IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Salvatore Vivolo
@SalvaAtWork

2021/09/19

Outline

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

Executive Summary

Methodologies

- Data collection with APIs and Web Scraping
- Data wrangling
- Exploratory Data Analysis (EDA) with visualizations and SQL
- Interactive visual analytics with Folium and Plotly Dash
- Predictive analysis with Classification Models

Results

- Exploratory Data Analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Introduction

Background

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

Therefore if we can determine if the first stage will land, we can determine the cost of a launch.
This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Problem

Our purpose is to collect, process and analyze available data from various public sources about Falcon 9 rocket launches, in order to be able to identify patterns, to see directly how variables might be related to each other, and which conditions Space X has to achieve to obtain the best landing success rate.

This will allow us to build, evaluate and refine predictive models to discover exciting insights about the data, and given a certain set of boundary conditions, to predict if the first stage of the Falcon 9 will land successfully (enabling reuse and therefore cost reduction).

Section 1

Methodology

Methodology

Executive Summary

- **Data collection**
  - REST APIs
  - Web Scraping with BeautifulSoup
- **Data wrangling**
  - Dealing with missing values
  - Turning Categorical values to numeric variables (One Hot Encoding)
- **Exploratory data analysis (EDA) with visualizations and SQL**
- **Interactive visual analytics with Folium and Plotly Dash**
- **Predictive analysis**
  - Selection of 4 classification models
  - Building, tuning and evaluation of classification models

Data Collection

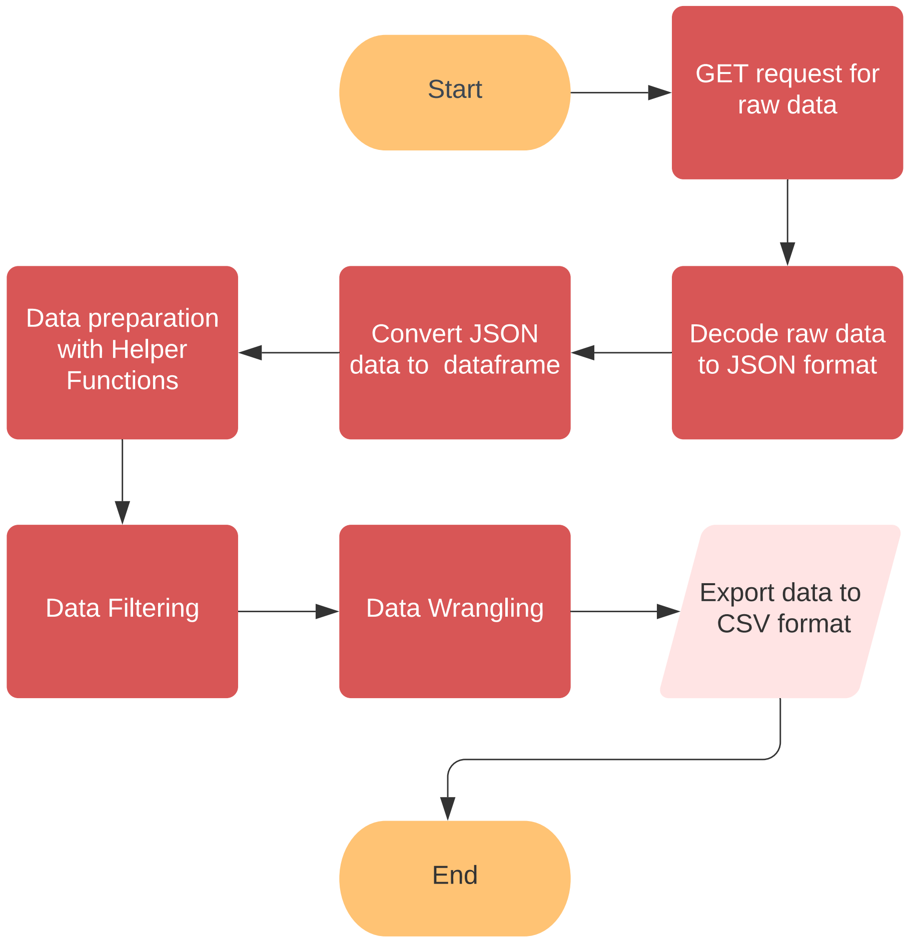
- We have collected SpaceX launch data including information about the rocket type, the delivered payload, launch specifications, landing specifications and the landing outcome.
- Our goal is to use this data to predict wheter SpaceX will attempt to land a rocket or not.
- To collect the data we have used two different techniques:
  - A REST API from SpaceX
  - Web Scraping on a Wikipedia page using BeautifulSoup

Data Collection "SpaceX API"

- **Process**
  - GET request to SpaceX API
  - Decode raw data to JSON format
  - Convert JSON data to Pandas dataframe
  - Data preparation (integrate additional data in the dataframe using Helper Functions)
  - Data Filtering (remove unnecessary data such as Falcon 1 lunches)
  - Data Wrangling (deal with missing values)
  - Export data to CSV

- **GitHub URL**
  https://github.com/salva1973/coursera-capstone-project/blob/master/01.%20Data%20Collection%20API.ipynb

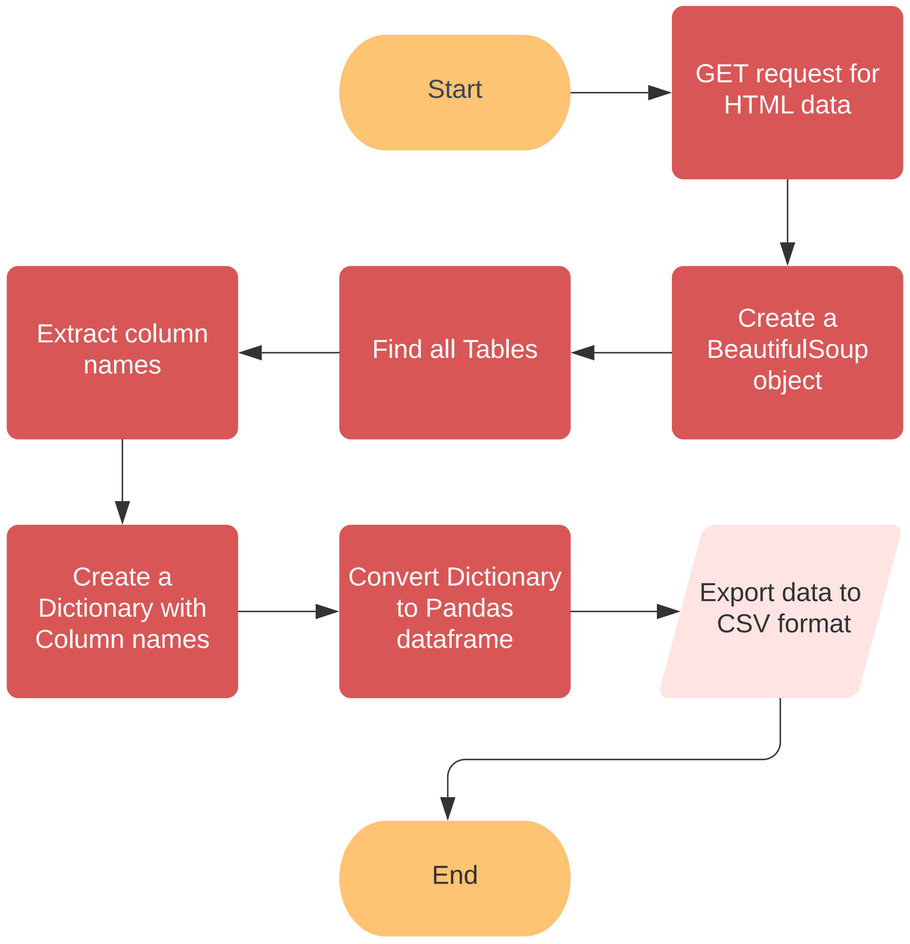
Data Collection with SpaceX REST API



Data Collection
"Web Scraping"

- **Process**
  - GET request to Wikipedia launches page
  - Create a BeautifulSoup object using the HTML
  - Find all tables in the object
  - Extract the column names
  - Create dictionary using column names as keys
  - Convert dictionary to Pandas dataframe using helper functions to process web scraped HTML table
  - Export data to CSV
- GitHub URL
  https://github.com/salva1973/coursera-capstone-project/blob/master/02.%20Data%20Collection%20with%20Web%20Scraping.ipynb

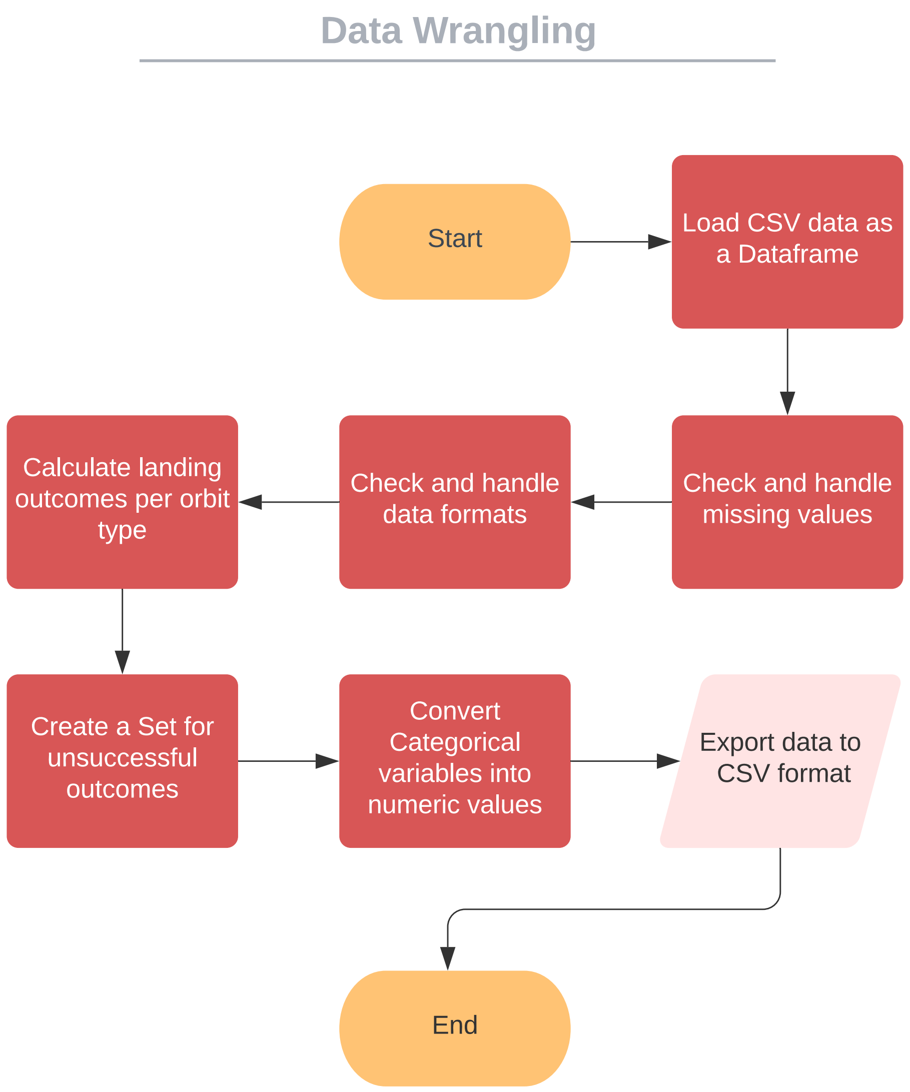
Data Collection with Web Scraping



Data Wrangling

- **Process**
    - Load CSV data as a dataframe
    - Check and handle missing values
    - Check and handle data formats
    - Calculate landing outcomes per orbit type
    - Create a set for unsuccessful outcomes
    - Convert Categorical variables into numeric values, using dummy variables and One Hot Encoding
    - Export data to CSV

- GitHub URL
  https://github.com/salva1973/coursera-capstone-project/blob/master/03.%20EDA%20(Exploratory%20Data%20Analysis)%20Data%20Wrangling.ipynb



EDA with Data Visualization

- Scatter point charts
  - Flight Number vs Payload Mass
  - Flight Number vs Launch Site
  - Payload Mass vs Launch Site
  - Flight Number vs Orbit Type
  - Payload Mass vs Orbit Type
- Why
  - Used to analyze the **correlation** between two variables for a large dataset

- Bar charts
  - Success rate of each orbit
- Why
  - Use to analyze the **frequency distribution** of a variable, represented in the Y axis, across different groups (in the X axis) typically created using data binning. It's useful to compare big sets of data from different groups at a glance.

- Line chart
  - Year vs Average Success Rate
- Why
  - Used to show **data trends** very clearly, and to predict future outcomes.

- GitHub URL
  https://github.com/salva1973/coursera-capstone-project/blob/master/05.%20EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

DATASET ANALYSYS WITH SQL QUERIES
1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first successful landing outcome in ground pad was achieved
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
9. List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
10. 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

- GitHub URL
  https://github.com/salva1973/coursera-capstone-project/blob/master/04.%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- We added the following objects to the Folium map
  - A circle at NASA Johnson Space Center with a popup label showing its name, as a first example
  - A circle for each launch site with popup labels showing their name, based on their coordinates (Lat, Long), to easily identify the sites on the map
  - A MarkerCluster object to include colored Markers for each launch (green icons for successful launches, red icons for the failed ones), to visualize successes and failures on the map
  - Mouse position to explore the proximity of each launch site and collect the coordinates of railways, highways, coastline, etc.
  - Lines to show calculated distances in km between launch sites and proximity places of interest (cities, coastline, etc.)
- We discovered the following
  - It seems launch sites are usually in close proximity to coastlines, but normally far away from cities, highways and railways, probably for security reasons.

- GitHub URL
  https://github.com/salva1973/coursera-capstone-project/blob/master/06.%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

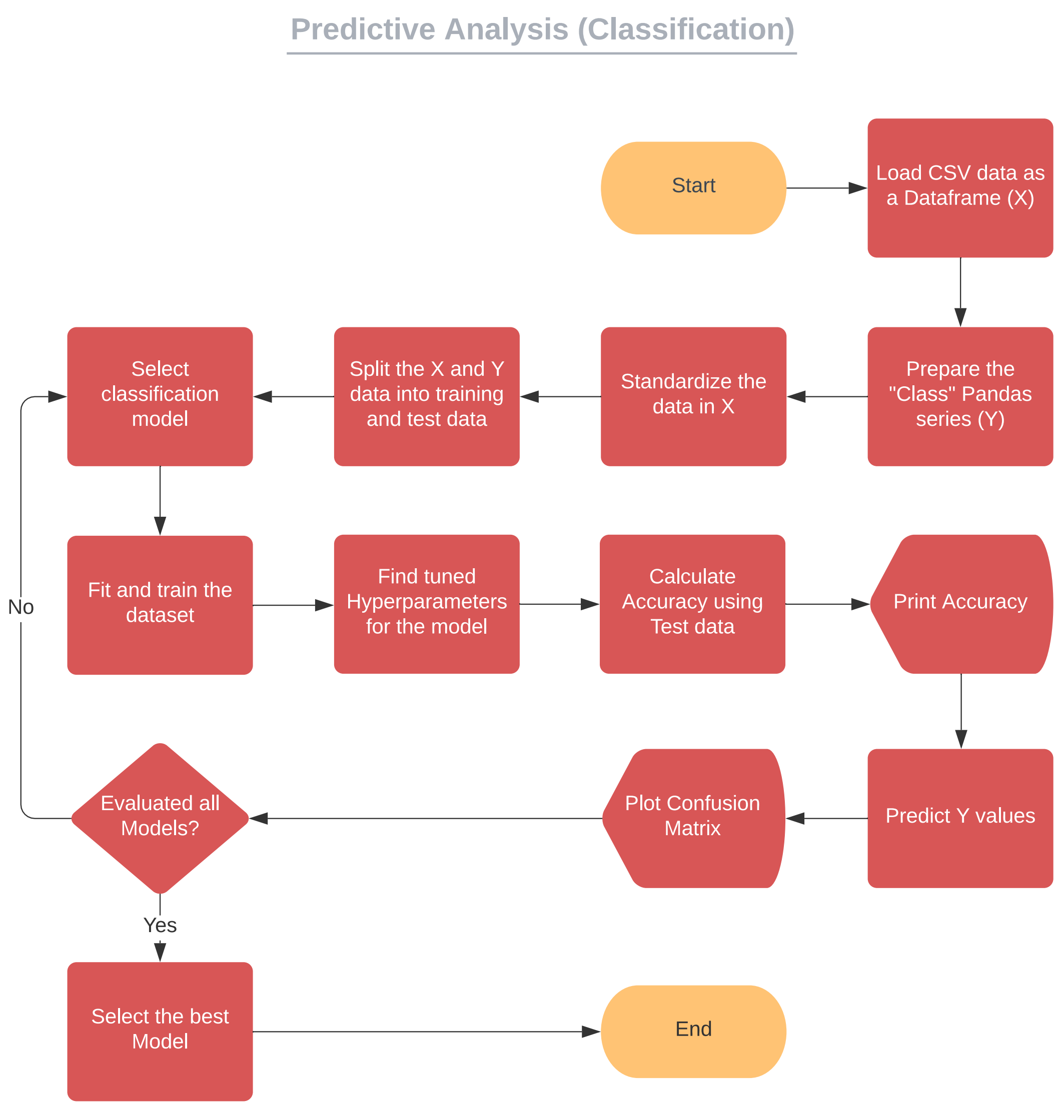
Build a Dashboard with Plotly Dash

- Dashboard components
  - Dropdown list to select the launch site
  - Interactive Pie Chart to show
    - The total successful launches count if all sites are selected
    - Success vs failed count if a specific launch site is selected

  (the dropdown can be used to change the chart)
  - Slider to select the payload mass range
  - Interactive Scatter point chart to show the correlation between the payload mass and the launch success, for the different Booster versions (both the dropdown and the slider can be used to change the chart)
- GitHub URL
  https://github.com/salva1973/coursera-capstone-project/blob/master/spacex_dash_app.py

Predictive Analysis "Classification"

- **Process**
  - Load the dataset into a Dataframe
  - Standardize the data
  - Split the data into training and test data
  - Select classification model (Logistic Regression, KNN, Decision Tree, SVN) (**BUILD**)
  - Fit and Train dataset (**EVALUATE**)
  - Find tuned Hyperparameters for the model
  - Calculate and print Accuracy using test data (**TUNE**)
  - Predict Y values and plot Confusion Matrix
  - Select model with best performance (Accuracy) (**SELECT**)
- GitHub URL
  https://github.com/salva1973/coursera-capstone-project/blob/master/07.%20Machine%20Learning%20Prediction.ipynb



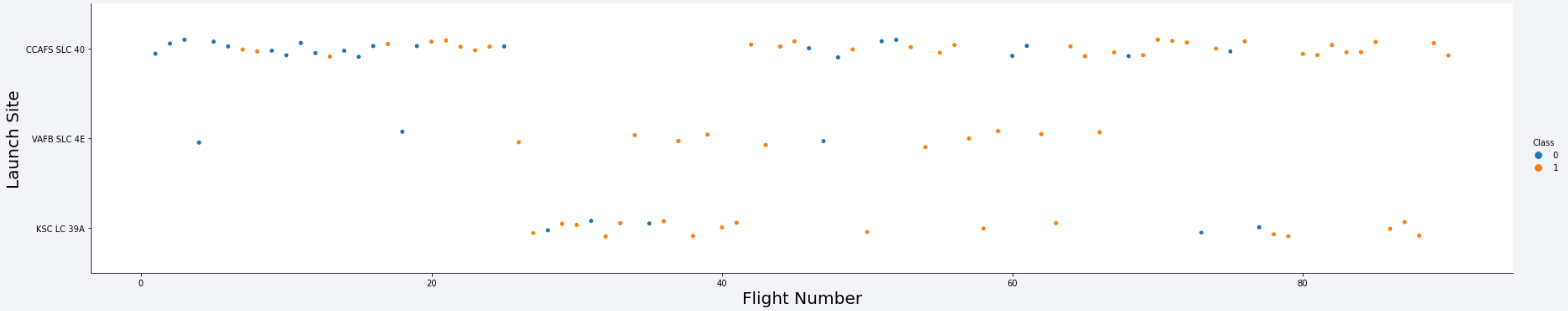
Results

- Exploratory Data Analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

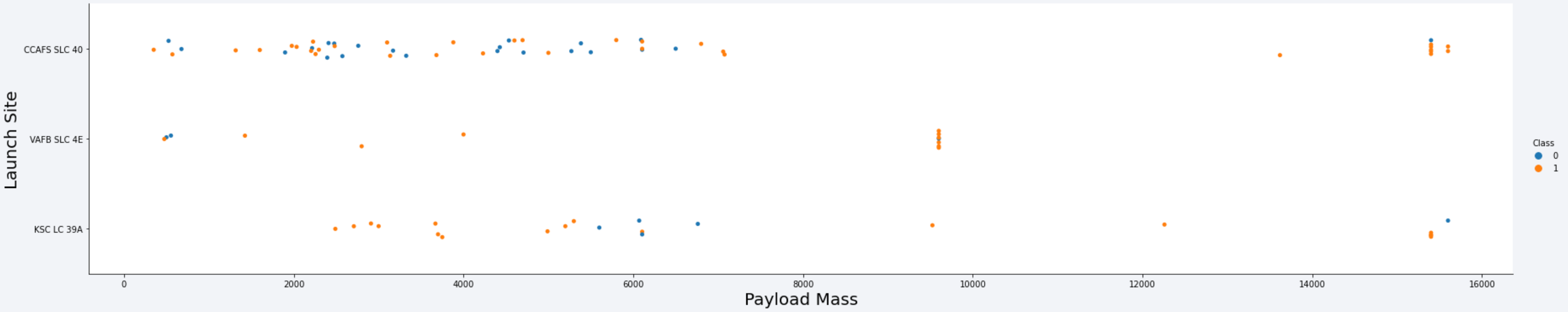


Class = 0 --> Failure, Class = 1--> Success

INSIGHTS

- As the flight number increases, it is more likely that there will be successful landings for each launching site
- CCAFS SLC-40 have more flights compared to the other launching sites
- VAFB SLC 4E has the least number of flights compared to the other launching sites

Payload vs. Launch Site

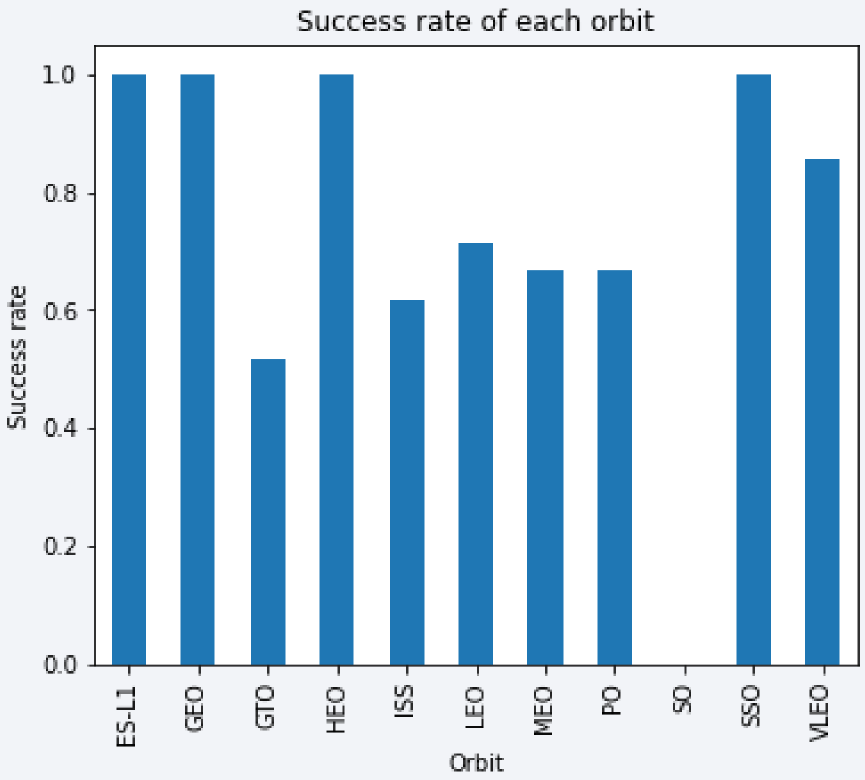


Class = 0 --> Failure, Class = 1--> Success

INSIGHTS

- For site CCAFS SLC-40 a higher payload seems to be related to a higher success rate
- There is no clear pattern for the other sites, concerning the correlation between the payload and the success rate

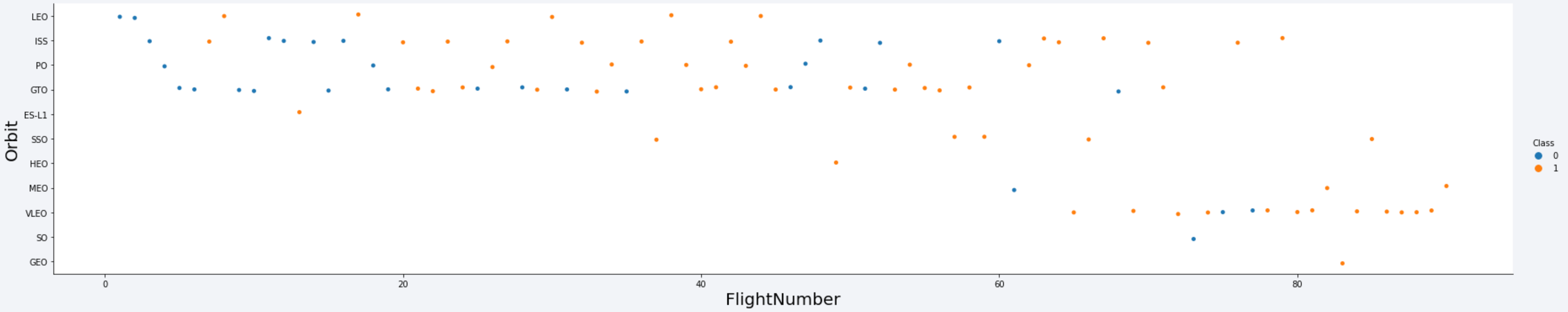
Success Rate vs. Orbit Type



INSIGHTS

- ES-L1, GEO, HEO and SSO have the highest success rate
- SO has the lowest success rate

Flight Number vs. Orbit Type

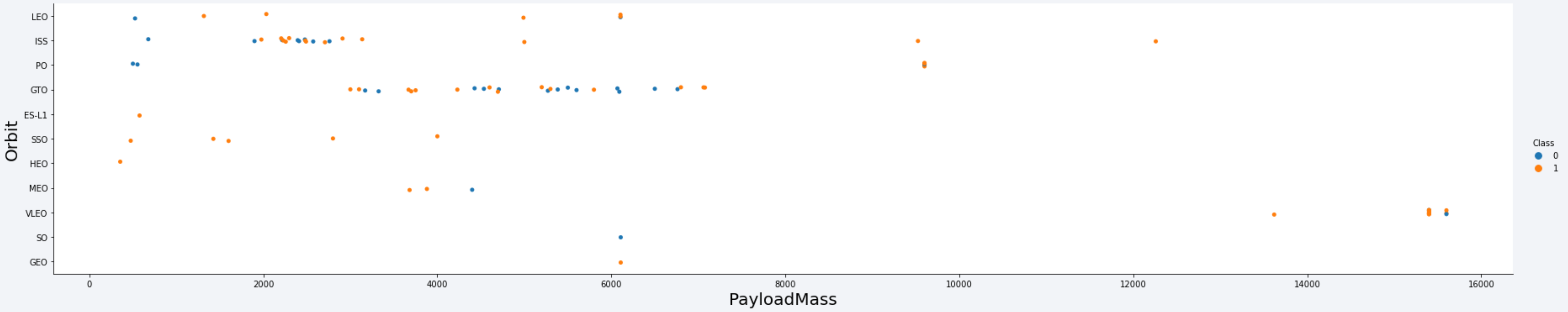


Class = 0 --> Failure, Class = 1--> Success

INSIGHTS

- In the LEO orbit the Success appears related to the number of flights
- There seems to be no relationship between flight number and success rate when in GTO orbit
- Orbits SO and GEO have only 1 launch in history

Payload vs. Orbit Type

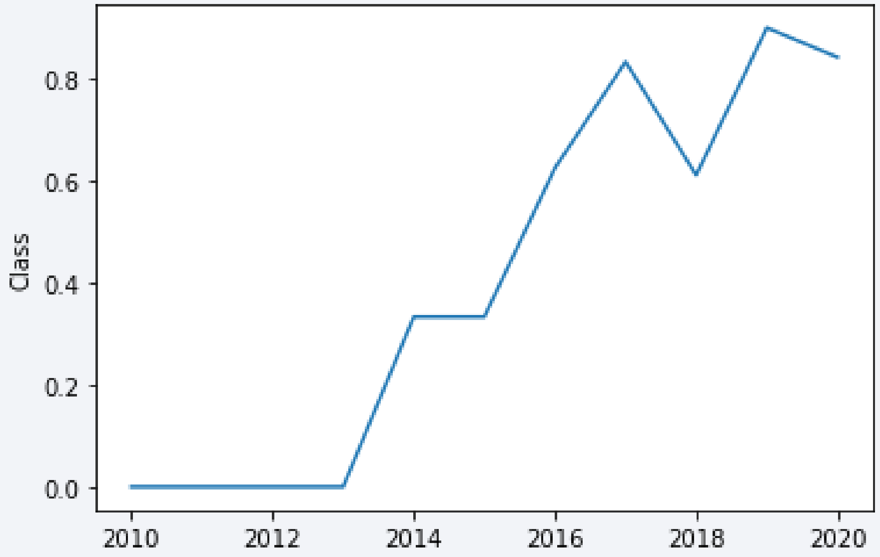


Class = 0 --> Failure, Class = 1--> Success

INSIGHTS

- Heavy payloads have a negative influence on GTO orbits and positive Polar LEO (ISS) orbits
- The highest payloads are associated to the VLEO orbit

Launch Success Yearly Trend



INSIGHTS

- The success rate since 2013 kept increasing until 2020

Section 3

EDA with SQL

All Launch Site Names

```
SELECT distinct(LAUNCH_SITE)
FROM SPACEXTBL
```

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

- The '**distinct**' in the query will return only unique values for the LAUNCH_SITE column, from the SPACEXTBL table

Launch Site Names beginning with 'CCA'

```
SELECT *
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5
```

- The '**LIMIT**' in the query will return only 5 records
- The '**LIKE**', using 'CCA' and the **wildcard '%'** will return all records beginning with 'CCA'

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass launched by NASA

```
SELECT SUM(PAYLOAD_MASS__KG_) AS PAYLOAD
FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)'
```

payload
45596

- The function **'SUM'** calculates the sum from the column 'PAYLOAD_MASS__KG_)
- **'AS'** is used to name the sum as 'payload'
- **'WHERE'** is used to filter only records corresponding to the customer 'NASA (CRS)', and use only those records for the calculation

Average Payload Mass by F9 v1.1

```
SELECT AVG(PAYLOAD_MASS__KG_) as avg_payload
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

avg_payload
2928.400000

- The function '**AVG**' calculates the average from the column 'PAYLOAD_MASS__KG_)
- '**AS**' is used to name the average as 'avg_payload'
- '**WHERE**' is used to filter only records corresponding to the Booster version 'F9 v1.1', and use only those records for the calculation

First Successful Ground Landing Date

```
SELECT MIN(DATE) as min_date
FROM SPACEXTBL
WHERE LANDING__OUTCOME LIKE '%Success%'
```

min_date
2015-12-22

- The function '**MIN**' find the minimum date from the column 'DATE')
- '**WHERE**' is used to filter only records with a 'LANDING__OUTCOME' equal to 'Success', and use only those records for the search with MIN

Successful Drone Ship Landing with Payload between 4000 and 6000 Kg

```
SELECT booster_version, *
FROM SPACEXTBL
WHERE landing__outcome = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

- '**WHERE**' is used to filter records with two conditions
  - 'landing_outcome' = 'Success (drone ship)'
  - 'PAYLOAD_MASS__KG_' BETWEEN 4000 and 6000

booster_version	DATE	time__utc_	booster_version_1	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__ou
F9 FT B1022	2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drc ship)
F9 FT B1026	2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drc ship)
F9 FT B1021.2	2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drc ship)
F9 FT B1031.2	2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drc ship)

Total Number of Successful and Failure Mission Outcomes

```
SELECT mission_outcome, COUNT(mission_outcome) as total
FROM SPACEXTBL
GROUP BY mission_outcome
```

- 'GROUP' is used to split the 'COUNT' results by 'mission_outcome'

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

Boosters Carried Maximum Payload

```
SELECT booster_version
FROM SPACEXTBL
WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_)
                           FROM SPACEXTBL)
```

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

- A subquery is used to filter the boosters that have carried the maximum payload mass

2015 Launch Records

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

```
SELECT landing__outcome, booster_version, launch_site
FROM SPACEXTBL
WHERE landing__outcome = 'Failure (drone ship)'
AND DATE LIKE '2015%'
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- '**WHERE**' is used to filter records with two conditions
  - 'landing_outcome' = 'Failure (drone ship)'
  - 'DATE' equals '2015'

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

```
SELECT landing__outcome, COUNT(landing__outcome) as landing_outcome
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY landing_outcome desc
```

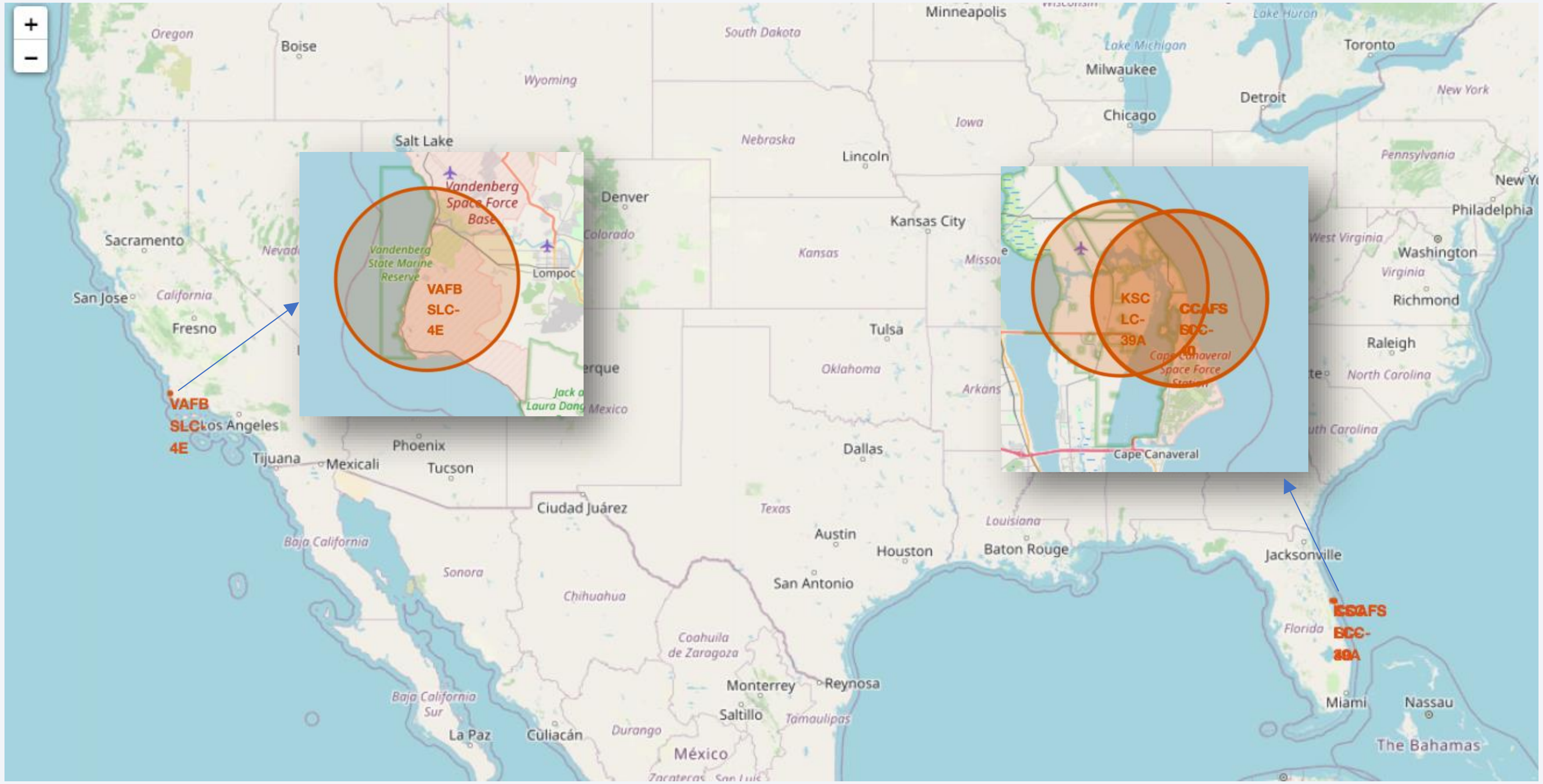
landing__outcome	landing_outcome
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- '**WHERE**' is to filter the records and return only the ones within the selected date range
- '**GROUP**' is used to split the '**COUNT**' results by 'landing_outcome'
- '**ORDER**' is used to order the records by 'landing_outcome' counts

Section 4

Launch Sites Proximities Analysis

All Launch Sites location

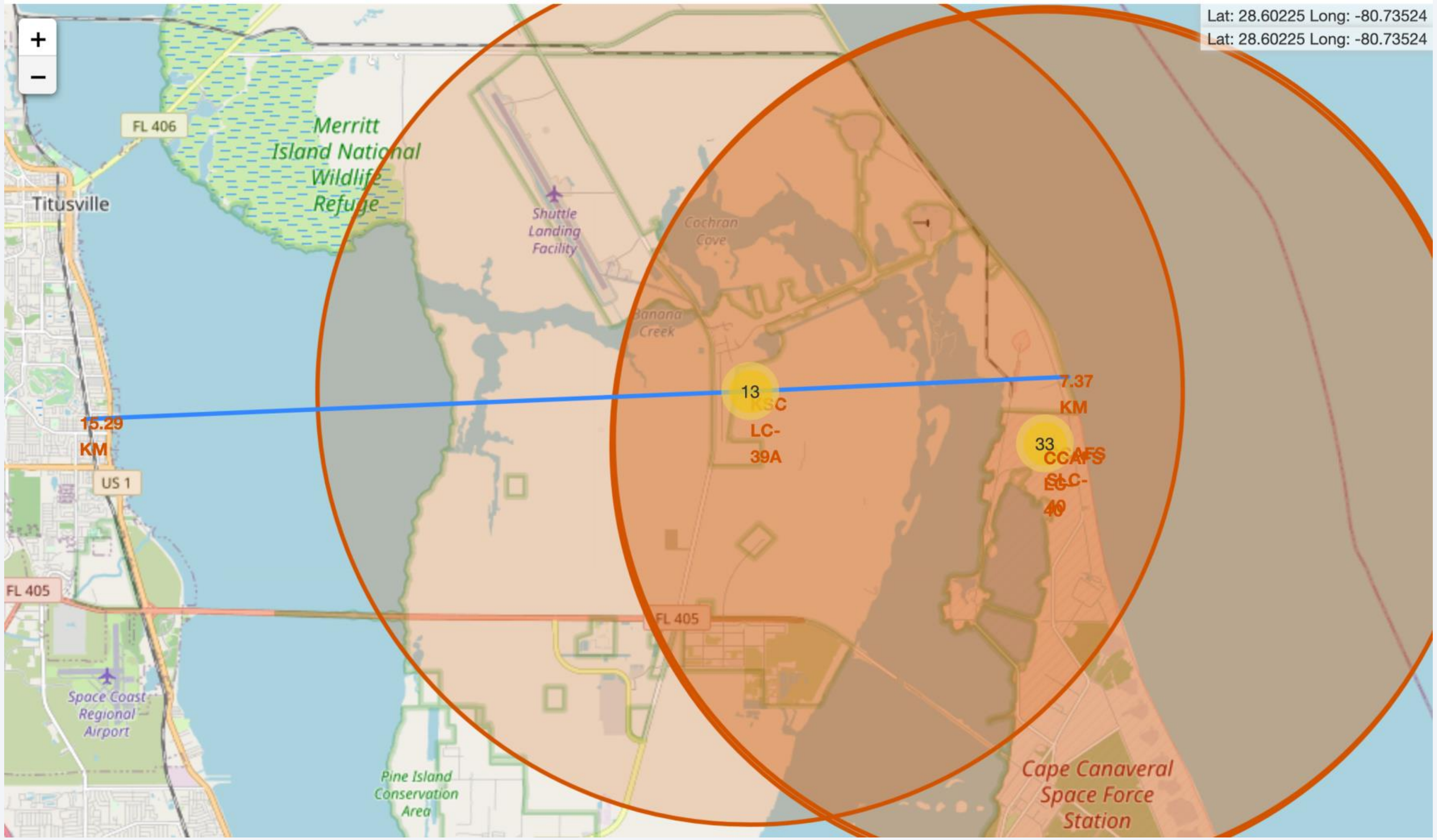


- All Launch Sites for SpaceX are in USA coasts, namely in California and Florida.
- California
  - "VAFB SLC-4E"
- Florida
  - "CCAFS LC-40"
  - "CCAFS SLC-40"
  - "KSC LC-39A"

Launch Outcomes

- Success rate
  - "VAFB SLV-4E"
    4 / 10 = 40%
  - "KSC LC-39A"
    10 / 13 = 77%
  - "CCAFS LC-40"
    7 / 26 = 27%
  - "CCAFS SLC-40"
    3 / 7 = 43%
- Highest success rate
  KSC LC-39A (77%)
- Lowest success rate
  CCAFS LC-40 (27%)

Launch Sites Proximities



- Selected Launch Site
  KSC LC-39A
- Distance to nearest Railway
  15.29 KM
- Distance to Coastline
  7.37 KM

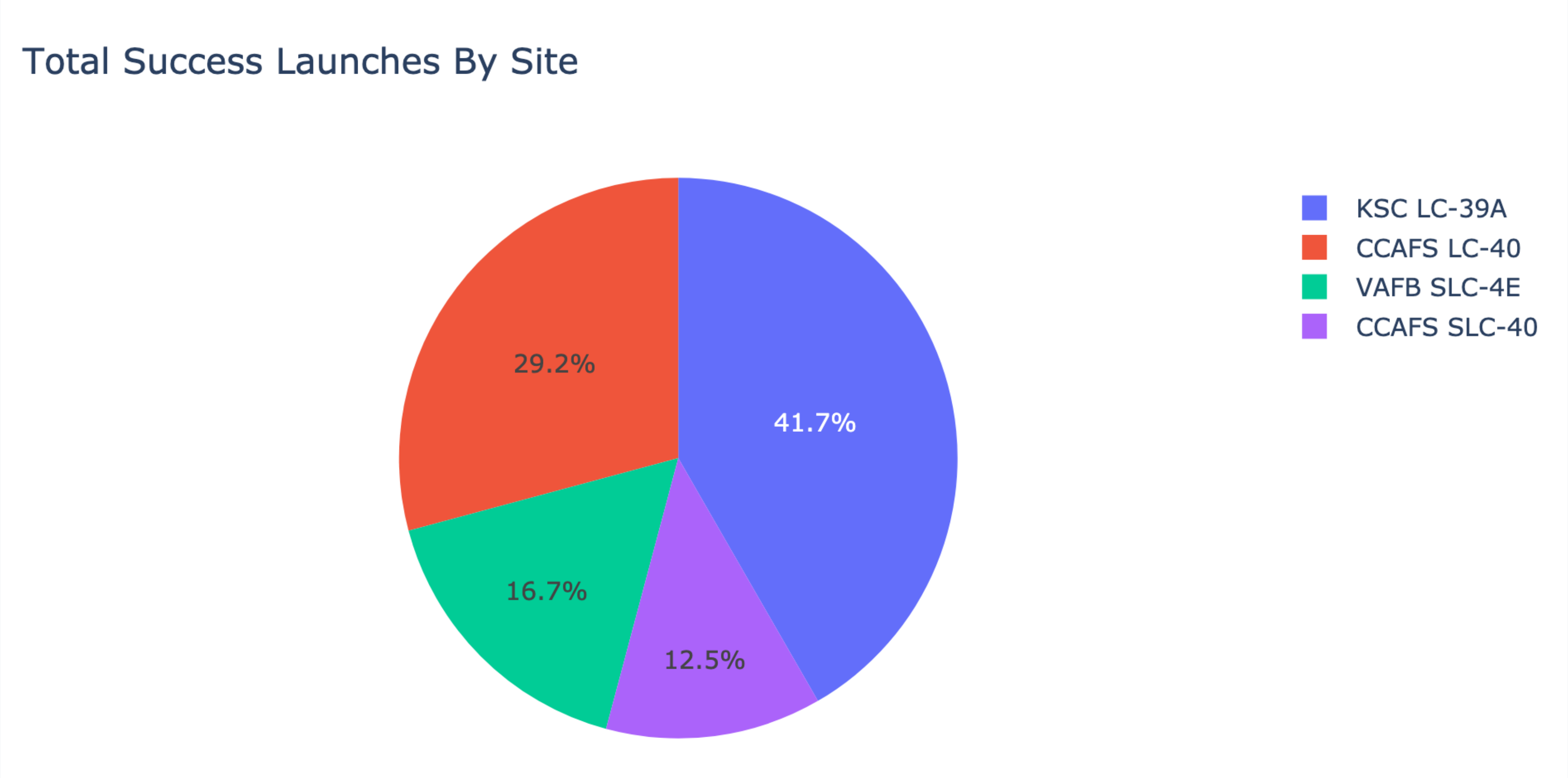
INSIGHTS

- Launch sites are never in close proximity to railways, highways or cities
- Launch sites are always in close proximity to the coastline

Section 5

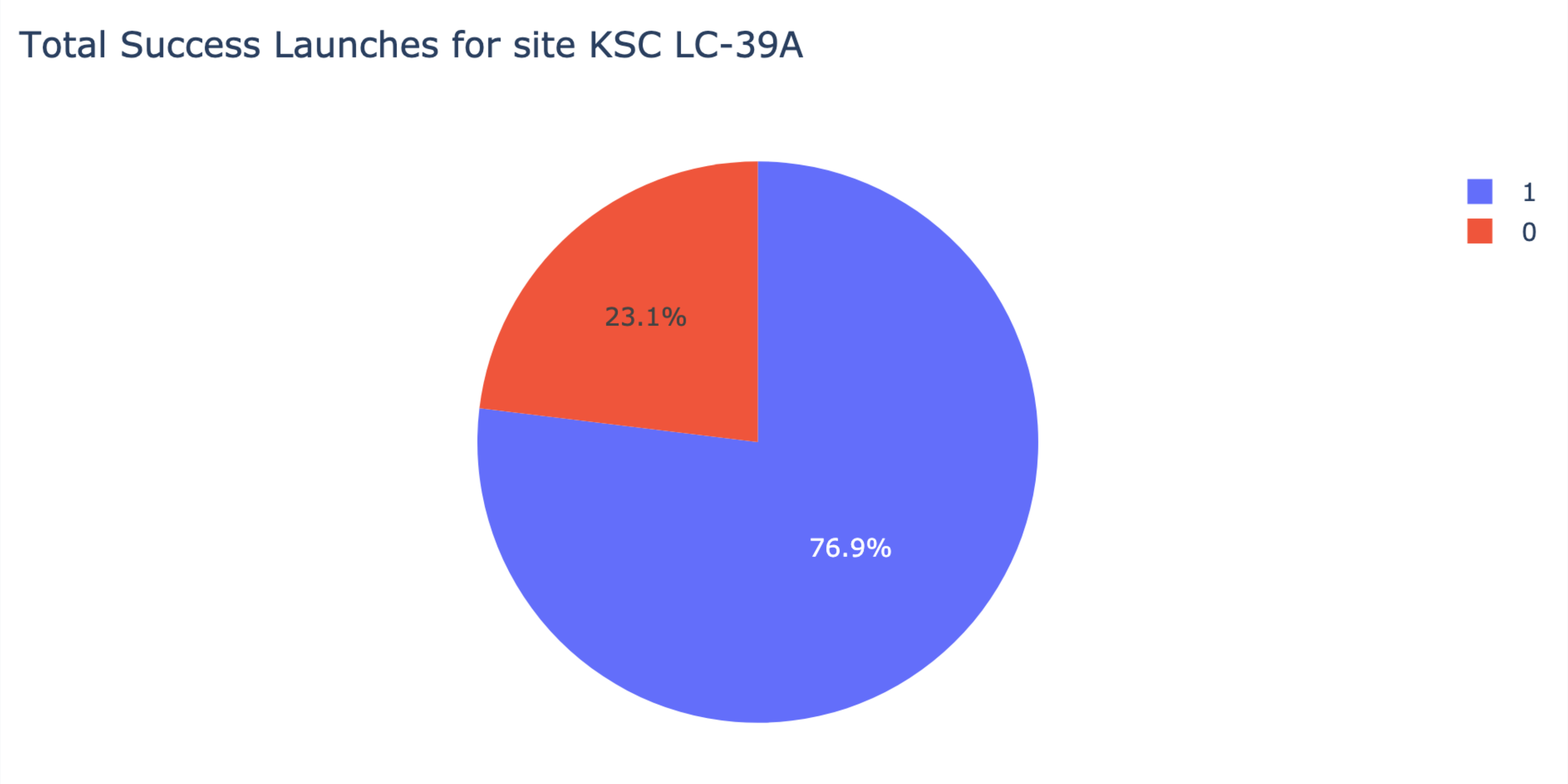
Build a Dashboard with Plotly Dash

Launch Success Count for All Sites



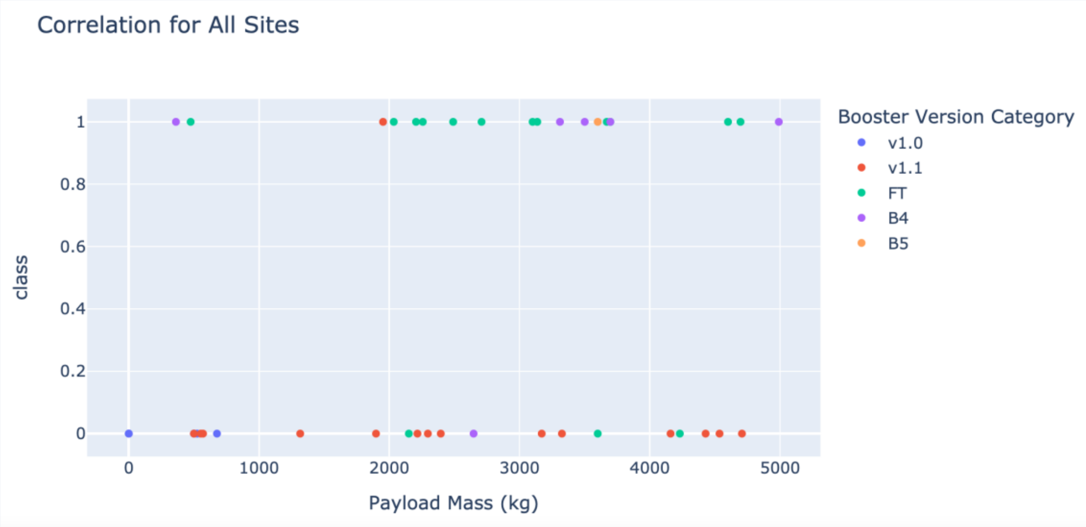
KSC LC-39A had the most successful launches from all sites

Launch Site with highest success ratio

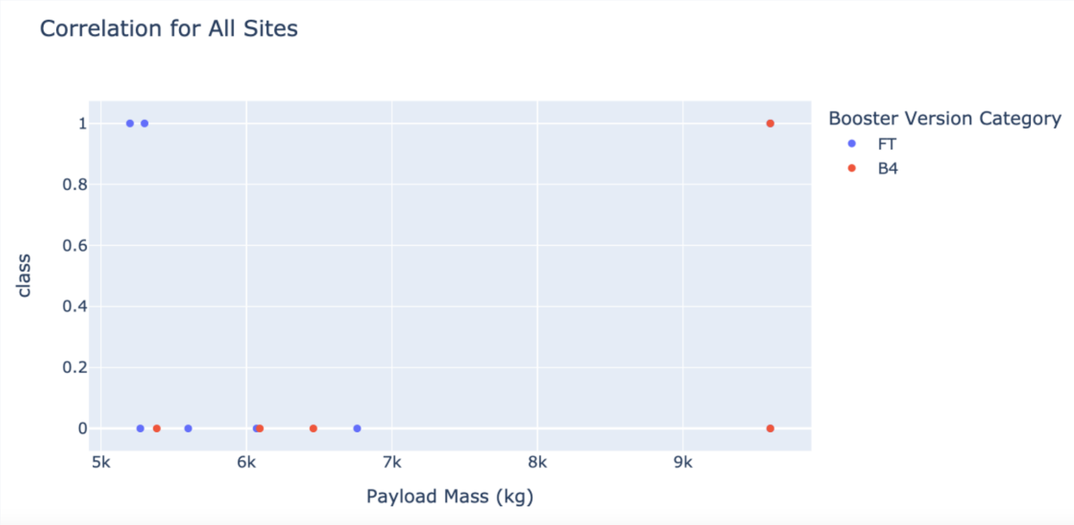


KSC LC-39A achieved a 76.9% success rate

Payload vs Outcome for All Sites



Lower Payloads



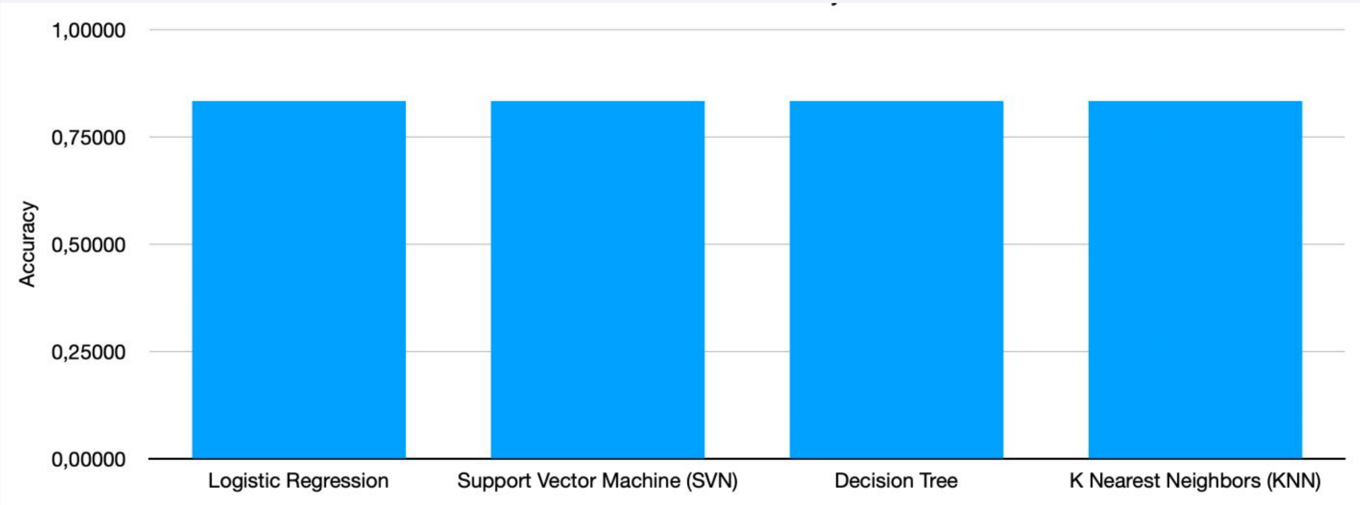
Higher Payloads

- The success rate is higher for lower payloads
- The higher success rate is for payloads **between 3100 and 3700 kg**
- The Booster version with the higher success rate is "**FT**"

Section 6

Predictive Analysis (Classification)

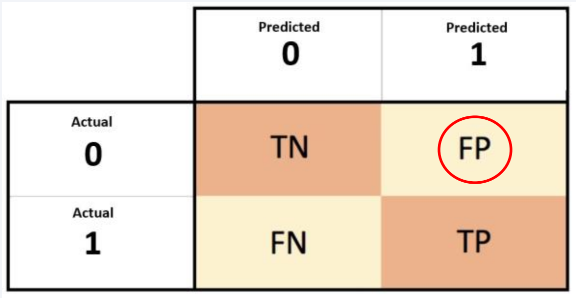
Classification Accuracy



	Accuracy		
Logistic Regression	0,83333		
Support Vector Machine (SVN)	0,83333		
Decision Tree	0,83333		
K Nearest Neighbors (KNN)	0,83333		

- The performance of the 4 classification models seems to be the same
- Since we had only 18 test samples, it could be that using a bigger test data set we would obtain more reliable results

Confusion Matrix



- Examining the confusion matrix, we see that all classification models can distinguish between the different classes
- We see that the major problem is **false positives**

Conclusions

Orbits ES-L1, GEO, HEO and SSO have the highest success rate

Since 2013 the success rate kept increasing over the years

The Launch Site with the highest success rate (76,9%) is "**KSC LC-39A**"

Launch sites are always in proximity of coastlines, but far from cities, highway and railways

The success rate is higher for lower payloads, especially between 3100 and 3700 kg

The Booster version with the higher success rate is "**FT**"

All classification models that were used have the same accuracy (83,3%)

Appendix

Appendix

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None
...	...	...	...	...	...	...	...	...	...	...	...	...
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc
93	90	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca

90 rows × 17 columns

Dataframe created collecting data with SpaceX API

Appendix

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

In [14]:
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = [0 if x in bad_outcomes else 1 for x in df['Outcome']]
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

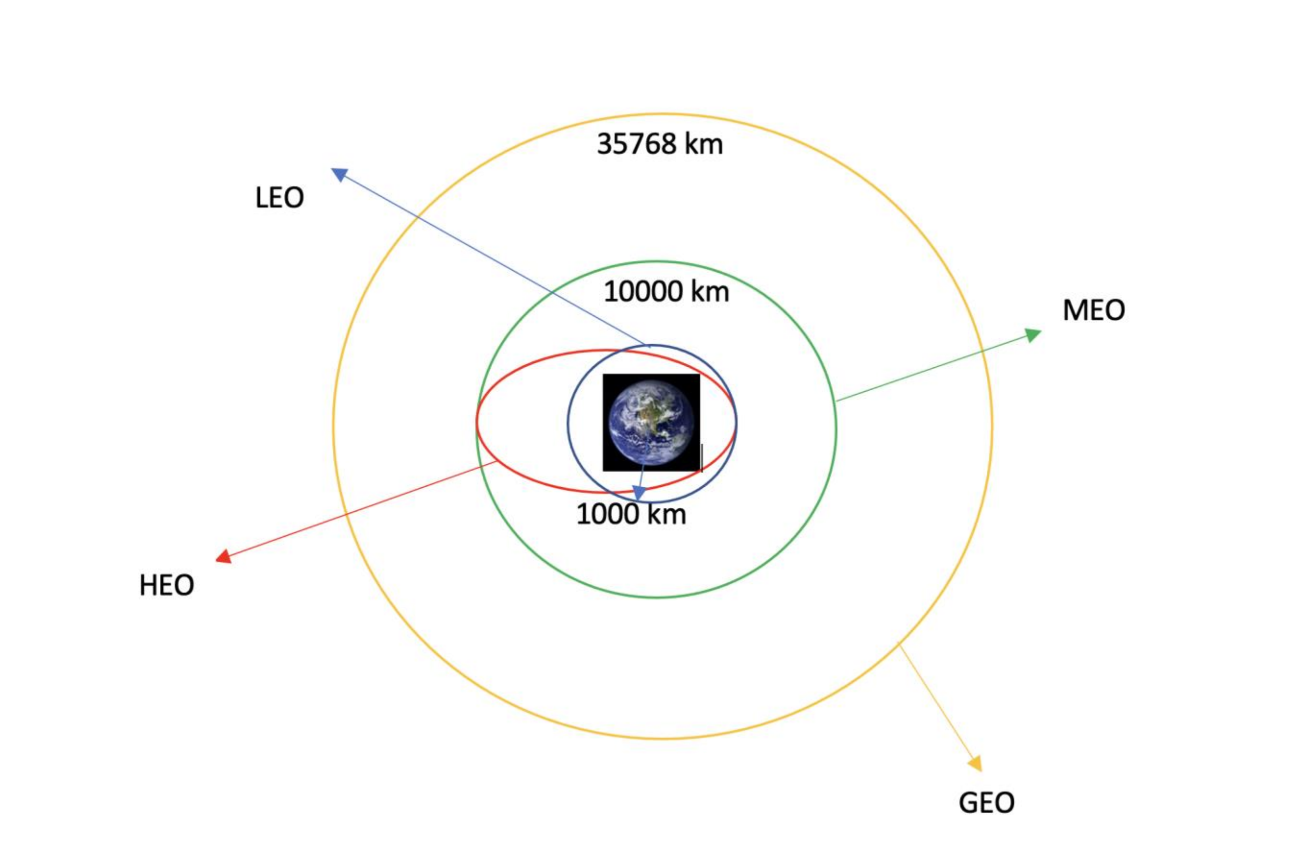
In [15]:
```
df['Class']=landing_class
df[['Class']].head(8)
```

Out[15]:

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

Turning Categorical variables into numeric values with One Hot Encoding

Appendix



Different Orbits for SpaceX launches

Appendix

Task 6

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

In [12]:
```
%%sql

SELECT booster_version, *
FROM SPACEXTBL
WHERE landing__outcome = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

```
Done.
```

Out[12]:

booster_version	DATE	time__utc_	booster_version_1	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__ou
F9 FT B1022	2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drc ship)
F9 FT B1026	2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drc ship)
F9 FT B1021.2	2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drc ship)
F9 FT B1031.2	2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drc ship)

Using a SQL query to discover the most successful Booster version within a certain payload range

Thank you!