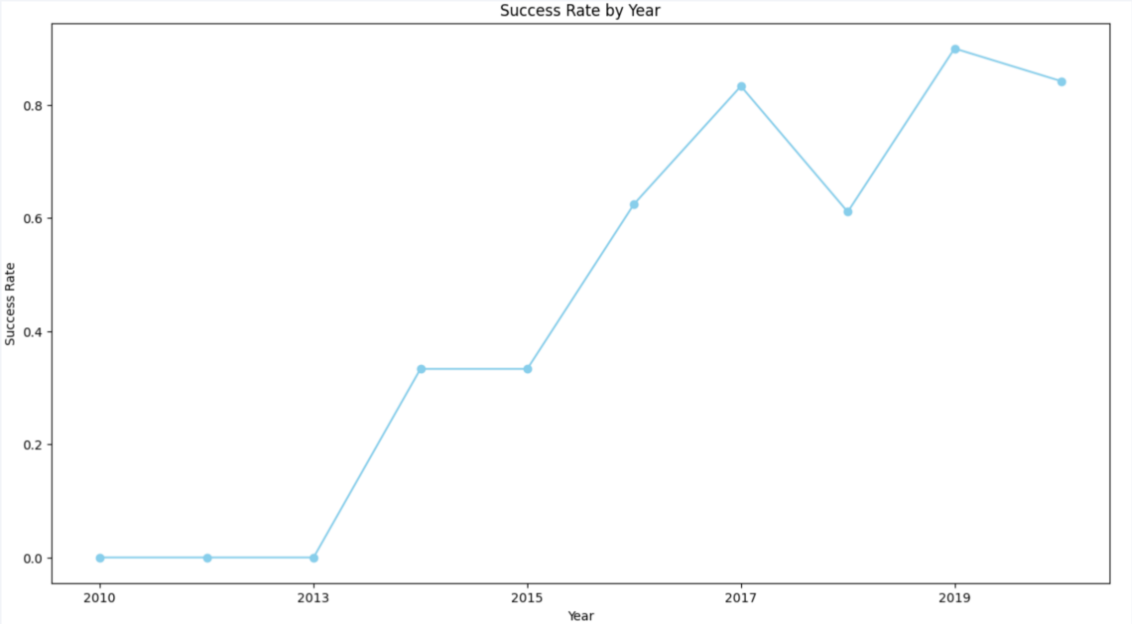
Payload vs. Orbit Type

- Higher end payload mass correlate to successful outcomes
  - Mid ranges have much higher variance
- VLEO: High amounts of success at top end payload mass



Launch Success Yearly Trend

- Success has risen generally each coming year



All Launch Site Names

- Created a list of all the launch sites
    - Distinct creates unique name labels showing the exhaustive list of launch sites for SpaceX

```
%sql SELECT DISTINCT(Launch_Site) FROM SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

- Found all launch sites which text begin with "CCA" string
  - Only presented 5

```
%%sql SELECT * FROM SPACEXTABLE
   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
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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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) has carried 45,596 kg in their SpaceX tracked flights

```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) as NASA_CRS_Total_Payload FROM SPACEXTABLE
    WHERE Customer='NASA (CRS)'
```

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

NASA_CRS_Total_Payload
45596

Average Payload Mass by F9 v1.1

- F9 v1.1 rockets have an average payload of 2928.4

```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) as F9v1_1__AVG_PAYLOAD FROM SPACEXTABLE
    WHERE Booster_Version = 'F9 v1.1'
```

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

F9v1_1__AVG_PAYLOAD
2928.4

First Successful Ground Landing Date

- Min selects the first date, where locks in on ground success

```
%%sql SELECT MIN(Date) as First_Ground_Success FROM SPACEXTABLE
    WHERE Landing_Outcome ='Success (ground pad)'
```

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

First_Ground_Success
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Large variety of successful drone ship landings between 4000 and 6000 kg

```
%%sql SELECT Booster_Version FROM SPACEXTABLE
    WHERE 4000 <= PAYLOAD_MASS__KG_ < 6000
    AND Landing_Outcome = 'Success (drone ship)'
```

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

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

- High number of successes on overall mission objectives
  - Outcomes were grouped and then totaled

```
%%sql SELECT Mission_Outcome, COUNT(*) as Total FROM SPACEXTABLE
   GROUP BY Mission_Outcome
```

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

Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Only F9 B5 versions have carried max payload

```
%%sql SELECT Booster_Version FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_ = (
        SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
                 )
```

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

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

2015 Launch Records

- Small amount of flights
  - No successes

```
%%sql SELECT substr(Date,6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE
    WHERE Landing_Outcome = 'Failure (drone ship)'
    AND substr(Date,0,5)='2015'
```

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

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

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

- Success is most common
  - And its variations
- No attempt is second most common
- Few failures (total)

```
%%sql SELECT Landing_Outcome, COUNT(*) as Total FROM SPACEXTABLE
    WHERE '2010-06-04' <= Date <= '2017-03-20'
    GROUP BY Landing_Outcome
    ORDER BY Total DESC
```

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

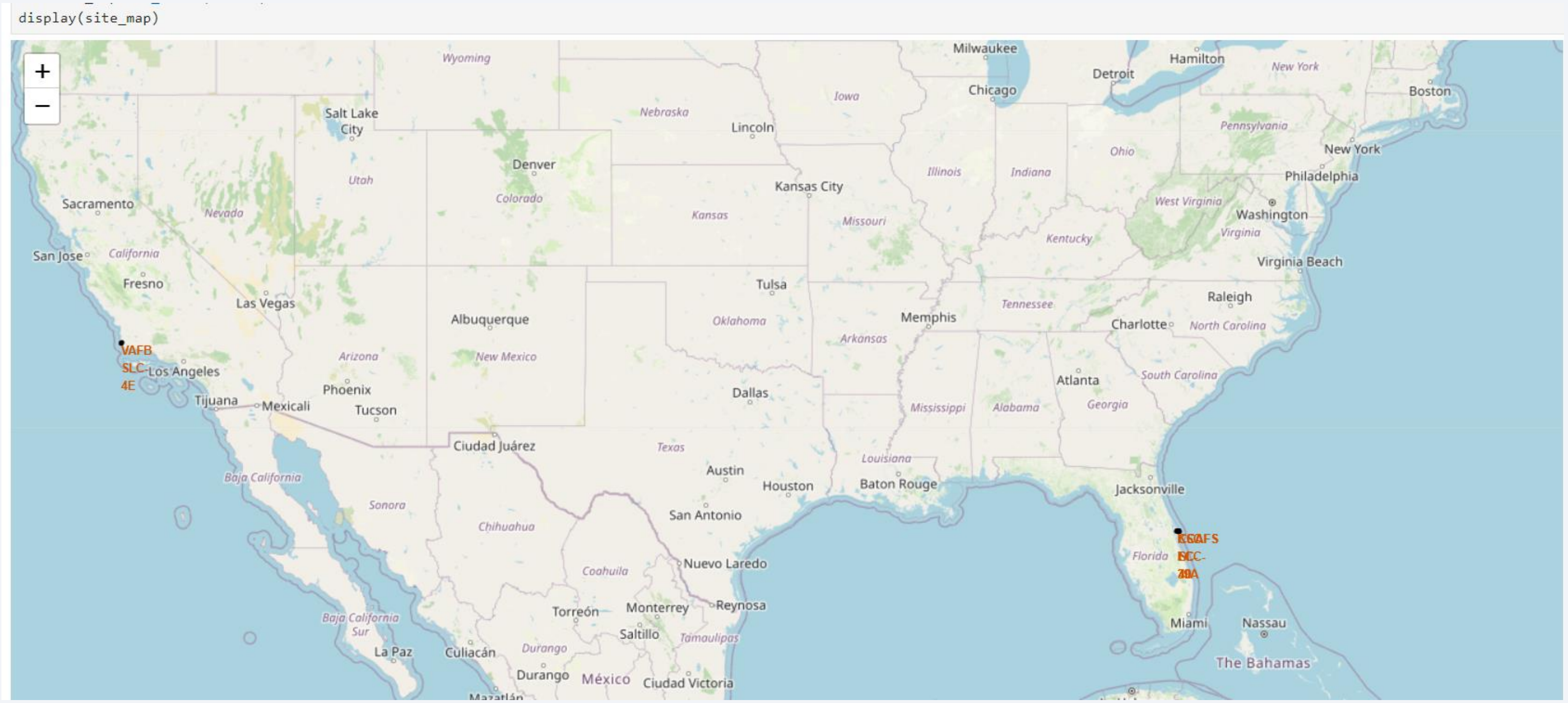
Landing_Outcome	Total
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

Section 3

Launch Sites Proximities Analysis

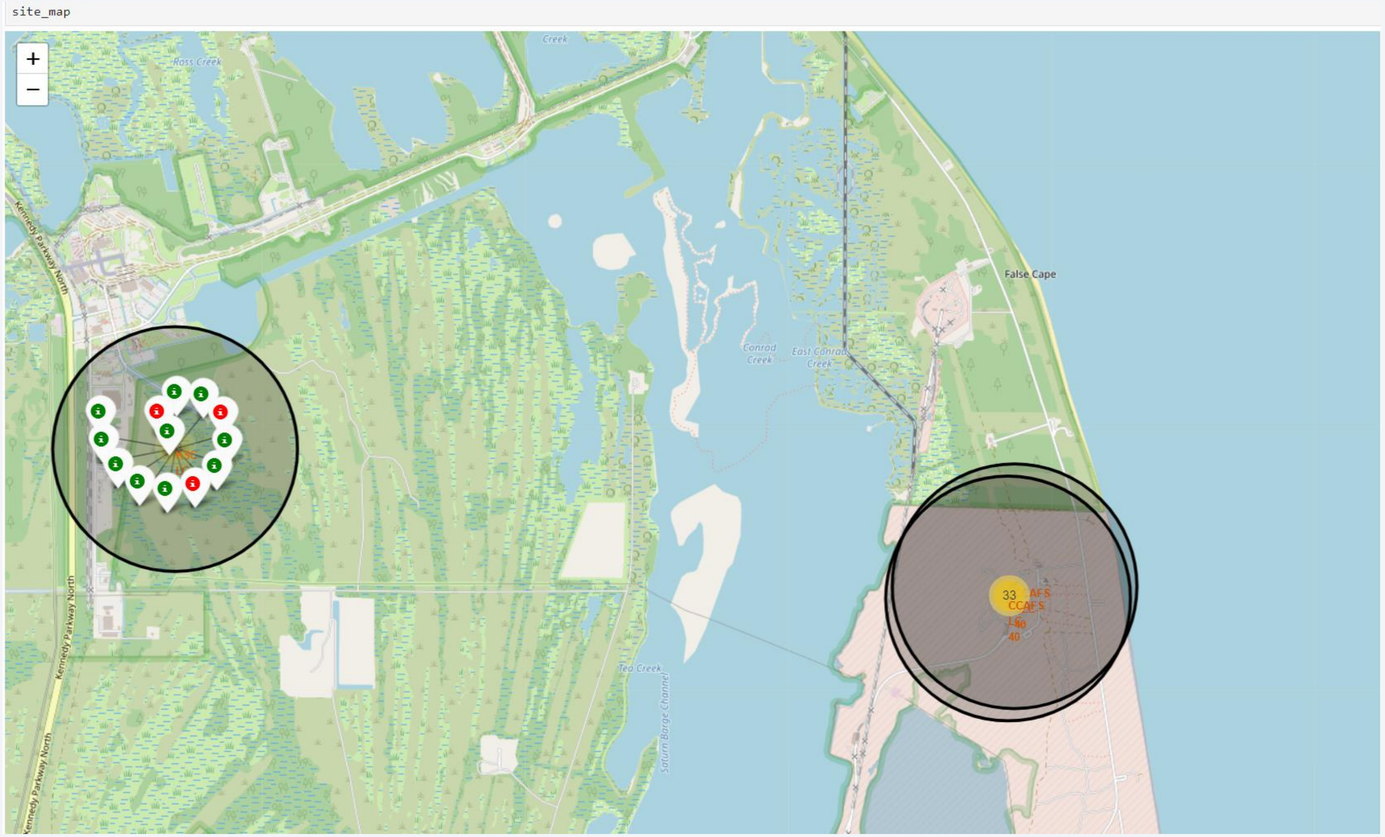
SpaceX Launch Site Locations

- USA
- Coastal
- Warmer Climate



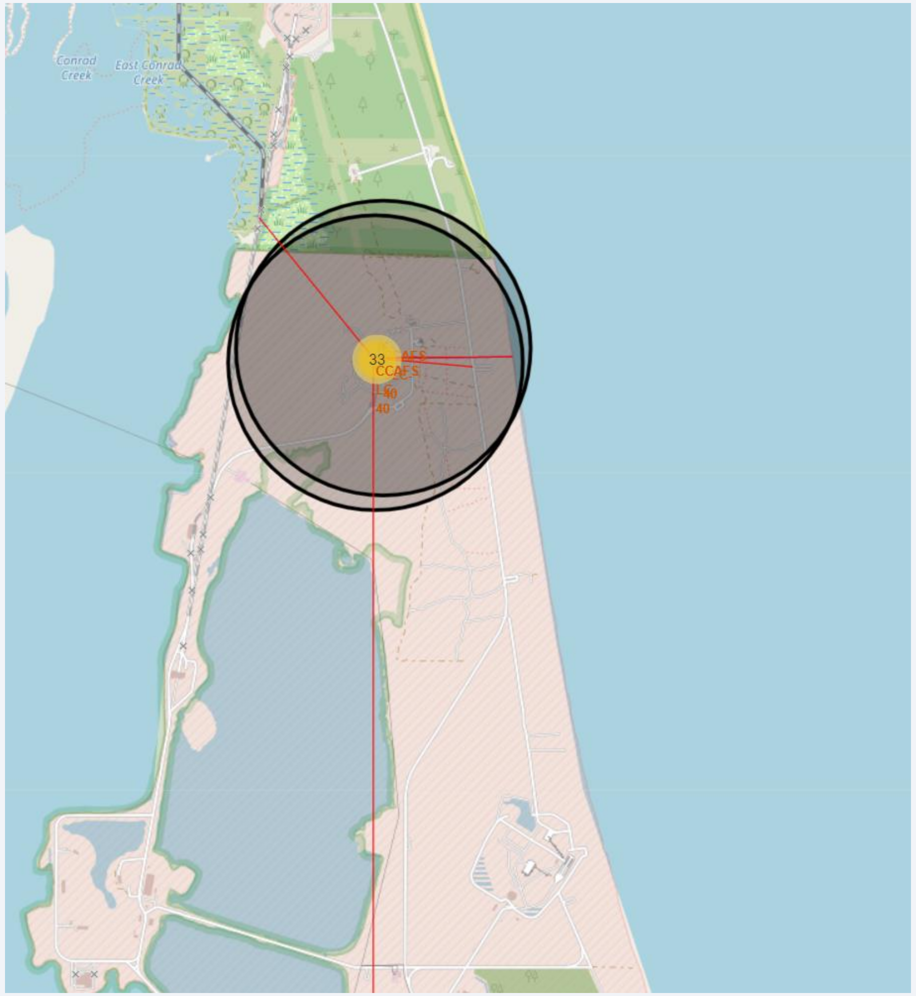
SpaceX Volume of Flights by Site

- KSC LC-39A: 13
- CCAFS SLC-40: 7
- CCAFS LC-40: 26
- VAFB SLC-4E: 10
- Successes are marked by green label



<Folium Map Screenshot 3>

- We saw that generally, launch sites had close proximity to:
  - Coast
  - Highway access
  - Railroads
- Florida bases were relatively close to major city, but VAFB SLC-4E was not

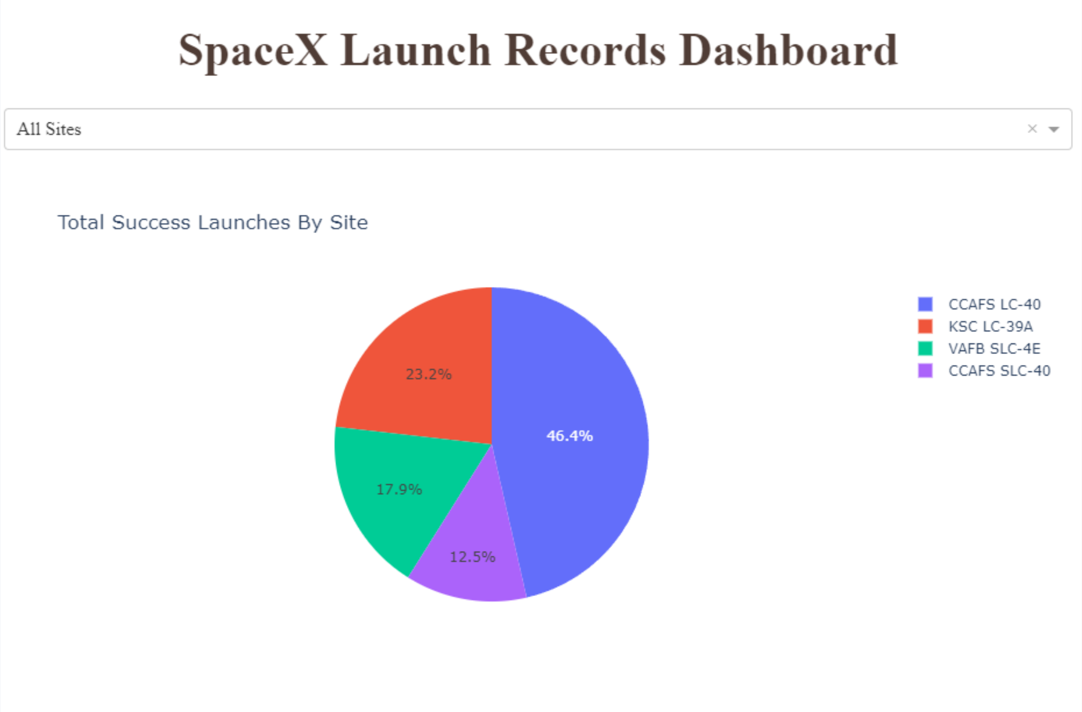


Section 4

Build a Dashboard with Plotly Dash

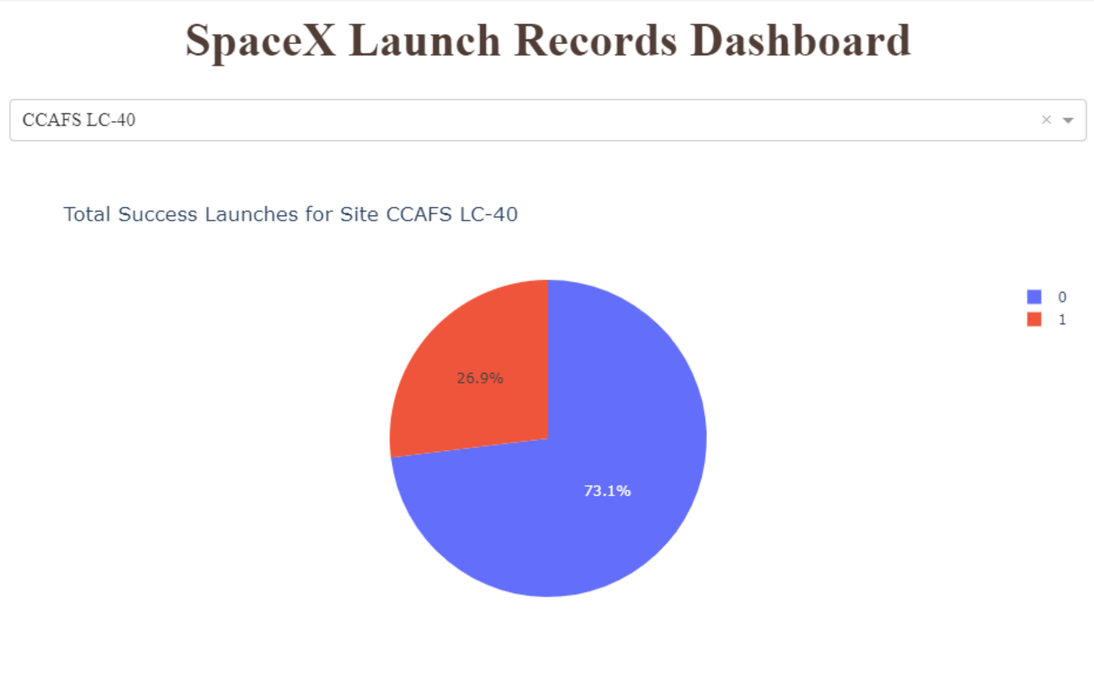
Launch Sites contribution to successes

- CCAFS LC-40 has highest total number of successes
- CCAFS SLC-40 has lowest total number of successes



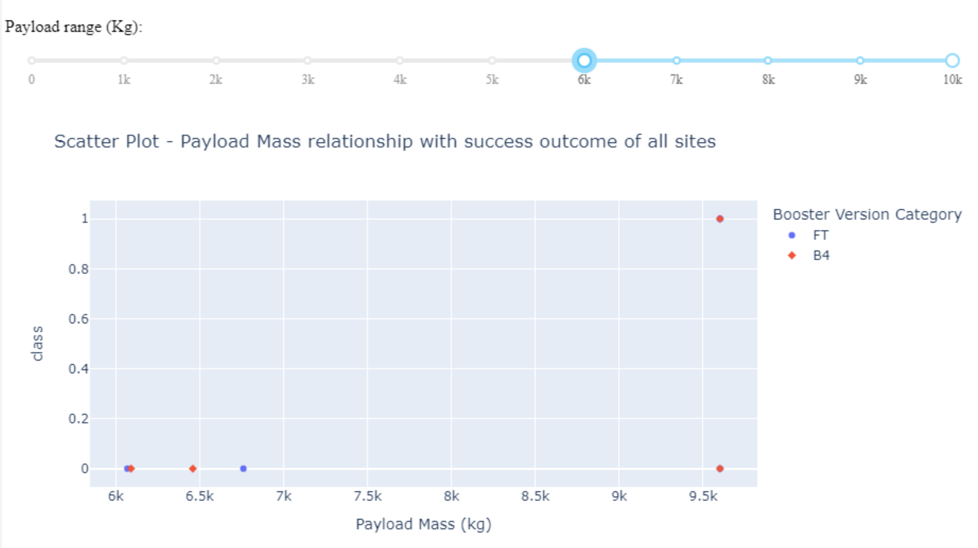
CCAFS LC-40 success rate

- High success rate
  - 2nd highest amongst all despite largest volume of launches



Payload and Success Rates of all sites

- FT boosters have best success
- B5 boosters have worst success rate
- Booster performance drops at higher weights

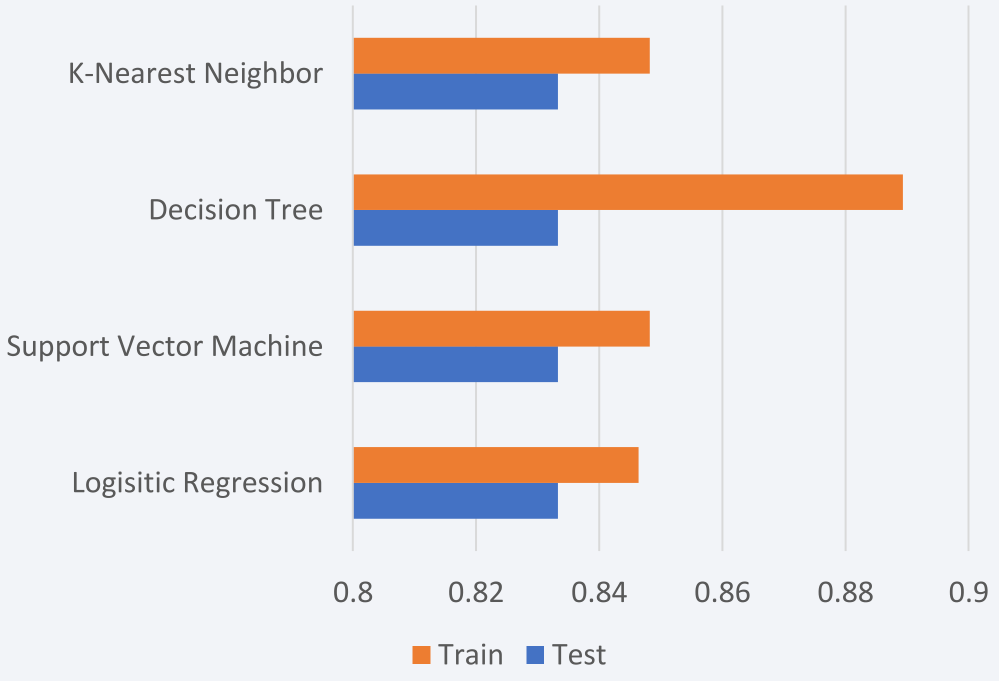


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Models performed at same rate
- Decision Tree had high training accuracy compared to test score
  - Possible overfit
- Reran models 5 times, reloading train test split
  - Decision Tree is only one which's accuracy shifted



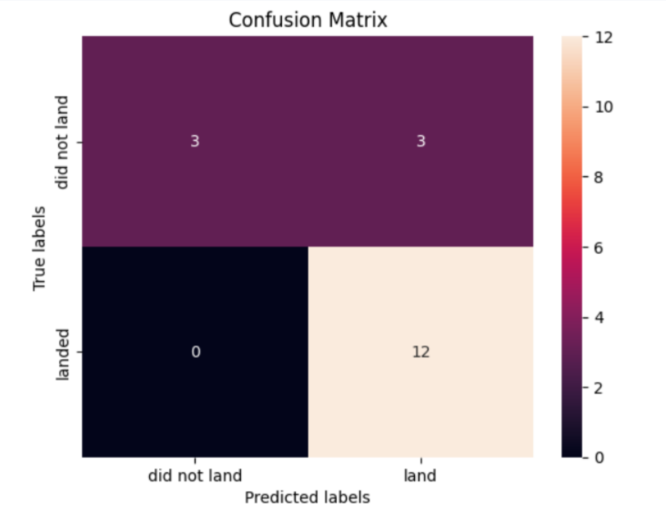
Find the method performs best:

```
acc_scores = {
    "Logisitic Regression":lr_acc,
    "Support Vector Machine":svm_acc,
    "Decision Tree":tree_acc,
    "K-Nearest Neighbor":knn_acc
}
best_method = max(acc_scores,key=acc_scores.get)
best_acc = acc_scores[best_method]
print(f"Best accuracy was {best_acc} by {best_method}")
```

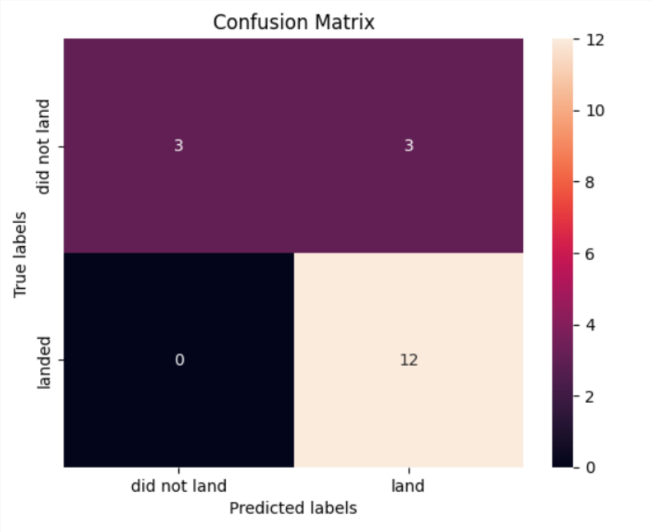
```
Best accuracy was 0.8333333333333334 by Logisitic Regression
```

Confusion Matricies

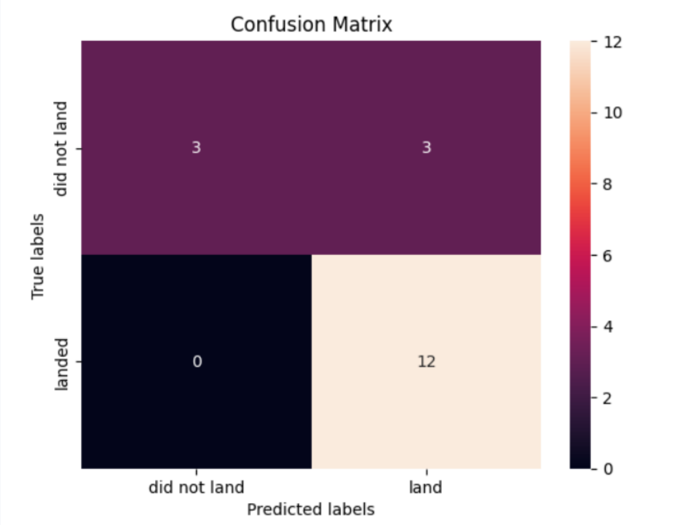
- Logistic Regression



- Support Vector Machine

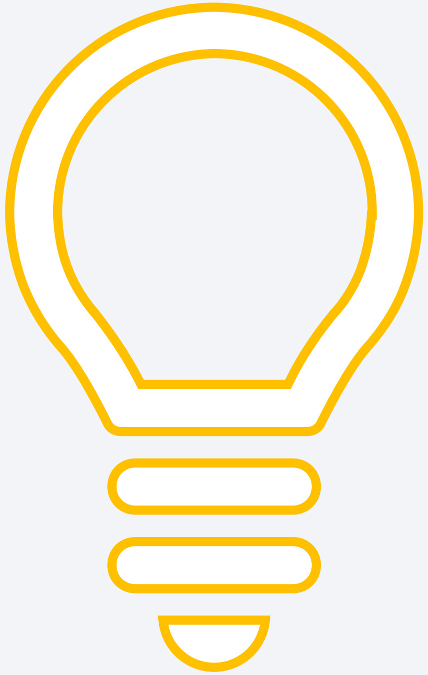


- K-Nearest Neighbor



Conclusions

- Flights should primarily originate CCAFS LC 40
  - Especially with top end payload
- VLEO and SSO orbits
  - Best combination of success and volume
- F9 B5 class boosters NEED to be used on any flights with max payload
- Decision Tree is eliminated as Classification option
  - Need to refine model, 3 way tie for best performance



Appendix

- Find all relevant data, code, and charts on the gordonrepository github page

Thank you!