IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pascale De Gheselle
2025-11-25

Outline

Executive Summary

- **Goal**:

  → Use data science methodology to predict if the **Falcon 9 first stage** will land successfully and be able to be reused (cost savings)

- **Data Science Methodology**:

  → Use of a **spaceX** dataset to extract patterns and develop a prediction model for success landing

    - Data collection (API/Web scraping) & data wrangling
    - Perform exploratory data analysis (EDA) using visualization and SQL
    - Perform interactive visual analytics using Folium and Plotly Dash
    - Perform predictive analysis using several classification models (logreg, svm, tree & knn)

- **Results:**
  - Choice of **launch sites' locations** is crucial:
    → close to Equator line, coastline, railways & highways but away from cities
    → "Best Launch Site" in the US: ***KSC LC-39A***
  - Optimal **rocket features**: booster Version ***F9 FT*** & **moderate payload** + **combination of features** to be taken into account
  - Four dominant **orbits** (*ES-L1, GEO, HEO and SSO*) but robustness to be further validated
  - Most accurate **classification** model: **decision tree**
  - **Increasing trend of success, new** technologies and **recent** available **databases** → **monitor trends** and changes

Introduction

- **Very high cost** for rocket launches that can be reduced in case of spaceX if the first stage can be reused
- **Goal**: predict if the Falcon 9 first stage will land successfully hence saving the cost for the 1th stage
- Use of **available datasets** gathering extensive information about the SpaceX launches, their specifications and outcomes
- Find **pattern and correlations** through the use of data science methodology
- **Answer questions**:
  - *Which **rocket specifications** are most favourable for success landing?*
  - *Which are the most optimal launch sites' **locations and proximities**?*
  - *Can we achieve a satisfying **success prediction** using **classification models**?*

Section 1

Methodology

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data from SpaceX collected using RESTful API & web scraping, then converted into a dataframe
- Perform data wrangling using Python's libraries such as Pandas and Numpy
  - Handling of missing values, sampling, encoding of categorical variables, determine Labels
- Perform exploratory data analysis (EDA) using visualization Charts and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models (logreg, svm, tree & knn)
  - Transform and split the data (training/test set), create different classification model objects, tune using GridSearchCV, evaluate classification models (accuracy/score, confusion matrix)

Data Collection

Data Collection - Overview

<u>Data collection process</u>

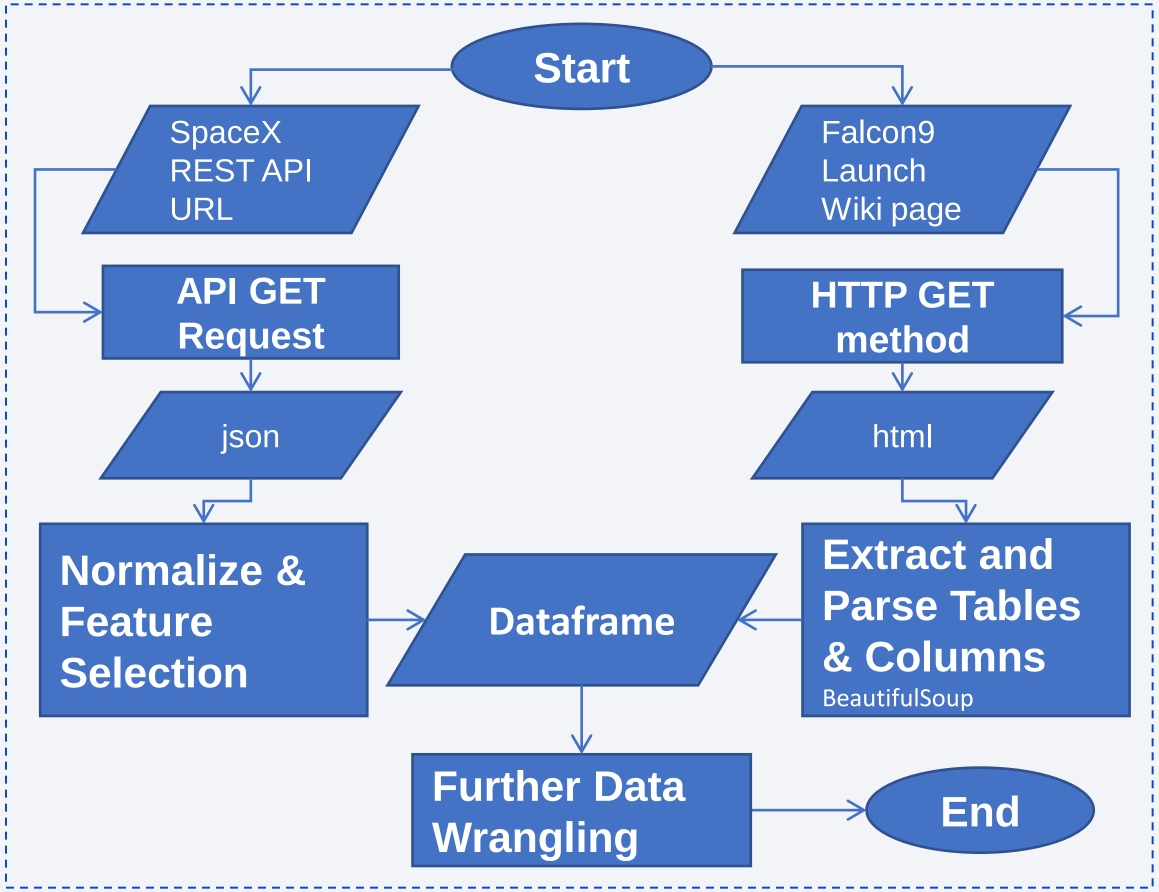
Two methods were used*:

1. **REST API**
   - Based on SpaceX URL
   - API Get Request →html
   - Web scrape with BeautifulSoup
2. **Web Scraping**
   - Based on Wiki page
   - HTTP Get Request → json
   - Normalize & Feature Selection

+ Additional Data Wrangling & Export as .csv

*See further details of both processes on following slides [#10](#) & [#12](#)

Overview: flowchart of Data Collection API vs Scraping*

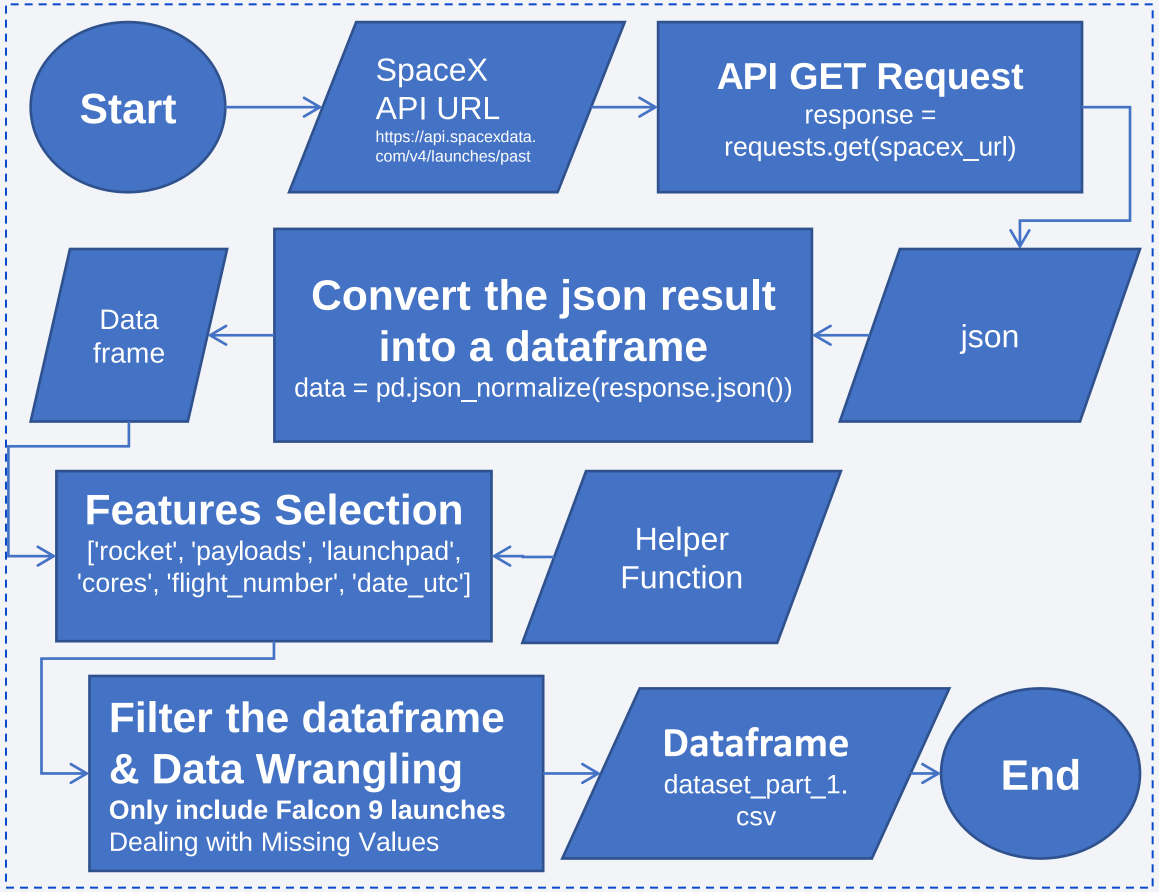


Data Collection – SpaceX API - Process

SpaceX API Data Collection Process:

1. Request and parse the SpaceX launch data using the GET request
    - Request rocket launch data from SpaceX API (https://api.spacexdata.com/v4/launches/past)
    - Decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()
    - Features Selection ['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']
2. Filter the dataframe to only include Falcon 9 launches
3. Data Wrangling & Formating: Dealing with Missing Values

flowchart of SpaceX API calls



See resulting Counts and Dataframe on slides #11

GitHub URL of the completed SpaceX API calls notebook: 202511251300_CourseraFinalProject_AppliedDataScienceCapstoneSpaceX/jupyter-labs-spacex-data-collection-api_COMPLETED.ipynb at main · pdeghesel/202511251300_CourseraFinalProject_AppliedDataScienceCapstoneSpaceX

Data Collection – SpaceX API - Outcome

Data Collection Outcome from SpaceX API

The first 5 lines of the resulting dataframe from the API Data Collection are shown in the table below:

```
data_falcon9.head(5)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

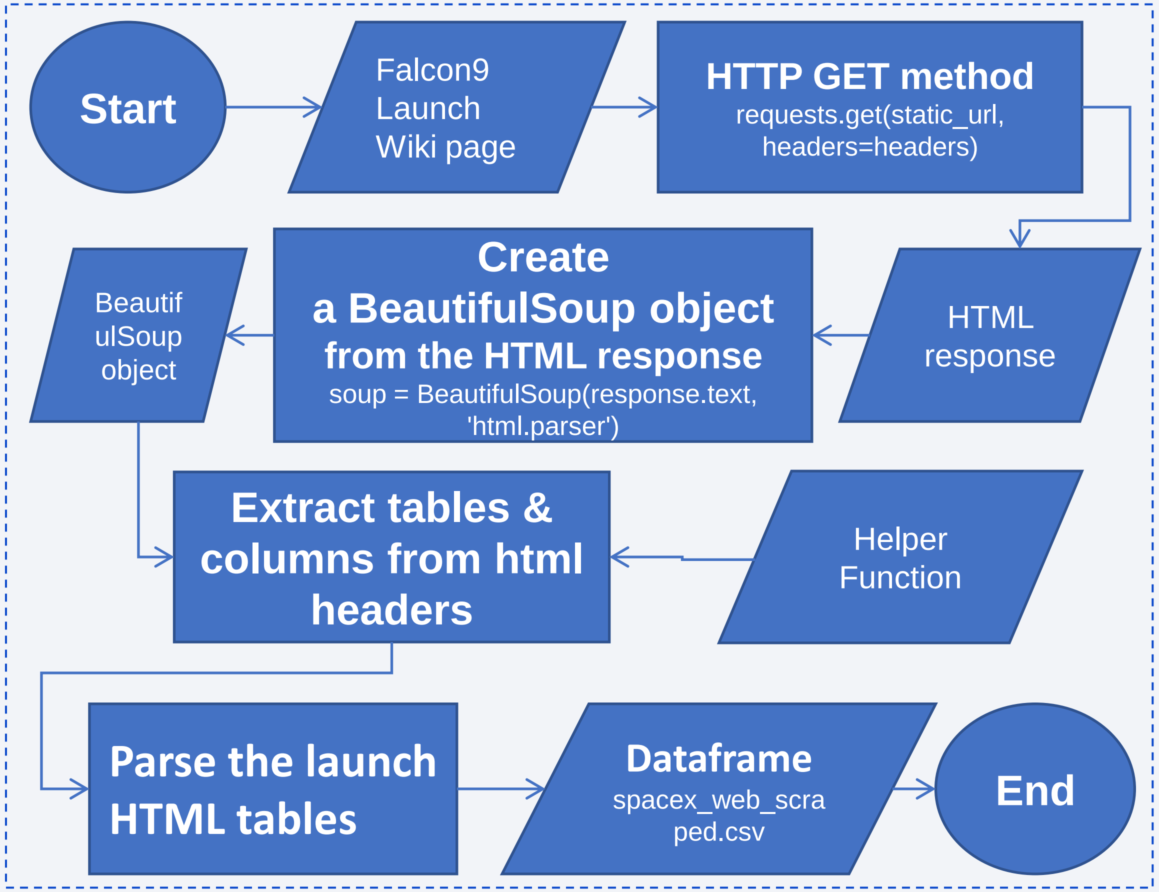
Dataframe exported as **dataset_part_1.csv**

Data Collection – Scraping - Process

Web scraping process:

1. Request the Falcon9 Launch Wikipedia page from its URL
2. Extract all column/variable names from the HTML table header
3. Create a Pandas data frame by parsing the launch HTML tables

flowchart of web scraping



See resulting Counts and Dataframe on slides [#13]

GitHub URL of the completed web scraping:
[202511251300_CourseraFinalProject_AppliedDataScienceCapstoneSpaceX/jupyter-labs-webscraping_COMPLETED.ipynb at main · pdeghesel/202511251300_CourseraFinalProject_AppliedDataScienceCapstoneSpaceX]

Data Collection – Scraping - Outcome

Data Collection Outcome from Web Scraping

The first 5 lines of the resulting dataframe from the Web Scraping are shown in the table below:

```
df.head(5)
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	flight_number	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	bv	NaN	date	time
1	1	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.07B0003.18	NaN	4 June 2010	18:45
2	1	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.07B0003.18	NaN	4 June 2010	18:45
3	2	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.07B0004.18	NaN	8 December 2010	15:43
4	3	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.07B0005.18	NaN	22 May 2012	07:44

Dataframe exported as **spacex_web_scraped.csv**

Data Wrangling

Data Wrangling - Process

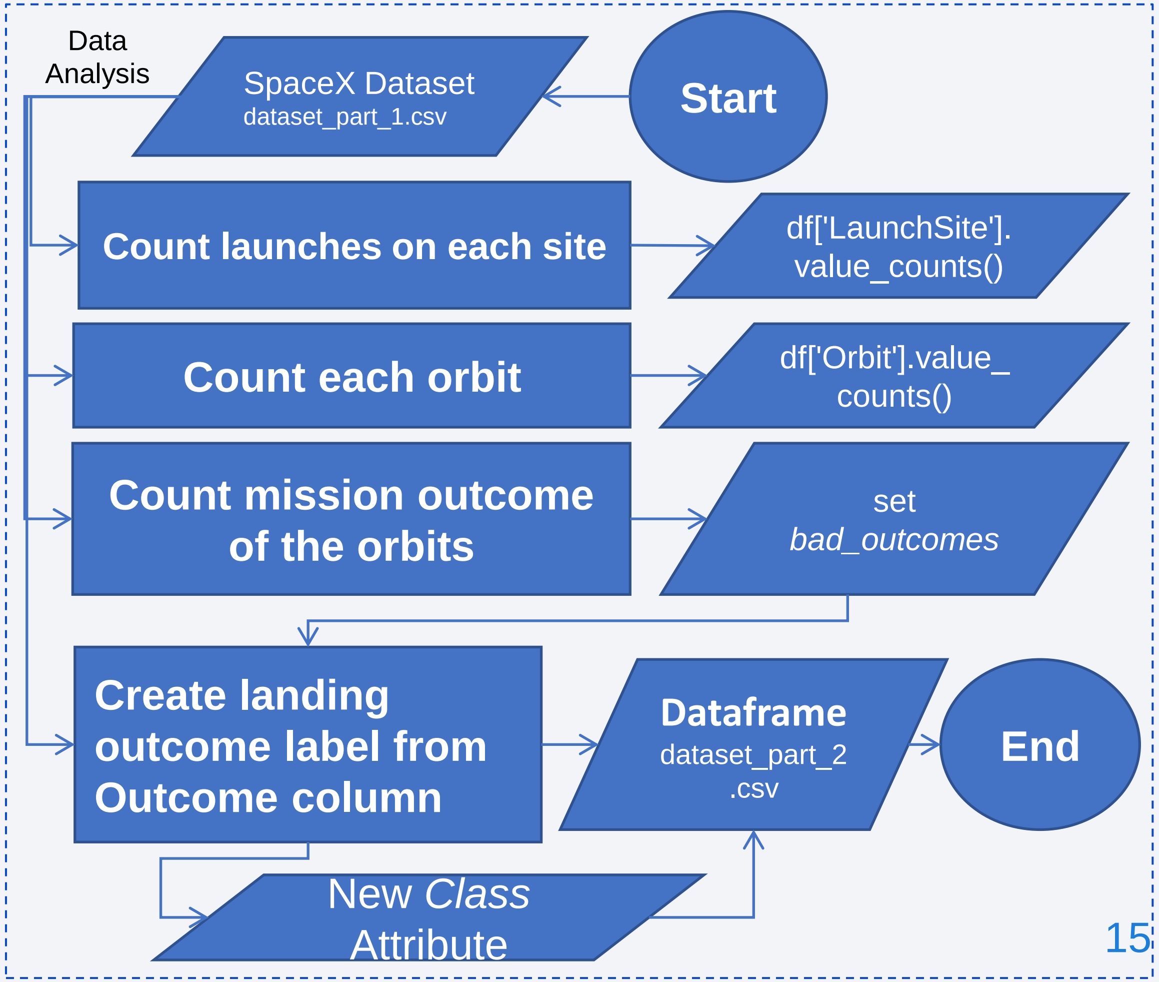
Data wrangling process:

EDA & Determine Training Labels

- Load Data & First Data Analysis/Checks
  - Load Space X dataset from API Data Collection
  - Check for Missing Values and Type of Data
  - Identify which columns are numerical and categorical
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome of the orbits
- Create a landing outcome label from Outcome column (→ New Class Attribute)

See resulting Counts and Dataframe on slides #16 & #17

flowchart of data wrangling



GitHub URL of the completed data wrangling: https://github.com/pdeghesel/202511251300_CourseraFinalProject_AppliedDataScienceCapstoneSpaceX/blob/main/labs-jupyter-spacex-Data%20wrangling_COMPLETED.ipynb

Data Wrangling – Outcome - Counts

Data Wrangling Outcome Counts

The first 5 lines of the resulting dataframe from the the data wrangling are shown in the table below:

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

```
Orbit
GTO       27
ISS       21
VLEO      14
PO         9
LEO        7
SSO        5
MEO        3
HEO        1
ES-L1      1
SO         1
GEO        1
Name: count, dtype: int64
```

```
df['Orbit'].value_counts().sum()
```

```
np.int64(90)
```

```
# Apply value_counts() on column LaunchSitedf['
df['LaunchSite'].value_counts()
```

```
LaunchSite
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: count, dtype: int64
```

```
landing_outcomes = df['Outcome'].value_counts()
```

```
print(landing_outcomes)
```

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

Data Wrangling – Outcome - Dataframe

Data Wrangling Outcome Dataframe

The first 5 lines of the resulting dataframe from the the data wrangling are shown in the table below:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

Dataframe exported as **dataset_part_2.csv**

EDA
SQL & Charts

EDA with Data Visualization

- **Summary of all charts:**
  - Flight Number vs. Launch Site (success outcome as hue parameter) in scatterplot
  - Payload vs. Launch Site (success outcome as hue parameter) in scatterplot
  - Success Rate vs. Orbit Type in bar chart
  - Flight Number vs. Orbit Type (success outcome as hue parameter) in scatterplot
  - Payload vs. Orbit Type (success outcome as hue parameter) in scatterplot
  - Launch Success Yearly Trend

→ Goal: **Identify correlations** between success outcomes and the other features of the plots

GitHub URL of the completed EDA with data visualization notebook:
https://github.com/pdeghesel/202511251300_CourseraFinalProject_AppliedDataScienceCapstoneSpaceX/blob/main/edadataviz_COMPLETED.ipynb

EDA with SQL

List of SQL queries that were 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 4t<payload mass<6t
- List the total number of successful and failure mission outcomes
- List all the booster_versions that have carried the maximum payload mass
- 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 landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

GitHub URL of the completed EDA with SQL notebook:
https://github.com/pdeghesel/202511251300_CourseraFinalProject_AppliedDataScienceCapstoneSpaceX/blob/main/jupyter-labs-eda-sql-coursera_sqllite_COMPLETED.ipynb

Interactive Analysis

Build an Interactive Map with Folium

- **Mark all launch sites on a map:**
  - Create a **folium.Map** object, with an initial center location to be NASA Johnson Space Center at Houston, Texas.
  - Use **folium.Circle** and **folium.Marker** to add a highlighted circle area with a text label on a specific coordinate.

  → **visualize all launch sites' locations at a glance and identify trend (proximity to coast/Equator line)**
- **Mark the launch outcomes (success/failure) for each site on the map:**
  - Create a **MarkerCluster object** (for launch records with exact same coordinate)
  - In combination with **folium.Icon** and **marker_color attribute** for defining colors based on the class value

  → **Identify sites with highest success rates**
- **Calculate the distances between a launch site to its proximities:**
  - Add a **MousePosition** on the map to get the coordinates for a mouse over any points of interests on the map.
  - Display the distances using "***folium.PolyLine***", "***folium.Marker"*** **objects** and the ***Icon*** **property**

  → **Identify relevance of proximities (coastline, highways, railways, cities) to launch sites**

GitHub URL of the completed notebook with interactive Folium map:
https://github.com/pdeghesel/202511251300_CourseraFinalProject_AppliedDataScienceCapstoneSpaceX/blob/main/lab_jupyter_launch_site_location_COMPLETED.ipynb

Build a Dashboard with Plotly Dash

- **Plots/graphs and interactions added to the dashboard with Plotly Dash:**
    - **Success Launches Ratio** for all sites and by site, in **pie chart**
        - Selection “All Sites” or “[Specific Site Selection]” through a **Dropdown Menu**
    - **Payload vs. Launch Outcome** for all sites and by Site, in **scatter plot**
        - With payload selection through a **range slider**
- **Questions to be answered:**
    - *Which site has the largest successful launches?*
    - *Which site has the highest launch success rate?*
    - *Which payload range(s) has the highest launch success rate?*
    - *Which payload range(s) has the lowest launch success rate?*
    - *Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate?*

GitHub URL of the completed Plotly Dash lab:
https://github.com/pdeghesel/202511251300_CourseraFinalProject_AppliedDataScienceCapstoneSpaceX/blob/main/spacex-dash-app_COMPLETED.py

Predictive Analysis

Predictive Analysis (Classification)

- **<u>Model development process:</u>**
  - **Build:**
    - Transform (using “*StandardScaler*” and “*fit_transform*” methods)
    - Split the data into training, validation and test data (using “*train_test_split”* function)
    - Create different classification model objects
      → logistic regression, support vector machnie (svm), decision tree & K-Nearest Neighbors (knn)
  - **Improve:**
    - Hyperparameter tuning using GridSearchCV
      (Grid of parameters & Cross-Validation)
  - Best Model **Selection** using accuracy/score on the validation data (“*.best_score_”* and “*.score*” methods)
  - Further **evaluation** using confusion matrix

GitHub URL of the completed predictive analysis lab:
https://github.com/pdeghesel/202511251300_CourseraFinalProject_AppliedDataScienceCapstoneSpaceX/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5_COMPLETED.ipynb

Results

Results

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

Section 2

Insights drawn from EDA

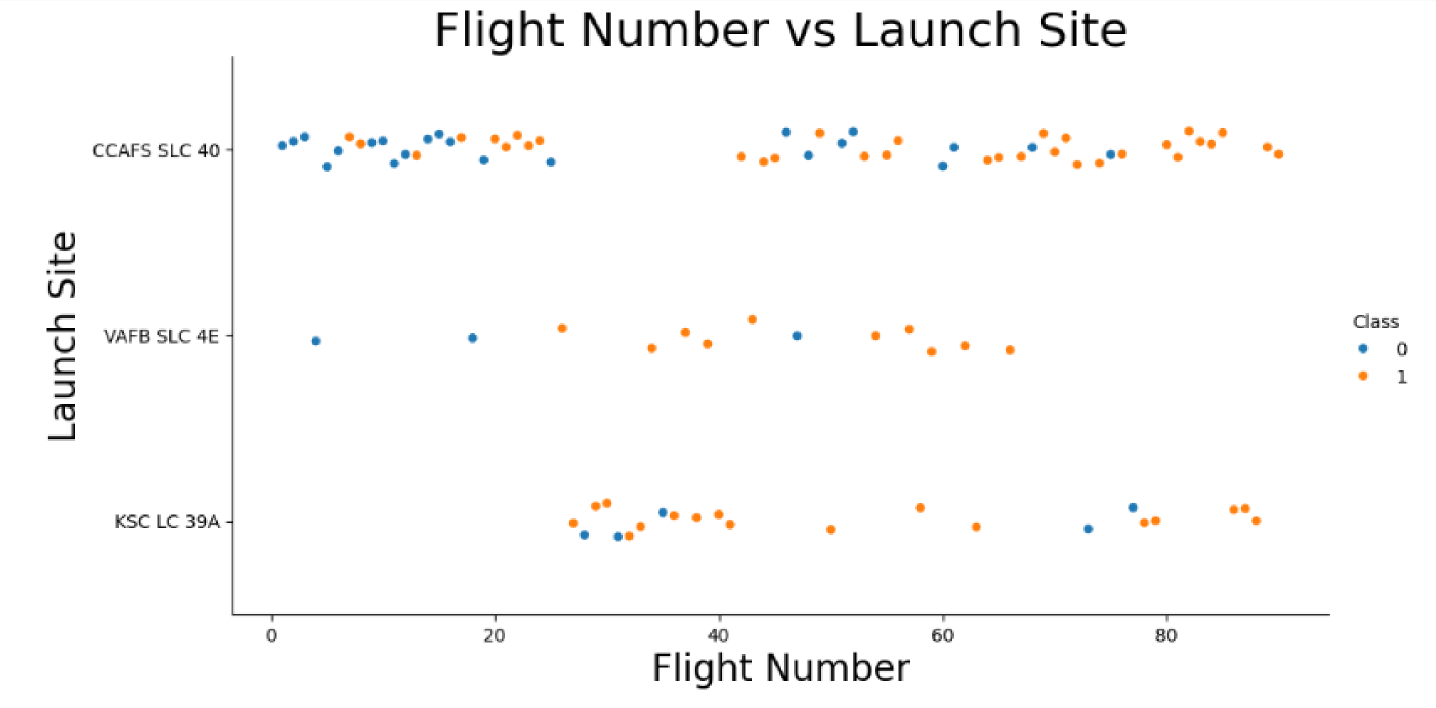
Insights drawn from EDA:

Visualization using Charts

Flight Number vs. Launch Site

- Very high failure rate in case of low flight number (<20)
- High success rate for “moderate” flight number (>20 and <80)
- Only Successes in case of high flight number (>80)

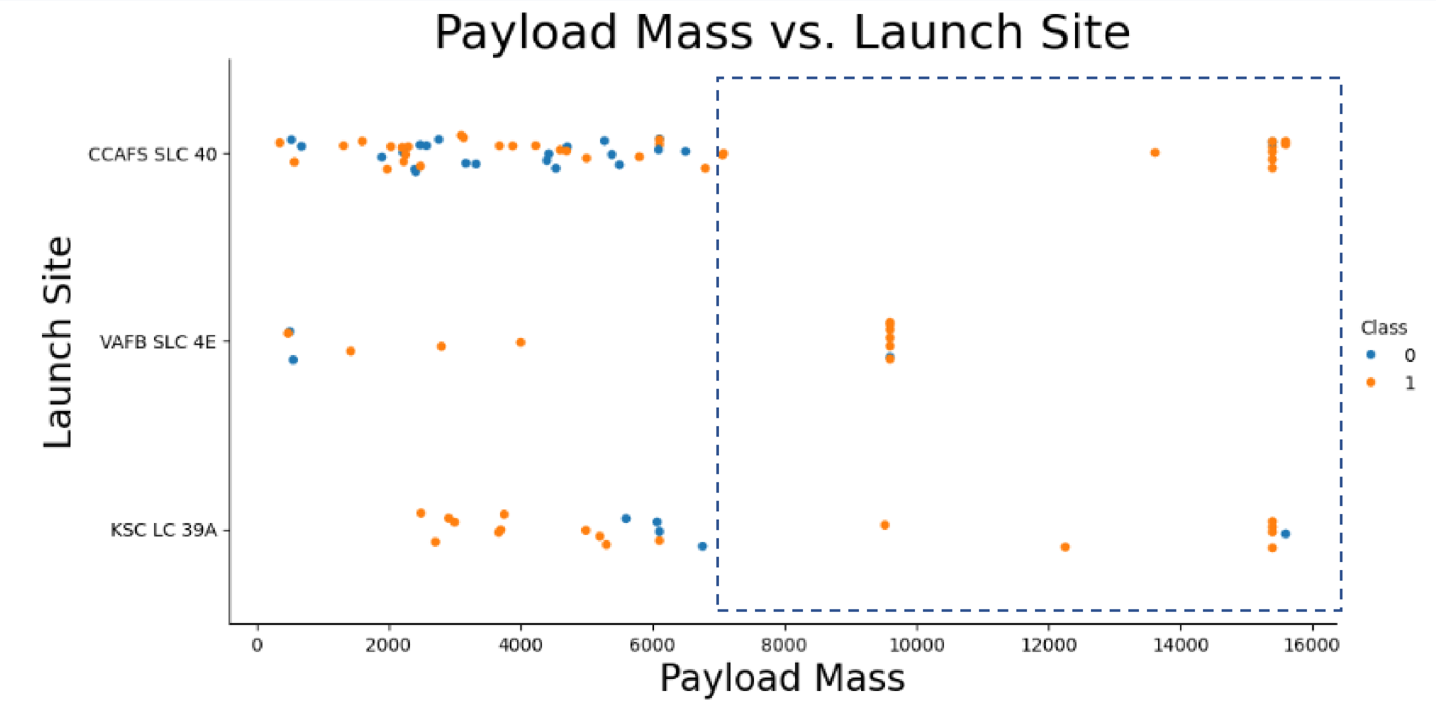
→ An increasing flight number correlates with an increasing landing success



Payload vs. Launch Site

- VAFB-SLC: No rockets launched for heavy payload mass (>10k)
- *CCAFS SLC-40: very high failure rate for low payload mass (<6k)*
- Generally very high success rate in case of high payload mass (>7k) and much higher than for lower masses

→ An increasing payload mass correlates with an increasing landing success



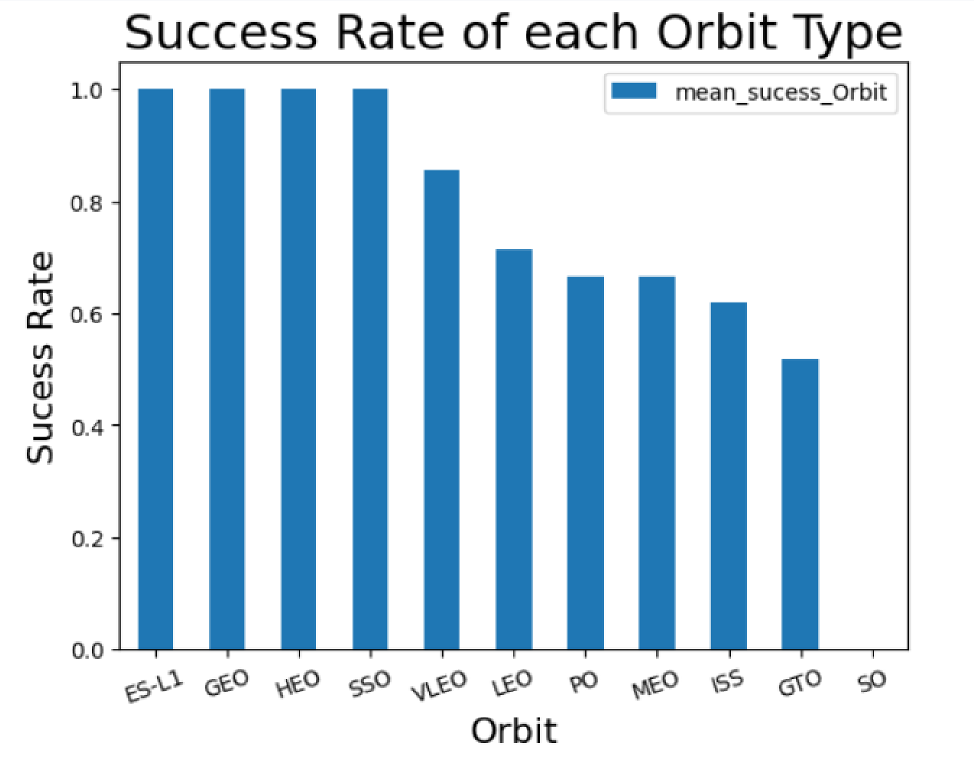
Success Rate vs. Orbit Type

- Highest success rates with **100%** for ***ES-L1***, ***GEO*** and ***HEO*** and ***SSO***
- ***ES-L1***, ***GEO*** and ***HEO*** have nevertheless only one occurrence

→ Questionable Robustness (results to be compared/validated with other databases)

Counts (Occurrence)

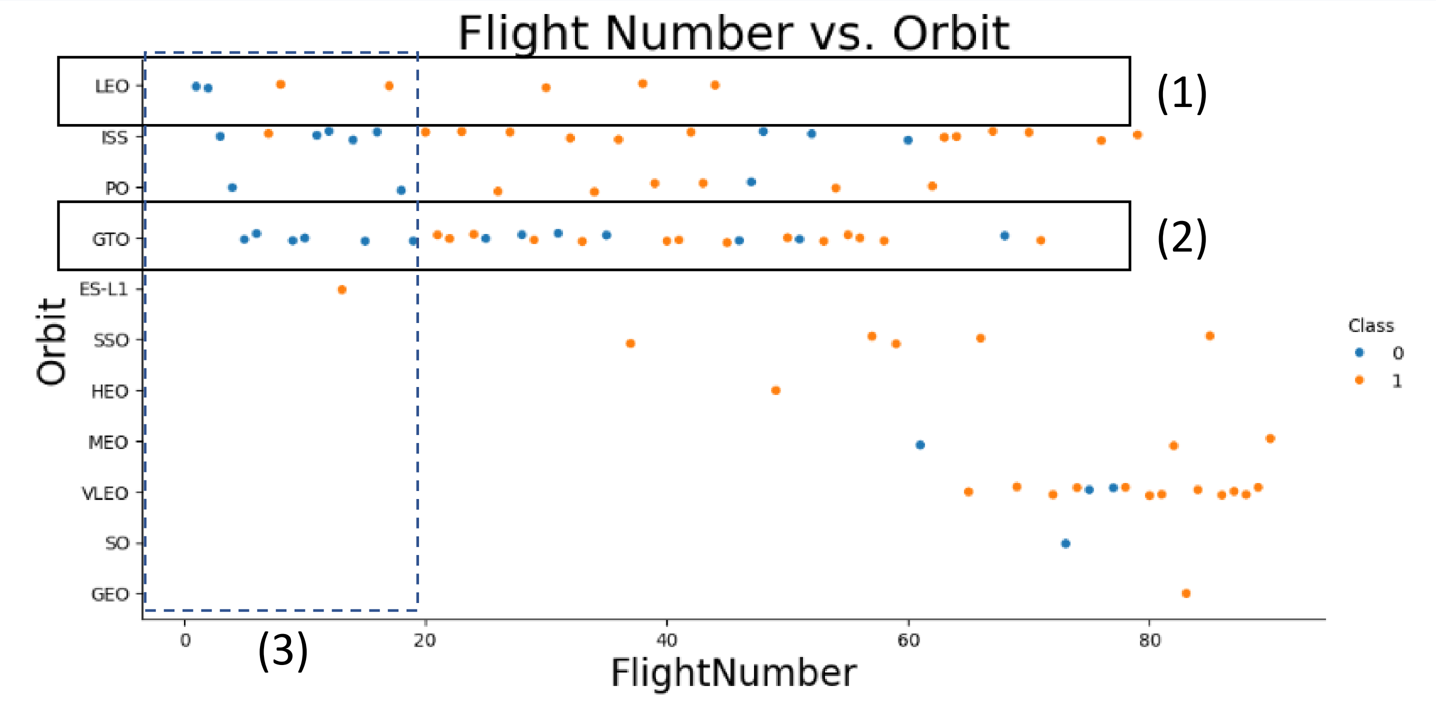
```
df['Orbit'].value_counts()

Orbit
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
ES-L1     1
SO        1
GEO       1
Name: count, dtype: int64
```



Flight Number vs. Orbit Type

- *LEO* orbit: success seems to be related to the number of flights (1)
- *GTO* orbit: no strict relationship between flight number and success, although low flight number (<20) systemically result in failure (2) quite like in *Polar* and *ISS* orbits (3)

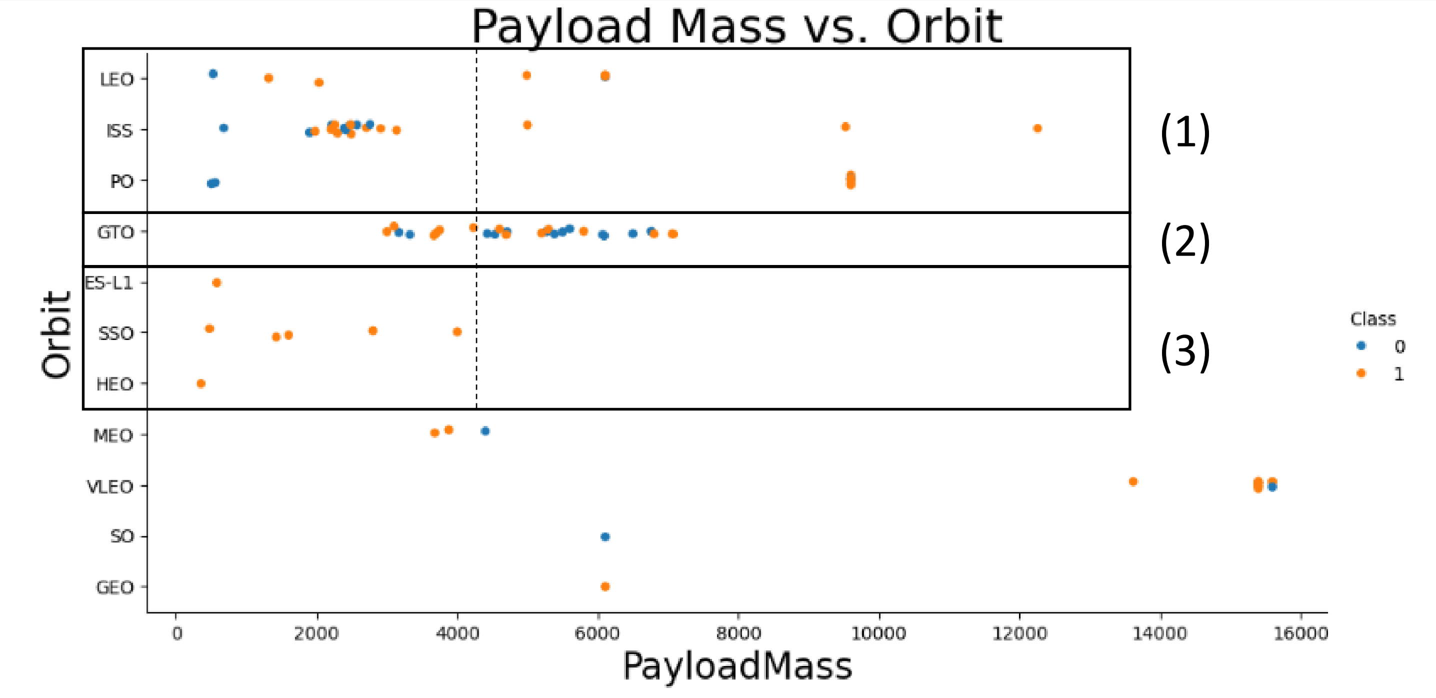
→ No generalized correlation between flight number and success when looking at orbit type, except that a very low flight number mostly fathers failures



Payload vs. Orbit Type

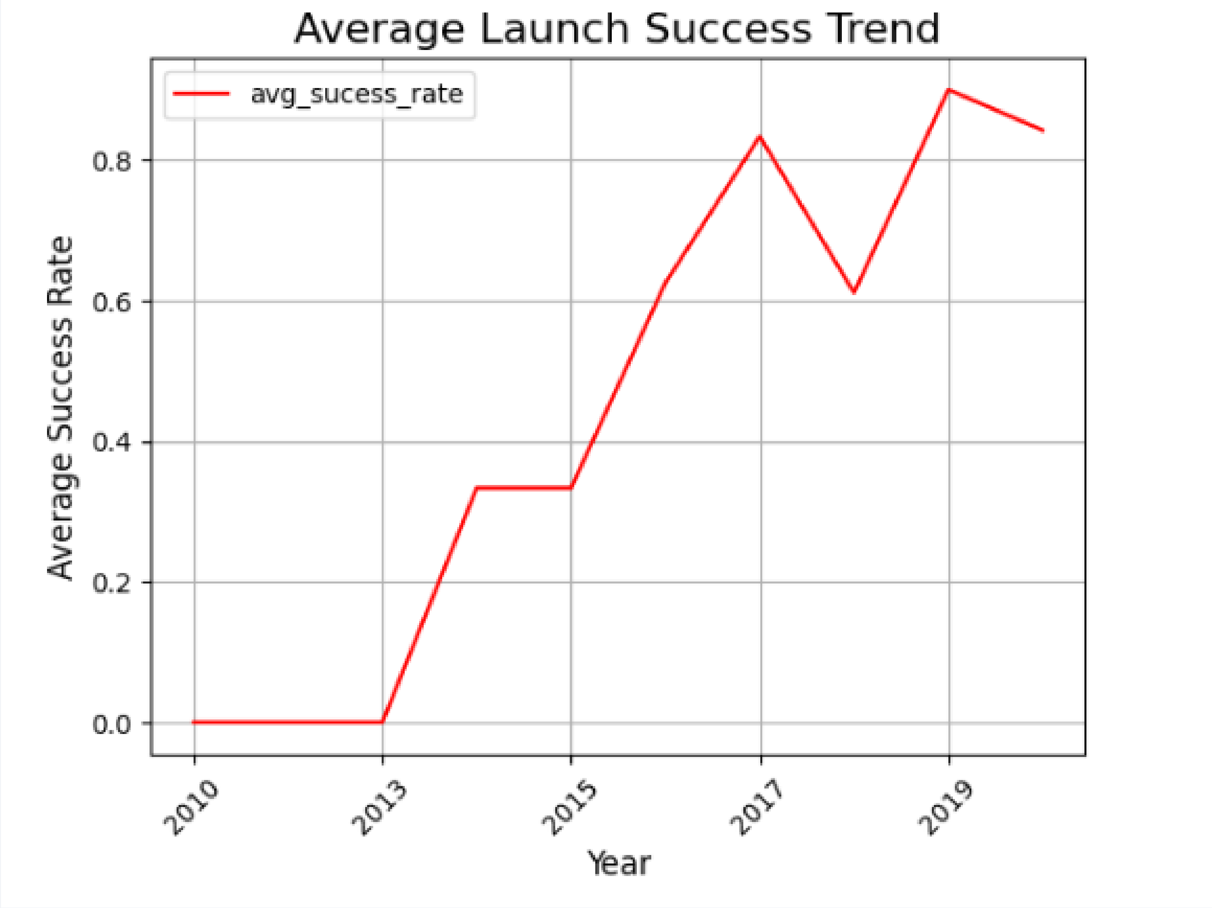
- Heavy payloads correlates with successful landing for *Polar*, *LEO* and *ISS* (1)
- However, no relationship in case of *GTO (2)*
- Only successes for *ES-L1, SSO* and *HEO* despite low payload masses – nut low occurrences (3)

→ No generalized correlation between Payload Mass and Success Landing in case of Orbits



Launch Success Yearly Trend

- The scatter plot shows a **trend of the launch success rate to increase** since 2013 till 2020



Conclusions – Insights drawn from EDA - Charts

- Low Flight Number (<20) → very low landing success rate
- Payload Mass
  - Launch Site vs Payload Mass: an increasing payload mass → increasing landing success
  - Payload vs. Orbit Type: no generalized correlation between Payload Mass and Success Landing

  → Combination of features to be considered for evaluation
- Orbits: 100% success rates for *ES-L1*, *GEO* and *HEO* and *SSO, but with only* one occurrence/count

  → Robustness to be validated with further databases
- Trend of the launch success rate to increase since 2013

Insights drawn from EDA:

Results from SQL Queries

All Launch Site Names

- Unique launch sites in the dataset of that space mission*:

	Launch_Site
0	CCAFS LC-40
1	VAFB SLC-4E
2	KSC LC-39A
3	CCAFS SLC-40

Details on Locations:

Cape Canaveral Space Force Station, Brevard County, Florida

Vandenberg Space Force Base, Santa Barbara County, California

Kennedy Space Center, Merritt Island, Florida

Cape Canaveral Space Force Station, Brevard County, Florida

→ Four different launch sites, three of them in Florida

→ Two sites starting with ***CCAFS*** (***Cape Canaveral Space Force Station***): **both sites** located on Cape Canaveral in **Brevard County, Florida**, then **very close to each other**.

(*: see details of query in Appendix)

Launch Site Names Begin with 'CCA'

- Sample of 5 records where launch sites begin with '*CCA*'*:

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

→ 5 records with *CCAFS LC-40* but with different other attributes (ex: Booster Version, Payload, Customers)

(*: see details of query in Appendix)

Total Payload Mass

- Total payload mass carried by boosters from NASA*:

	Sum_PLM
0	45596

→ The total payload mass reached almost 46t

(*: see details of query in Appendix)

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1*:

	avg_PLM
0	2928.4

→ The average payload carried by booster version F9 v1.1 reached almost 3t

(*: see details of query in Appendix)

First Successful Ground Landing Date

- Date of the first successful landing outcome on ground pad*:

	Date
0	2015-12-22

→The first successful landing outcome on ground pad was in year 2015

(*: see details of query in Appendix)

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000*

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

	COUNT(Booster_Version)
0	101

→Four such boosters, named ***F9 FT***, (out of 101) can be found in the dataset

(*: see details of query in Appendix)

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes*:

	Sum_Sucess
0	100

	Sum_Failure
0	1

	COUNT(Mission_Outcome)
0	101

→1Failure for 100 Successes (<1%)

Mission Outcomes distinguished as followed:

	Mission_Outcome
0	Success
1	Failure (in flight)
2	Success (payload status unclear)
3	Success

(*: see details of query in Appendix)

Boosters Carried Maximum Payload

- Maximum payload mass: **15600kg**

- The names of the booster which have carried the maximum payload mass are presented in the table*

→ All of them *F9* ***B5***

→12 distinct boosters (out of 101) with this payload can be found in the dataset (~12%)

(*: see details of query in Appendix)

	Booster_Version	PAYLOAD_MASS__KG_
0	F9 B5 B1048.4	15600
1	F9 B5 B1049.4	15600
2	F9 B5 B1051.3	15600
3	F9 B5 B1056.4	15600
4	F9 B5 B1048.5	15600
5	F9 B5 B1051.4	15600
6	F9 B5 B1049.5	15600
7	F9 B5 B1060.2	15600
8	F9 B5 B1058.3	15600
9	F9 B5 B1051.6	15600
10	F9 B5 B1060.3	15600
11	F9 B5 B1049.7	15600

2015 Launch Records

- List of the **failed landing outcomes in drone ship for 2015** (including their booster versions, and launch site names)*

	month_name	Landing_Outcome	Booster_Version	Launch_Site
0	January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Two records: one failure outcome in drone ship in January and one in April of year 2015, both using ***F9 v1.1*** boosters and on Launch Site ***CCAFS LC-40***

(*: see details of query in Appendix)

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

- Rank (count) of landing outcomes between the date 2010-06-04 and 2017-03-20 is presented here*
  - In the **top 4** we can find either **successes** or no attempts (1)
  - Then with lower amounts we can mostly find failures, no success (2)

(*: see details of query in Appendix)

	Landing_Outcome	count_Outcome
0	Success	38
1	No attempt	21
2	Success (drone ship)	14
3	Success (ground pad)	9
4	Failure (drone ship)	5
5	Controlled (ocean)	5
6	Failure	3
7	Uncontrolled (ocean)	2
8	Failure (parachute)	2
9	Precluded (drone ship)	1
10	No attempt	1

(1)

(2)

Conclusions – Insights drawn from EDA - SQL

- Analysis of records focused on **features** such as:
  Date, Time, Booster Version, Payload, Payload Mass, Orbit, Customer, Mission & Landing Outcome
- Four different **launch sites**, three of them in Florida, two directly in same region (Brevard County), thus very close to each other.
- Maximum payload mass/booster: **15.6t** (carried by ***F9 B5***)
- **Average** of **3t** payload mass carried by booster version ***F9 v1.1***
- **Total** payload mass **~46t**
- **First successful landing** outcome on ground pad was in year **2015**
- Successful Drone Ship Landing with Payload between 4t & 6t: with ***F9 FT***
- High rate of **successful** mission outcomes (>99%)
- Two records of failure outcome in drone ship in 2015 (January & April, ***F9 v1.1*** boosters, Launch Site ***CCAFS LC-40)***
- **Top 4 Landing Outcome** between 2010 and 2017 are **successes**

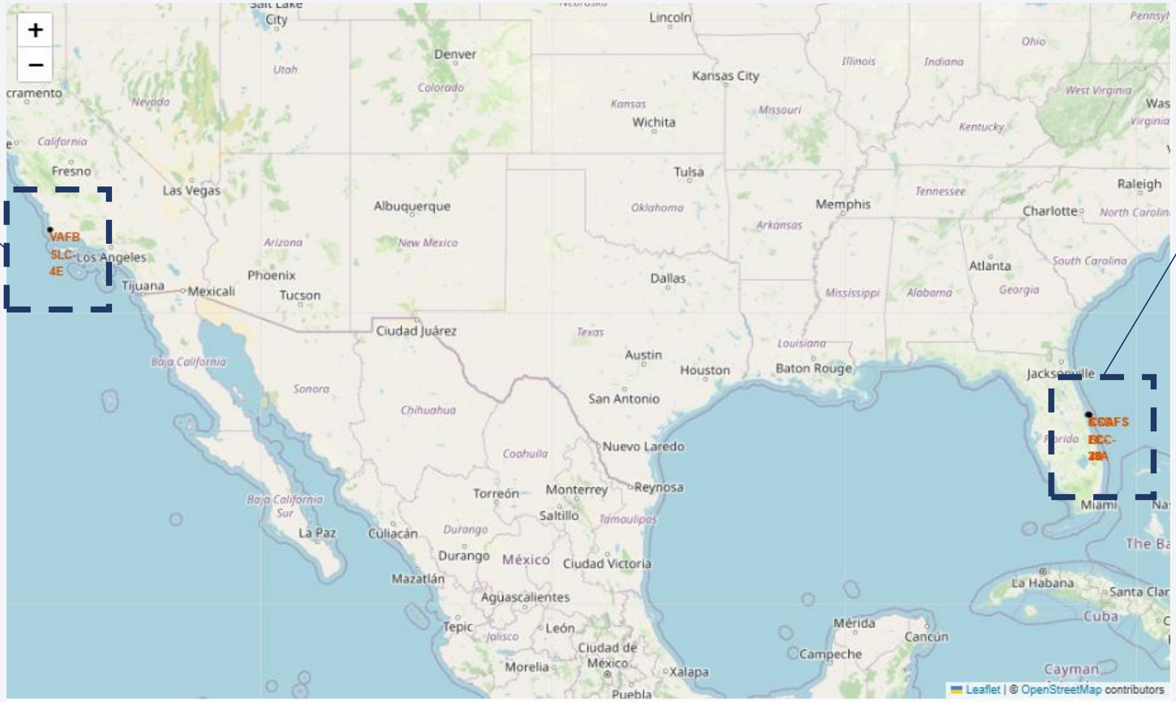
Section 3

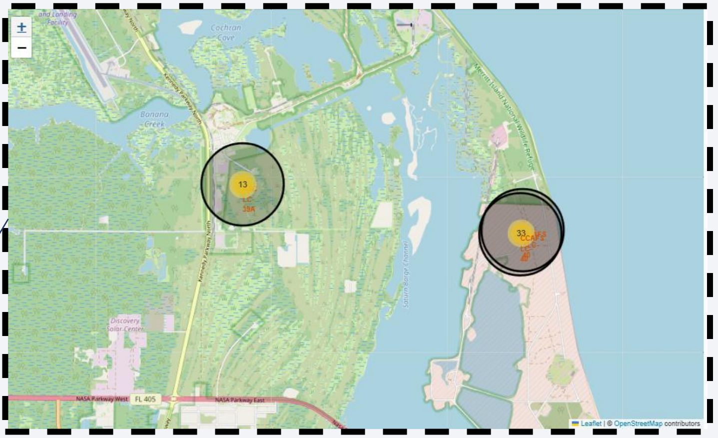
Launch Sites Proximities Analysis

Launch Site Proximities:

Interactive Visual Analytics with Folium

Overview of all launch sites’ location





VAFB SLC-4E

KSC LC-39A

CCAFS SLC-40

CCAFS LC-40

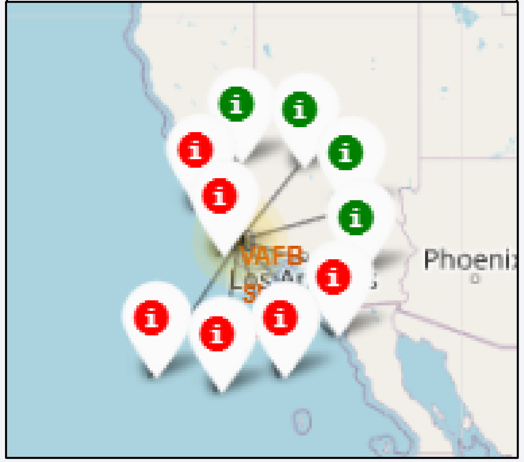
- All launch sites are located in the south of the US, in **proximity to the Equator line** (Rotational Speed Boost)
- All launch sites are also in very close **proximity to the coast** (Rotational Speed Boost, Logistic, Security, Weather Conditions)

Launch outcomes of each launch site

Launch outcomes “Success vs. Failure” for each four launch sites

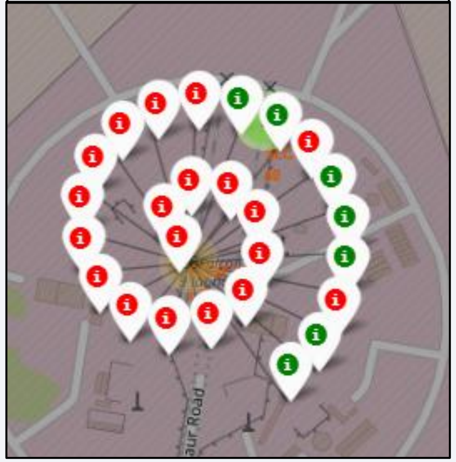
Success Failure

VAFB SLC-4E

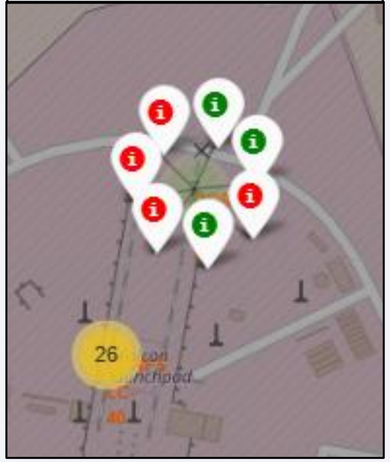


KSC LC-39A

CCAFS LC-40



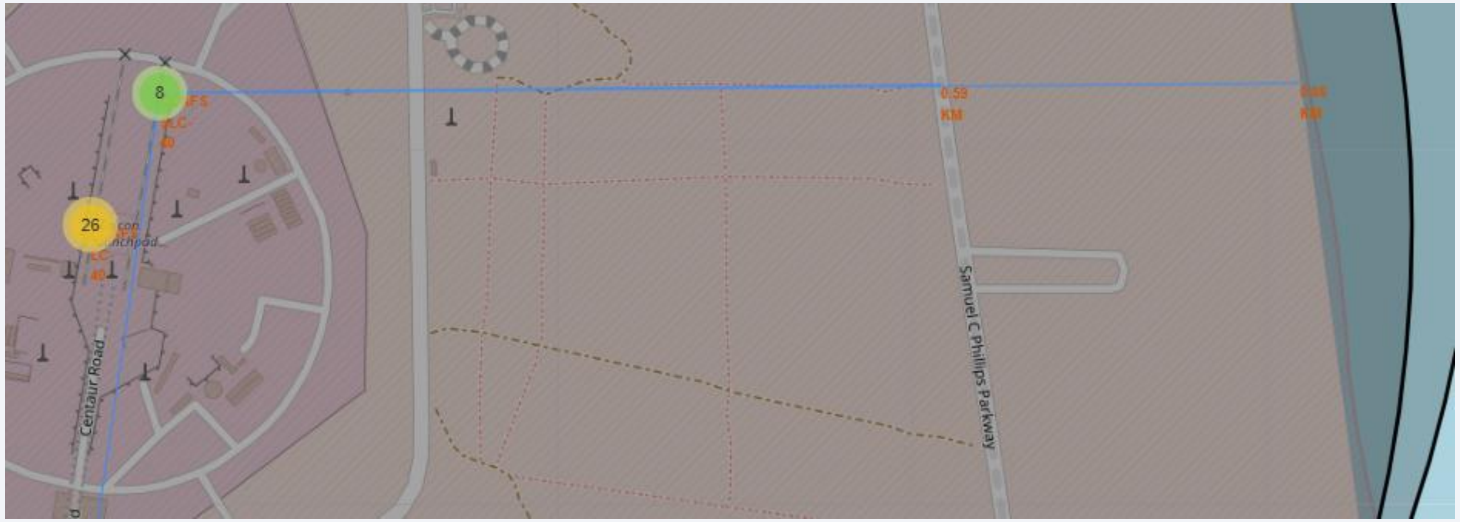
CCAFS SLC-40



- The color-labeled markers in marker clusters show that the site ***KSC LC-39A*** has from far the highest success rate, with a majority of success launches and with a relatively high number of launches.
- All other sites have more failed than success launches.
- *CCAFS LC-40* has the higher number of launches, but also the highest number of failures

Proximities of launch sites*

Proximity to coastline, highways & railways*



Proximity to cities*

- Launch site* in very close proximity to railways (0km)
- Launch site* in close proximity to highways & coastline (<1km)
- Launch site* keeps certain distance away from cities (~20km)

Example; results based on CCAFS SLC-40

Conclusions – Interactive Visual Analytics with Folium

- **Importance of launch sites’ location & proximities :**
  - Close proximity to the Equator line

    → Rotational Speed Boost
  - Close proximity to the coast

    → Rotational Speed Boost, Logistic, Security, Weather Conditions
  - Close proximity to railways & highways

    → Logistic
  - Certain distance away from cities

    → Security

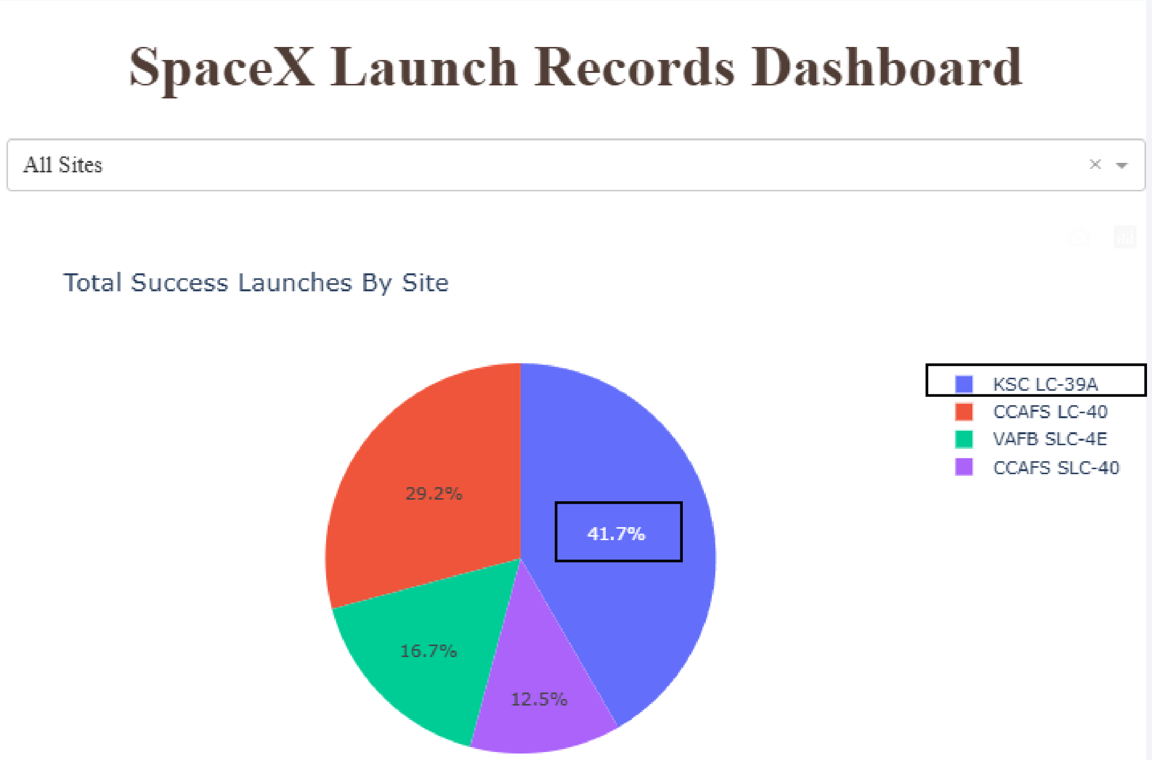
Section 4

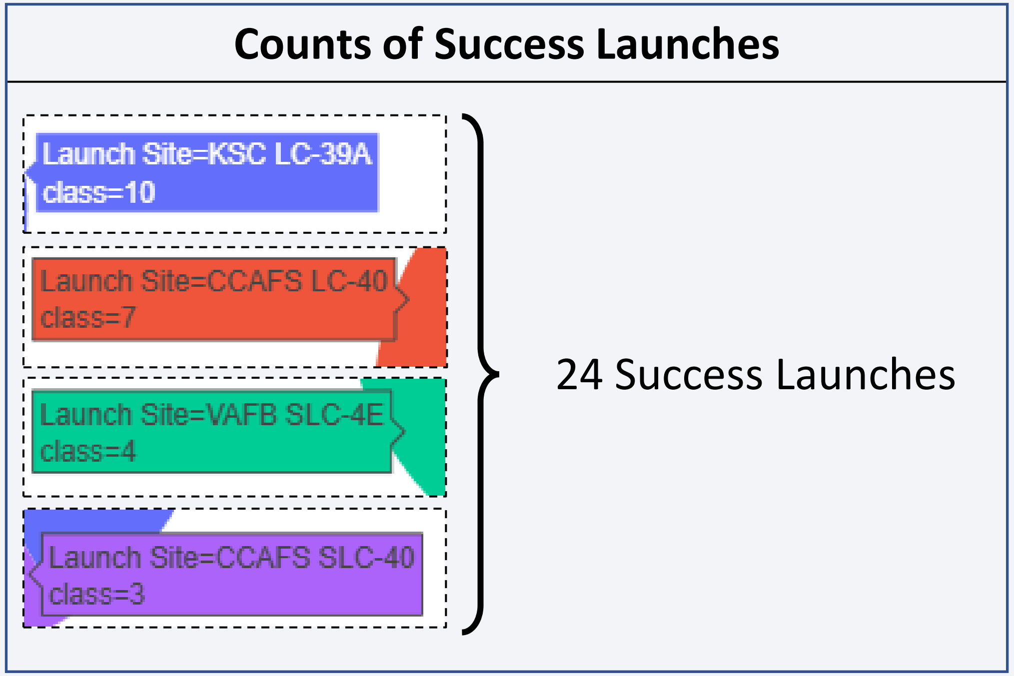
Build a Dashboard with Plotly Dash

Build a Dashboard with Plotly Dash:

- Total Success Launches By Site
- Total Success Launches for “Best Site”
- Payload vs. Launch Outcome for all sites

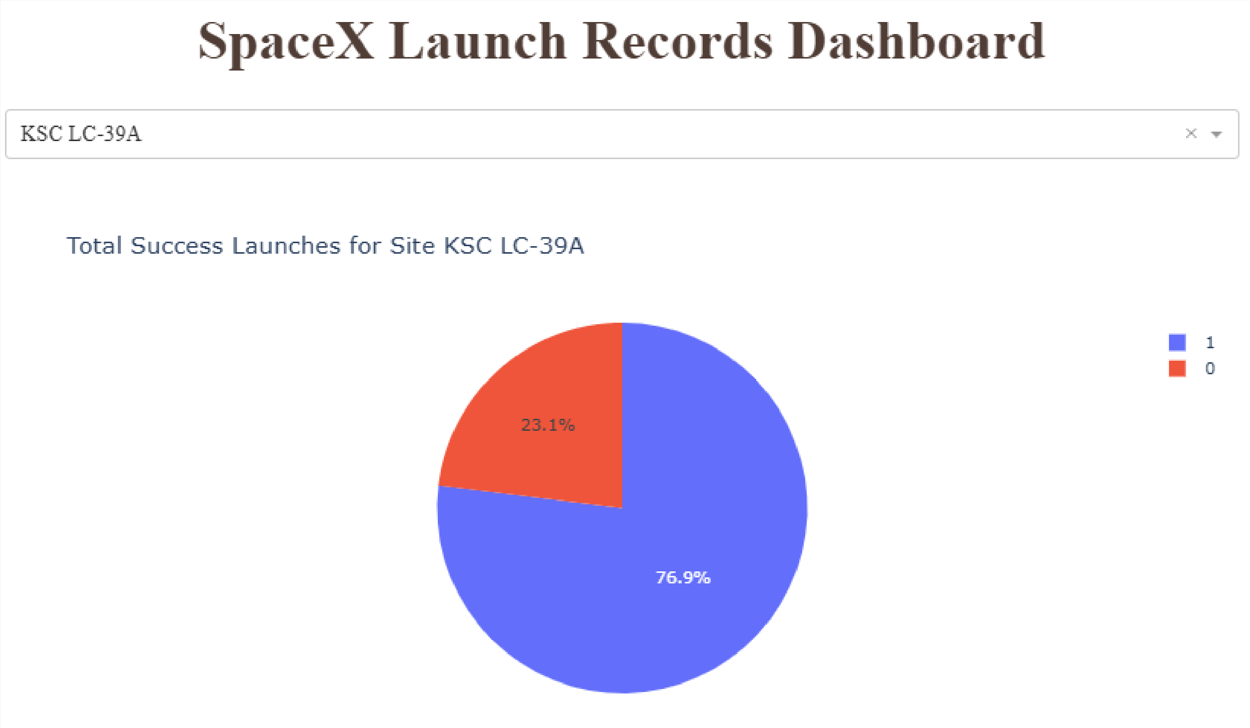
Total Success Launches By Site





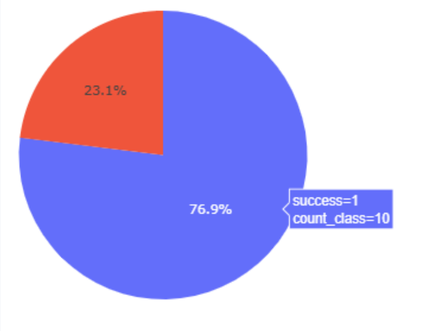
- One **dominant** launch **site** with almost **42%** of all success launches : ***KSC LC-39A***
- Poor Rate (<17%) for VAFB SLC-4E and CCAFS SLC-40

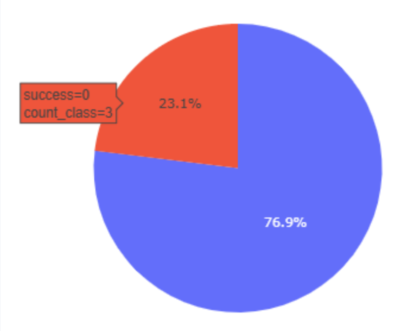
Total Success Launches for “Best Site”



KSC LC-39A

- *13 launches*
- *10 Successes vs. 3 Failures*
- ~77% Success vs. 23% Failures





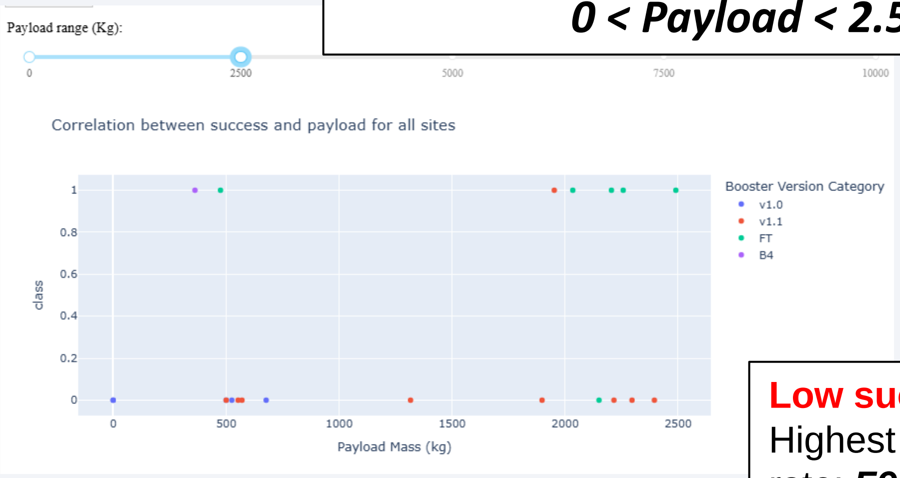
- The Launch Site ***KSC LC-39A*** has the highest launch success ratio:
  → ~77% Successes (10) vs. 23% Failures (3)
  → high majority of successes over failures

Payload vs. Launch Outcome for all sites

0 < Payload < 10k

Highest launch success rate:
→ ***F9 FT***

0 < Payload < 2.5k



Low success rate
Highest launch success rate: ***F9 FT***

2.5k < Payload < 5k

Range with **highest success rate**
Best Booster:
→ ***F9 FT***

5k < Payload < 10k

Low success rate
High Payload masses only with ***FT*** and **B4**

Findings & Insights “Build a Dashboard with Plotly Dash”

<u>Summary of Visual Insights – Analysis of SpaceX launch data:</u>

- Site with the largest successful launches: ***KSC LC-39A***
- Site with the highest launch success rate: ***KSC LC-39A***
- Payload range(s) with the highest launch success rate: ***2.5k < Payload < 5k***
- Payload range(s) with the lowest launch success rate: ***Payload < 2.5k & Payload > 5k***
- F9 Booster version with the highest launch success rate: ***FT***

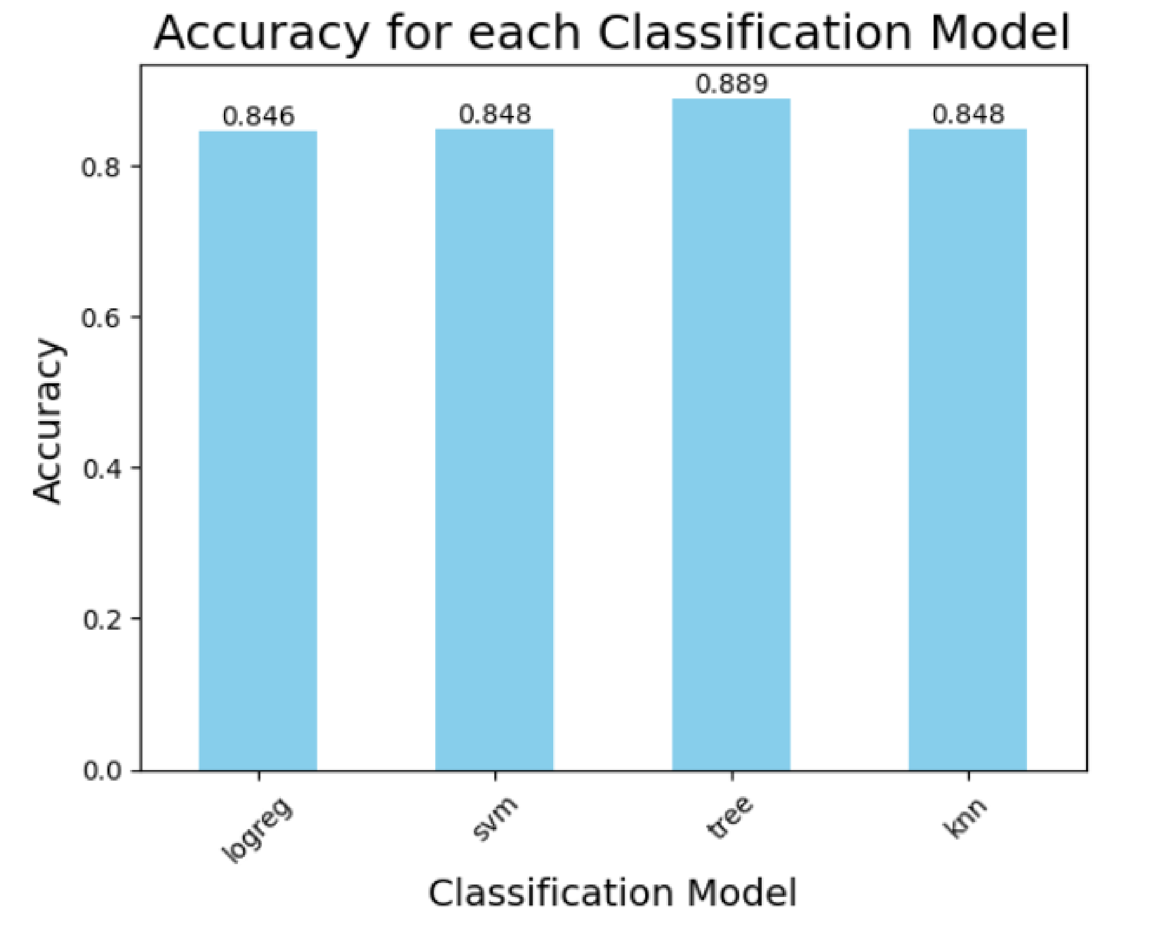
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Logreg, svm and knn deliver almost the same accuracy: ~85%
- Tree has with ~89% the highest accuracy. About 4% more than the other 3 models

→ Highest classification accuracy for the tree model



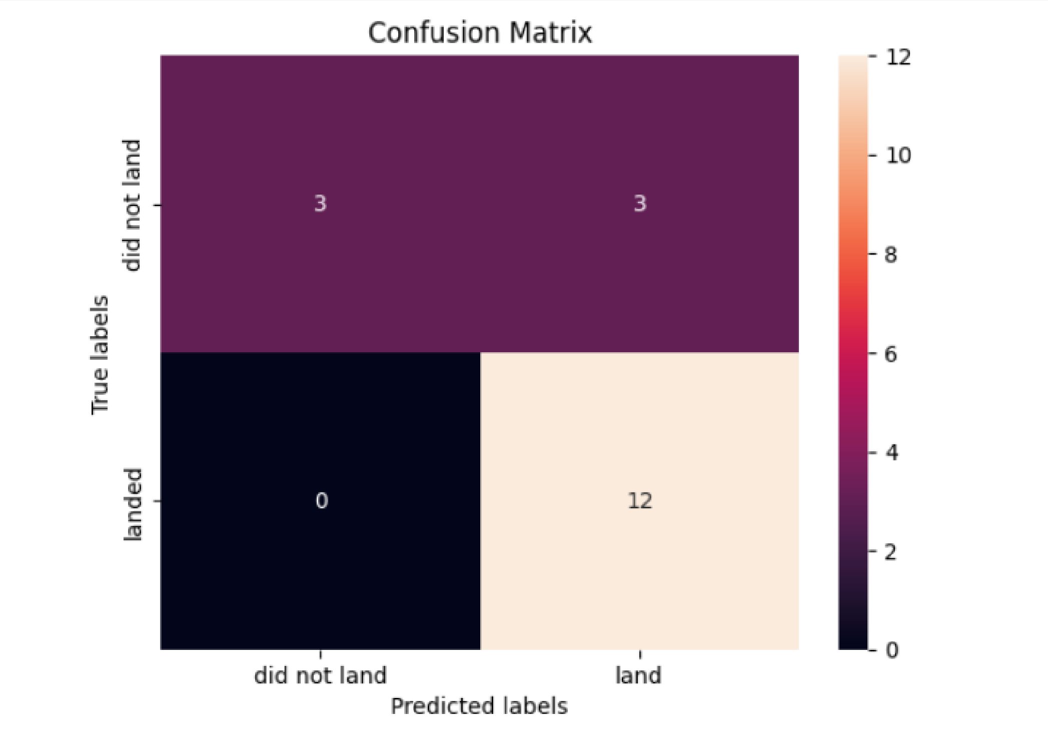
Confusion Matrix

- **<u>Confusion matrix of the decision tree model</u>**

  (best performing one)

  Well distinction between the different classes:

  - True Positive: 12
  - True Negative: 3
  - **False Positive: 3 → Issue**
  - False Negative: 0



Conclusions – Predictive Analysis

- The highest classification accuracy can be found for the tree model classification.
- The Confusion Matrix actually is similar with all 4 models.
- Examining the confusion matrix, we see that the models can well distinguish between the different classes.
- The confusion matrix shows that the problem is false positives (no false negative).

Discussion & Conclusion

Discussion – Findings & Implications

- Importance of launch **sites' location & proximities** :
  - Close proximity to Equator line, coast, railways & highways

    → Rotational Speed Boost, Logistic, Security, Weather Conditions
  - Distance way from cities

    → Security
- Best Launch Site in the US: ***KSC LC-39A***
- Best rocket features: Booster Version ***F9 FT***, ***2.5k < Payload < 5k***
- 4 Orbits with 100% success rate (***ES-L1***, ***GEO*** and ***HEO*** and ***SSO***) *but robustness to be validated with further databases due to small counts*
- ***Combination of features*** *– instead of isolated consideration - may be relevant/further analysed (ex. correlation using payload mass depends on other rocket or launch specifications)*
- Best prediction using the **Decision Tree Classification Model** (with about 90% of accuracy)
- Increasing trend of success and newer technologies → apply prediction model, **monitor trend** and detect changes

Conclusions

- **Optimal launch sites' locations:**
  - **Close to Equator** line, **coastline**, **railways** & **highways**

  → "Best **Launch Site**" in the US: ***KSC LC-39A***
  - **Away from cities**
- **Optimal rocket characteristics:**
  - Payload Mass (*KSC LC-39A*) ***2.5k < Payload < 5k***
  - Booster Version ***F9 FT***
  - ***Orbits: ES-L1***, ***GEO***, ***HEO*** and ***SSO*** *(but robustness to be further validated)*
  - **Combination of features** to be taken into account (payload mass & other specifications)
- **Best Predictive model:**
  - Decision Tree Classification model
- **Next Steps:**
  - Evaluate more **recent databases** (after year 2020)

    → use model, monitor trends, detect changes, evaluate further & newer metrics

Appendix

Appendix

- EDA SQL Queries
- Folium Supplements
- References
- Acknowledgments

Appendix:

EDA SQL Queries

All Launch Site Names

Appendix: Detailed SQL Queries

- Find the names of the unique launch sites

```
QUERY = "SELECT DISTINCT Launch_Site FROM SPACEXTABLE"
df_querry = pd.read_sql_query(QUERY, con)
print(df_querry)
```

```
    Launch_Site
0   CCAFS LC-40
1   VAFB SLC-4E
2    KSC LC-39A
3  CCAFS SLC-40
```

Launch Site Names Begin with 'CCA'

Appendix: Detailed SQL Queries

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

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

```
QUERY = """
SELECT *
FROM SPACEXTABLE
WHERE Launch_Site LIKE 'CCA%' LIMIT 5
"""

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	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

Appendix: Detailed SQL Queries

- Calculate the total payload carried by boosters from NASA

Task 3

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

QUERY = """ SELECT * FROM SPACEXTABLE WHERE Customer= 'NASA (CRS)' """

df_querry = pd.read_sql_query(QUERY, con) #print(df_querry) df_querry

```
QUERY = """
SELECT SUM(PAYLOAD_MASS__KG_) as Sum_PLM
FROM SPACEXTABLE
WHERE Customer= 'NASA (CRS)'
"""

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	Sum_PLM
0	45596

Average Payload Mass by F9 v1.1

Appendix: Detailed SQL Queries

- Calculate the average payload mass carried by booster version F9 v1.1

```
QUERY = """
SELECT AVG(PAYLOAD_MASS__KG_) as avg_PLM
FROM SPACEXTABLE
WHERE Booster_Version = 'F9 v1.1'
"""

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	avg_PLM
0	2928.4

First Successful Ground Landing Date

Appendix: Detailed SQL Queries

- Find the dates of the first successful landing outcome on ground pad

```
QUERY = """
SELECT Date
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)'
ORDER BY Date ASC LIMIT 1
"""

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	Date
0	2015-12-22

Successful Ground Landing Records

Appendix: Detailed SQL Queries

- List of successful landing outcome on ground pad

```
QUERY = """
SELECT *
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (ground pad)'
"""

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)
1	2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2	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)
3	2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
4	2017-06-03	21:07:00	F9 FT B1035.1	KSC LC-39A	SpaceX CRS-11	2708	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
5	2017-08-14	16:31:00	F9 B4 B1039.1	KSC LC-39A	SpaceX CRS-12	3310	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
6	2017-09-07	14:00:00	F9 B4 B1040.1	KSC LC-39A	Boeing X-37B OTV-5	4990	LEO	U.S. Air Force	Success	Success (ground pad)
7	2017-12-15	15:36:00	F9 FT B1035.2	CCAFS SLC-40	SpaceX CRS-13	2205	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
8	2018-01-08	1:00:00	F9 B4 B1043.1	CCAFS SLC-40	Zuma	5000	LEO	Northrop Grumman	Success (payload status unclear)	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

Appendix: Detailed SQL Queries

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

```
QUERY = """
SELECT Booster_Version
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (drone ship)'
    AND PAYLOAD_MASS__KG_ > 4000
        AND  PAYLOAD_MASS__KG_ < 6000
"""

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	Booster_Version
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

```
QUERY = """
SELECT COUNT(Booster_Version)
FROM SPACEXTABLE
"""

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	COUNT(Booster_Version)
0	101

Total Number of Successful and Failure Mission Outcomes

Appendix: Detailed SQL Queries

- Calculate the total number of successful and failure mission outcomes

```
QUERY = """
SELECT COUNT(Mission_Outcome) as Sum_Sucess
FROM SPACEXTABLE
WHERE Mission_Outcome LIKE 'Success%'
"""

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	Sum_Sucess
0	100

```
QUERY = """
SELECT COUNT(Mission_Outcome) as Sum_Failure
FROM SPACEXTABLE
WHERE Mission_Outcome LIKE 'Failure%'
"""

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	Sum_Failure
0	1

```
QUERY = "SELECT COUNT(Mission_Outcome) FROM SPACEXTABLE"

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	COUNT(Mission_Outcome)
0	101

Boosters Carried Maximum Payload

Appendix: Detailed SQL Queries

- Calculus of max Payload Mass

```
QUERY = "SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE"

df_querry = pd.read_sql_query(QUERY, con)
df_querry
```

	MAX(PAYLOAD_MASS__KG_)
0	15600

- List the names of the booster which have carried the maximum payload mass

```
QUERY = """
SELECT Booster_Version, PAYLOAD_MASS__KG_
FROM SPACEXTABLE
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
"""

df_querry = pd.read_sql_query(QUERY, con)
#print(df_querry)
df_querry
```

	Booster_Version	PAYLOAD_MASS__KG_
0	F9 B5 B1048.4	15600
1	F9 B5 B1049.4	15600
2	F9 B5 B1051.3	15600
3	F9 B5 B1056.4	15600
4	F9 B5 B1048.5	15600
5	F9 B5 B1051.4	15600
6	F9 B5 B1049.5	15600
7	F9 B5 B1060.2	15600
8	F9 B5 B1058.3	15600
9	F9 B5 B1051.6	15600
10	F9 B5 B1060.3	15600
11	F9 B5 B1049.7	15600

2015 Launch Records

Appendix: Detailed SQL Queries

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
QUERY = """
SELECT
    CASE strftime('%m', Date)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
    END AS month_name,
    Landing_Outcome, Booster_Version, Launch_Site
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Failure (drone ship)'
    AND strftime('%Y', Date) = "2015"
"""
df_querry = pd.read_sql_query(QUERY, con)
df_querry
```

	month_name	Landing_Outcome	Booster_Version	Launch_Site
0	January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
1	April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

```
QUERY = """
SELECT
    Date,
    strftime('%Y', Date) AS year,
    strftime('%m', Date) AS month,
    strftime('%d', Date) AS day,
    CASE strftime('%m', Date)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
    END AS month_name,
    Mission_Outcome, Landing_Outcome, Launch_Site
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Failure (drone ship)'
    AND strftime('%Y', Date) = "2015"
"""
df_querry = pd.read_sql_query(QUERY, con)
df_querry
```

	Date	year	month	day	month_name	Mission_Outcome	Landing_Outcome	Launch_Site
0	2015-01-10	2015	01	10	January	Success	Failure (drone ship)	CCAFS LC-40
1	2015-04-14	2015	04	14	April	Success	Failure (drone ship)	CCAFS LC-40

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

Appendix: Detailed SQL Queries

- 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

```
QUERY = """
SELECT
    Landing_Outcome, COUNT(Landing_Outcome) as count_Outcome
FROM SPACEXTABLE
GROUP BY Landing_Outcome
ORDER BY  COUNT(Landing_Outcome) DESC
"""
df_querry = pd.read_sql_query(QUERY, con)
df_querry
```

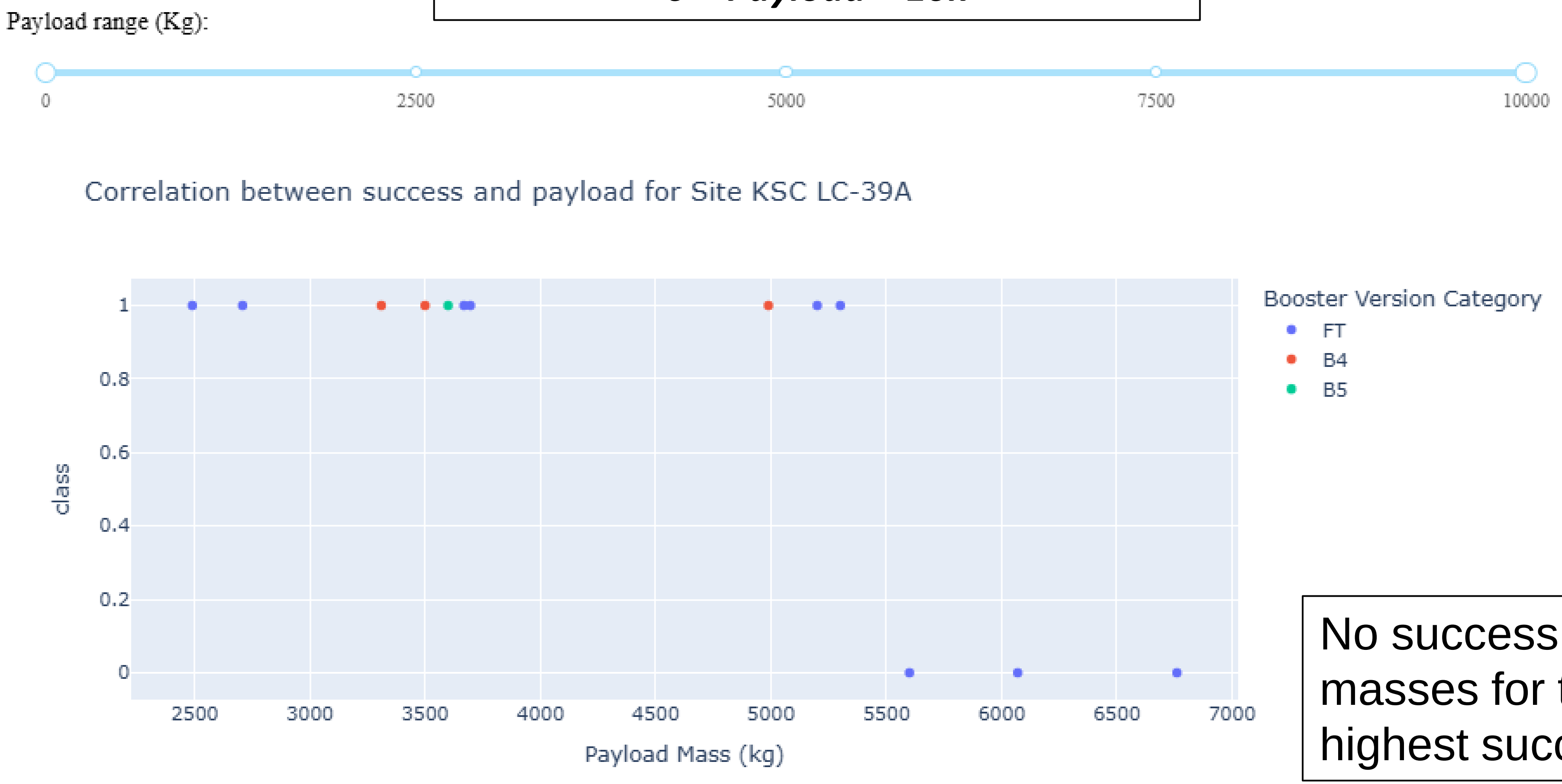
	Landing_Outcome	count_Outcome
0	Success	38
1	No attempt	21
2	Success (drone ship)	14
3	Success (ground pad)	9
4	Failure (drone ship)	5
5	Controlled (ocean)	5
6	Failure	3
7	Uncontrolled (ocean)	2
8	Failure (parachute)	2
9	Precluded (drone ship)	1
10	No attempt	1

Appendix:

Insights using Folium - Supplements

Payload vs. Launch Outcome for KSC LC-39A

0 < Payload < 10k



No success with high payload masses for the site with the highest success launch rate

Appendix:

References & Acknowledgments

References

Appendix: References & Acknowledgments

Dataset Source:

- Dataset “List of Falcon 9 and Falcon Heavy launches” provided by SpaceX:

→ https://api.spacexdata.com/v4/launches/past (SpaceX API)

→ https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922 (Wikipedia)

- Modified datasets from the IBM Team for the purposes of that project, such as:

→ https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json (API LAB)

→ https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv (Data Wrangling LAB)

→ https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv (EDA LAB)

→ https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_geo.csv (Folium LAB)

Acknowledgments

Appendix: References & Acknowledgments

Many Thanks to:

- **SpaceX** who made the “List of Falcon 9 and Falcon Heavy launches” available through API and on Wikipedia

- **IBM** for elaborating and providing us students with that Capstone Project

Thank you!