Jonathan Chong
14th January 2022

Outline

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

Executive Summary

- Summary of methodologies
- Data Collection
- Data wrangling
- EDA with Data Visualisation
- EDA with SQL
- Built an interactive map with Folium
- Built a dashboard with Plotly Dash
- Predictive Analysis (ML Classification model)

- Summary of all results
- Exploratory analysis results
- Screenshots of interactive analytics demo
- Predictive analysis results
- Conclusion

Introduction

Project background and context:

- We conducted this analysis to predict whether Falcon 9 first stages will land successfully. If we can predict accurately whether the first stage will land, we can more accurately determine the cost of a launch. This data analysis can be used by a new company which wants to compete with SpaceX.

Problems we want to find answers to:

- What are the variables which influence successful landings?
- If we know what influences successful landings, what are the optimum conditions we must have to achieve a successful landing?

Section 1

Methodology

Methodology

- Executive Summary
- Data collection methodology:
- Data was collected through SpaceX API and webscraping from SpaceX Wikipedia page
- Performed data wrangling
- Data was processed and transformed through one hot encoding and dropping irrelevant columns
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models
- Built, tuned, evaluated classification models

Data Collection– SpaceX API

1. Get response from API

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

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

2. Convert to .json file

```
# Use json_normalize meethod to convert the json result into a dataframe
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Apply custom functions to clean data

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

4. Assign list to dictionary then dataframe

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
# Create a data from launch_dict
df = pd.DataFrame.from_dict(launch_dict)
```

5. Filter and save to .csv

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

[Github link for Data Collection]()

Data Collection- Scraping

[Github link to Webscraping notebook]

1. Get response from HTML

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

2. Create BeautifulSoup Object

```
# Use BeautifulSoup() to create a BeautifulSoup obj
soup = BeautifulSoup(response.text, 'html.parser')
```

3. Find Tables

```
# Assign the result to a list called
html_tables = soup.find_all('table')
```

4. Get column names

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

5. Create Dictionary

```
launch_dict= dict.fromkeys(column_names)

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

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

6. Append data to Keys (too long so code in link)

7. Convert dictionary to Dataframe

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

8. Convert dataframe to .CSV

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

Data Wrangling

- We cleaned up data so only information that we needed to predict a successful launch of Falcon 9 remained.
- We identified the most common orbit types using the ‘value_count’ command. We found the most common were GTO, ISS and VLEO respectively.
- We identified ‘bad_outcomes’ in the ‘landing_outcomes’ column. The results in this column were a bit convoluted with different types of bad outcomes so we sought to create a simpler variable for outcome.
- We created a simpler ‘landing_class’ column which displayed ’1’ for successful launches and ‘0’ for failures. (This later becomes the new landing_outcome column in future notebooks in this project).

- Github Data Wrangling Notebook

EDA with Data Visualization

Summary of charts Used

Scatter plots:

- Flight Number vs Payload Mass
- Flight Number vs Launch Site
- Payload vs Launch Site
- Orbit vs Flight Number
- Orbit vs Payload
- Orbit vs PayloadMass

Box plots:

- Mean success rate vs Orbit

Line graph:

- Success rate vs Year

[Github Data Visualization Notebook]

EDA with SQL

- 10 questions were asked which we found information about

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 EDA SQL Notebook]

Build an Interactive Map with Folium

- Took latitude and longitude coordinates of launch sites.
- Added a circle marker and label to each launch site.
- Added Green and Red markers (using classes 0 and 1) to the map using markercluster()
- Added lines to various landmarks which also displayed distance between the launch site and landmark (used Haversine distance formula).

Github Interactive Visual Folium Notebook

Build a Dashboard withPlotly Dash

- We built a dashboard withPlotly Dash using the given SN labs virtual environment where we can launch applications
- Pie chart graph:
    - Showsthe percentage of successful launch outcomes for all the sites, and each of the sites individually
- Scatter graph:
    - Shows therelationship between successful launch outcome and payload mass (kg) for the different Booster Versions
    - Minimum and maximum payload mass can be changed using slider
    - Both charts provide easy observation for those looking at charts

[Github Data dashboard]

Predictive Analysis (Classification)

Stage 1 – Building:

- Transformed data, split data into train and test sets, set parameters for KNN, Decision Tree, and Logistic Regression algorithms, and used GridSearchCV.

Stage 2 – Evaluation:

- Hyperparameters for each model, accuracy score, and plotted confusion matrix

Stage 3 – Improve:

- Feature engineering and algorithm tuning

Stage 4 – Find best performing:

- Compared accuracy scores of different algorithm models

• Github Predictive Analysis Notebook

Results

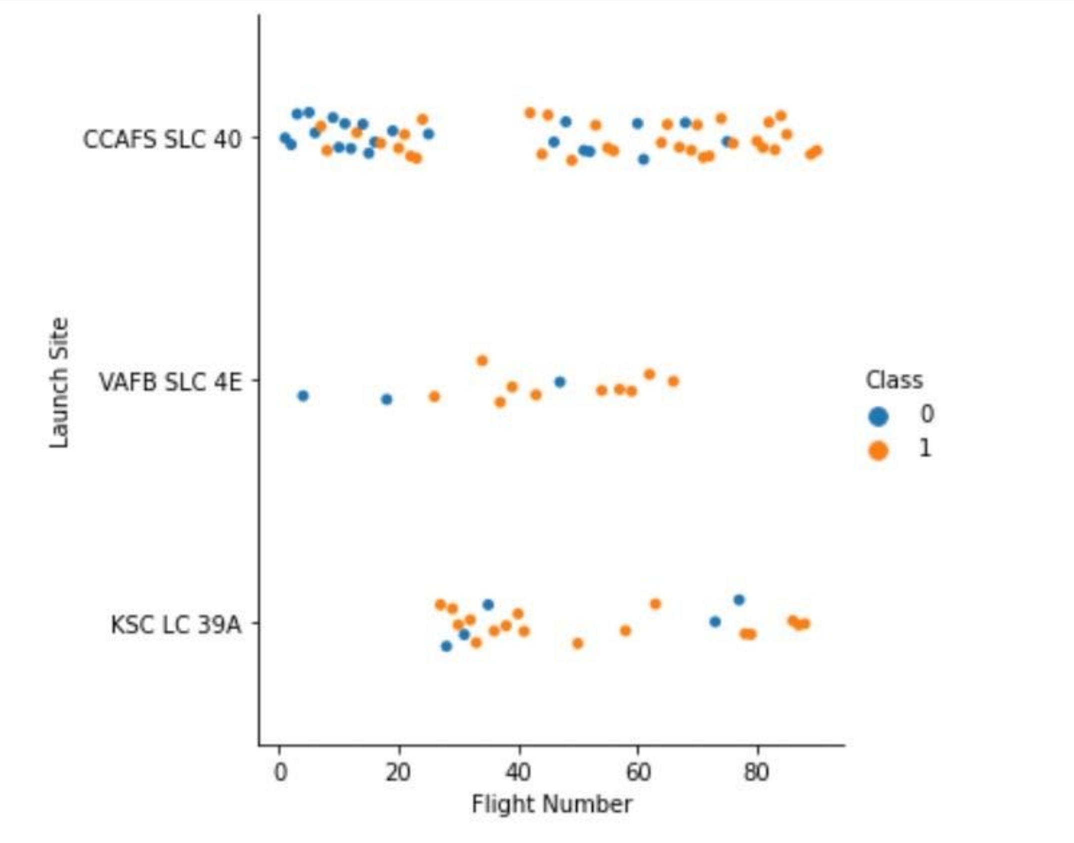
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

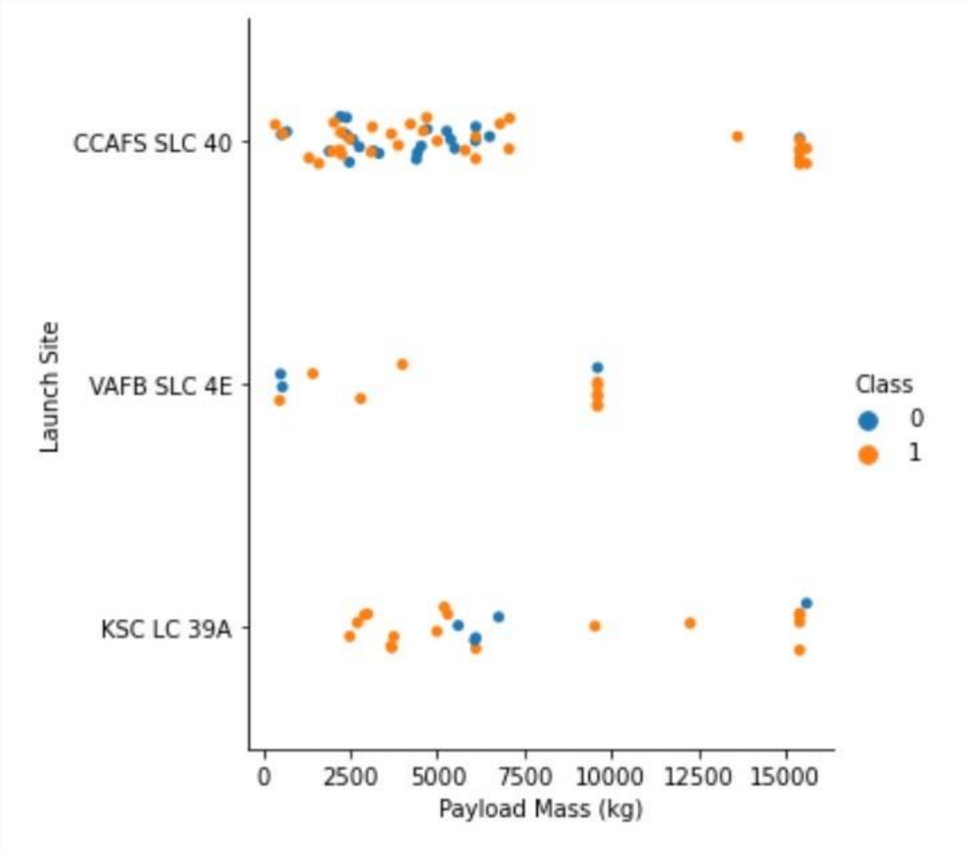
Flight Number vs. Launch Site

- Orange dots are successful landings, and blue dots are unsuccessful landings.
- As the flight number increases, the likelihood of a successful flight has increased.
- CCAFS has the most successful launches, but also the greatest number of unsuccessful launches
- There is no single launch site which is necessarily better than the others.



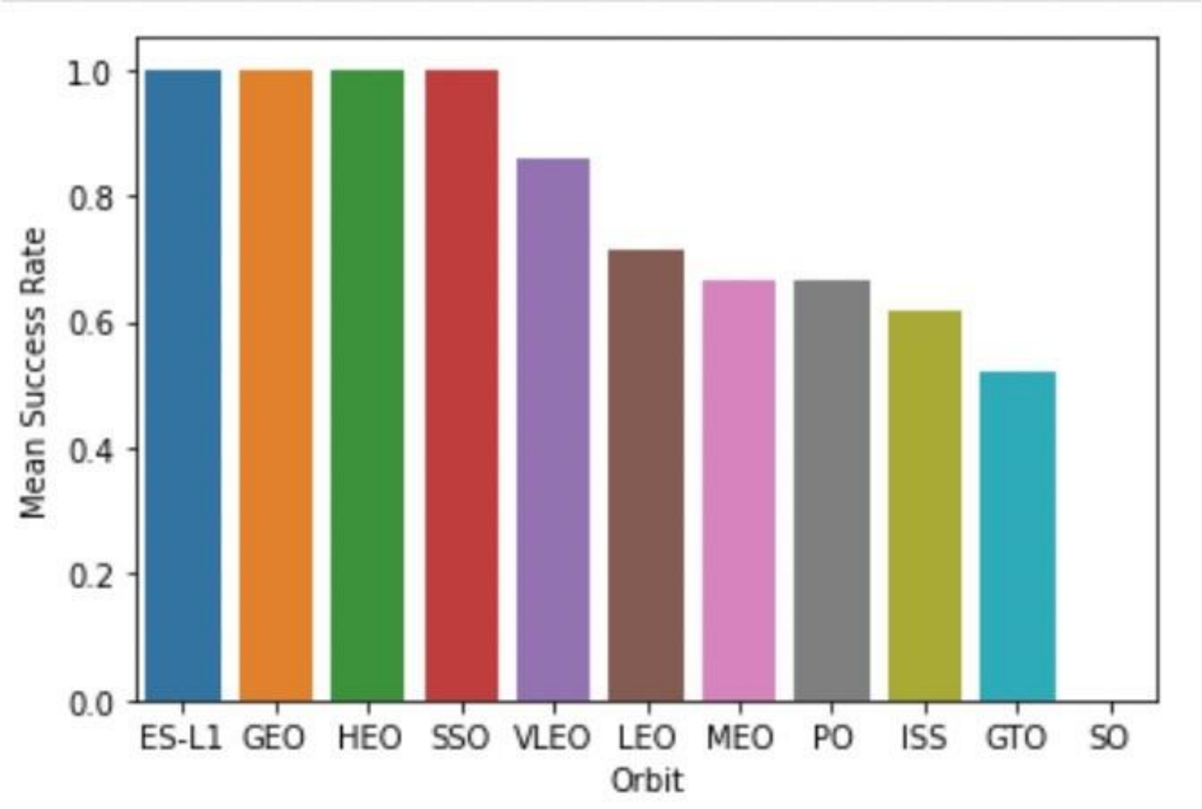
Payload vs. Launch Site

- Successful launches are made likelier by heavier payloads. This is the case across all e launches.



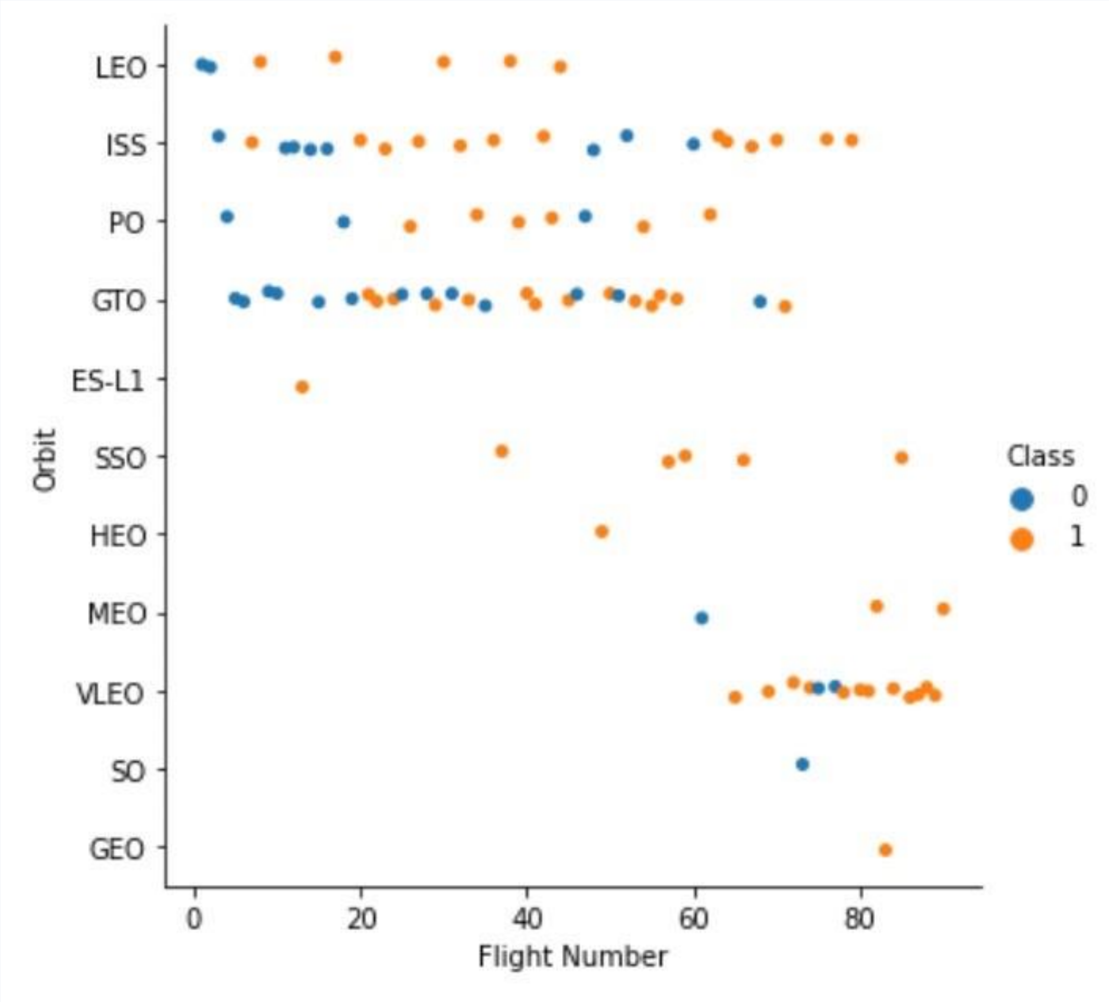
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO are the most successful Orbit types



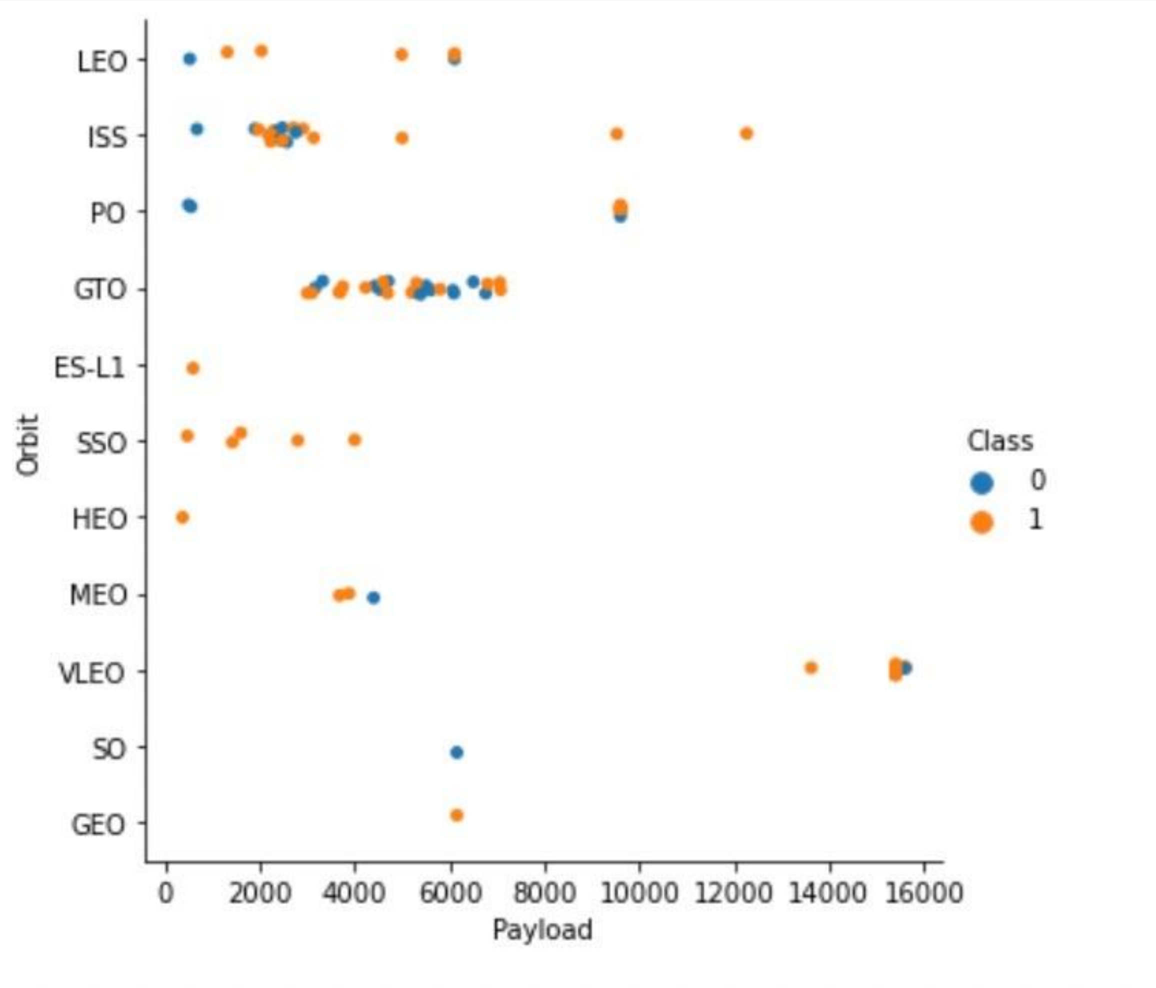
Flight Number vs. Orbit Type

- Many of the early launches at LEO, ISS, PO, and GTO were unsuccessful. These were launch sites of the early flights when SpaceX had less experience.
- VLEO orbit has the most successful launches, mainly due to the fact that it has only been used in more recent flights.
- There has never been an unsuccessful launch with SSO orbit.



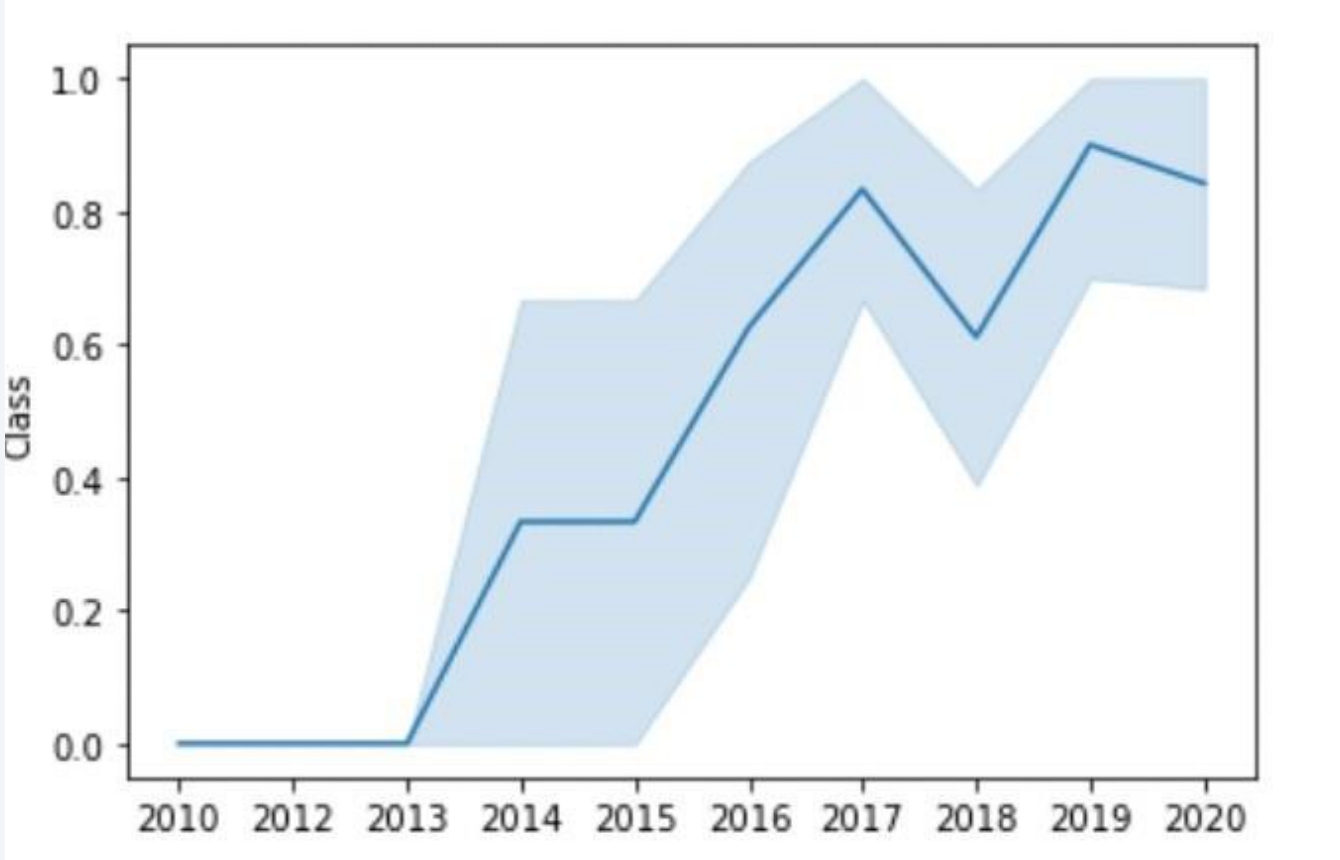
Payload vs. OrbitType

- Heavier payloads result in more success for LEO and ISS orbits.
- GTO orbit does not necessarily benefit from heavier payload.



Launch Success Yearly Trend

- The trend of the launch rate has steadily increased between 2013 and 2019.
- Therewas a large dip in successrate between 2017 and 2018.
- Therewas a small dip in success rate between 2019 and 2020, but the success rate remained high



All Launch Site Names

Display the names of the unique launch sites in the space

```
%sql select distinct launch_site from SPACEX
```

```
 * ibm_db_sa://tzh14394:***@0c77d6f2-5da9-48a9-81f
Done.
```

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

The command ’Distinct’ shows only unique values from the ‘launch_site’ column.

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEX where launch_site like 'CCA%' limit 5
```

```
 * ibm_db_sa://tzh14394:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

DATE	Time (UTC)	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	Landing _Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	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

- Command ‘limit 5’ shows only top 5 results
- Command ‘like ‘CCA%’’ shows only results from launch_site column which start with CCA.

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEX where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://tzh14394:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqbl
Done.
```

1
45596

Command ‘sum’ gives us the total payload mass which is displayed as 45596(kg).

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEX where booster_version like 'F9 v1.1%'
```

```
 * ibm_db_sa://tzh14394:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.dat
Done.
```

1
2534

This query gives us the average payload mass in KG for booster version F9 v1.1

The result is 2534 kg.

First Successful Ground Landing Date

```
%sql select min(DATE) DATE from SPACEXTBL where Landing_OUTCOME = 'Success (drone ship)'

 * ibm_db_sa://tzh14394:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.da
Done.
```

DATE
2016-04-08

- The first successful landing was on 2016-04-08
- The 'min' function gives us the earliest date there was a successful landing

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select booster_version from SPACEXTBL where LANDING_OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

```
 * ibm_db_sa://tzh14394:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

]:

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

- Selected booster versions which resulted in successful drone ship landings
- The ’and’ command helps us filter results which are between 4000 and 6000.
- Thereare 4 individual booster versions which fulfil these conditions

Total Number of Successful and Failure Mission Outcomes

```
%sql select mission_outcome, count(*) from SPACEX group by mission_outcome
```

```
 * ibm_db_sa://tzh14394:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90
Done.
```

]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

In total there were 100 successful mission outcomes and only 1 failure.

Boosters Carried Maximum Payload

```
%sql select booster_version, payload_mass__kg_ from SPACEX where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEX)
```

```
 * ibm_db_sa://tzh14394:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

]:

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

Use of sub-query to only display booster versions which carried the maximum payload mass. The maximum payload mass is 15600kg.

2015 Launch Records

```
%sql select Landing_Outcome, booster_version, launch_site from SPACEXTBL where date like '2015%' and landing_outcome like 'Failure (drone ship)'
```

```
 * ibm_db_sa://tzh14394:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

9]:

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

Question 9. asked to list the failedlanding outcomes in drone ship, their booster versions, andlaunch site names in year 2015

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

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 ¶

```
%sql select * from SPACEXTBL where Landing_Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```

```
 * ibm_db_sa://tzh14394:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

]:

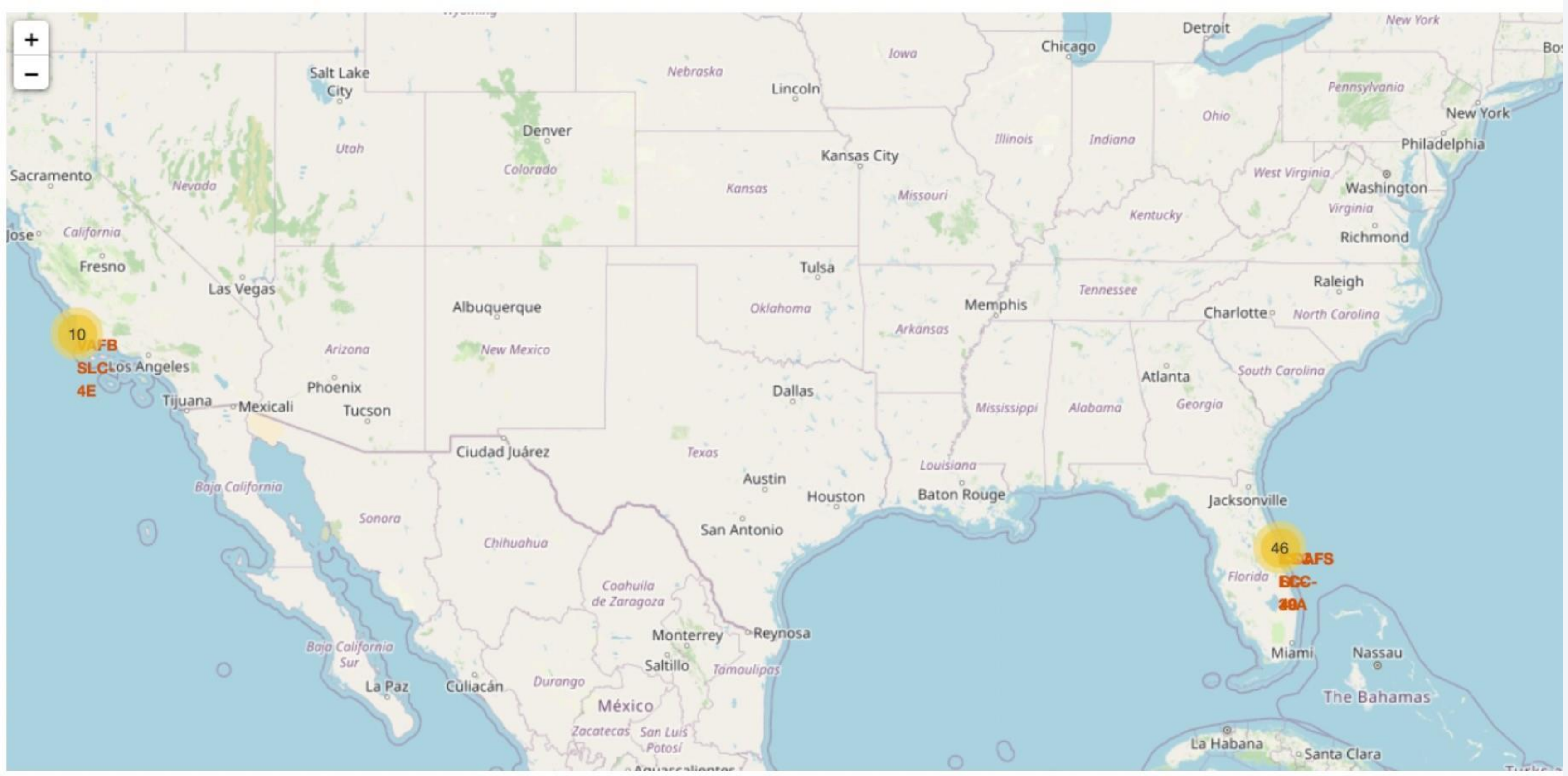
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing_outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Ranked the count of landing outcomes in descending order between the date 2010-06-04 and 2017-03-20.

Section 4

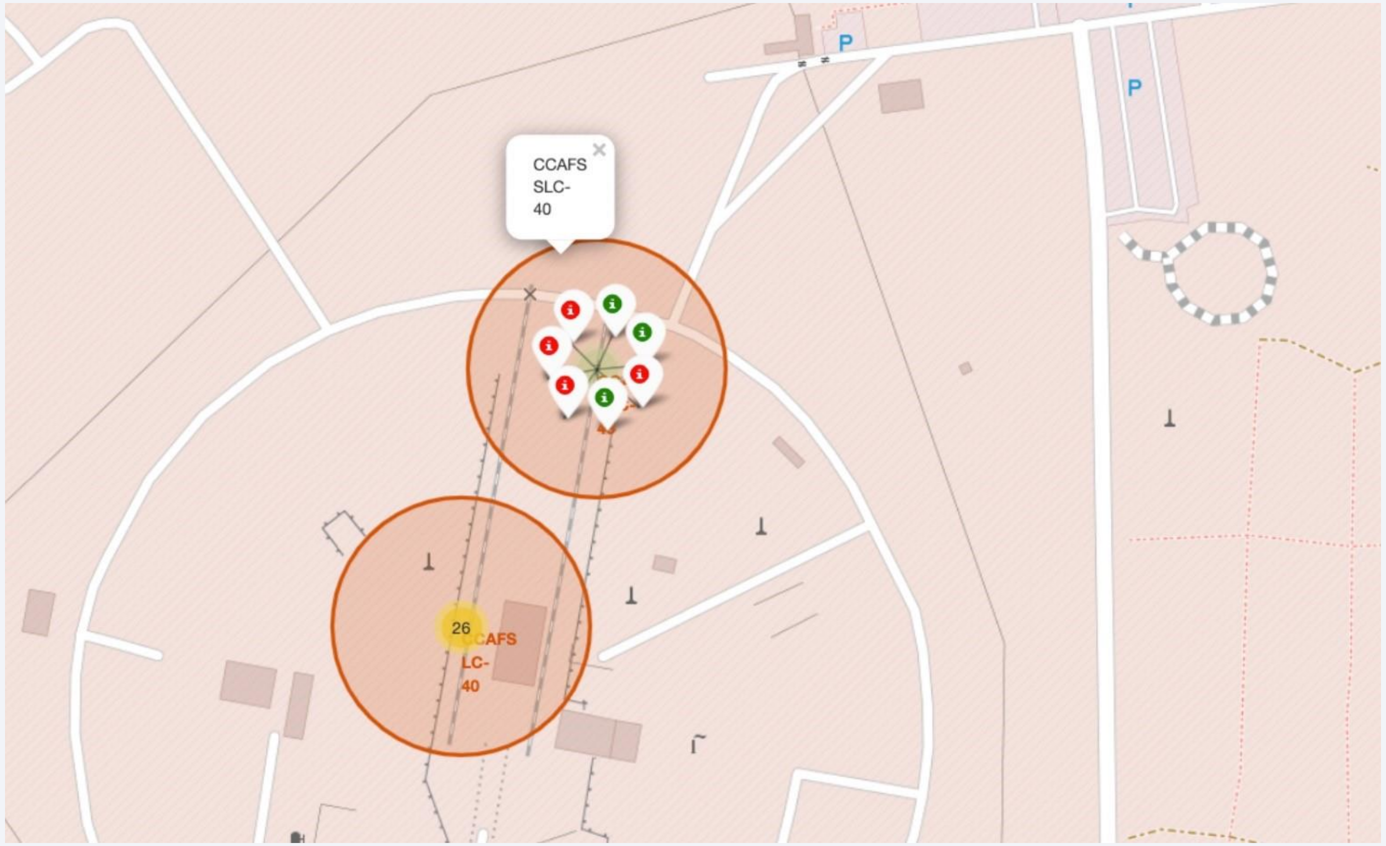
Launch Sites Proximities Analysis

Map of both launch locations for SpaceX



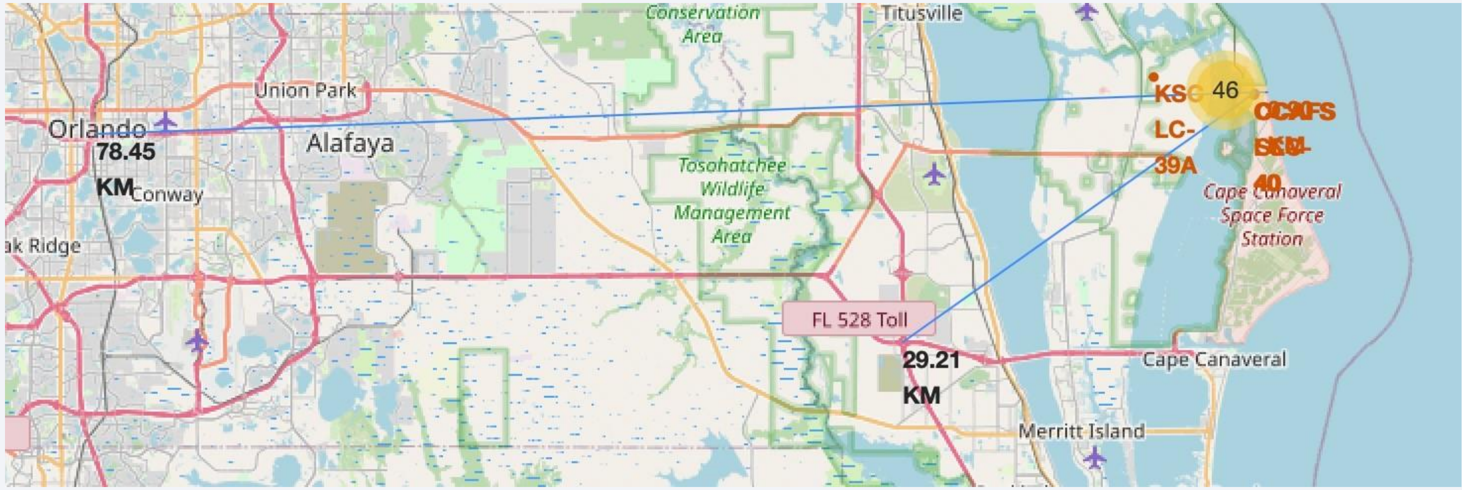
SpaceX has two rocket launch sites both located in the USA, one in California, and one in Florida

Color labelled markers



- Successful launches are colored green, whereas failures are red.
- Markers are placed in circle clusters depending on the launch site. Click on clusters to see markers.

Launch site proximity to landmarks

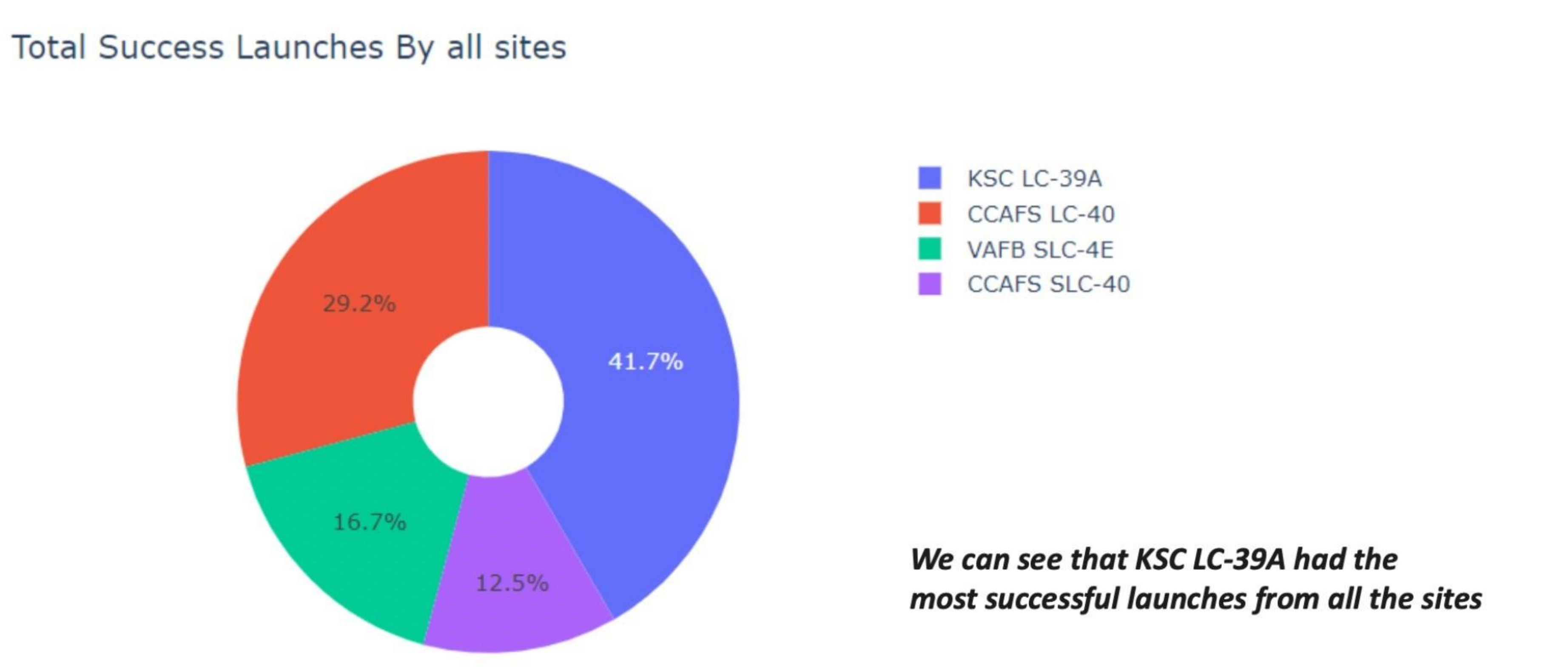


- The closest launch site to a coastline is only 0.9km away.
- The closest city to this launch site is Orlando which is 78.45 km away and the closest motorway is the FL 528 Toll which is 29.21km away

Section 5

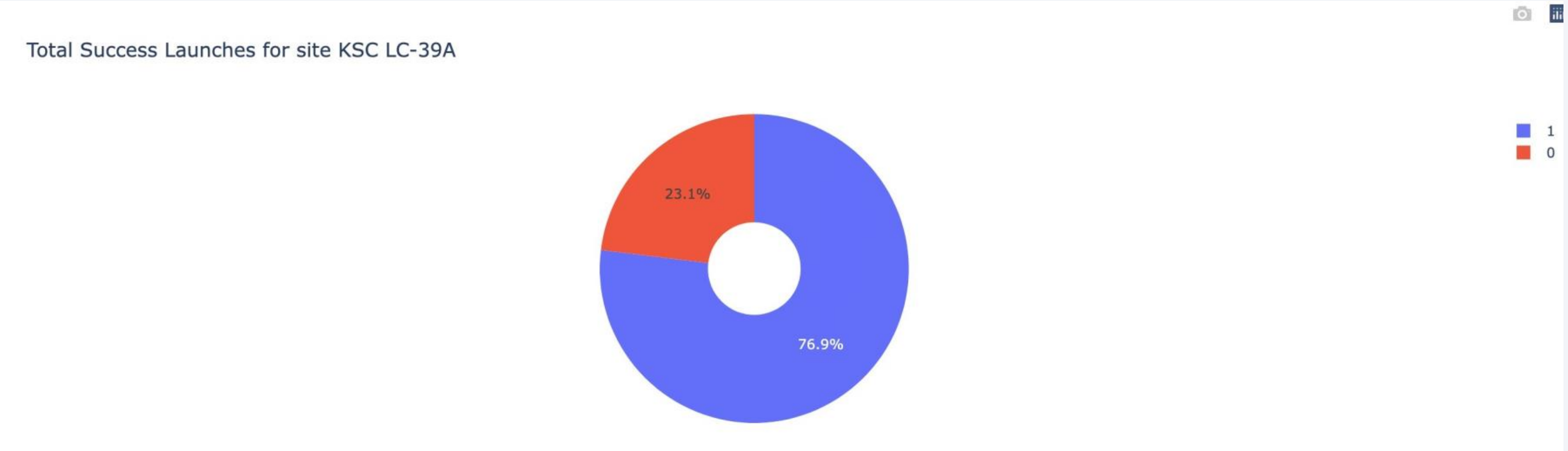
Build a Dashboard with Plotly Dash

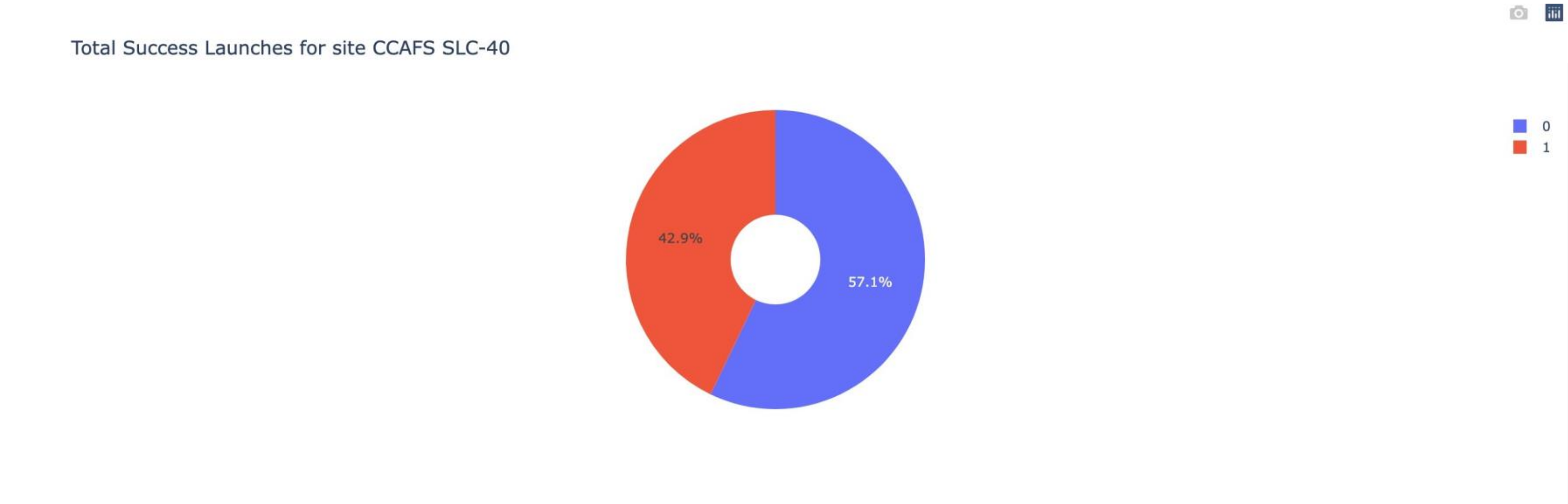
Dashboard showing success percentage of each site



- KSC LC-39A was the most successful launch site with 41.7% of all successful launches.
- CCAFS SLC-40 was the least successful with only 12.5% of all successful launches.

Further review of KSC LC-39A and CCAFS SLC-40





76.9% of launches were successful at the best performing launch site KSC LC-39A compared to only 42.9% at the worst performing launch site CCAFS SLC-40.

Payloadmasshas an effect on launch outcome

- There was a higher percentage of successful launches at payload masses below 4000kg than between 4000kg and 10000kg.
- There were no successful launches above 5500kg.

Section 6

Predictive Analysis (Classification)

Classification Accuracy

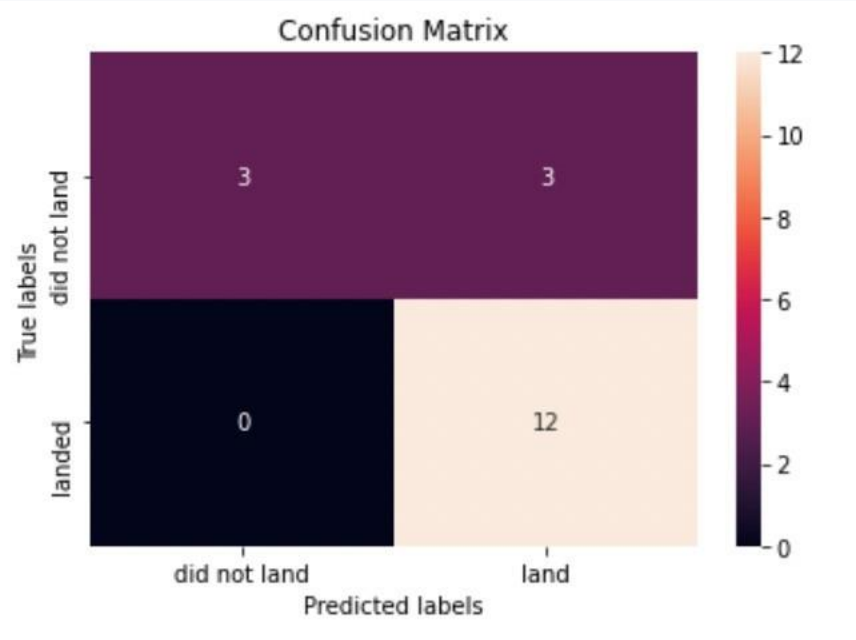
- The bar chart shows the accuracy of different classification algorithms with model type on the x axis and accuracy on the y axis.

- Decision Tree classifier model is the model which hasthe highest classificationaccuracy:

```
Best Algorithm is Tree with a score of 0.9017857142857144
Best Params is : {'criterion': 'gini', 'max_depth': 8, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'best'}
```

Confusion Matrix

- This is the confusion matrix for our decision tree model. The main problem the predictive model has is false positives. There are no false negatives.



Conclusions

– The decision tree classifier provides the best model for us to predict successful landings

– The likelihood of a successful landing is increasing as SpaceX continues to launch more flights

– KSC LC-39A is the most successful launch site, with 71.9% of launches successful

– Orbit types GEO, HEO, SSO and ES-L1 have the best success rates.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!