IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Igor Cornejo
2026-Jan-25

Outline

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

Executive Summary

Summary of Methodologies

This project applied a full data science workflow to analyze SpaceX Falcon 9 performance and predict successful landings. Data was collected via the SpaceX API and web-scraped from Wikipedia, then cleaned and integrated using the Pandas library in Python. Exploratory analysis was performed using SQL, Seaborn visualizations, Folium GIS mapping, and an interactive dashboard using the Plotly-Dash Python library.

Summary of all results

The analysis covers launches from **June 4, 2010 to November 11, 2020**. Key findings include the introduction of the fully reusable **Falcon 9 Block 5** booster on **May 11, 2018 (Flight 54)** and a sharp rise in landing success after **Flight 40**. Most launches occurred from **CCAFS SLC-40**, followed by **KSC LC-39A** and **VAFB SLC-4E**, driven by orbit requirements and crewed vs. uncrewed missions. Multiple machine learning models were trained and evaluated using scikit-learn, including Logistic Regression, SVM, Decision Tree, and KNN. The **Decision Tree model** delivered the best predictive performance with **89% accuracy**.

Introduction

Project background and context

SpaceX has become the dominant leader in commercial spaceflight, driven largely by its breakthrough in reusable rocket technology. As of December 2025, SpaceX lists the price of a Falcon 9 launch at approximately **$69.75 million**, compared to the **$165 million or more** typically required for missions using traditional expendable rockets. This cost advantage has reshaped market expectations and intensified competition in the commercial launch sector.

To enter this market, **SpaceY** aims to analyze publicly available data on SpaceX launches and first-stage landings. A successful SpaceX booster recovery directly reduces launch cost. Predicting landing outcomes is critical for estimating potential cost competitiveness. By leveraging insights from SpaceX's performance, **SpaceY** intends to design its own reusable launch system and strategically price missions to compete effectively in the global launch marketplace.

Problem definition

- What are the key technical, operational, and environmental factors that determine the success or failure of a Falcon 9 first-stage landing?
- Can these factors be systematically analyzed to develop a reliable predictive model?
- Which machine learning model delivers the highest accuracy in predicting successful rocket landings?

Section 1

Methodology

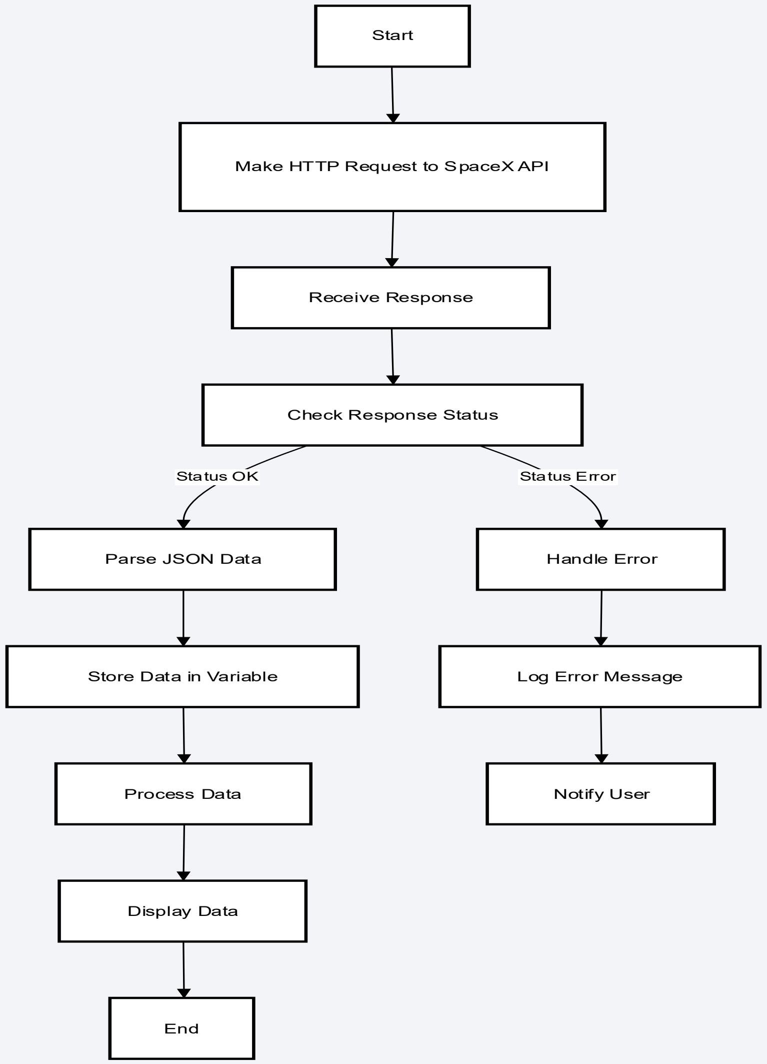
Methodology

Executive Summary

- Data collection methodology:
  - Collected and curated data from the **SpaceX REST API** and web-scraped the **List of Falcon 9 and Falcon Heavy launches** from Wikipedia.
- Perform data wrangling
  - Cleaned, standardized, and merged datasets using Python and Pandas.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Trained multiple classification models (Logistic Regression, SVM, Decision Tree, and KNN). Tuned models using scikit-learn's GridSearchCV feature to identify the best model parameters. Selected the best model based on the accuracy score on the test data.

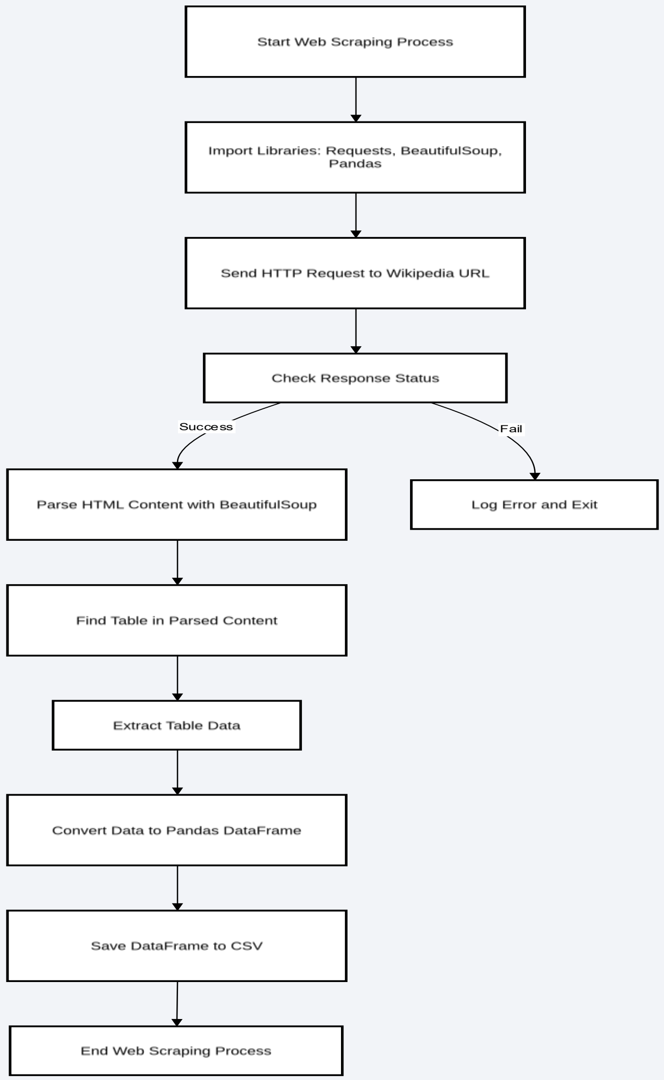
Data Collection – SpaceX API

- The following flowchart represents the steps used to download data from the SpaceX REST API.
- Refer to notebook: 01_jupyter-labs-spacex-data-collection-api-v2.ipynb to review the detailed steps and processes to download and clean the SpaceX REST API dataset.



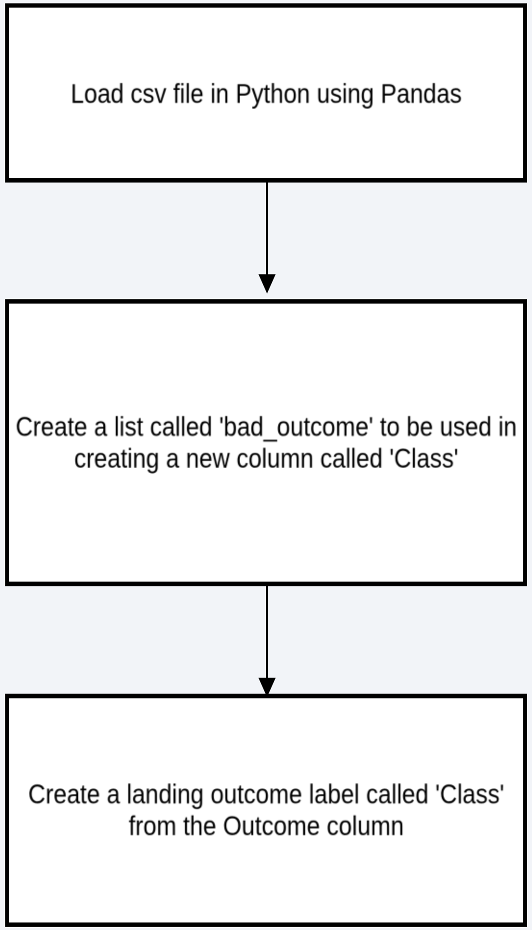
Data Collection - Scraping

- The following flowchart represents the steps used to web-scrape data from the Wikipedia website: "List of Falcon 9 and Falcon Heavy Launches".
- Refer to notebook: 02_jupyter-labs-webscraping-v2.ipynb to review the detailed steps and processes to scrape the data and create a dataset.



Data Wrangling

- For this project, the primary objective of data wrangling is to create a new feature called **Class**. **Class** summarizes the launch and landing outcomes found in the dataset into 2 discrete values:
    - If the booster landing was not successful, **Class** = 0
    - If the booster landing was successful, **Class** = 1
- The **Class** column will be the **Target** used to analyze and predict launch / landing outcomes.
- In addition to creating this Target column, the following operations were performed in this notebook:
    - Calculate the number of launches on each site
    - Calculate the number and occurrence of each orbit
    - Calculate the number and occurrence of mission outcome of the orbits
- Refer to notebook: 03_jupyter-labs-spacex-Data wrangling-v2.ipynb to review the details of these calculations and the creation of the **Class** feature.



EDA with Data Visualization

- In order to understand the relationship between dataset features, and identify any correlations to successful landings, the following graphs will be produced:

1. Categorical Plots
    - Relationship between Flight Number and Pay Load Mass
2. Scatter Plots
    - Relationship between Flight Number and Launch Site
    - Relationship between Pay Load Mass and Launch Site
    - Relationship between Pay Load Mass and Orbit
3. Bar Charts
    - Relationship between Success Rate and Orbit Type
    - Relationship between Success Rate and Falcon 9 Blockk Number
4. Line Graph
    - Relationship between Year and Success Rate

EDA with Data Visualization

- Summary of key findings:
  - The Falcon 9 booster with the best success rate is the Block 2 booster, followed by the Block 5
  - However, the only booster **designed to be fully reusable** is the **Falcon 9 Block 5**
- Refer to notebook: 05_jupyter-labs-eda-dataviz-v2.ipynb for graphs and details

EDA with SQL

- The EDA exercise also included performing the following SQL queries on a database created in SQLite, using the csv created by accessing the SpaceX API and web-scraping Wikipedia:
    - Display the names of the unique launch sites in the space mission
    - Display 5 records where launch sites begin with the string 'KSC'
    - 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 where the successful landing outcome in drone ship was achieved
    - List the names of the boosters which have success in ground pad and have payload mass greater than 4,000 but less than 6,000
    - List the total number of successful and failure mission outcomes
    - List all the booster_versions that have carried the maximum payload mass. Use a subquery
    - List the records which will display the month names, successful landing_outcomes in ground pad, booster versions, and launch_site for the months in year 2017
    - Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

EDA with SQL

- Summary of key findings:
  - The first successful drone ship landing was: April 8, 2016
  - The Falcon 9 Block 5 booster is the only booster version that has carried the maximum payload mass of 15, 600 kg
  - 99 out of 103 launches successfully delivered their payloads into orbit
- Refer to notebook: 04_jupyter-labs-eda-sql-edx-sqllite-v2.ipynb for details

Build an Interactive Map with Folium

- A GIS analysis was performed using the Python Folium library. The following map objects were used to perform this analysis:
  - Circles were used to help locate the following regions of interest:
    - NASA Johnson Space Center
    - Cape Canaveral Air Force Station Space Launch Complex 40
    - Kennedy Space Center Launch Complex 39A
    - Vandenberg Air Force Base Space Launch Complex 4E
  - Markers and Marker Clusters
    - Markers were used to mark between successful landings (Green) cs unsuccessful landings (Red), at each launch site.
    - Marker were also used to identify closest coast line, city, railway, highway and a luanch site
    - Marker Clusters were used to group landings together in an organized fashion, at each launch site.
  - Lines
    - Lines were used to denote distances between launch sites, and closest coast line, city, railway, highway
  - Property Boundaries
    - The boundary of each launch site was added as a GeoJSON file to the map
- These objects aided in assessing any relationship between launch site and landing success, in addition to logistical planning of transporting material to the launch site.
- Refer to notebook: 06_jupyter-labs-launch-site-location-v2.ipynb for details.

Build a Dashboard with Plotly Dash

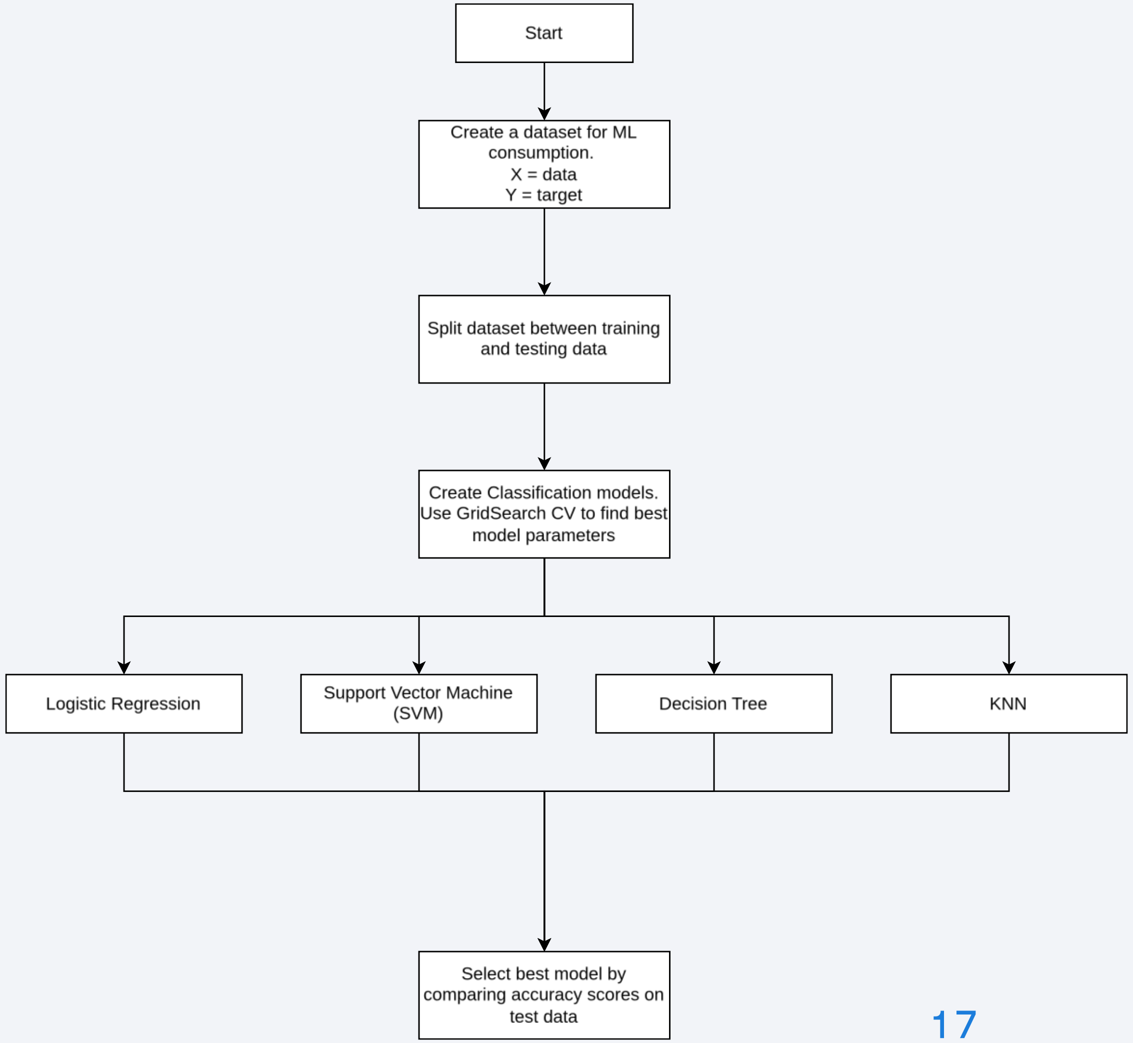
- An interactive dashboard was created using the Plotly Dash Python library
- A Pie-chart is used to understand which launch site has the best success rate
    - Users are able to select between the following launch sites:
        - All (The % success rate of each launch site as an average of the 3 launch sites)
        - CCAFS SLC-40 (% Comparison between successful vs. unsuccessful launches)
        - KSC LC-LC 39A (% Comparison between successful vs. unsuccessful launches)
        - VAFB SLC-4E (% Comparison between successful vs. unsuccessful launches)
- A Scatter-plot is used to understand the relationship between pay load mass and successful launches.
    - In addition to selecting a launch site, users are able to select a pay load mass between 0 and 9600 kg using a range slider
- These plots are meant to understand if there is a relationship between successful launches, payload mass, and booster version for the user selected launch site

Build a Dashboard with Plotly Dash

- Summary of key findings:
  - VAFB SLC-4E is the launch site with the worst success rate
  - The Falcon 9 Block 5 booster has only launched from KSC LC-39A
  - Payloads up to 5,500 kg have 100% success rates. Payloads > 5,500 kg have the lowest success rate
- Refer to:
  - Notebook: 07_jupyter-labs-spacex_dash-app.ipynb for details on the code
  - Spacex-dash-app.py for the python version of the file that can be run as a script

Predictive Analysis (Classification)

- The flowchart summarizes the steps and process used to build and select a model
- Machine learning models were built using the Scikit-Learn Python library. Scikit-Learn was also used to automate the process of choosing the best model parameters
- A model was chosen by comparing model performance accuracy, on test data, and selecting the best performing model
- Refer to notebook: 08_jupyter-labs-SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb for details on model building, and selection

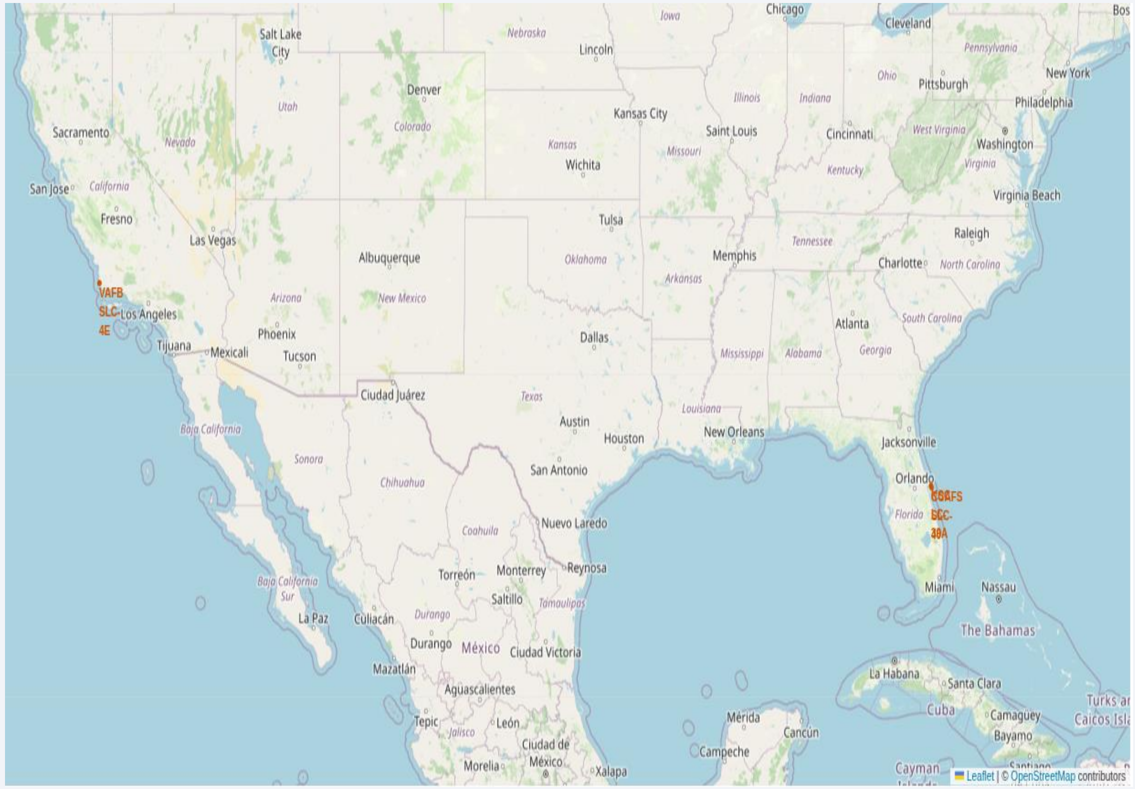


Results

- Exploratory data analysis results
  - The best predictor of a successful landing is the Falcon 9 Booster Version
  - The Falcon 9 Booster with the best success rate is the Block 2 booster, followed by the Block 5
  - The **Falcon 9 Block 5** is the only booster designed to be fully reusable
  - The first Block 5 mission was May 11, 2018 (Flight number: 54)
  - Block 5 rockets have an 85% success rate

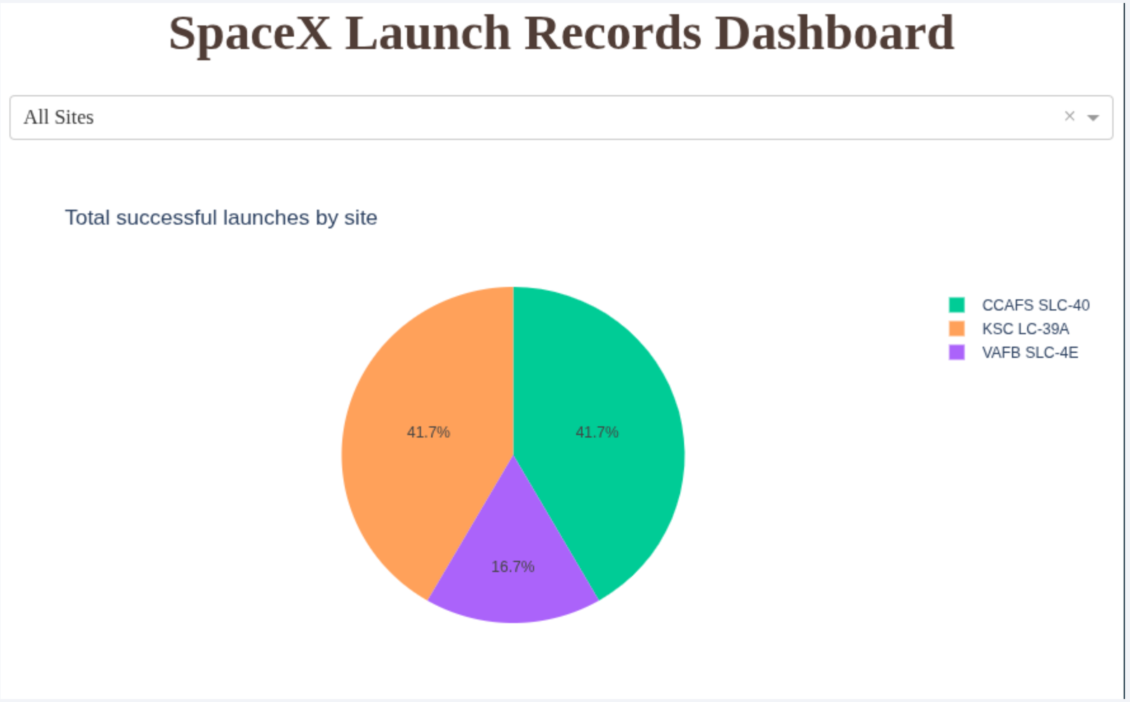
Results

- Interactive analytics demo in screenshots
  - SpaceX uses these 3 launch sites:
    - Cape Canaveral Air Force Station Space Launch Complex 40 (CCAFS SLC-40)
    - Kennedy Space Center Launch Complex 39A (KSC LC-39A)
    - Vandenberg Air Force Base Space Launch Complex 4E (VAFB SLC-4E)
  - The key driver to selecting a launch site are Orbit and mission requirements



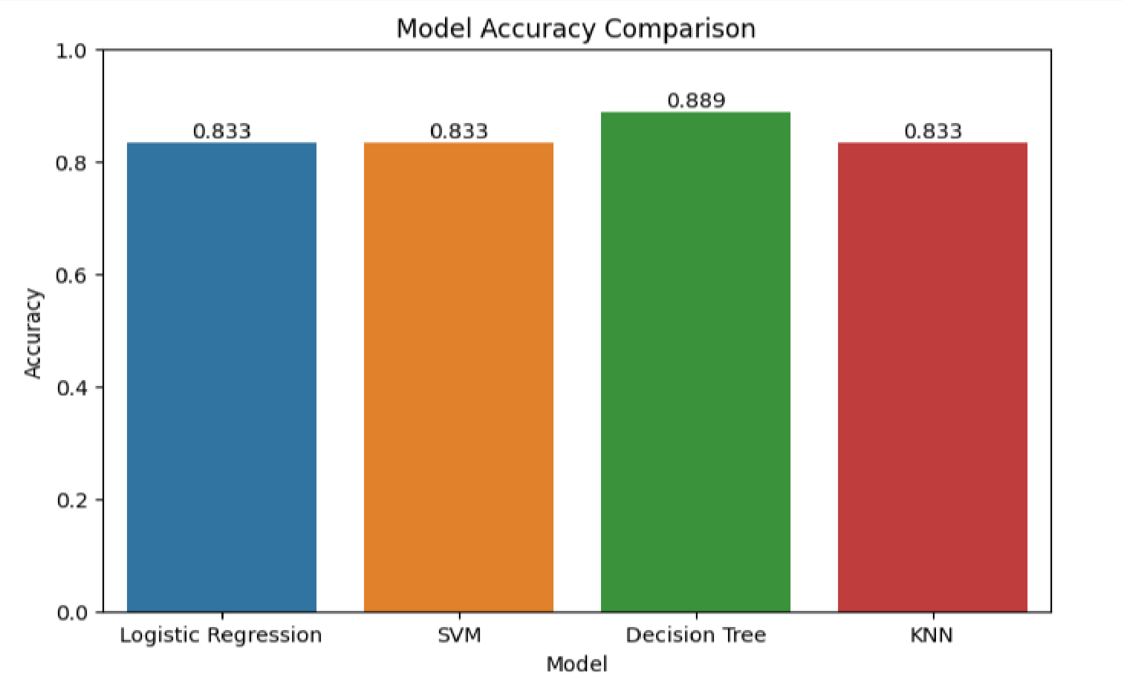
Results

- Interactive analytics demo in screenshots
    - CCAFS SLC-40 and KSC LC-39A have the same success rate



Results

- Predictive analysis results
  - All models performed well
  - Decision Tree classifier was chosen for having the best model performance

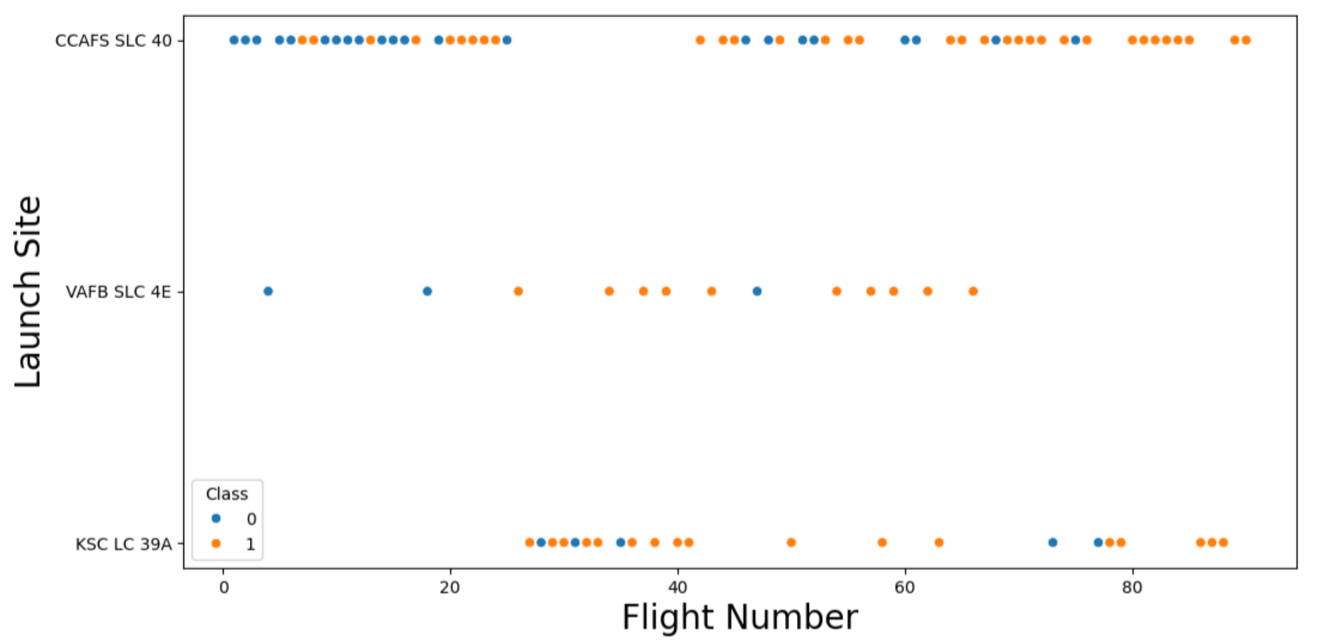


Section 2

Insights drawn from EDA

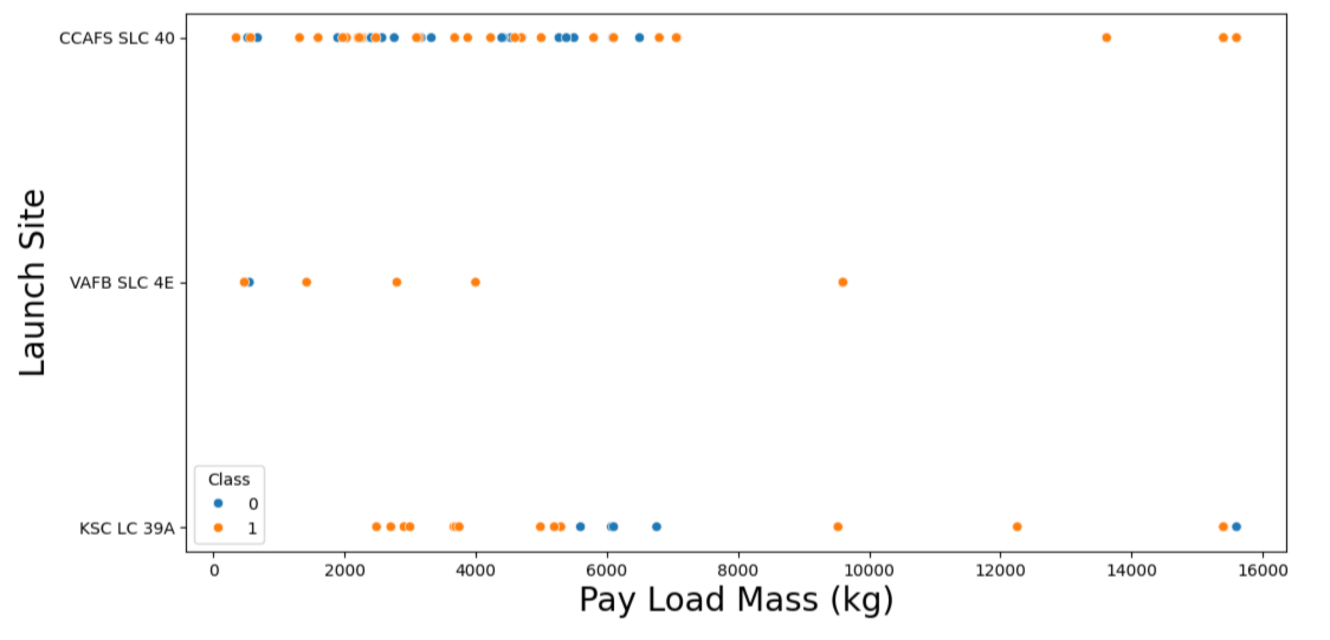
Flight Number vs. Launch Site

- Most launches have occurred at SLC-40, followed by LC-39A
- As expected, increasing flight numbers increases the likely hood of success
- Best success rates are in flight numbers > 40



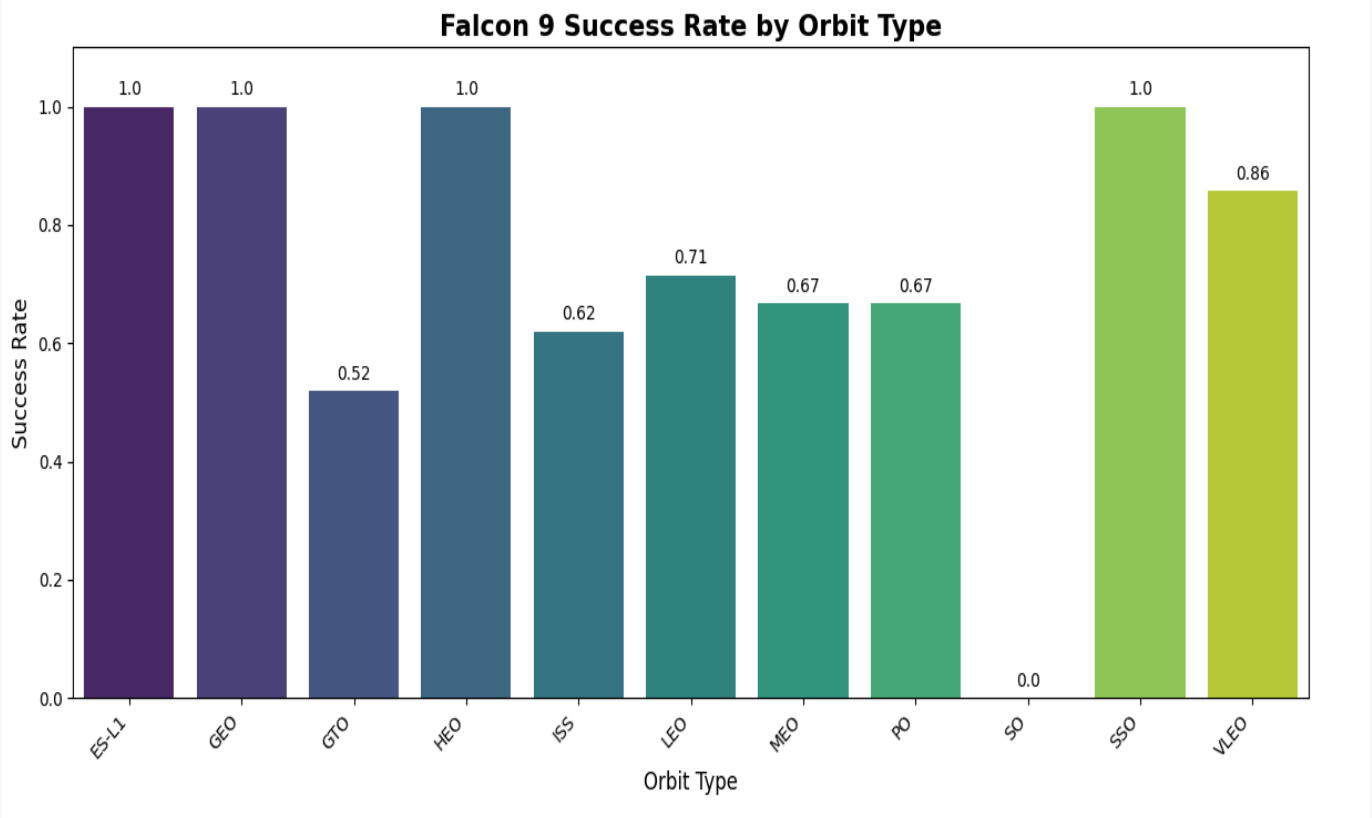
Payload vs. Launch Site

- CCAFS SLC-40 has launched the most diverse range of payload mass
- There is no correlation between payload mass and launch site



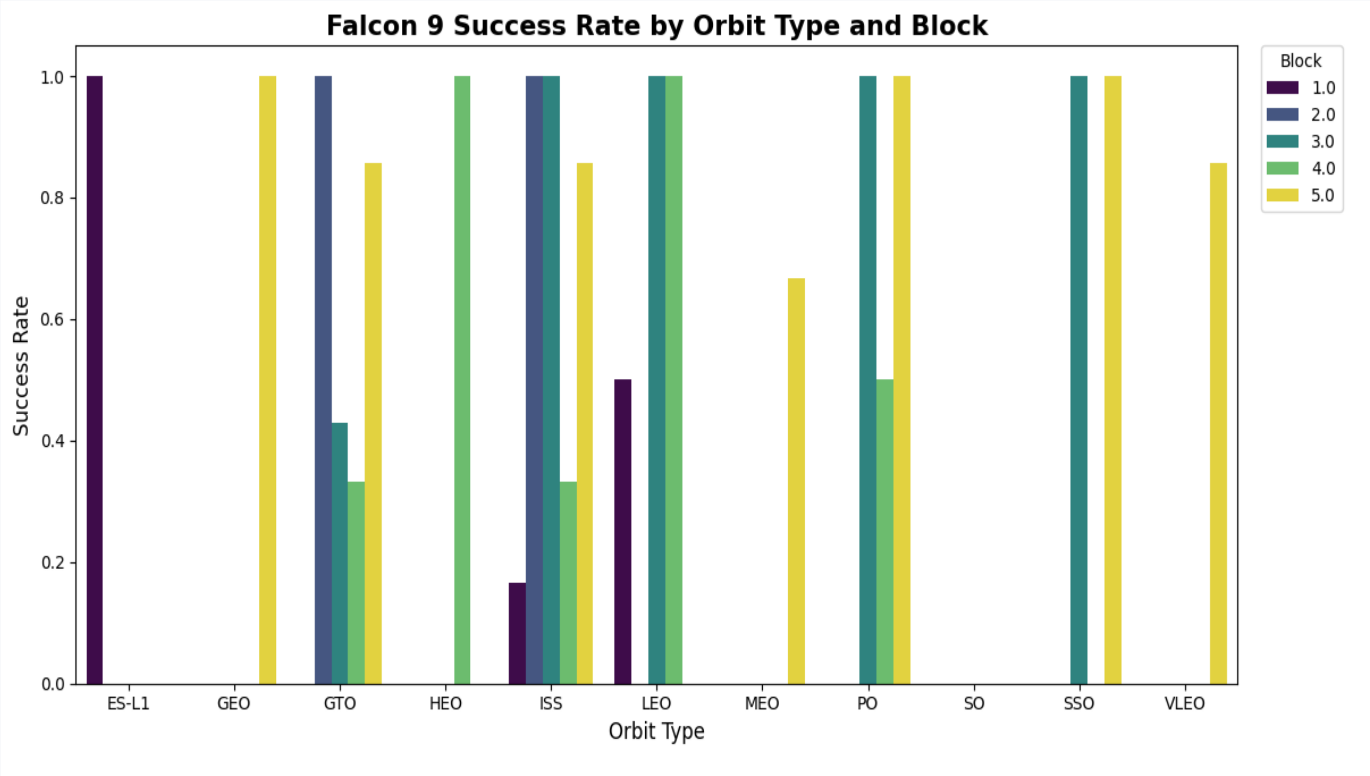
Success Rate vs. Orbit Type

- Some orbits have 100% success: ES-L1, GEO, HEO, SSO
- To date, launches to SO orbit have not been successful
- Orbit should be considered a factor in success
- Understanding the frequency of launches to each orbit, and block number would help in better understanding significance of this factor



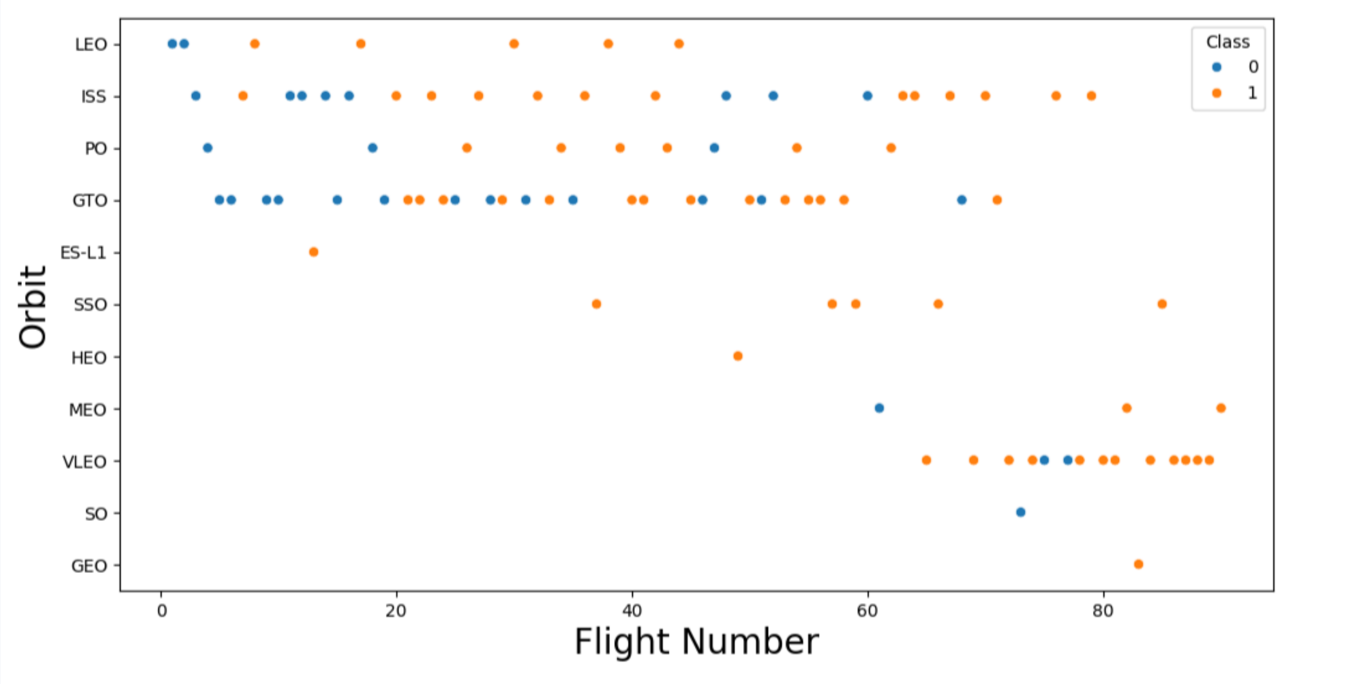
Success Rate vs. Orbit Type and Block

- The Block 5 has a high success rate in several orbits
- Block 5 boosters is key to having success



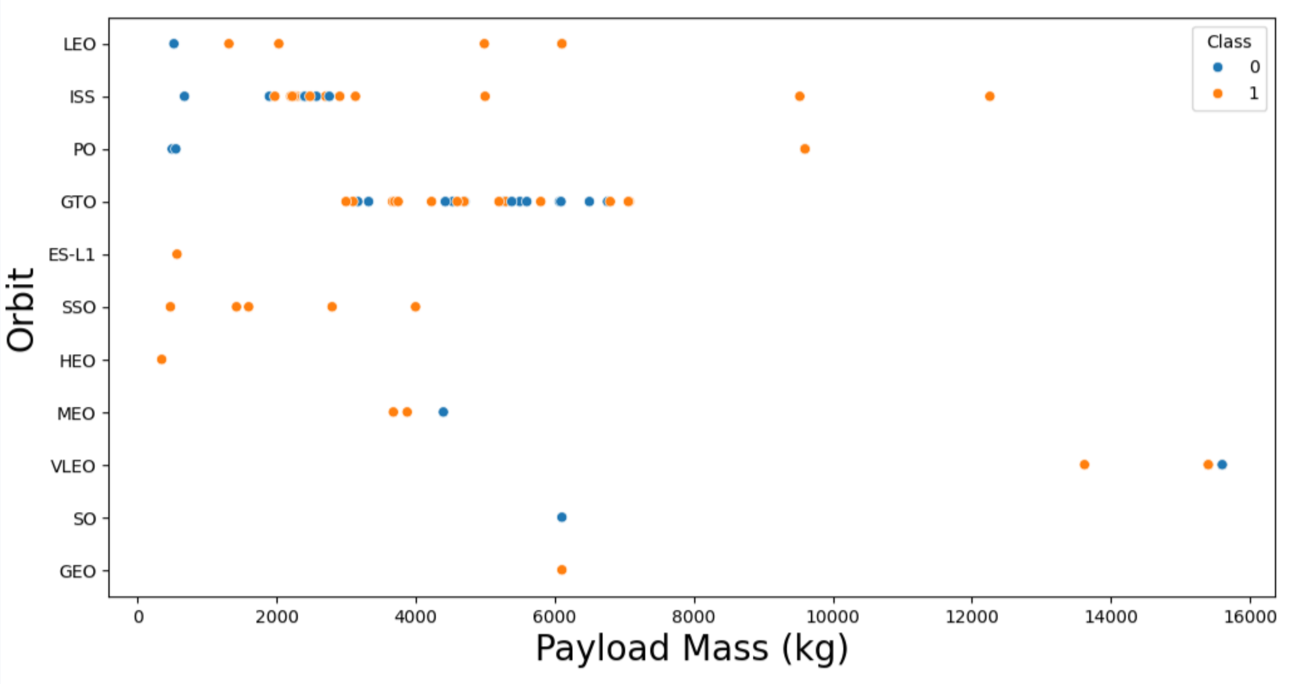
Flight Number vs. Orbit Type

- As expected, higher flight numbers have the best chance of success
- Flight Numbers > 40 have the most success



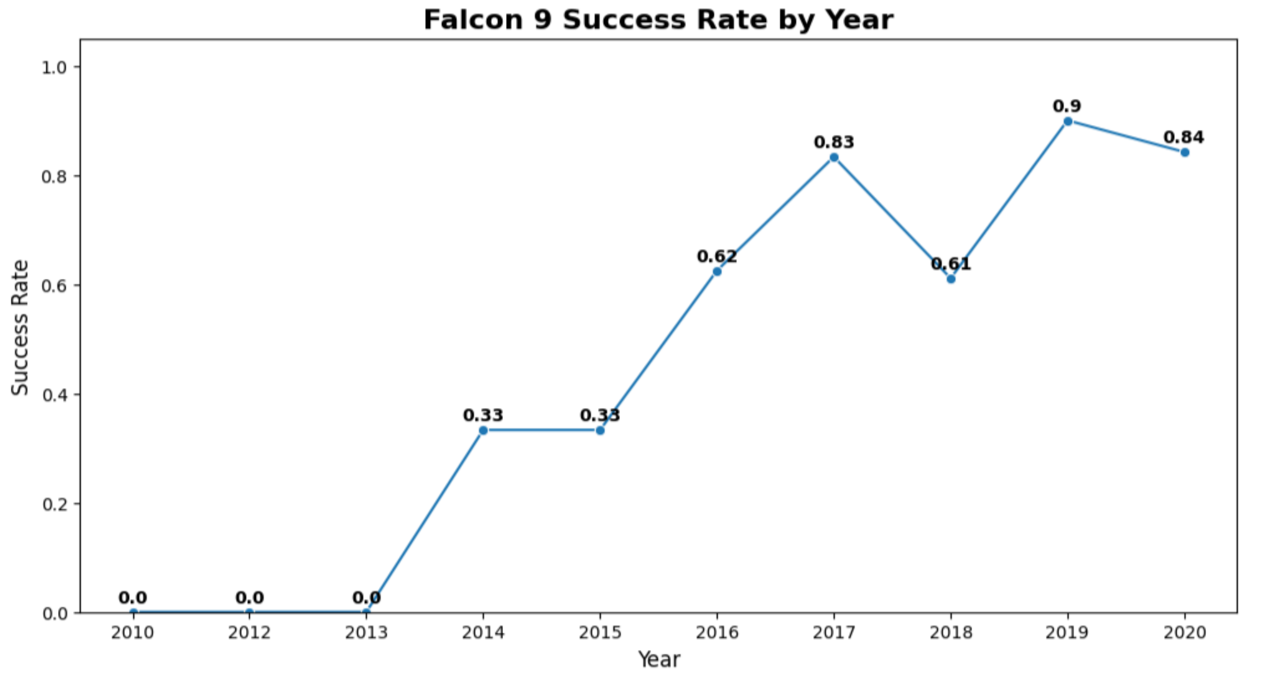
Payload vs. Orbit Type

- There is no correlation between Payload Mass and Orbit Type



Launch Success Yearly Trend

- Since 2013, success rates significantly increased
- After 2015, Falcon 9 boosters have had a high success rate
- The year over year increase in success rates can be explained by the introduction of new Booster Versions, ones designed to be fully reusable



All Launch Site Names

- The names of the unique launch sites used by SpaceX Falcon 9 are:
    - Cape Canaveral Air Force Station Space Launch Complex 40 (CCAFS SLC-40)
    - Kennedy Space Center Launch Complex 39A (KSC LC-39A)
    - Vandenberg Air Force Base Space Launch Complex 4E (VAFB SLC-4E)
- The following SQL query was used to obtain these results
- **NOTE: CCAFS LC-40 and CCAFS SLC-40 are the same launch site. LC-40 was renamed SLC-40 when SpaceX started leasing this site**

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

[12]:
```
%%sql
SELECT DISTINCT launch_site
    FROM spacextbl
    GROUP BY launch_site;
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

[12]:

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

Launch Site Names Begin with 'KSC'

- SQL was used to query the database with specific search criteria
- Below is a sample query, and the result, of finding **Launch Site** names that being with 'KSC'

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

[13]:
```
%%sql
SELECT *
    FROM spacextbl
    WHERE launch_site
    LIKE 'KSC%'
    LIMIT 5;
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

[13]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

- SQL was used to calculate the total mass launched by NASA CRS
- Below is the query and the calculated mass (45,596 kg)

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

[14]:
```
%%sql
SELECT SUM(payload_mass__kg_) as "Total payload mass (kg) launched by NASA (CRS)"
    FROM spacextbl
    WHERE customer
    LIKE 'NASA (CRS)';
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

[14]:

Total payload mass (kg) launched by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

- Using SQL, we calculated the average payload mass launched by the F9 v1.1 booster version
- Below is the SQL and query result (2,928.4 kg)

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

[15]:
```
%%sql
SELECT AVG(payload_mass__kg_) AS "Average payload mass (kg) launched by booster version F9v1.1"
    FROM spacextbl
    WHERE booster_version
    LIKE 'F9 v1.1';
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

[15]:

Average payload mass (kg) launched by booster version F9v1.1
2928.4

First Successful Ground Landing Date

- The database was queried to find the dates of the first successful landing outcome on a drone ship
- Below is the query and result (2016-04-08)

List the date where the succesful landing outcome in drone ship was acheived

Hint:Use min function

[16]:
```
%%sql
SELECT MIN(date) as "First successful drone ship landing (YYYY-MM-DD)"
    FROM spacextbl
    WHERE landing_outcome
    LIKE 'Success (drone ship)';
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

[16]:

First successful drone ship landing (YYYY-MM-DD)
2016-04-08

Successful Ground Pad Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on a ground pad and had payload mass greater than 4000 but less than 6000 is shown below
- Below is the SQL query, and query result. Notice the booster versions that have successfully landed on a ground pad are either F9 B4 (Block 4) or F9 FT (Full Thrust).

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

[17]:
```
%%sql
SELECT booster_version, payload_mass__kg_
    FROM spacextbl
    WHERE payload_mass__kg_ > 4000
    AND payload_mass__kg_ < 6000
    AND landing_outcome LIKE 'Success (ground pad)'
    GROUP BY payload_mass__kg_;
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

[17]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B4 B1040.1	4990
F9 B4 B1043.1	5000
F9 FT B1032.1	5300

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes is shown below
- The following SQL query, and result, demonstrates the high success rate of SpaceX Falcon 9 boosters

List the total number of successful and failure mission outcomes

[18]:
```
%%sql
SELECT TRIM(mission_outcome) AS mission_outcome, COUNT(mission_outcome)
    FROM spacextbl
    GROUP BY TRIM(mission_outcome);
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

[18]:

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

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass is shown in the following image
- The following SQL query, and query result, show the script used to produce the list
- Notice that booster version are all Falcon 9 Block 5 (B5) boosters?

List all the booster_versions that have carried the maximum payload mass. Use a subquery.

[19]:
```
%%sql
SELECT booster_version, payload_mass__kg_
    FROM SPACEXTBL
    WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL)
    GROUP BY booster_version;
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

[19]:

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

2017 Launch Records

- The following image is the output of an SQL query that lists the records containing: successful landing outcomes in ground pad, booster version, and launch site by month in 2017
- The complete SQL query is also shown in the image
- The majority of these launches were from KSC LC-39A

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

Note: SQLLite does not support monthnames. So you need to use substr(Date,6,2) for month, substr(Date,9,2) for date, substr(Date,0,5),='2017' for year.

[20]:

```
%%sql
SELECT
    -- Convert the month digit to a month name
    CASE
        CAST(STRFTIME('%m', Date) AS INTEGER)
        WHEN 1 THEN 'January'
        WHEN 2 THEN 'February'
        WHEN 3 THEN 'March'
        WHEN 4 THEN 'April'
        WHEN 5 THEN 'May'
        WHEN 6 THEN 'June'
        WHEN 7 THEN 'July'
        WHEN 8 THEN 'August'
        WHEN 9 THEN 'September'
        WHEN 10 THEN 'October'
        WHEN 11 THEN 'November'
        WHEN 12 THEN 'December'
    END AS Month,
    Booster_Version,
    Launch_Site,
    Landing_Outcome
FROM spacextbl
WHERE
    STRFTIME('%Y', Date) = '2017'
    AND LOWER(Landing_Outcome) LIKE '%success%'
    AND LOWER(Landing_Outcome) LIKE '%ground pad%'
ORDER BY CAST(STRFTIME('%m', Date) AS INTEGER);
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

[20]:

Month	Booster_Version	Launch_Site	Landing_Outcome
February	F9 FT B1031.1	KSC LC-39A	Success (ground pad)
May	F9 FT B1032.1	KSC LC-39A	Success (ground pad)
June	F9 FT B1035.1	KSC LC-39A	Success (ground pad)
August	F9 B4 B1039.1	KSC LC-39A	Success (ground pad)
September	F9 B4 B1040.1	KSC LC-39A	Success (ground pad)
December	F9 FT B1035.2	CCAFS SLC-40	Success (ground pad)

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

- The rank and count of landing outcomes such as, Failure (drone ship) or Success (ground pad)), between the date 2010-06-04 and 2017-03-20, in descending order, is shown in the image
- The SQL query and query result is shown in the image
- The 'No attempt' is more likely to have occurred between 2010 and 2013

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

[21]:

```
%%sql
SELECT landing_outcome AS 'Landing Outcome', COUNT(landing_outcome) AS 'Total Count'
        FROM spacextbl
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY landing_outcome
        ORDER BY COUNT(landing_outcome) DESC;
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

[21]:

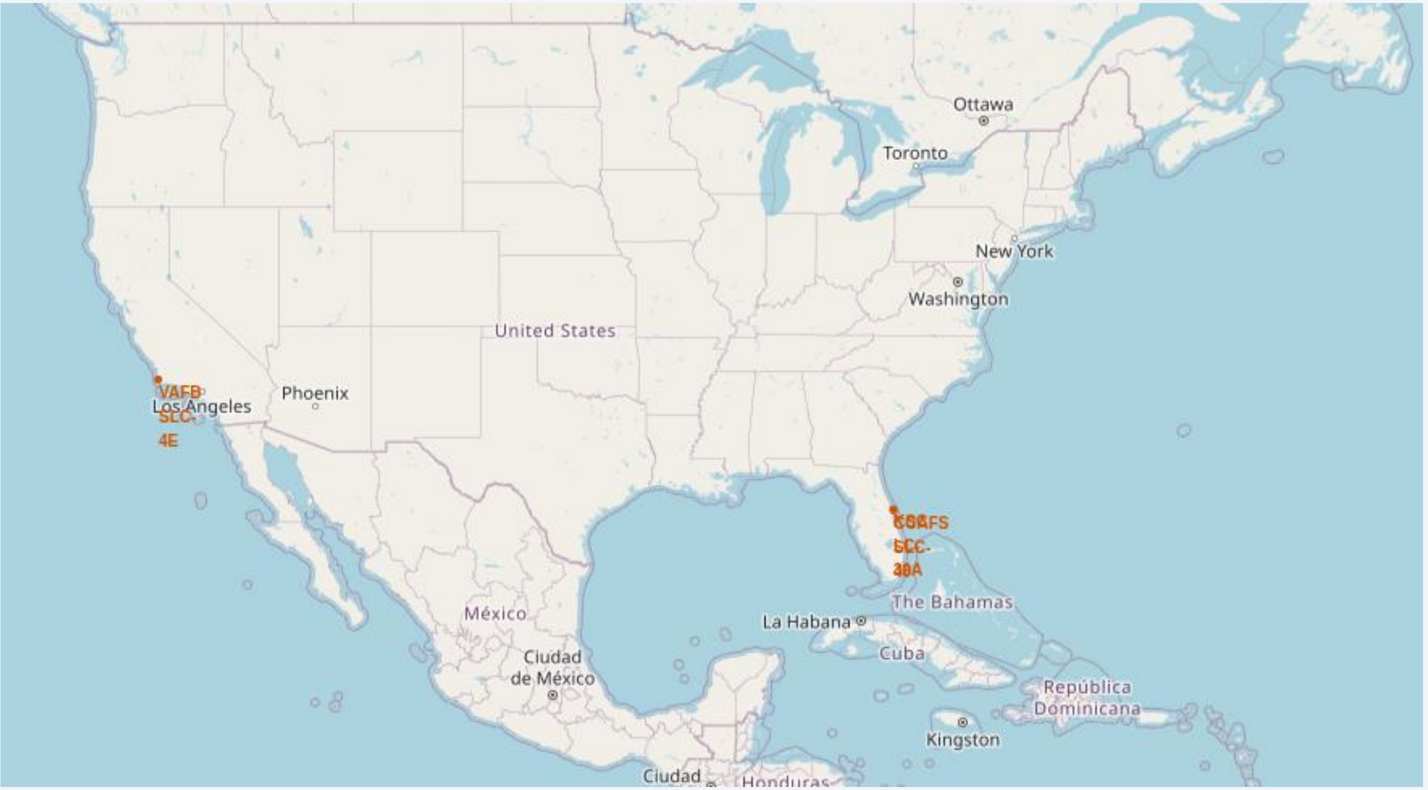
Landing Outcome	Total Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

SpaceX Falcon 9 Launch Sites

- SpaceX Falcon 9 boosters can be launched from either the west coast or east coast
- West coast launches occur at:
  - Vandenberg Air Force Base (VAFS SLC-4E)
- East coast launches occur at:
  - Cape Canaveral Air Force Station Space Launch Complex 40 (CCAFS SLC-40)
  - Kennedy Space Center Launch Complex 39A (KSC LC-39A)
- The decision on the launch site is based on the Orbit and Mission requirements



SpaceX Falcon 9 Launch Sites

- The following table is from SpaceX' Falcon User's Guide (Version 8)
- There are some orbit types that require a specific launch site

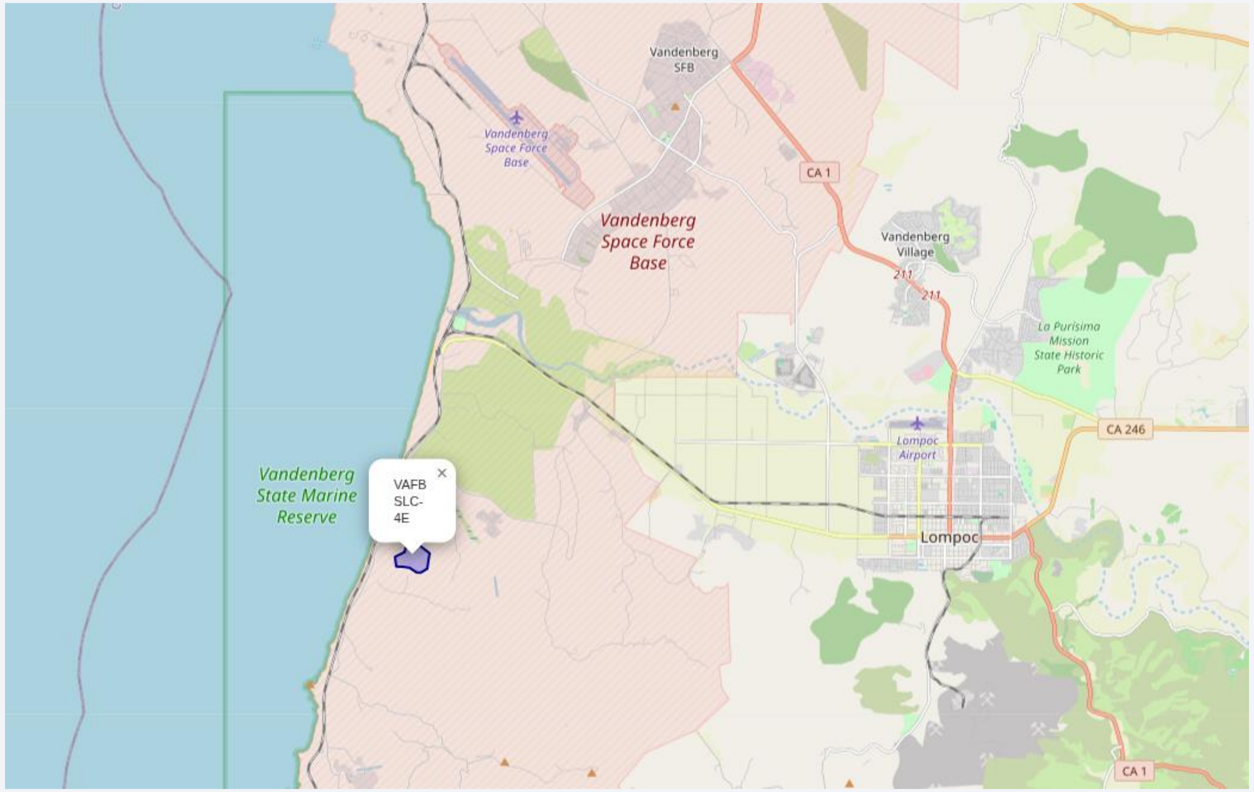
Table 3-1: Falcon 9 and Falcon Heavy Launch Services

Insertion Orbit	Inclination Range	Vehicle	Launch Site(s)
LEO	28.5 – 55 deg	Falcon 9 or Falcon Heavy	Eastern Range
LEO	55 – 65 deg*	Falcon 9 or Falcon Heavy (Eastern Range only)	Eastern or Western Range
LEO	65 – 85 deg	Falcon 9	Western Range
LEO / Retrograde	105+ deg	Falcon 9	Western Range
LEO Polar / SSO	85 – 105 deg*	Falcon 9	Western or Eastern Range
GTO	Up to 28.5 deg	Falcon 9 or Falcon Heavy	Eastern Range
GSO	Up to 28.5 deg	Falcon Heavy	Eastern Range
Earth escape	N/A	Falcon 9 or Falcon Heavy	Western or Eastern Range

*Subject to mission-specific performance considerations. Falcon Heavy is only available from the Eastern Range

SpaceX Falcon 9 Launch Site – West Coast

- Vandenberg Air Force Base (VAFS SLC-4E) is used for LEO orbits that require high inclinations
- VAFS SLC-4E can also be used with the following Orbits: PO/SSO, and earth escape trajectories

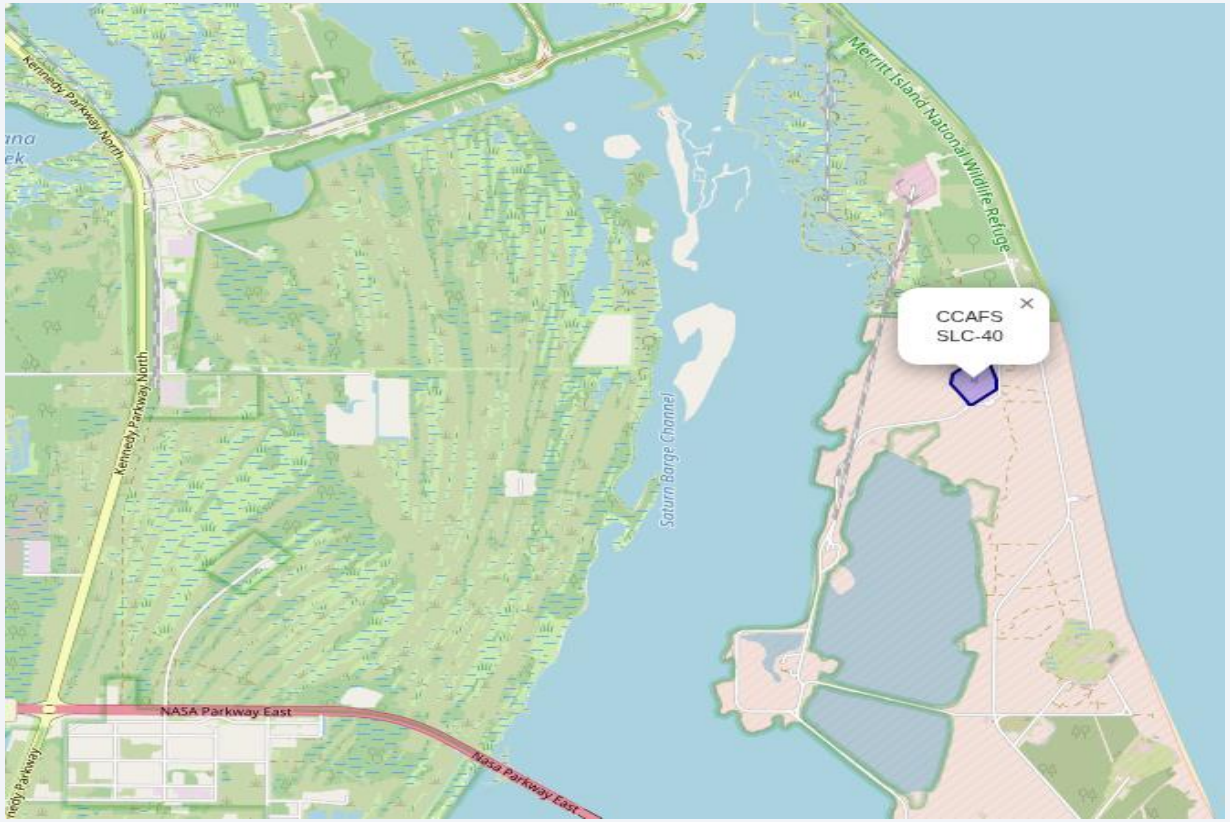


SpaceX Falcon 9 Launch Site – East Coast

- East coast launches occur at either:
  - Cape Canaveral Air Force Station Space Launch Complex 40 (CCAFS SLC-40)
  - Kennedy Space Center Launch Complex 39A (KSC LC-39A)
- KSC LC-39A launches manned Falcon 9 flight missions, missions to the ISS, and Falcon Heavy launches
- CCAFS SCL-40 launches all other Falcon 9 missions. Most Falcon 9 launches have occurred at SLC-40

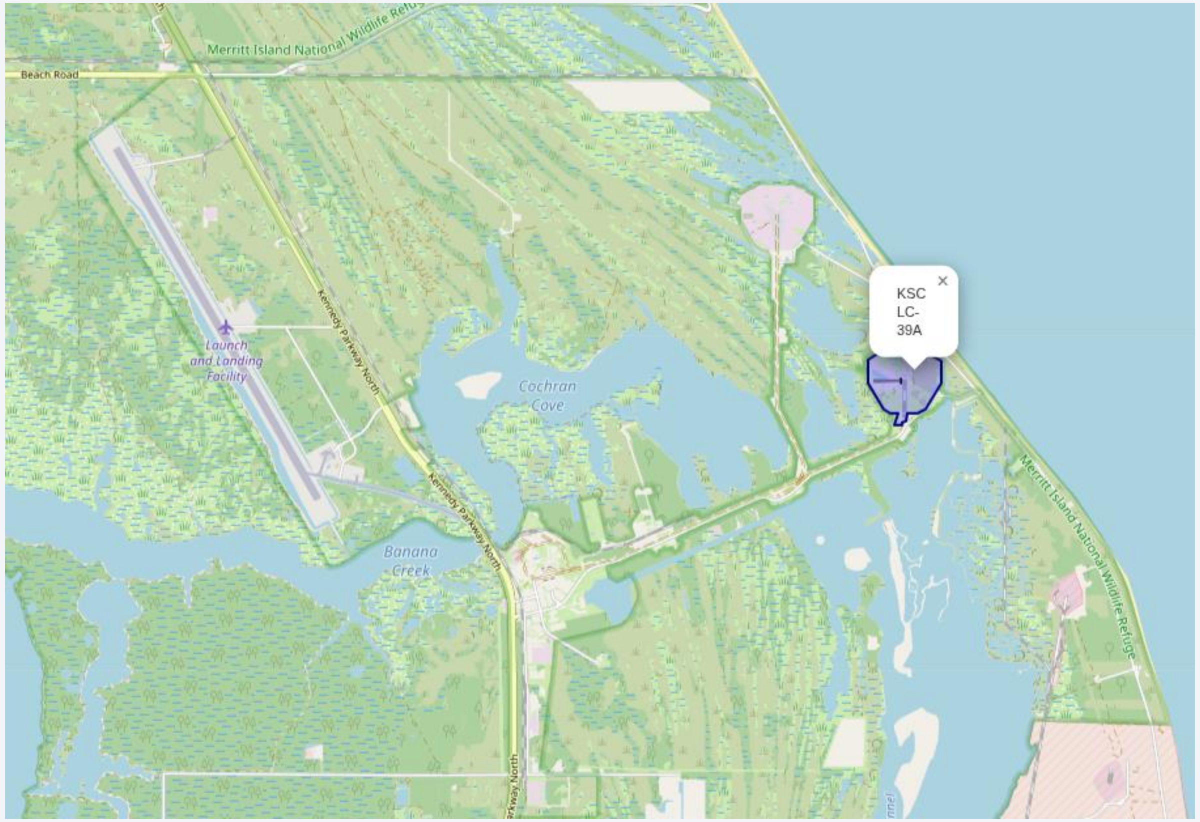
SpaceX Falcon 9 Launch Site – East Coast

- CCAFS SLC-40 is SpaceX's work horse launch site
- Over 60% of all SpaceX launches have occurred at this launch site
- CCAFS SLC-40 is configured to operate a high volume of launches



SpaceX Falcon 9 Launch Site – East Coast

- KSC LC-39A is configured to handle manned crews
- This launch site was previously used by the Space Shuttle and Apollo space programs
- Today SpaceX also uses this launch site for missions to the ISS, and Falcon Heavy launches

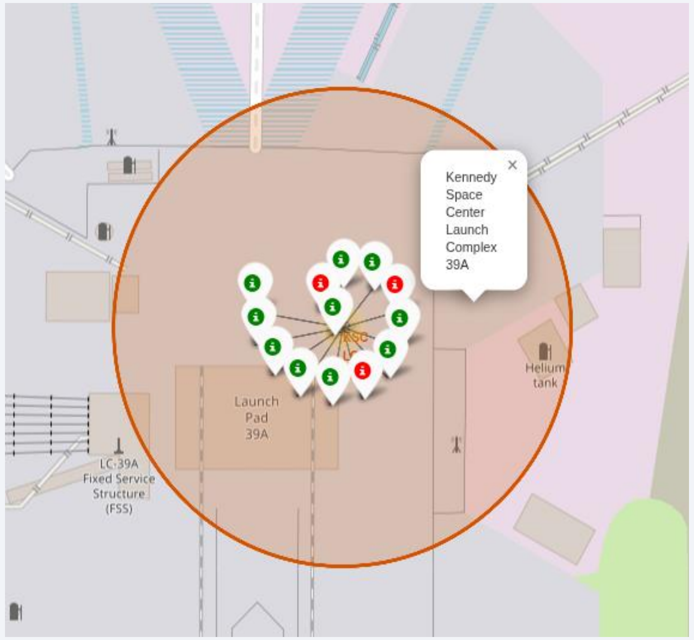


Launch Count and Outcome by Launch Site

- The total launches at each launch site have been color coded:
  - Red = Unsuccessful landing
  - Green = Successful Landing
- SLC-40 has launched the most rockets, LC-39A has the best success rate

CCAFS SLC-40

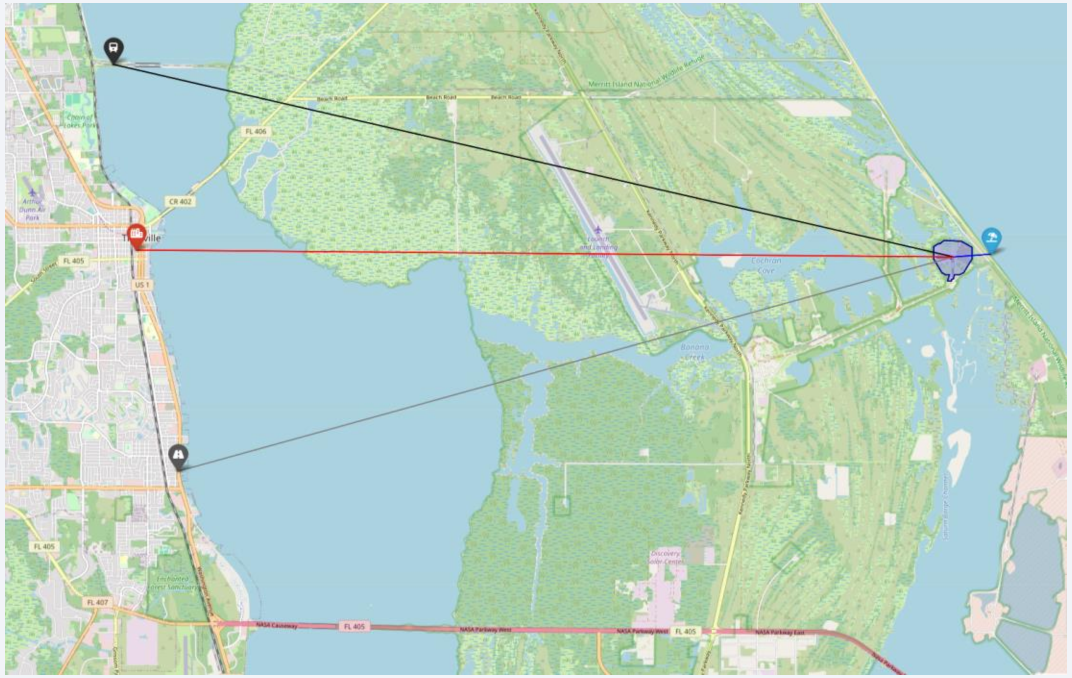
KSC LC-39A



VAFB SLC-4E

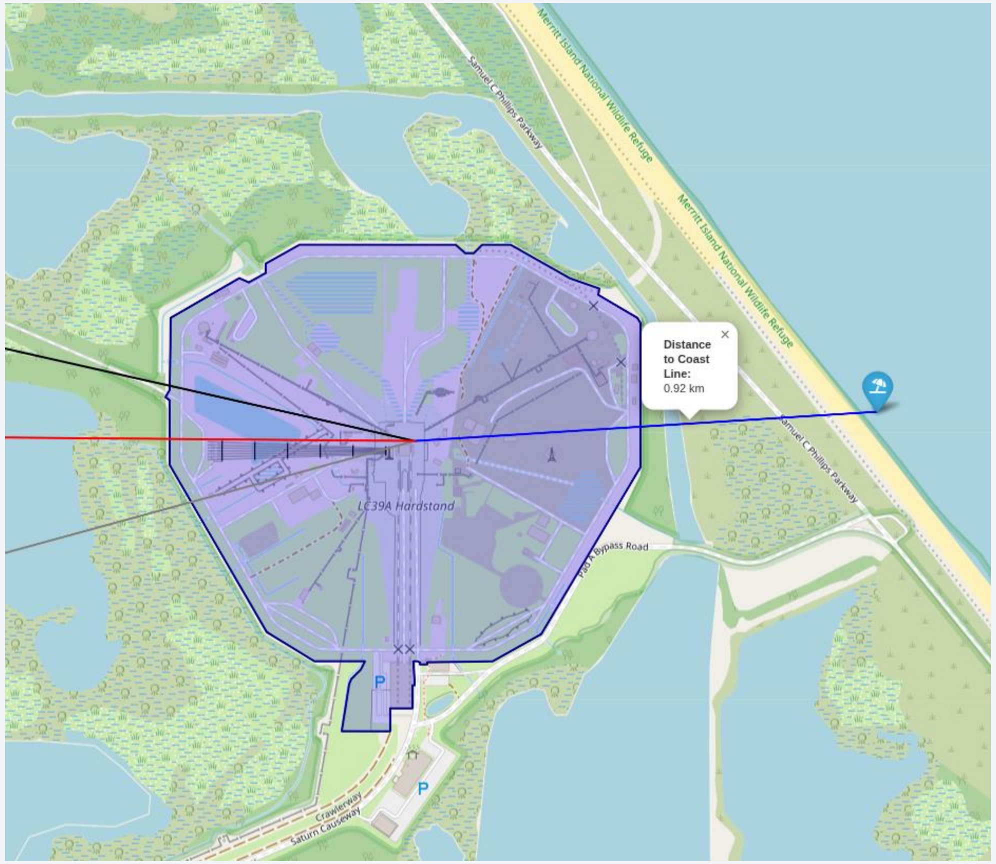
KSC LC-39A Proximity to railway, highway, and coastline

- KSC LC-39A was previously used by the Space Shuttle and Apollo programs
- This makes KSC LC-39A an ideal launch site, given the in-place infrastructure that can be used to transport and assemble rocket components and payload at this site



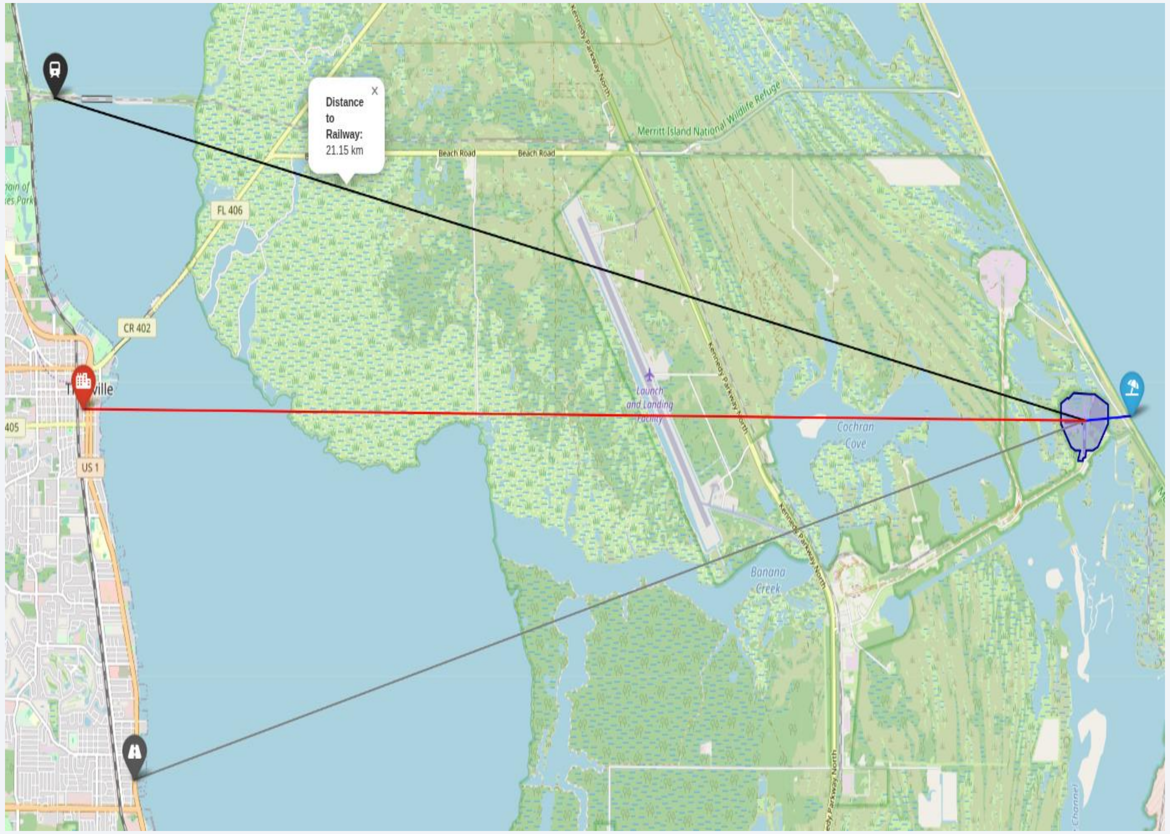
KSC LC-39A Proximity to railway, highway, and coastline

- The distance from KSC LC-39A to the coast is 0.92 km



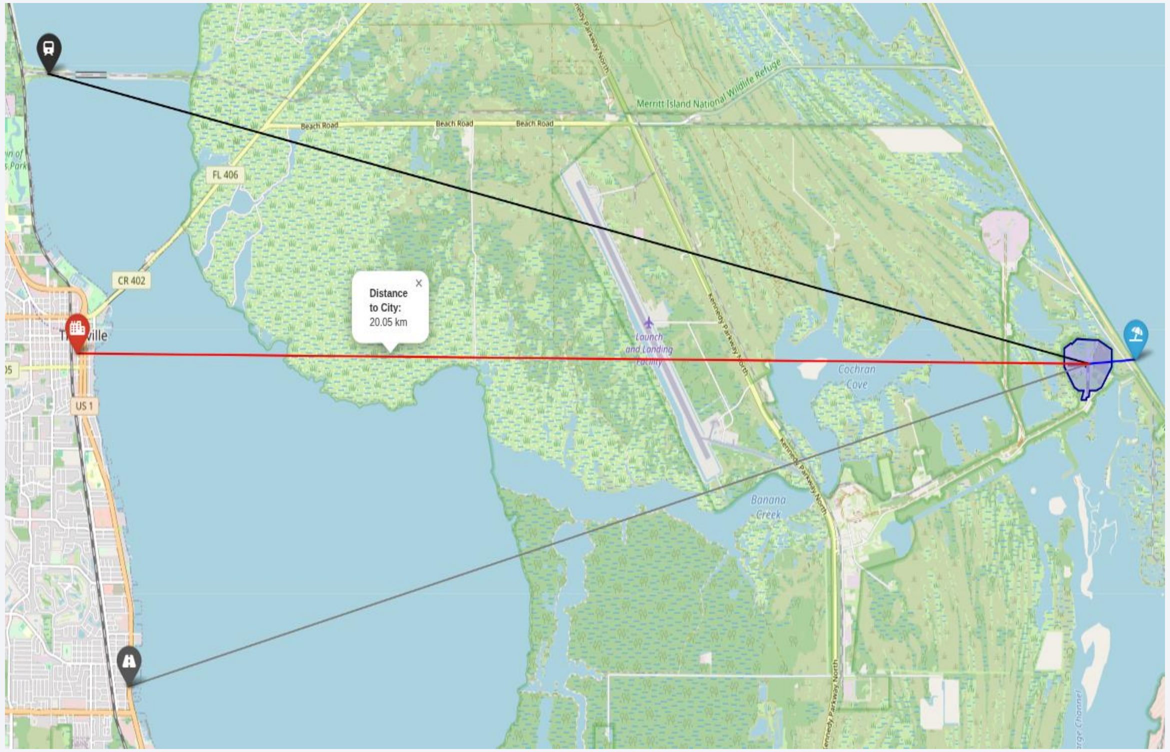
KSC LC-39A Proximity to railway, highway, and coastline

- The distance from KSC LC-39A to the closest railway hub is 25 km



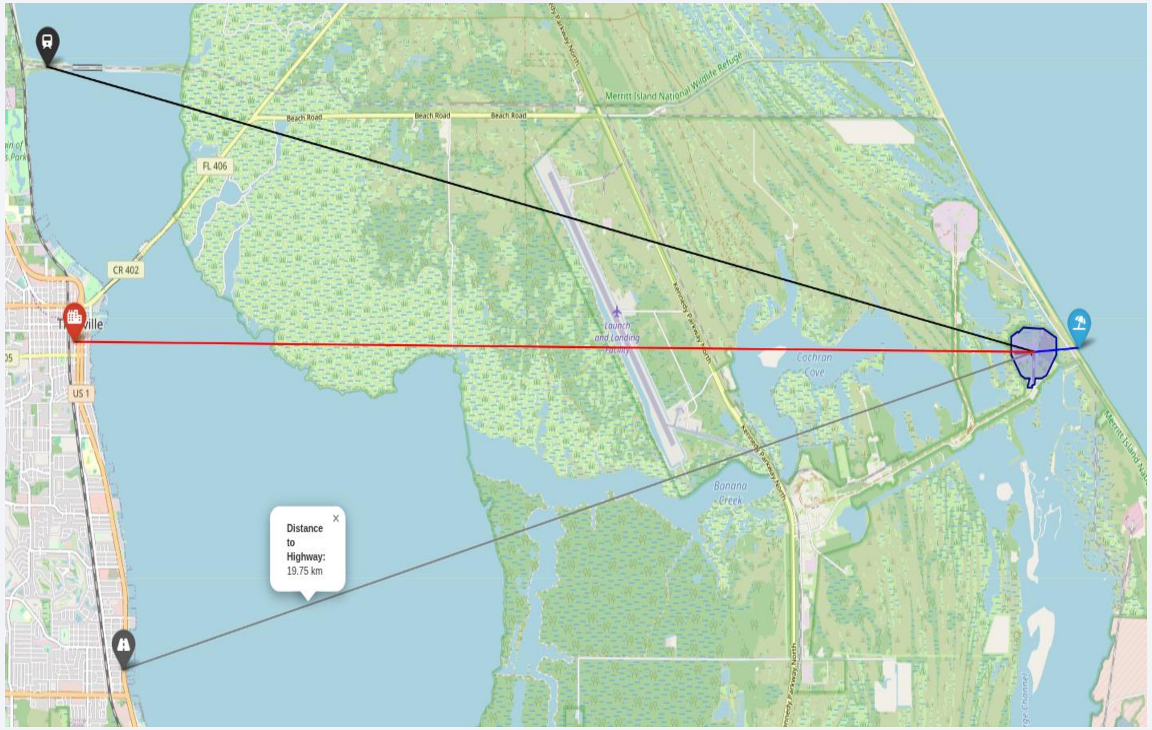
KSC LC-39A Proximity to railway, highway, and coastline

- The distance from KSC LC-39A to the closest city (Titusville) is 20 km



KSC LC-39A Proximity to railway, highway, and coastline

- The distance from KSC LC-39A to the closest highway is ~20 km

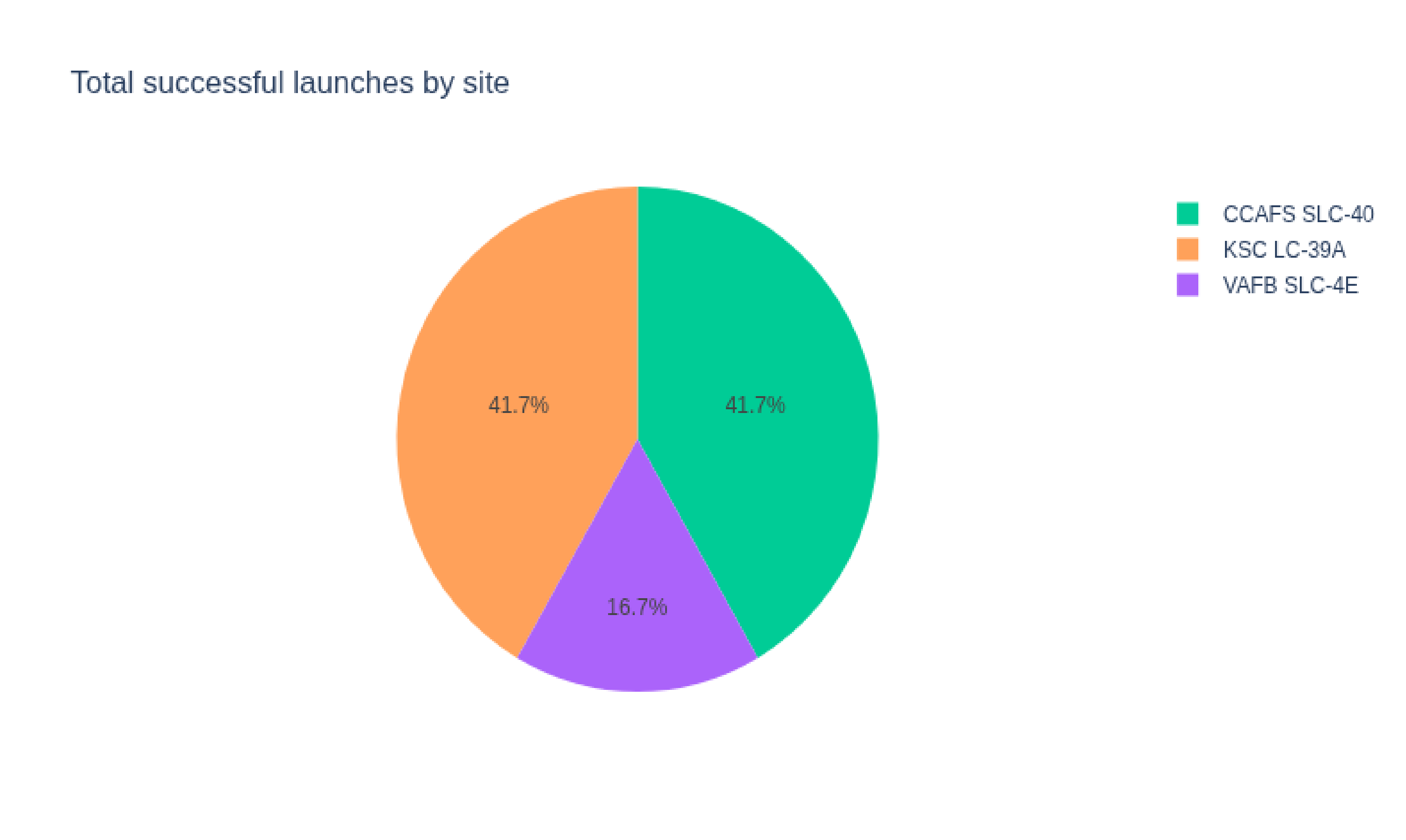


Section 4

Build a Dashboard with Plotly Dash

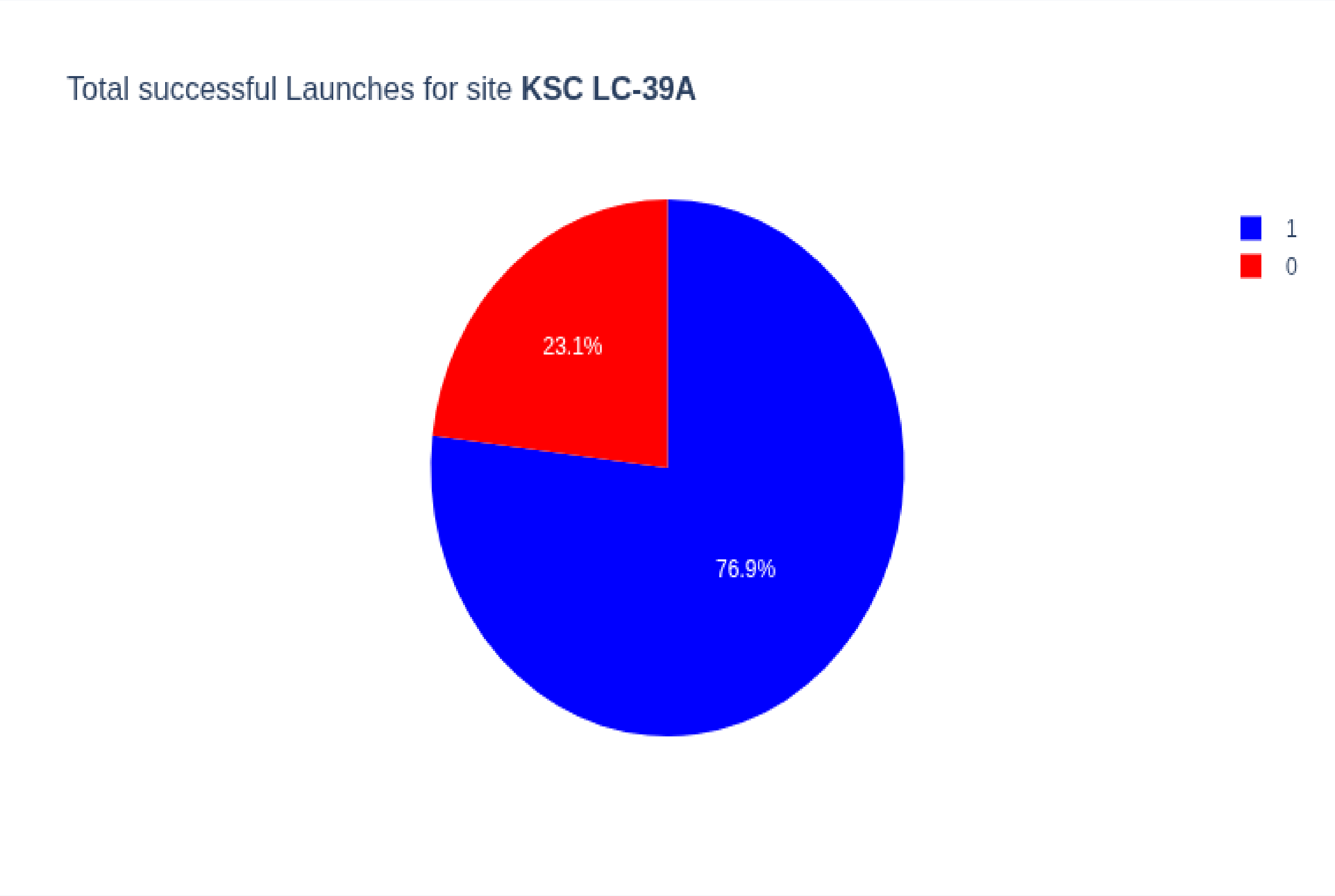
Successful launches by site

- CCAFS SLC-40 and KSC LC-39A have the same launch success rate
- The breakdown of total launches by site is:
  - CCAFS SLC-40 = 33
  - KSC LC-39A = 13
  - VAFB SLC-4E = 10
- The success rate is expected to significantly improve once SpaceX has fully migrated all its Falcon 9 boosters to the Block 5 version



Launch site with highest success ratio

- KSC LC-39A is the launch site with the highest success ratio
- Out of a total of 13 launches, there were **10 successful launches**, with a success ratio of 10:13
- There were only 3 unsuccessful launches
- The success ratios for the other launch sites are:
  - CCAFS SLC-40 (10:33)
  - VAFB SLC-4E (4:10)



Payload Mass (kg) and Success Rate (All Launch Sites)

- This dataset contains a total of 56 launches
  - The overall launch success rate is 0.43
  - The overall success ratio is 3:7
- The low success rate is explainable by the number of **test boosters** vs. **production boosters** found in the data set
- The B4 and B5 boosters are the only Falcon 9 boosters designed to be reusable. The B5 is the only booster designed to be fully reusable

Payload Mass (<= 5000 kg)

- There have been a total of 42 Falcon 9 launches with a payload mass <= 5000 kg
  - The overall launch success rate is 0.45
  - The overall success ratio is 19:42
- 7 of these launches were either B4 or B5 boosters
  - The overall launch success rate is 0.86
  - The overall success ratio is 6:7

Payload Mass (>= 5000 kg)

- There have been a total of 14 Falcon 9 launches with a payload mass >= 5000 kg
  - The overall launch success rate is 0.36
  - The overall success ratio is 5:14
- 5 of these launches were B4 boosters (no B5 in this range)
  - The overall launch success rate is 0.20
  - The overall success ratio is 1:5

Observations

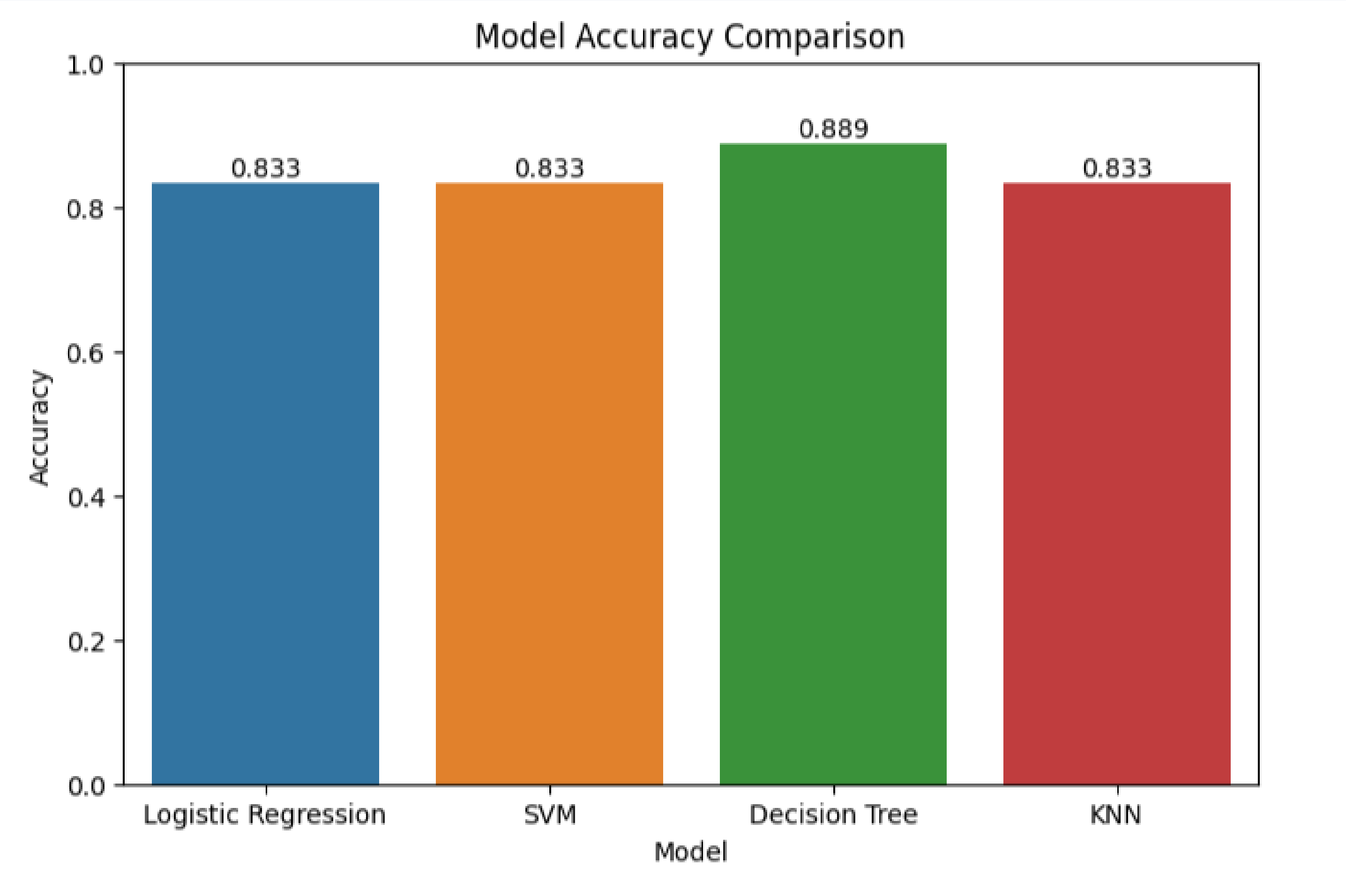
- According to the data, the best likelihood of a successful launch is using a Falcon 9 B5 booster carrying a Payload Mass up to 5000 kg

Section 5

Predictive Analysis (Classification)

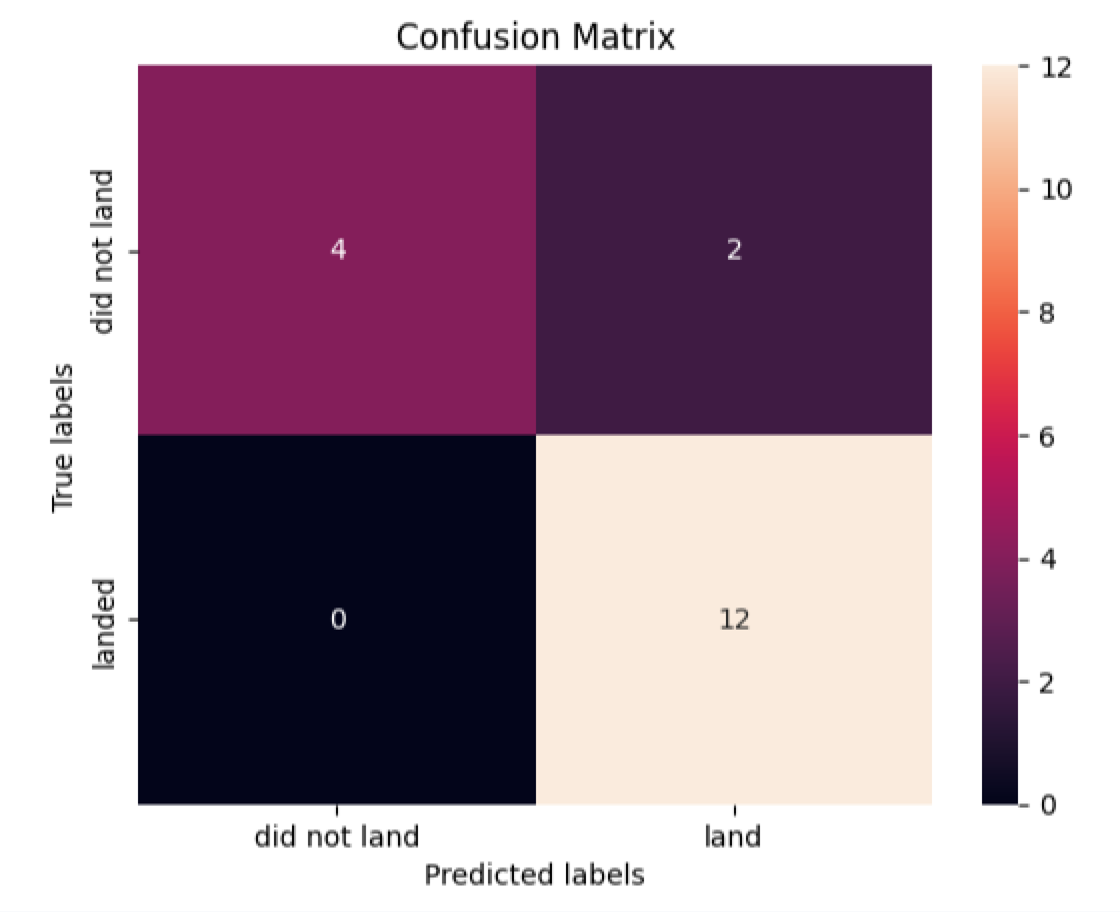
Classification Accuracy

- 4 classification models were built and trained
- The **Decision Tree** classifier was the best performing model, as is the recommended model to use in deployment



Confusion Matrix

- The **Decision Tree** classifier performed better at predicting unsuccessful landings (TP) in comparison to the other classifiers
- This classifier also had a better (lower) score in False Negative (FN) perdictions



Conclusions

- The Falcon 9 Block 5 booster version is the most critical feature to predicting a successful booster landing
  - The success rate of the Block 5 booster is 0.85, success ratio is: 33:39
  - The first Block 5 mission was May 11, 2018 (Flight number: 54)
  - The Block 5 booster is the only Falcon 9 booster version designed to be fully reusable
  - It only took SpaceX 8 years to design a fully reusable booster
  - Other factors like: Launch Site location, orbit, and payload mass did not make a significant difference to success outcomes, especially when considering that the first few booster versions were proof of concept designs, not production versions
- A machine learning model was built to predict landing outcomes, with an accuracy of 89%
- The model needs to be updated quarterly (every 3 months as a minimum), ensuring most recent launch data is incorporated in model predictions
- Considering the amount of Block 5 boosters currently in service, future analytics and machine learning projects should consider **calculating the probability of success based on: payload mass, orbit, number of booster reuses, and any date related historical trends or patterns**

Appendices

- **Appendix 1** – Working with SQL in a Jupyter Notebook
- **Appendix 2** – Working with months in SQLite
- **Appendix 3** – Falcon 9 Success Rate by Orbit Type and Block
- **Appendix 4** – Falcon 9 Success Rate by Block
- **Appendix 5** – Base maps of launch locations
- **Appendix 6** – Orbits vs. Launch Locations
- **Appendix 7** – Building a Tree Classifier model

Appendix 1 – Working with SQL in a Jupyter Notebook

- There is an error in the **Hands-on Lab: Complete the EDA with SQL** notebook. Below are the correct libraries to install:

Correct configuration

The following is the correct configuration.

```
# This configuration WILL NOT WORK, it generates errors
# CONVERT THIS CELL TO 'CODE'.
# ONLY RUN IF YOU DIDN'T INSTALL THE LIBRARIES USING THE `requirements.txt` FILE
!pip install sqlalchemy
!pip install jupysql # Correct library for using SQL in Jupyter
!pip install pandas
```

- Please refer to the notebook for complete details

Appendix 2 – Working with months in SQLite

- The following SQL script was used to complete the task of displaying sorting data by month in SQLite
- Please refer to the notebook for complete details

Task 9

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

Note: SQLLite does not support monthnames. So you need to use substr(Date,6,2) for month, substr(Date,9,2) for date, substr(Date,0,5),='2017' for year.

[20]:

```sql
%%sql
SELECT
    -- Convert the month digit to a month name
    CASE
        CAST(STRFTIME('%m', Date) AS INTEGER)
        WHEN 1 THEN 'January'
        WHEN 2 THEN 'February'
        WHEN 3 THEN 'March'
        WHEN 4 THEN 'April'
        WHEN 5 THEN 'May'
        WHEN 6 THEN 'June'
        WHEN 7 THEN 'July'
        WHEN 8 THEN 'August'
        WHEN 9 THEN 'September'
        WHEN 10 THEN 'October'
        WHEN 11 THEN 'November'
        WHEN 12 THEN 'December'
    END AS Month,
    Booster_Version,
    Launch_Site,
    Landing_Outcome
FROM spacextbl
WHERE
    STRFTIME('%Y', Date) = '2017'
    AND LOWER(Landing_Outcome) LIKE '%success%'
    AND LOWER(Landing_Outcome) LIKE '%ground pad%'
ORDER BY CAST(STRFTIME('%m', Date) AS INTEGER);
```

Running query in 'sqlite:///../data/00_Datasets/00_Raw/SQLite/spacex_sql.db'

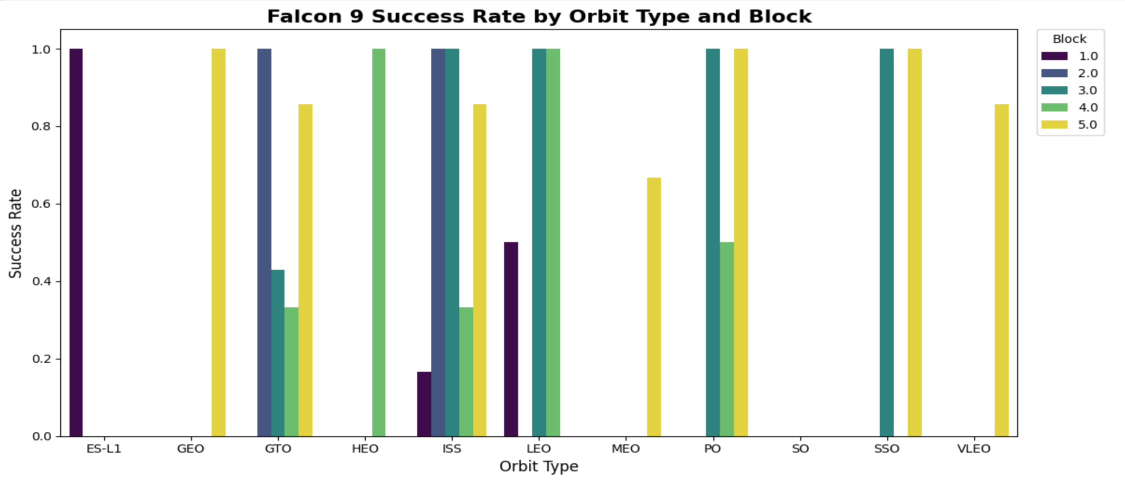
[20]:

Month	Booster_Version	Launch_Site	Landing_Outcome
February	F9 FT B1031.1	KSC LC-39A	Success (ground pad)
May	F9 FT B1032.1	KSC LC-39A	Success (ground pad)
June	F9 FT B1035.1	KSC LC-39A	Success (ground pad)
August	F9 B4 B1039.1	KSC LC-39A	Success (ground pad)
September	F9 B4 B1040.1	KSC LC-39A	Success (ground pad)
December	F9 FT B1035.2	CCAFS SLC-40	Success (ground pad)

Appendix 3 – Falcon 9 Success Rate by Orbit Type and Block

- The following script was used to create a relationship of success rate by orbit type and block
- It clearly demonstrates the high success rate of the Block 5 booster
- Refer to the notebook for details

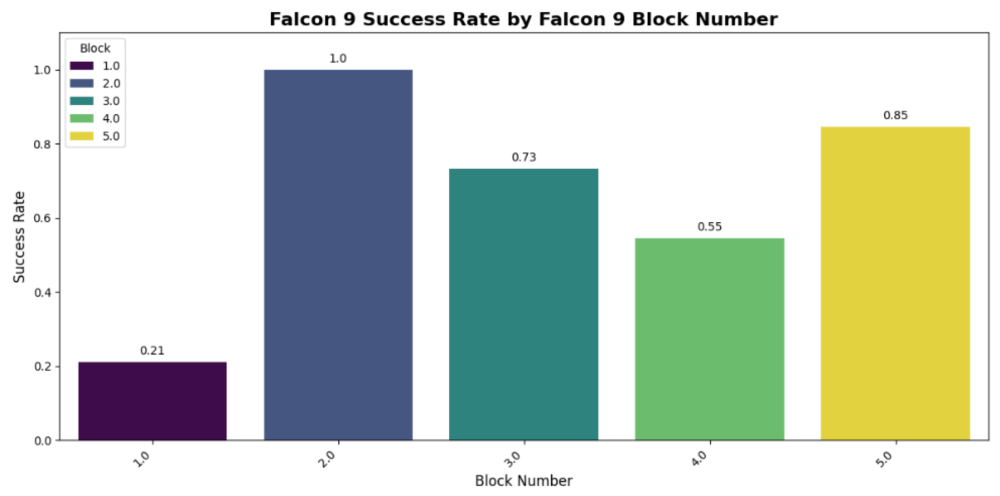
```python
# HINT use groupby method on Orbit column and get the mean of Class column
# Calculate success rate by orbit
success_rate_by_orbit_and_block_df = df.groupby(['Orbit', 'Block'])['Class'].mean().reset_index()

# Rename the 'Class' column to something clearer
success_rate_by_orbit_and_block_df.rename(columns = {'Class': 'SuccessRate'}, inplace = True)
# Set the figure size (important when moving legend outside)
plt.figure(figsize = (12, 6))

# Create the bar plot
ax = sns.barplot(
    data = success_rate_by_orbit_and_block_df,
    x = 'Orbit',
    y = 'SuccessRate',
    hue = 'Block',
    palette = 'viridis'
)

# Customize the plot
ax.set_title(
    'Falcon 9 Success Rate by Orbit Type and Block',
    fontsize = 16,
    fontweight = 'bold'
)
ax.set_xlabel('Orbit Type', fontsize = 12)
ax.set_ylabel('Success Rate', fontsize = 12)

# Move legend to top-right, outside the plot
ax.legend(
    title = 'Block',
    loc = 'upper left',
    bbox_to_anchor = (1.02, 1),  # push legend outside
    borderaxespad = 0
)

# Adjust layout so legend is not clipped
plt.tight_layout()
plt.show()
```



Appendix 4 – Falcon 9 Success Rate by Block

- The following script was used to create a relationship of success rate by block
- It shows the high success rate of the Block 5 booster
- Refer to the notebook for details

```python
# HINT use groupby method on Block column and get the mean of Class column
# Calculate success rate by orbit
success_rate_by_block_df = df.groupby('Block')['Class'].mean().reset_index()

# Rename the 'Class' column to something clearer
success_rate_by_block_df.rename(columns = {'Class': 'SuccessRate'}, inplace = True)
# Set the figure size
plt.figure(figsize = (12, 6))

# Create the bar plot
sns.barplot(
    data = success_rate_by_block_df,
    x = 'Block',
    y = 'SuccessRate',
    hue = 'Block',
    palette = 'viridis'
)

# Customize the plot
plt.title('Falcon 9 Success Rate by Falcon 9 Block Number', fontsize = 16, fontweight = 'bold')
plt.xlabel('Block Number', fontsize = 12)
plt.ylabel('Success Rate', fontsize = 12)
plt.ylim(0, 1.1)  # Success rate ranges from 0 to 1
plt.xticks(rotation = 45, ha = 'right')  # Rotate x-axis labels for readability

# Add actual success rate on top of bars
for i, row in success_rate_by_block_df.iterrows():
    plt.text(
        i,
        row['SuccessRate'] + 0.02,
        f"{row['SuccessRate']:.2}",  # Format as percentage
        ha = 'center',
        fontsize = 10
    )

plt.tight_layout()
plt.show()
```



Appendix 5 – Base maps of launch locations

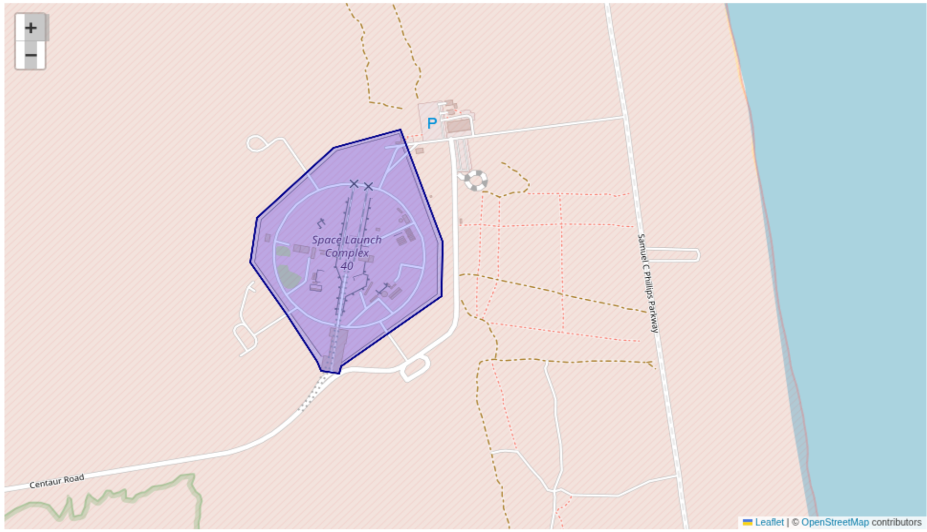
- In order to enhance the analytics, base maps of each launch location was obtained. The following script and base map is the CCAFS SLC-40 location
- The basemaps come from <u>Overpass Turbo</u>, an open source website containing GIS data
- Refer to the notebook for details

Create the base map of the CCAFS SLC-40 area

[28]:
```python
# Start location is SLC-40 launch complex
ccafs_slc_40_coordinate = [
    launch_sites_df.loc[0, 'Lat'],
    launch_sites_df.loc[0, 'Long']
]

location_name = launch_sites_df.loc[0, 'Launch Site']

# Create a map object
ccafs_slc_40_site_map = folium.Map(
    location = ccafs_slc_40_coordinate,
    zoom_start = 10
)

# Path to geojson file of SLC-40
geojson_path = "../data/00_Datasets/02_GIS/CCAFS SLC-40.geojson"

# Add GeoJSON overlay to the map
folium.GeoJson(
    geojson_path,
    name = "SLC-40 Boundary",
    style_function = lambda x : {
        "fillColor" : "blue",
        "color" : "darkblue",
        "weight" : 2,
        "fillOpacity" : 0.2,
    }
).add_to(ccafs_slc_40_site_map).add_child(folium.Popup(location_name));
```

[29]:
```python
# Generate map
ccafs_slc_40_site_map
```

[29]:



Appendix 6 – Orbits vs. Launch Locations

- The following table was used to understand the relationship between launch site and orbit requirements
- The table is from SpaceX's Falcon User's Guide (Version 8)
- A local copy of this guide can be found here
- Refer to the guide for details

Table 3-1: Falcon 9 and Falcon Heavy Launch Services

Insertion Orbit	Inclination Range	Vehicle	Launch Site(s)
LEO	28.5 – 55 deg	Falcon 9 or Falcon Heavy	Eastern Range
LEO	55 – 65 deg*	Falcon 9 or Falcon Heavy (Eastern Range only)	Eastern or Western Range
LEO	65 – 85 deg	Falcon 9	Western Range
LEO / Retrograde	105+ deg	Falcon 9	Western Range
LEO Polar / SSO	85 – 105 deg*	Falcon 9	Western or Eastern Range
GTO	Up to 28.5 deg	Falcon 9 or Falcon Heavy	Eastern Range
GSO	Up to 28.5 deg	Falcon Heavy	Eastern Range
Earth escape	N/A	Falcon 9 or Falcon Heavy	Western or Eastern Range

*Subject to mission-specific performance considerations. Falcon Heavy is only available from the Eastern Range

Appendix 7 – Building a Tree Classifier model

- The following scripts were used to create a decision tree classifier
- Some modifications were needed to the originally provided scripts, due to changes in more recent versions of Scikit-Learn
- Refer to the notebook for details

Create a decision tree classifier object then create a `GridSearchCV` object `tree_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

[25]:
```python
# GridSearchCV parameters to use with Decision Tree
tree_parameters_dct = {
    'criterion' : ['gini', 'entropy'],
    'splitter' : ['best', 'random'],
    'max_depth' : [2*n for n in range(1,10)],
    'max_features' : ['sqrt', 'log2', None], # Removed 'auto', deprecated in scikit-learn versions > 1.1. Added 'log2' and None
    'min_samples_leaf' : [1, 2, 4],
    'min_samples_split' : [2, 5, 10]
}

# Create an empty model
tree = DecisionTreeClassifier()
```

[26]:
```python
# Create a GridSrearch CV object
tree_cv = GridSearchCV(
    estimator = tree,
    param_grid = tree_parameters_dct,
    cv = 10
)
```

[27]:
```python
# Train the model
tree_cv.fit(X_train, Y_train)
```

[27]:

GridSearchCV

best_estimator_: DecisionTreeClassifier

DecisionTreeClassifier

▼ Parameters

criterion	'gini'
splitter	'random'
max_depth	8
min_samples_split	10
min_samples_leaf	2
	0.0
max_features	'sqrt'
random_state	None
max_leaf_nodes	None
min_impurity_decrease	0.0
class_weight	None
ccp_alpha	0.0
monotonic_cst	None

[28]:
```python
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_spl
it': 10, 'splitter': 'random'}
accuracy : 0.8875
```

Thank you!