IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

IBM Applied Data Science Capstone Project

Donald Mears
April 14, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis (EDA) with Data Visualization
  - EDA with SQL
  - Building an Interactive Map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive Analysis (Classification)
- Summary of all results
  - EDA Results
  - Interactive Analytics Demo in Screenshots
  - Predictive Analysis Results

Introduction

- Project background and context
  - SpaceX advertises that a Falcon 9 rocket launches on its website for a cost of $62 million each while other providers cost upwards of $165 million.
  - A significant driver of Falcon 9 cost savings are due to the fact that SpaceX can reuse the first stage.
  - However SpaceX isn’t always able to successfully recover the first stage.
  - We will determine if SpaceX will reuse the first stage using a machine learning model and publicly available information.

- Problems you want to find answers
  - What are the drivers of successful launches?
  - What conditions lead to the highest successful landing rates?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was gathered from the SpaceX API and web scraping the Falcon 9 and Falcon Heavy launch records from Wikipedia
- Data Wrangling
  - Converted outcome feature into Boolean target label for use in classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Found best hyperparameter for Support Vector Machines (SVM), Decision Trees, and Logistic Regression models and evaluated accuracy metrics

Data Collection

- Data sets were collected from two sources:
  - SpaceX API Process: 1

Calls to the SpaceX API → Transform response data to JSON format → Normalize JSON file into a Pandas dataframe → Select subsets and make file type conversions → Data Wrangling (e.g., handling missing values) → Save to csv format

  - Wikipedia via Web Scraping Process: 2

Use the HTTP GET method to request the Falcon 9 Launch Wiki page → Create a BeautifulSoup object → Parse data into a dictionary → Create a Pandas dataframe from the dictionary → Save to csv format

SpaceX API Columns: 1

FlightNumber
Date
BoosterVersion
PayloadMass
Orbit
LaunchSite
Outcome
Flights
GridFins
Reused
Legs
LandingPad
Block
ReusedCount
Serial
Longitude
Latitude

Web Scraping Columns: 2

Flight No.
Launch site
Payload
Payload mass
Orbit
Customer
Launch outcome
Version Booster
Booster landing
Date
Time

Data Collection – SpaceX API

1) Request rocket launch data from SpaceX API

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

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

2) Convert JSON File into Pandas Dataframe using the normalize method

```
data = pd.json_normalize(response.json())
```

3) Select subsets and make file type conversions

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boos
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list an
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

4) Data Wrangling

```
data_falcon9.isnull().sum()
```

```
# Calculate the mean value of PayloadMass column
PayloadMass_mean = data_falcon9.PayloadMass.mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, PayloadMass_mean)
```

5) Save to a csv file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

GitHub Link to Jupyter Notebook File:
https://github.com/donaldmears/IBM_Data_Science_Certification/blob/main/Data%20Collection%20API.ipynb

Data Collection - Scraping

1) Getting html data

```
html_data = requests.get(static_url).text
```

2) Create BeautifulSoup object

```
soup = BeautifulSoup(html_data, 'lxml')
```

3) Parse data into a dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

4) Create Pandas dataframe from the dictionary

```
df=pd.DataFrame(launch_dict)
```

2) Save to csv format

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

GitHub Link to Jupyter Notebook File:
https://github.com/donaldmears/IBM_Data_Science_Certification/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- The Outcome field contains both the outcome and landing site and needed to be transformed into a Boolean Class (0 or 1) field for use in later data visualization and machine learning work
- The outcome success portion of the Outcome field has the following values:
  - True – successful
  - False – not successful
  - None – not successful
- The landing site portion of the Outcome field has the following values:
  - ASDS – drone ship
  - Ocean – specific region of ocean
  - RTLS – ground pad
  - None – not available however failed to land

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

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

```
landing_class = []
for outcome in df.Outcome:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

```
df['Class']=landing_class
df[['Class']].head(8)
```

GitHub Link to Jupyter Notebook File:
https://github.com/donaldmears/IBM_Data_Science_Certification/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Scatter Charts	Bar Chart	Line Chart
• Flight Number vs Pay Load Mass (kg) • Flight Number vs Launch Site • Pay Load Mass (kg) vs Launch Site • Flight Number vs Orbit Type	• Orbit Type vs Success Rate	• Year vs Success Rate
With the hue set to class, these scatter plots become useful in determining potentially predictive features that can be used in machine learning	Users can see which Orbit Types have the highest success rates and which have the lowest for further investigation	This line chart shows a clear improvement trend over time from 0% success rate in 2012 and 2013 to over 80% in three of the four most recent years

GitHub Link to Jupyter Notebook File:
https://github.com/donaldmears/IBM_Data_Science_Certification/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

SQL queries performed:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- 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.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

GitHub Link to Jupyter Notebook File: https://github.com/donaldmears/IBM_Data_Science_Certification/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- An interactive map using Folium was created with several objects
  - Markers identifying launch sites on the map
  - Markers that indicate success/failure of launches for each of these sites
  - Lines showing the distances to various features near the launch sites, including:
    - Cities
    - Coastline
    - Highways
    - Railways

GitHub Link to Jupyter Notebook File (maps not available):
https://github.com/donaldmears/IBM_Data_Science_Certification/blob/main/lab_jupyter_launch_site_location.ipynb

IBM Link (maps may be available for a limited time)
https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/ba642df4-fc19-4785-81a3-2eaef83e7649/view?access_token=c9c1c62d78eb80f52ffc111cda0b2994d4f9a275e0eca9d9930050ad850fcd06

Build a Dashboard with Plotly Dash

Pie Chart	Scatter Chart
Shows percentage of success launches by site to determine success rates of launch sites	Shows relationship between Outcomes and Payload Mass to help determine how success depends on the launch point, payload mass, and booster version
Users can select all sites or specific sites to see a breakdown	Users can use a slider to see different ranges of Payload Range (Kg)

GitHub Link to Jupyter Notebook File:
https://github.com/donaldmears/IBM_Data_Science_Certification/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

Process Followed:

Data prepared / features and target (Y) identified → X data scaled / fit / transformed → Train / Test Split with test size param set to 20% → Logistic Regression model → Support Vector Machine model → Decision Tree model → K-Nearest Neighbors model → Compare accuracy and confusion matrix results

Machine Learning Models

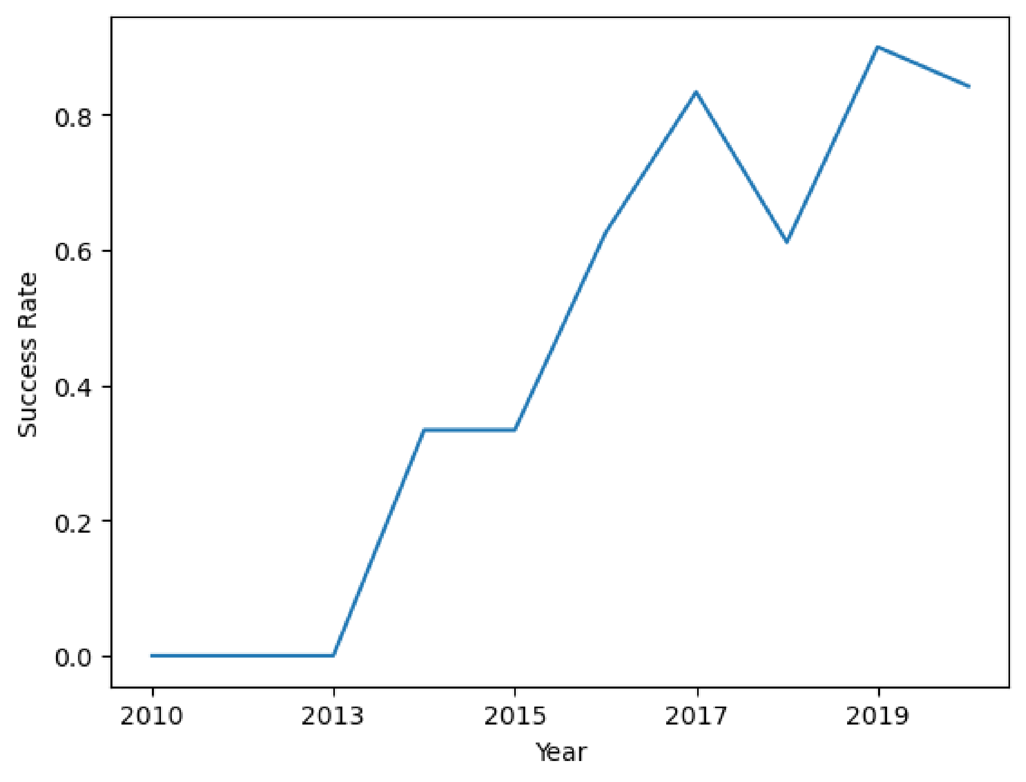
- All models performed with the same accuracy likely due to the small sample size

GitHub Link to Jupyter Notebook File:
https://github.com/donaldmears/IBM_Data_Science_Certification/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

- Landing success has increased over the past 10 years to more than 80% in three of the four most recent years
- It is encouraging that all of the predictive models show high accuracy of 83%
- A deeper look at all of the exploratory analysis and data visualizations will be presented in Section 2 of the report

Landing Success Rate

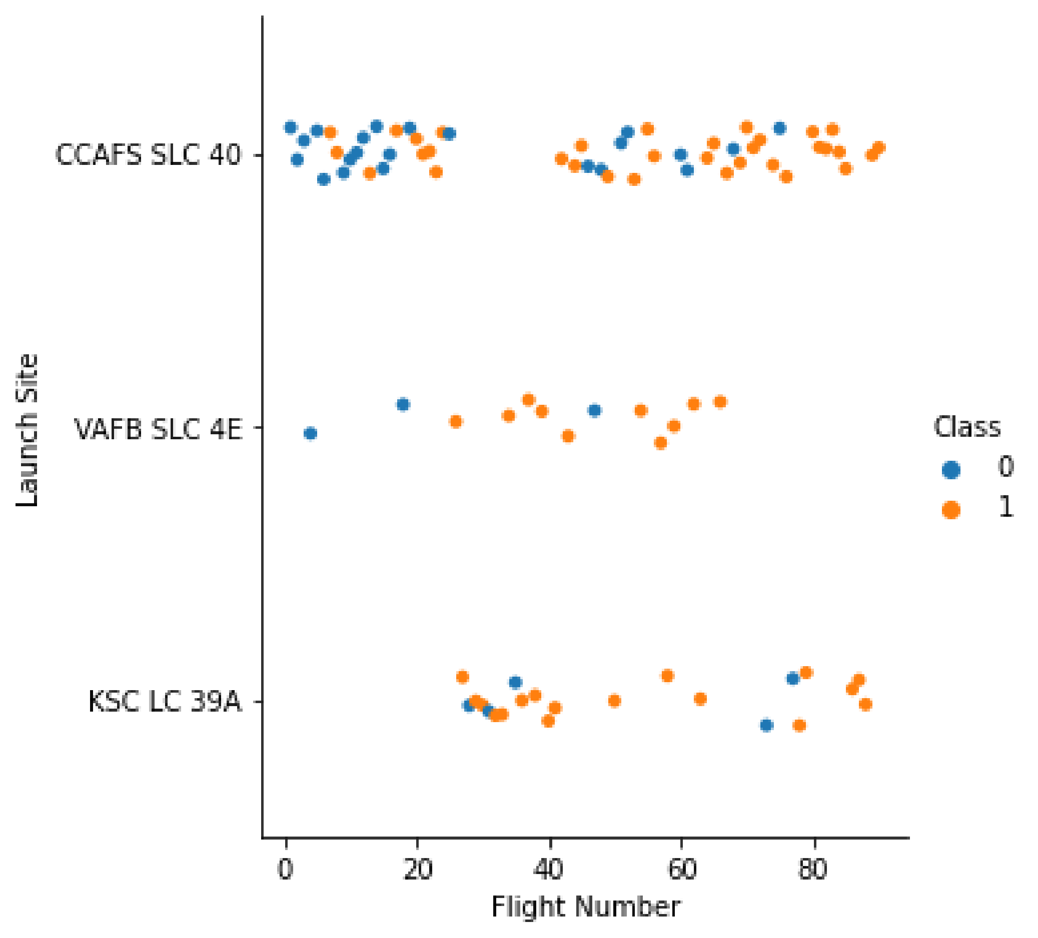


Section 2

Insights drawn from EDA

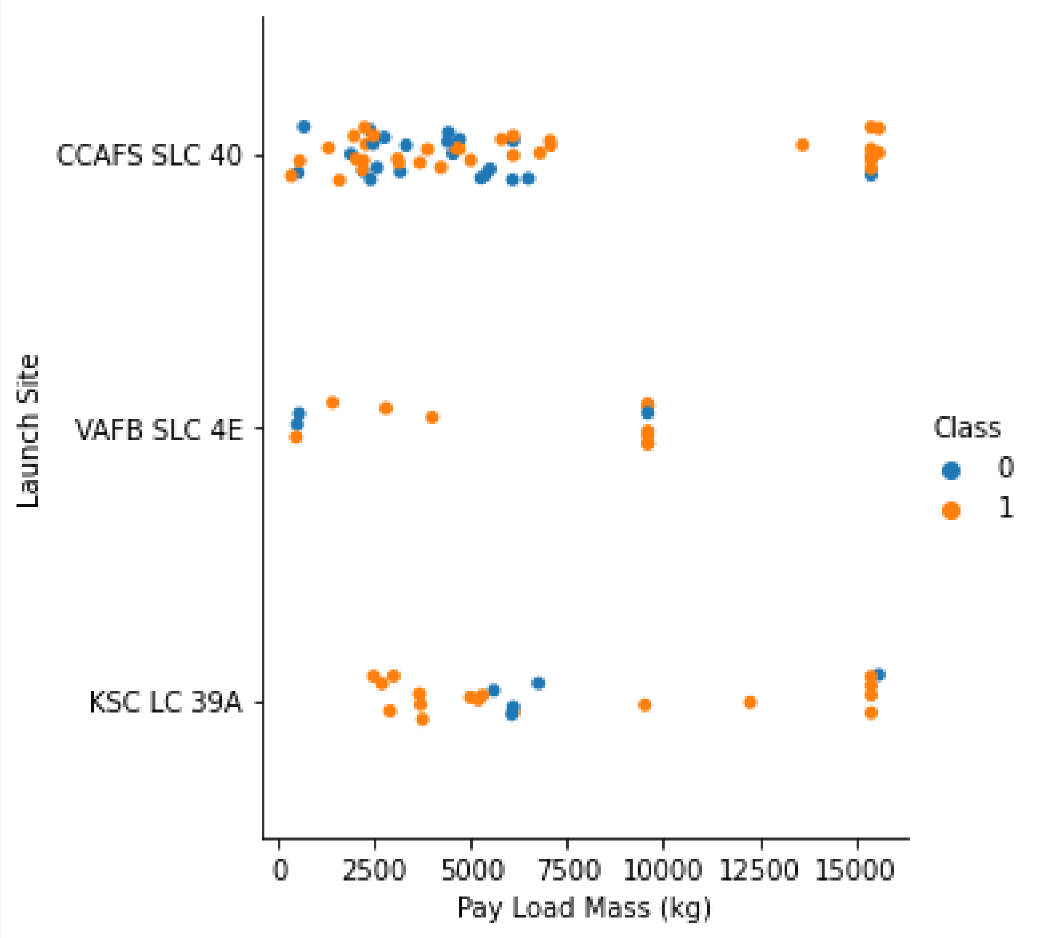
Flight Number vs. Launch Site

- Similar to the overall improvement trend over the past 10 years, this chart shows improvement as Flight Numbers increase (fewer blue dots which are unsuccessful)
- CCAFS SLC 40 is the most popular launch site and there seems to have been a period where KSC LC 39A was used (around Flight Number 25-40)
- It was around this time that improvements seem to have been made



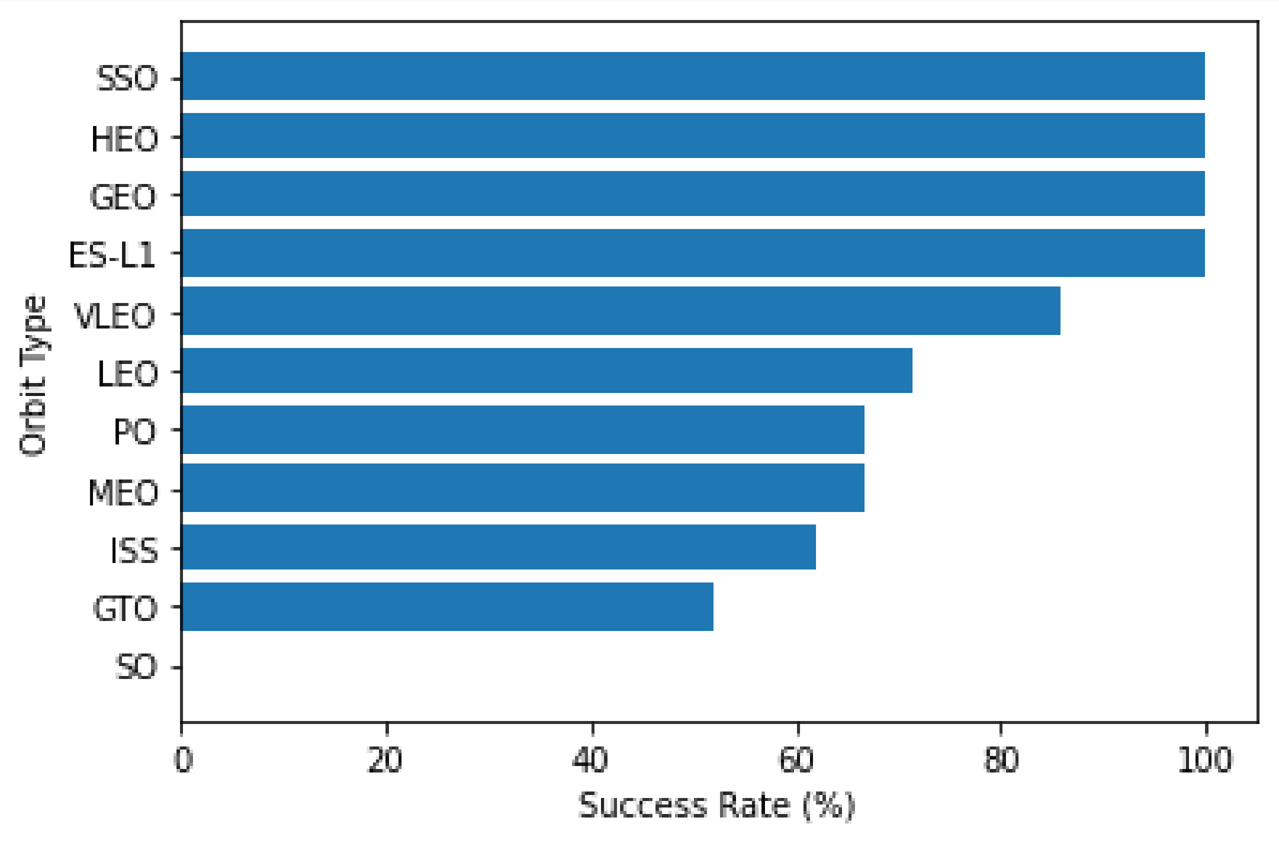
Payload vs. Launch Site

- Low Pay Load launches from KSC LC 39A seem to be more successful while higher Pay Load launches from CCAFS SLC 40 seem to be more successful
- Sample sizes are likely too low to prove that Pay Load plays a direct role in launch success



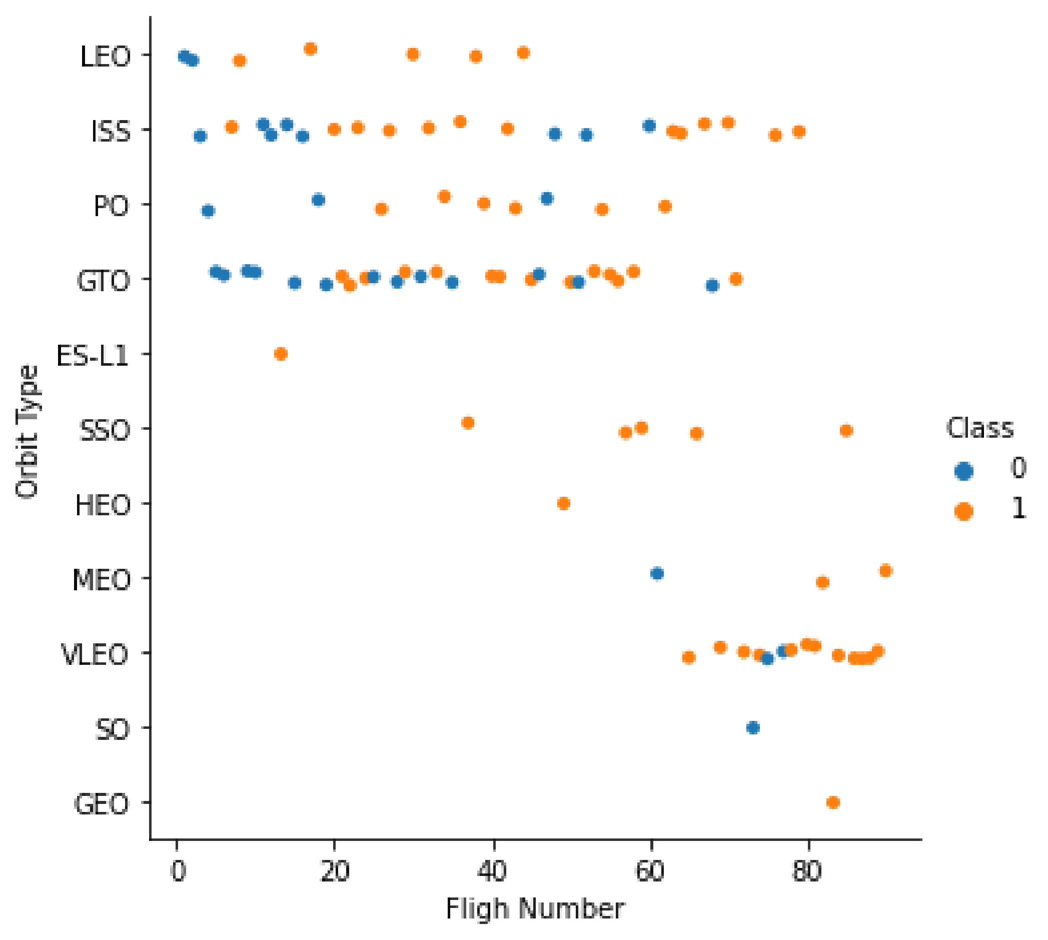
Success Rate vs. Orbit Type

- SSO, HEO, GEO, and ES-L1 are the Orbit Types with 100% success rates
- The number and recency of launches (success rates have improved in the past several years) likely play a role
- Similarly, SO has a 0% success rate however only one launch



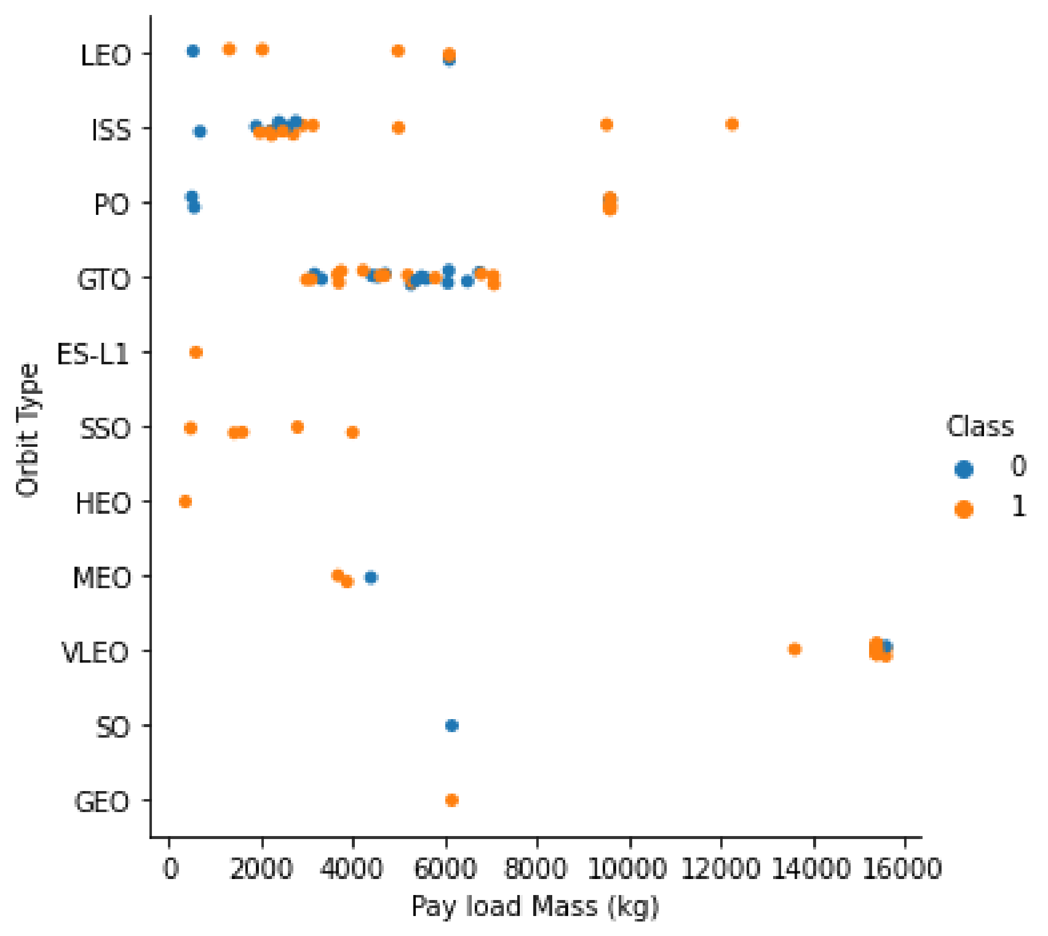
Flight Number vs. Orbit Type

- Similar to previous scatter plot charts, as Flight Number increases, success increases
- There have only been a few launches at the historically higher volume Orbit Types (LEO, ISS, PO, GTO) and new Orbit Types have been introduced, potentially playing a role in the improvement
- There has also been gradual improvement seen in the higher volume Orbit Types



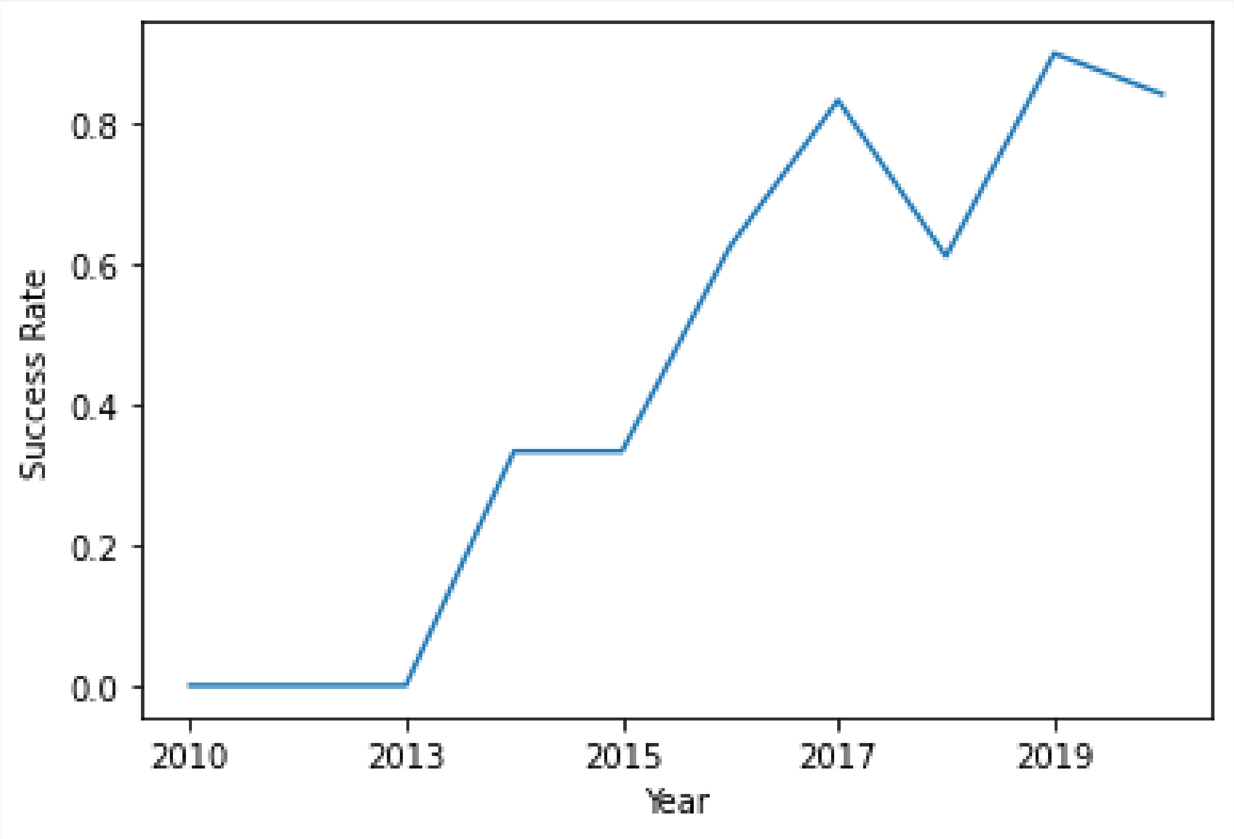
Payload vs. Orbit Type

- Payloads tend to be in a fairly tight range by Orbit Type with a few exceptions (ISS, PO, GTO, LEO)
- Regardless, this is an unremarkable scatter plot



Launch Success Yearly Trend

- This line chart shows a clear improvement trend over time from 0% success rate in 2012 and 2013 to over 80% in three of the four most recent years
- 2018 saw more than a 20 point reduction in Success Rate



All Launch Site Names

- Using the SQL SELECT DISTINCT clause allows us to identify the four launch sites

```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

```
 * 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'

```
%%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

- With the LIKE operator and percent sign following “CCA” coupled with the LIMIT 5 clause, this allows us to query five records pertaining to sites that begin with “CCA”

Total Payload Mass

- The SUM function allows us to see the sum of the Payload Mass
- The WHERE clause allows us to filter the customer field to use only NASA (CRS) records in the query

```
%%sql
SELECT sum(payload_mass__kg_) as total_payload_mass_kg
from spacextbl
where customer = 'NASA (CRS)'
```

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

total_payload_mass_kg
45596

Average Payload Mass by F9 v1.1

- The AVG function allows us to see the average of the Payload Mass
- The WHERE clause allows us to filter the Booster Version field to use only F9 v1.1 records in the query

```
%%sql
SELECT avg(PAYLOAD_MASS__KG_) AS avg_payload_mass_kg
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

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

avg_payload_mass_kg
2928.4

First Successful Ground Landing Date

- The MIN function allows us to see the minimum or first date
- The WHERE clause allows us to filter the Landing Outcome field to use only successful ground pad records in the query

```
%%sql
SELECT MIN(Date) AS FIRST_SUCCESSFUL_LANDING_DATE
FROM SPACEXTBL
WHERE "Landing _Outcome" = "Success (ground pad)"
```

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

FIRST_SUCCESSFUL_LANDING_DATE
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
select BOOSTER_VERSION from SPACEXTBL
where "Landing _Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4001 and 5999
```

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

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

- The WHERE clause allows us to filter the Landing Outcome field to use only successful drone ship records and the AND operator allows us to add another condition to filter records between 4000 and 6000 of payload mass in the query

Total Number of Successful and Failure Mission Outcomes

- The COUNT function allows us to calculate the number of records
- The GROUP BY statement groups the Mission_Outcome field in the query

```
%%sql
SELECT mission_outcome, count(*) as total_number
from spacextbl
group by mission_outcome
```

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

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

Boosters Carried Maximum Payload

- A subquery in the WHERE clause allows us to filter based on the MAX function on the Payload Mass Kg column

```
%%sql
SELECT distinct booster_version, payload_mass__kg_
from spacextbl
where payload_mass__kg_ = (select max(payload_mass__kg_) from spacextbl)
```

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

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

2015 Launch Records

```
%%sql
SELECT substr(Date, 4, 2) as month,booster_version,"Landing _Outcome"
from SPACEXTBL where "Landing _Outcome" ='Failure (drone ship)' and substr(Date,7,4)='2015'
```

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

month	Booster_Version	Landing _Outcome
01	F9 v1.1 B1012	Failure (drone ship)
04	F9 v1.1 B1015	Failure (drone ship)

- The WHERE clause allows us to filter the Landing Outcome field to use only drone ship failure records in the query
- The AND operator then allows us to additionally filter the year 2015
- With sqlite, it’s necessary to use the SUBSTR function to extract the year from the date instead of using the YEAR function on the date

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

```
%%sql
SELECT "Landing _Outcome", COUNT(*) AS COUNT_LAUNCHES
FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE "%Success%" AND DATE BETWEEN '04-06-2010' AND '20-03-2017'
GROUP BY "Landing _Outcome"
ORDER BY COUNT_LAUNCHES DESC;
```

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

Landing _Outcome	COUNT_LAUNCHES
Success	20
Success (drone ship)	8
Success (ground pad)	6

- The WHERE clause allows us to filter the Landing Outcome field to successful records in the query using the LIKE clause and wildcards around the word success
- The AND operator then allows us to additionally filter the requested date range

Section 3

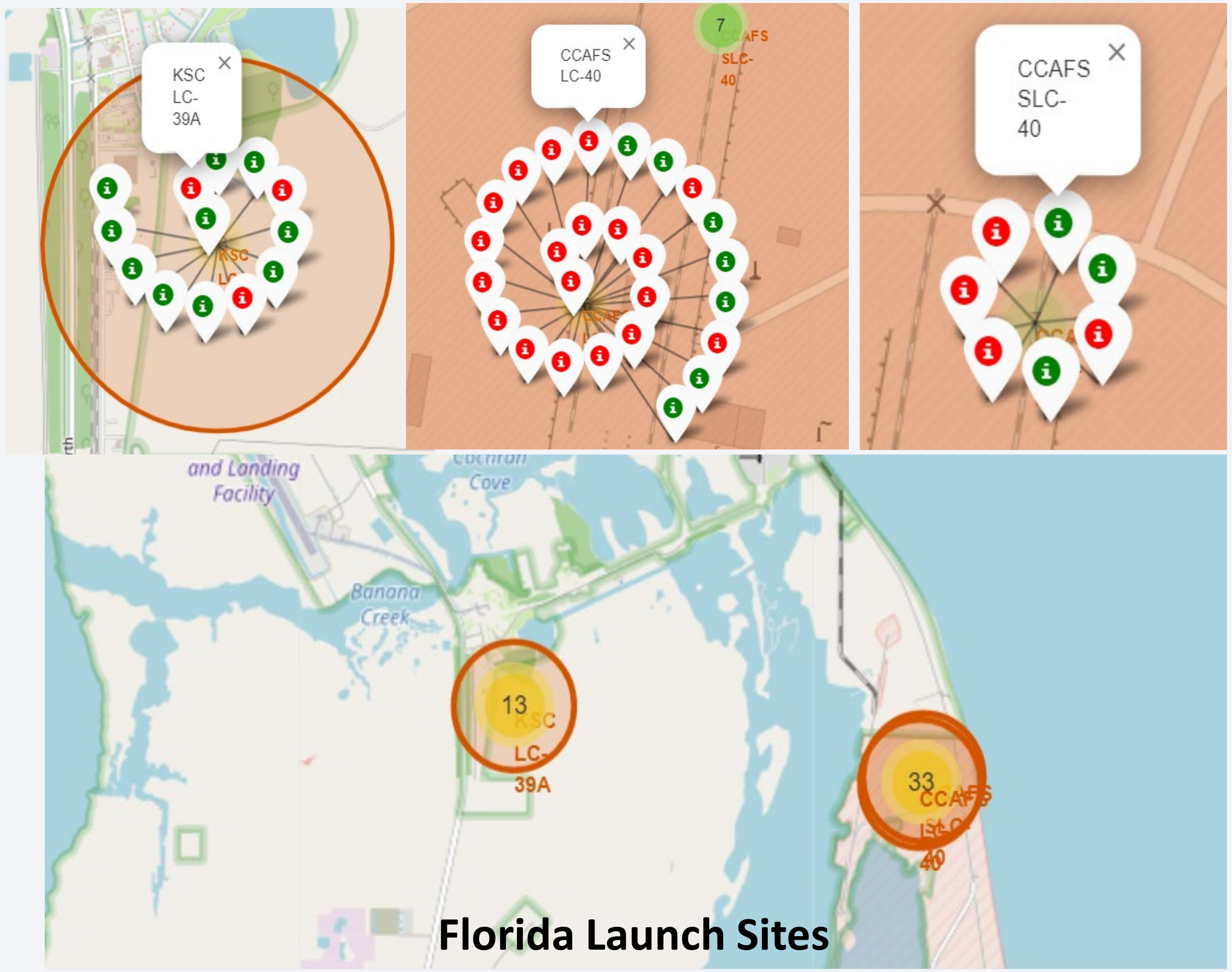
Launch Sites Proximities Analysis

Folium Map Showing All Launch Sites

- In the interactive Folium map, it is possible to zoom in to very detailed features and all the way out to the world map
- In this map, the two Launch Sites can be seen in California and Florida

California and Florida Launch Sites

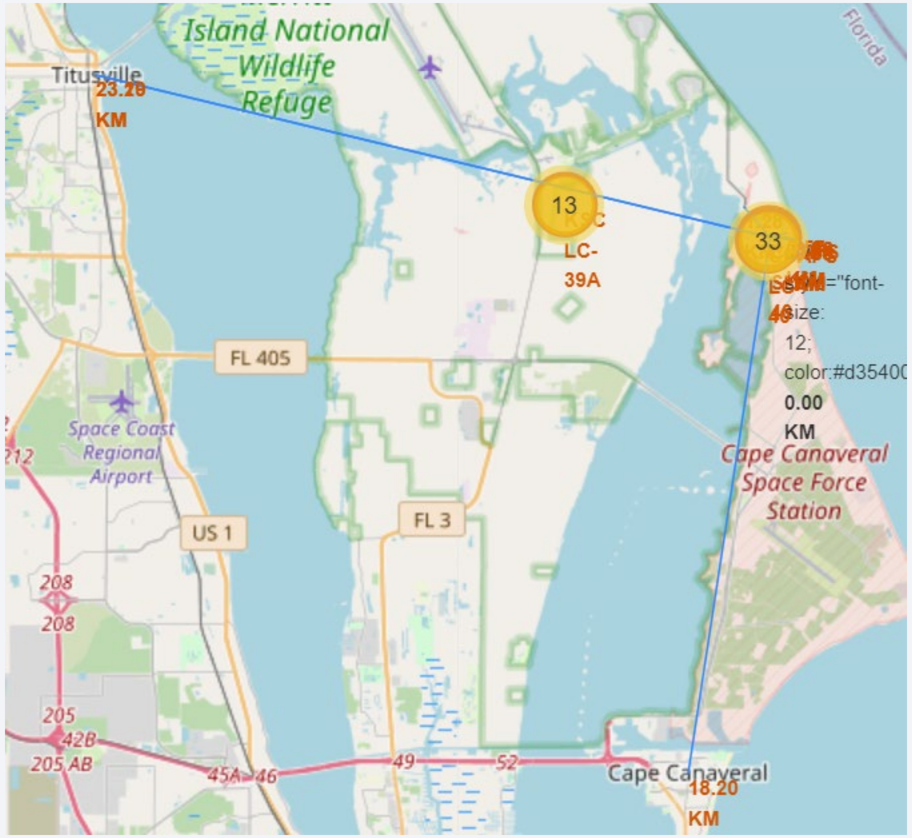

California Launch Site



Florida Launch Sites

- Clicking on launch sites, successful and failed landings are indicated by the green and red indicators, respectively

Launch Site Proximities



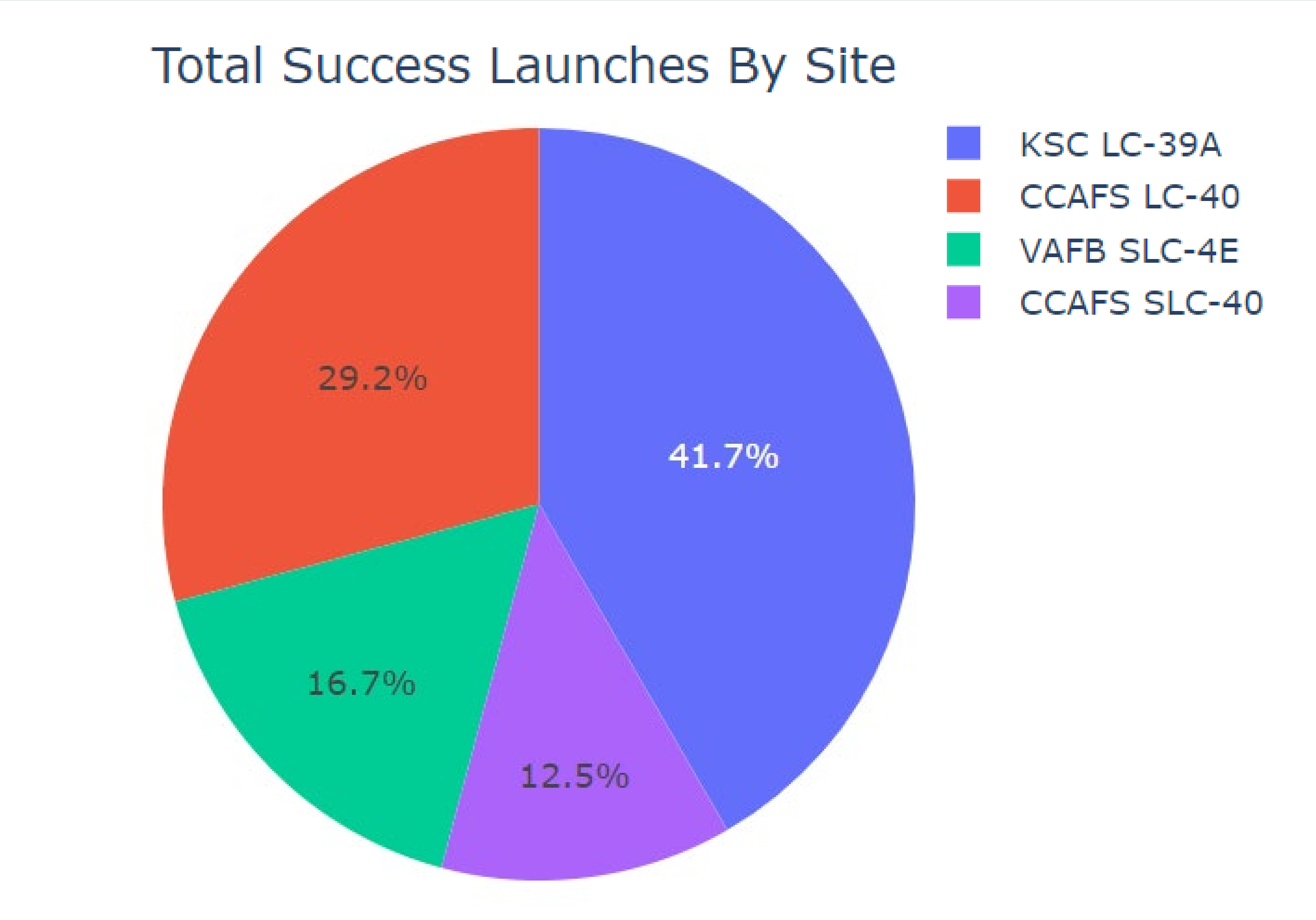
- Launch sites are close to railways, highways, and the coastline – all distances are calculated and shown in the interactive map
- Additionally, the launch sites are a good distance away from cities for safety reasons

Section 4

Build a Dashboard with Plotly Dash

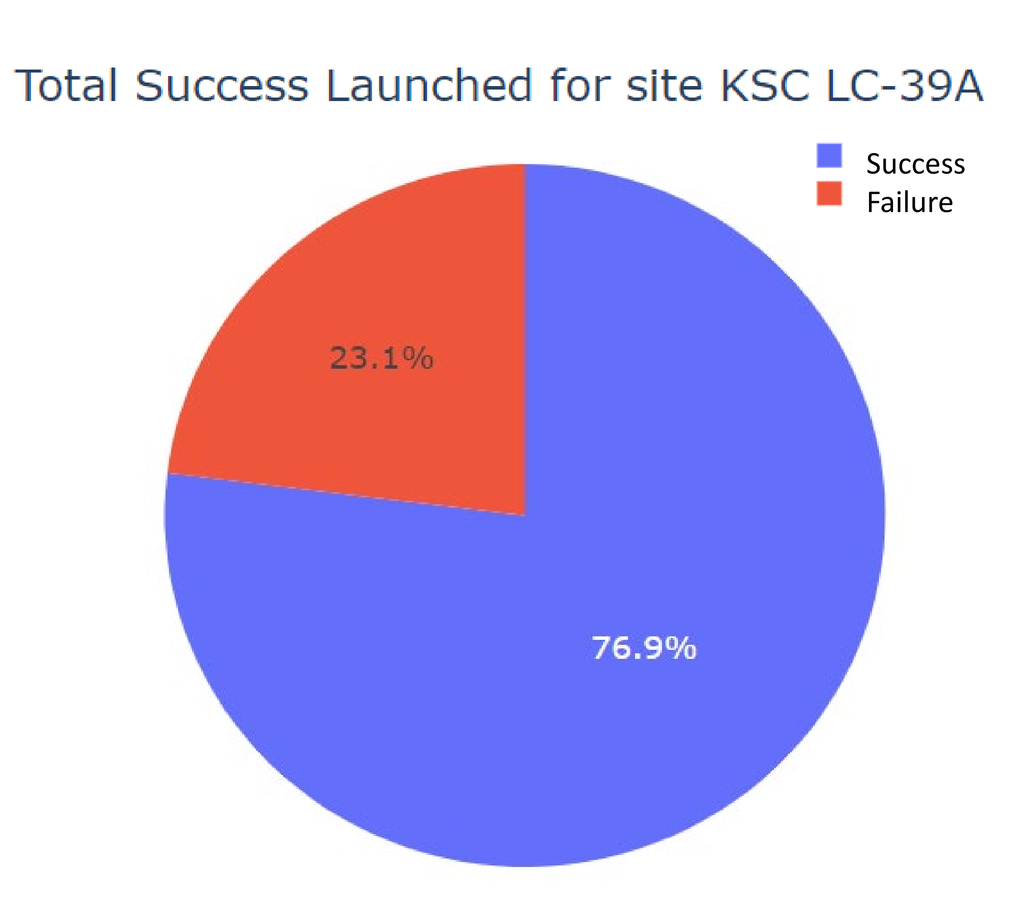
Overall Successful Launches by Site

- KSC LC-39A makes up 41.7% of overall successful launches followed by CCAFS C-40 with 29.2%
- The other two sites contribute the least percentage of successful launches

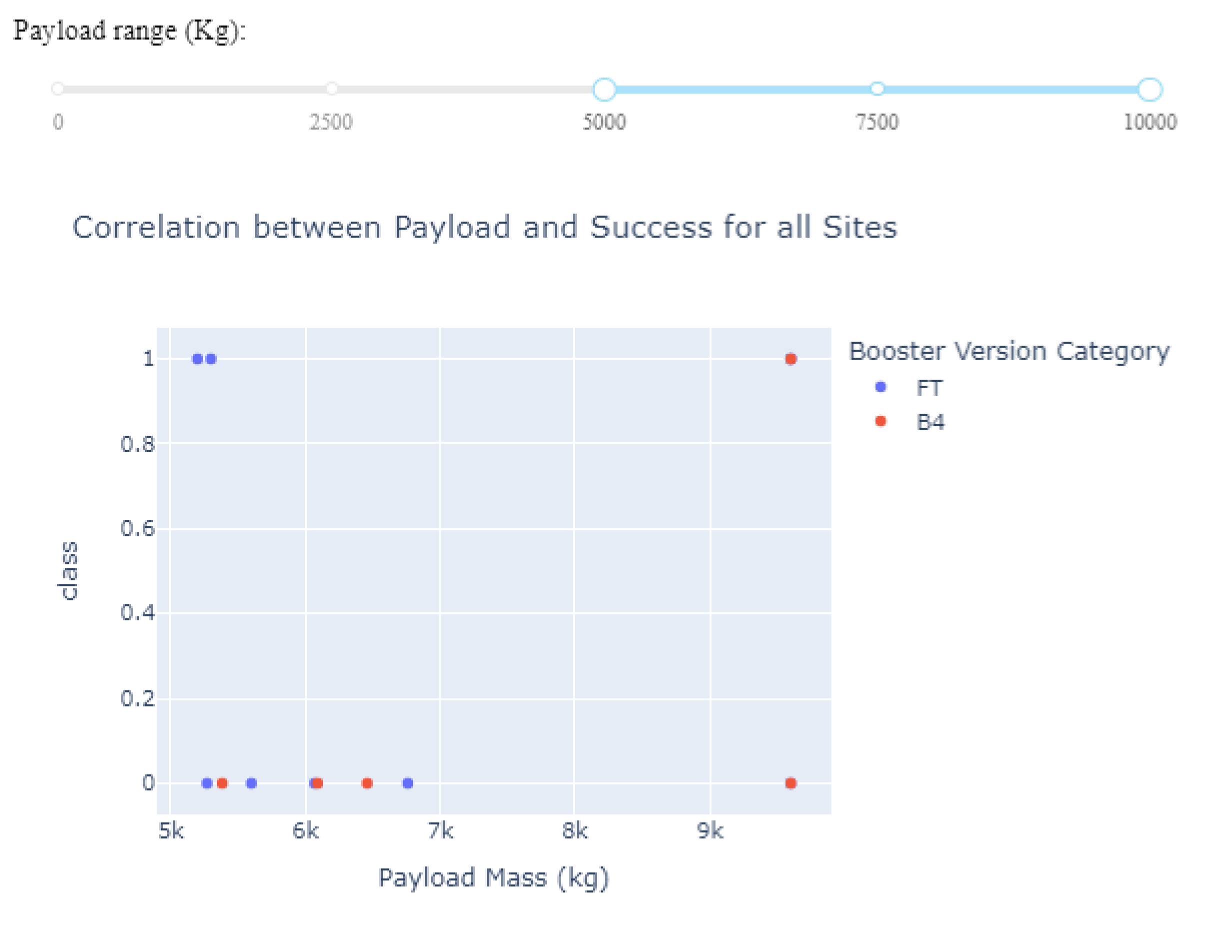


Most Successful Launch Site

- KSC LC-39A has the highest success rate at 76.9%
- The site has had 10 landing successes and 3 failures



Payloads and Success Rates



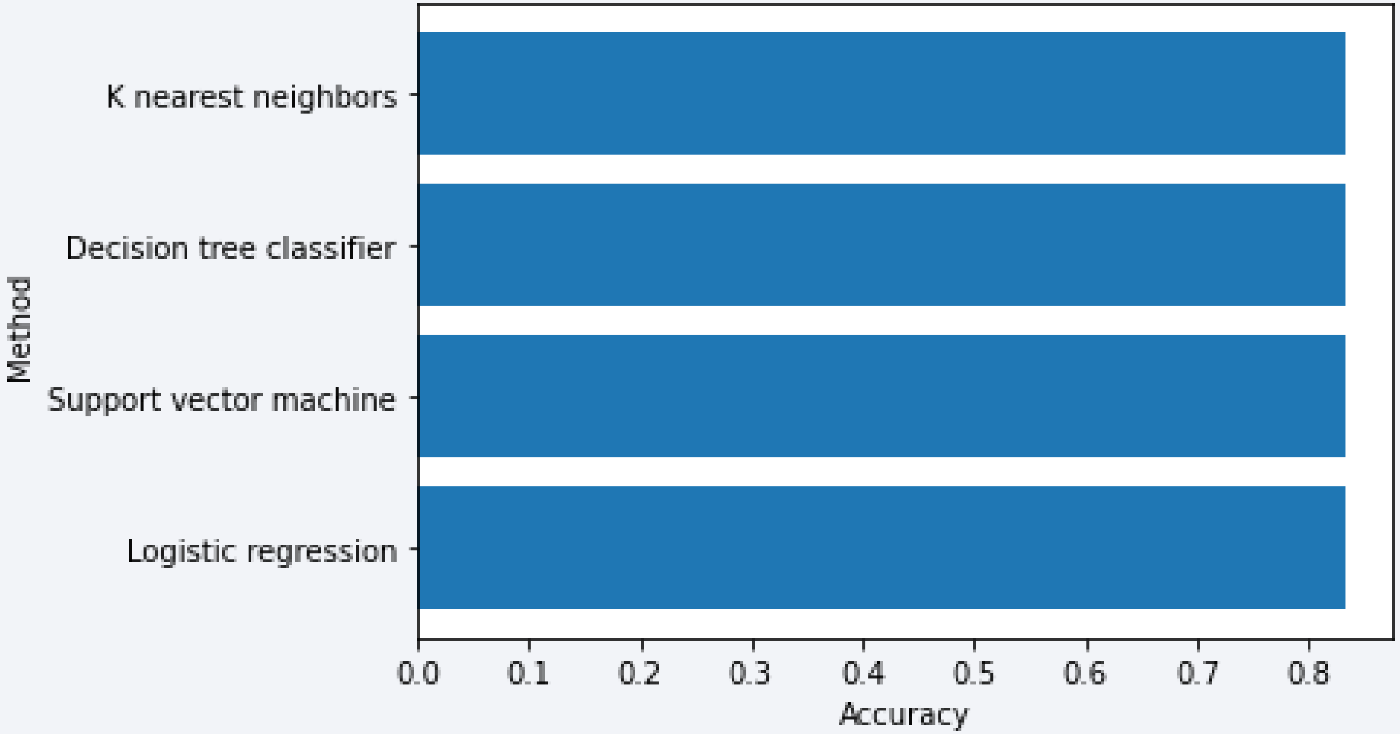
- Launch success rates for heavier payloads is lower although the number of heavier payload launches is lower

Section 5

Predictive Analysis (Classification)

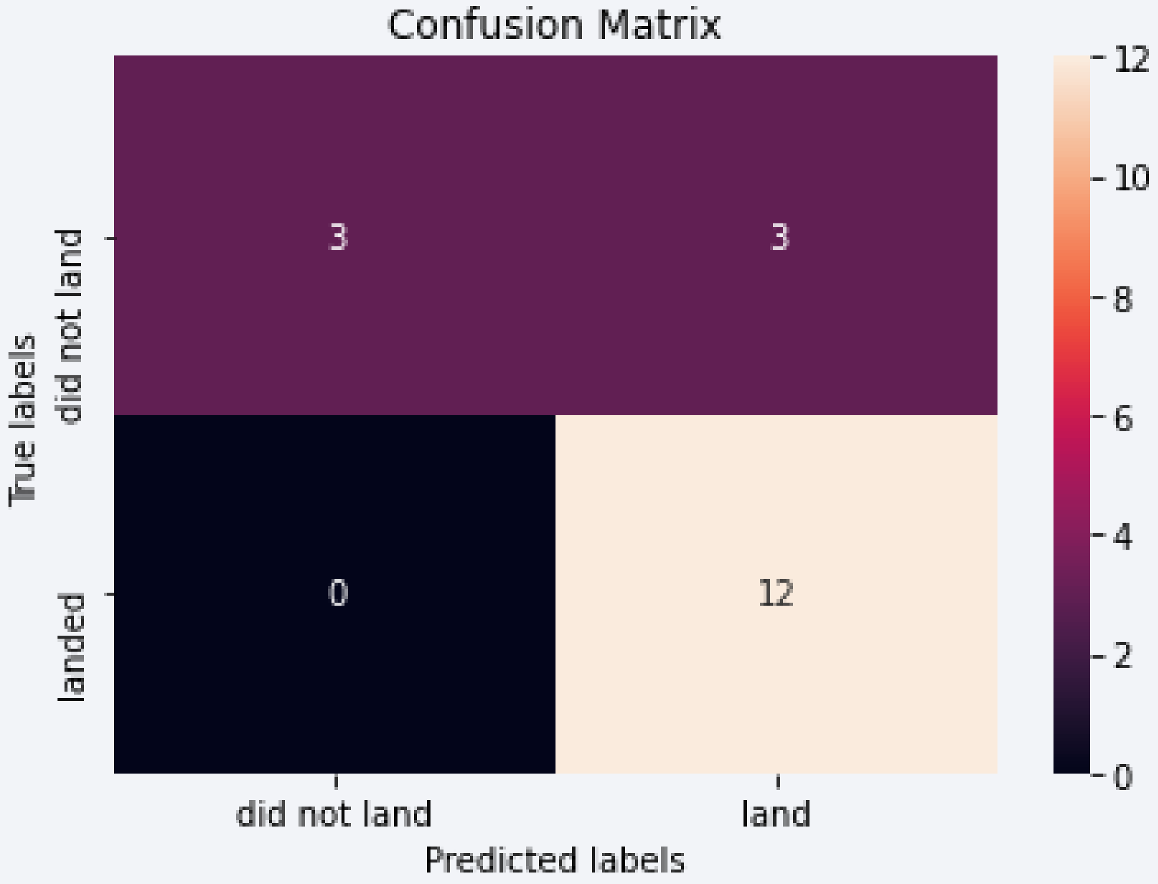
Classification Accuracy

- All four machine learning models provided the same accuracy of 83.3% which is very encouraging
- This could be caused by a low relative test size and it may be beneficial to have more data



Confusion Matrix

- Similar to the accuracy metrics, all four machine learning models produced the same confusion matrix
- The main area of concern is the three successful landing predictions that actually did not land successfully
- Overall the models perform well however more data may be helpful



Conclusions

- Success rates have been trending up over the past 10 years however there is still opportunity for improvement
- It may be helpful to investigate the 2018 dip in success rates
- The machine learning models provide good accuracy however it seems that more data would be helpful
- There have only been a few launches at the historically higher volume Orbit Types (LEO, ISS, PO, GTO) and new Orbit Types have been introduced, potentially playing a role in the improvement
- CCAFS SLC 40 is the most popular launch site and it also has the highest success rate among all sites

Appendix

- GitHub site with all files:
  https://github.com/donaldmears/IBM_Data_Science_Certification

Thank you!