IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

L. Soroko
04 April 2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis
  - Interactive Visual Analytics with a Map
  - Interactive Dashboard
  - Predictive Analysis
- Summary of all results
  - Insights drawn from EDA
  - Interactive analysis with screenshots
  - Results from predictive analysis

Introduction

- Project background and context

  Space Y would like to compete with SpaceX in launching rockets into space. First step into reaching this goal is to determine whether the first stage of the rocket will be reusable or not, depending on the mission parameters. This will determine much of the price of each launch.

  The aim of this project is to gather and analyse public information to build and train a machine learning model in order to predict whether SpaceX will reuse the first stage of the rocket.

- Problems I want to find answers to:
  - Which mission parameters are responsible for success/failure?
  - How to optimize the success rate of recovering the first stage of the rocket?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was obtained from public sources through the SpaceX Rest API as well as through web scraping from the Falcon 9 and Falcon Heavy Launches Records from Wikipedia
- Perform data wrangling
  - Exploratory Data Analysis was performed and training labels were determined where 1 means the booster was successfully retrieved and 0 means unsuccessful.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Various models were created, tuned and tested for performance using cross-validation

Data Collection – SpaceX API

- During this step I collected the data by making a request to the SpaceX API, normalized the JSON response and parsed it into a Data Frame.
  The IDs in this Data Frame were exchanged by readable data and the columns were combined into a Dictionary. After removing Falcon1 data and replacing the missing payload mass data with the mean payload mass, the Data Frame was saved into 'dataset_part_1.csv' for future use.
- GitHub URL of the completed [SpaceX API]() calls notebook

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)

In [11]: # Use json_normalize meethod to convert the js
         pd.json_normalize(response.json())

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

In [23]: # Create a data from launch_dict
         df = pd.DataFrame.from_dict(launch_dict)

In [25]: # Hint data['BoosterVersion']!='Falcon 1'
         data_falcon9 = df[df['BoosterVersion']!='Falcon 1']

In [30]: data_falcon9['PayloadMass'].replace(np.nan, value=payloadmass_mean, inplace=True)
```

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

Data Collection – Web Scraping

- For this stage I extracted all Falcon launch records from the HTML table and parsed the table using BeautifulSoup. After extracting the column names I parsed the launch HTML tables and created a Data Frame from the data, which is saved as 'spacex_web_scraped.csv'
- GitHub URL of the completed Web Scraping Notebook

```
# use requests.get() method with the provided static_url
data = requests.get(static_url).text
# assign the response to a object
soup = BeautifulSoup(data, 'html5lib')
```

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

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
table_data = first_launch_table.find_all("th")
# table_data
for row in table_data:
    name = extract_column_from_header(row)
    #name = row.
    if(name != None and len(name) > 0):
        column_names.append(name)
```

```
In [13]:
# df=pd.DataFrame(launch_dict)
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

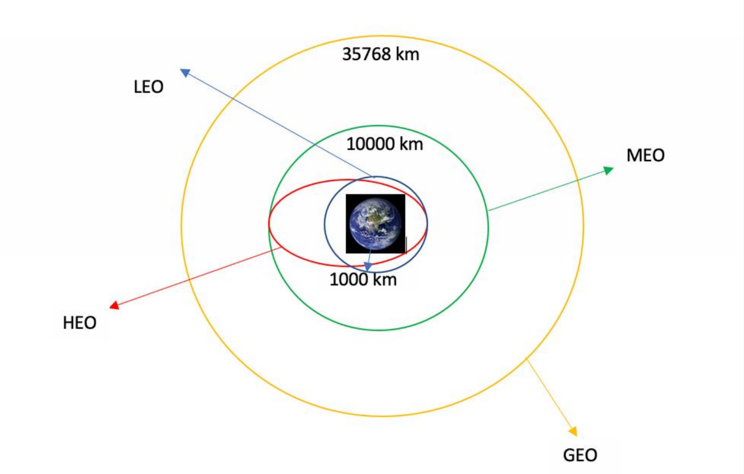
Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.1	Failure	4 June 2010	18:45
2	CCAFS	Dragon	0	LEO	NASA (COTS)\nNRO	Success	F9 v1.1	Failure	8 December 2010	15:43
3	CCAFS	Dragon	525 kg	LEO	NASA (COTS)	Success	F9 v1.1	No attempt\n	22 May 2012	07:44
4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA (CRS)	Success\n	F9 v1.1	No attempt	8 October 2012	00:35
5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA (CRS)	Success\n	F9 v1.1	No attempt\n	1 March 2013	15:10

Data Wrangling

- During the Data Wrangling process the following steps were taken:
  - Exploratory Data analysis
  - Calculation of number of launches at each site and the number of launches in each orbit
  - Creation of a landing outcome label and adding this column to the dataset
  - Exported the result to 'dataset_part_2.csv' for future use.
- GitHub URL of the completed [data wrangling notebook]

```
In [5]: # Apply value_counts() on column LaunchSite
        df[['LaunchSite']].value_counts()

Out[5]: LaunchSite
        CCAFS SLC 40    55
        KSC LC 39A      22
        VAFB SLC 4E     13
        dtype: int64
```

```
In [6]: # Apply value_counts on Orbit column
        df[['Orbit']].value_counts()

Out[6]: Orbit
        GTO      27
        ISS      21
        VLEO     14
        PO        9
        LEO       7
        SSO       5
        MEO       3
        ES-L1     1
        GEO       1
        HEO       1
        SO        1
```



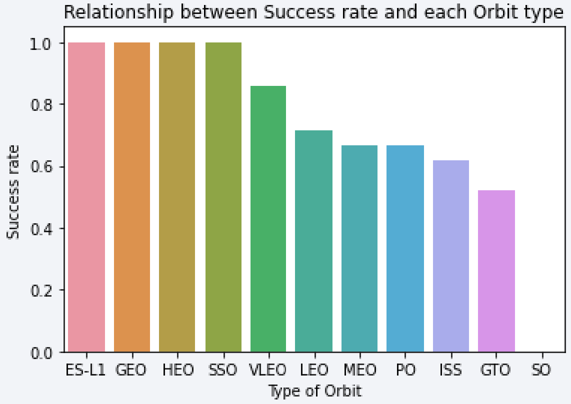

```
In [9]: bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
        bad_outcomes

Out[9]: {('False ASDS',),
         ('False Ocean',),
         ('False RTLS',),
         ('None ASDS',),
         ('None None',)}
```

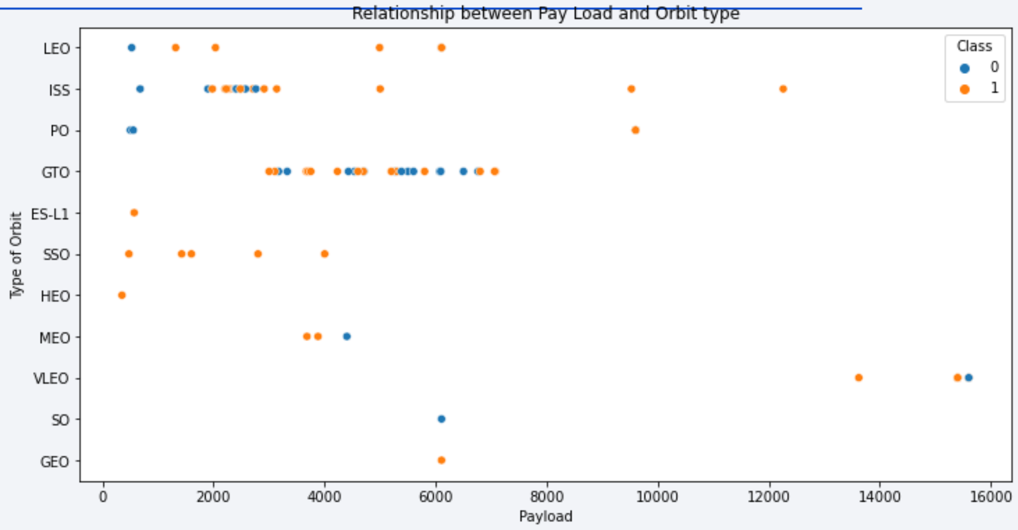
```
for x in df.index:
    if df['Outcome'][x] in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

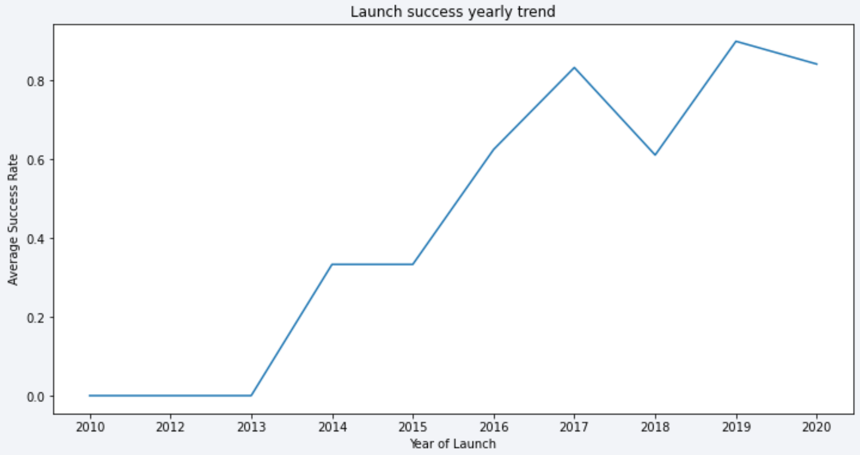
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 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

EDA with Data Visualization

- During the Exploratory Data Analysis with Data Visualization the following Graphs have been drawn:
  - Flight Number vs Payload
  - Flight Number vs Launch Site
  - Payload vs Launch Site
  - Success rate and Orbit type
  - Payload and Orbit type
  - Launch Success yearly trend







- Graphs can be found in Section 2 of this presentation
- During Features Engineering I applied OneHotEncoding for the columns Orbits, LaunchSite, LandingPad and Serial. I exported the result to ‘dataset_part_3.csv’ for future use.
- GitHub URL of the [EDA with data visualization notebook]

EDA with SQL

- I loaded the SpaceX dataset .CSV file into a DB2 Database on the IBM platform. After creating the database I performed several SQL queries in order to:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass.
  - List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 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

- Full queries and outcomes can be found in Section 2 of this presentation

- GitHub URL of the completed [EDA with SQL notebook]

Build an Interactive Map with Folium

- I built an Interactive Map with Folium to visualize the launch sites, and per launch site the successful launches and failures.
  - I used folium.Circle and folium.map.Marker for the launch sites.
  - For the success/failed launches I used a MarkerCluster per site and for each launch a white Marker was created with a green (success) or red (fail) icon.
- Furthermore I used the map draw lines to analyze the distances from launch sites to the nearest railway, highway, city and coastal line. From this info I was able to answer the following questions:
  - Are launch sites in close proximity to railways? Yes for duty railways (transport of material), No for commercial railways.
  - Are launch sites in close proximity to highways? No
  - Are launch sites in close proximity to coastline? Yes
  - Do launch sites keep certain distance away from cities? Yes
- GitHub URL of my completed [interactive map with Folium map]

Build a Dashboard with Plotly Dash

- The dashboard consists of a drop-down menu to select a launch site. Initial value is all launch sites.
- The second chart is a pie chart, which indicates:
  - The total of success launches per site (in percentage of all successful launches), or
  - The total of success launches for the selected site (success/fail rate)
- Below the pie chart is a slider which allows you to select a range of payloads which triggers the last chart
- The last chart is a scatter plot which shows success/failure per Booster Version within the payload range selected with the slider mentioned above.
- The dashboard will answer questions like ‘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? and which F9 Booster version has the highest launch success rate?
- An overview of some screenshots and the answers to the questions you will find in Section 4.
- You can find the source code of my completed Plotly Dash application (spacex_dash_app.py) at [this Github link].

Predictive Analysis (Classification)

- Data from previous steps was loaded into dataframe X (data) and NumPy array Y (target)
- Data in dataframe X was standardized and the data and target were split into a training set (80%) and a test set (20%).
- Various Machine Learning models were build (LogisticRegression, SupportVectorMachine, DecisionTreeClassifier and KNearestNeighborsClassifier), fitted with parameters from a dictionary and through GridSearchCV found the best parameters.
- For each model the accuracy was calculated and a confusion matrix was built.
- After comparison the best model was selected with the optimal values for the best parameters.
- GitHub URL of my completed predictive analysis lab notebook

Results

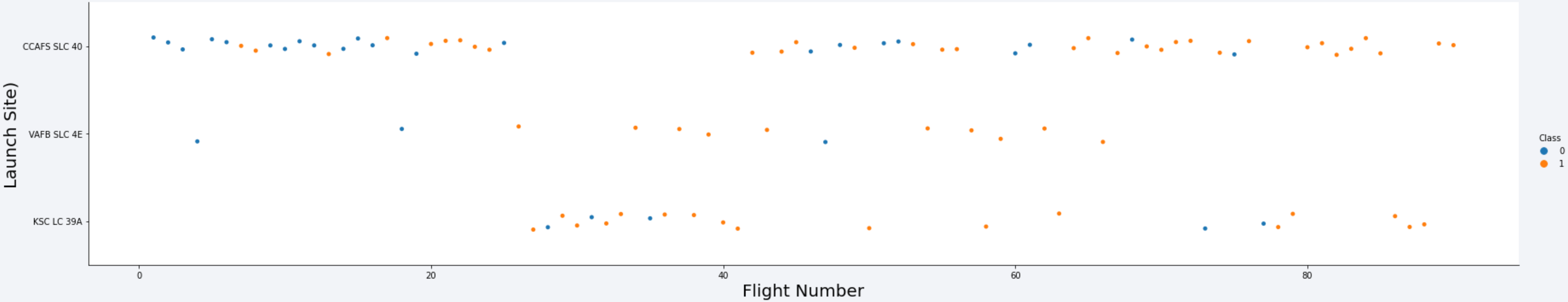
In the next sections the following results are presented:

- Section 2 - Exploratory data analysis results
- Section 3 - Interactive analytics demo in screenshots
- Section 4 – Screenshots from the Dashboard Application
- Section 5 - Predictive analysis results and Conclusions

Section 2

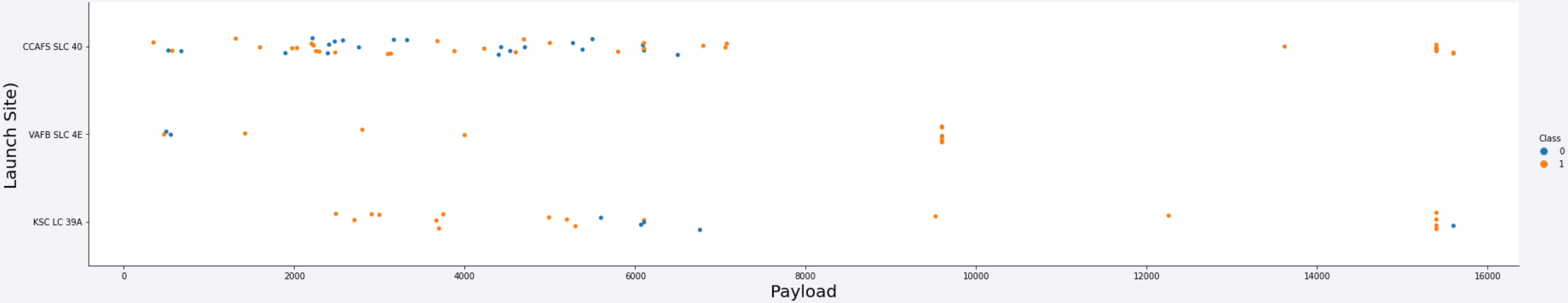
Insights drawn from EDA

Flight Number vs. Launch Site



- It is clearly visible that in the beginning (lower flight numbers) the success rate is low (CCAFS SLC 40 launch site).
- The last 20 launches from launch site CCAFS SLC 40 were mainly successful.
- Only a few flights were launched from VAFB SLC 4E, but they were almost all successful.

Payload vs. Launch Site

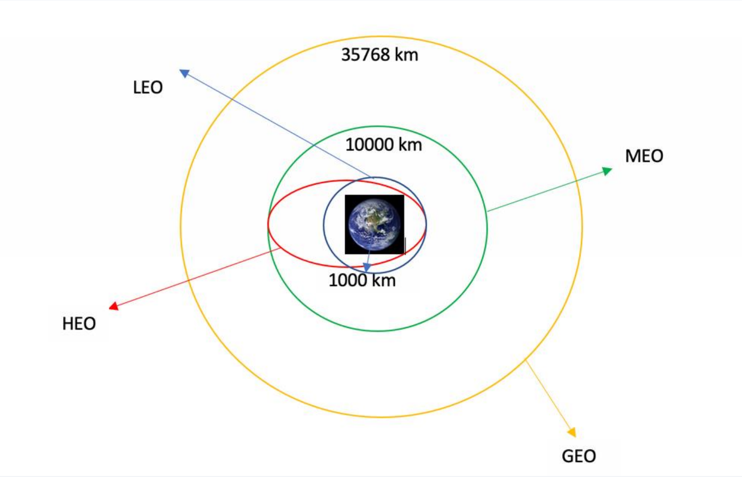


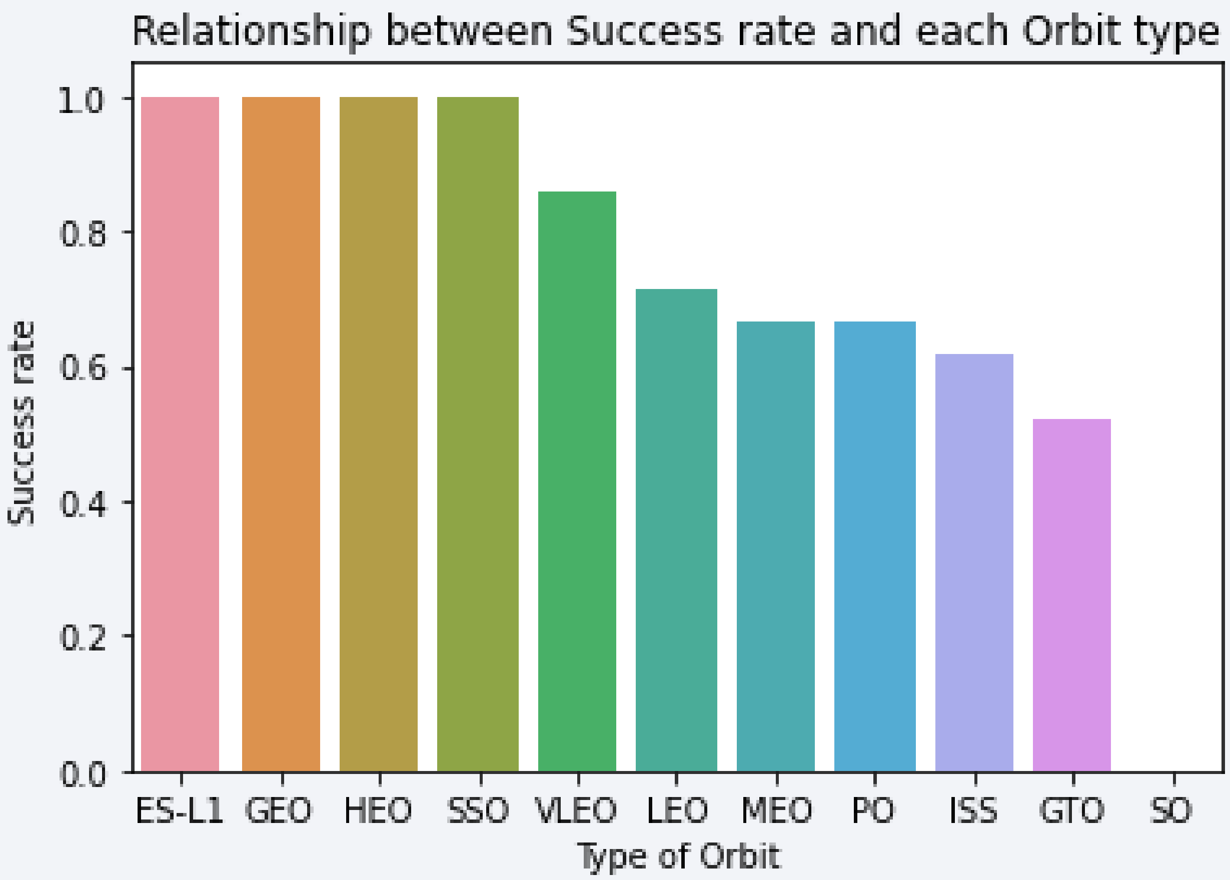
- In general we can say that the higher the payload, the more successful the launches are.
- Launches from VAFB SLC 4E were almost all successful AND no launch with a payload above 10000 kgs were made, therefore no pertinent conclusions can be drawn from this graph.

Success Rate vs. Orbit Type

- This bar chart shows us that launches into
  - ES-L1
  - GEO
  - HEO
  - SSO

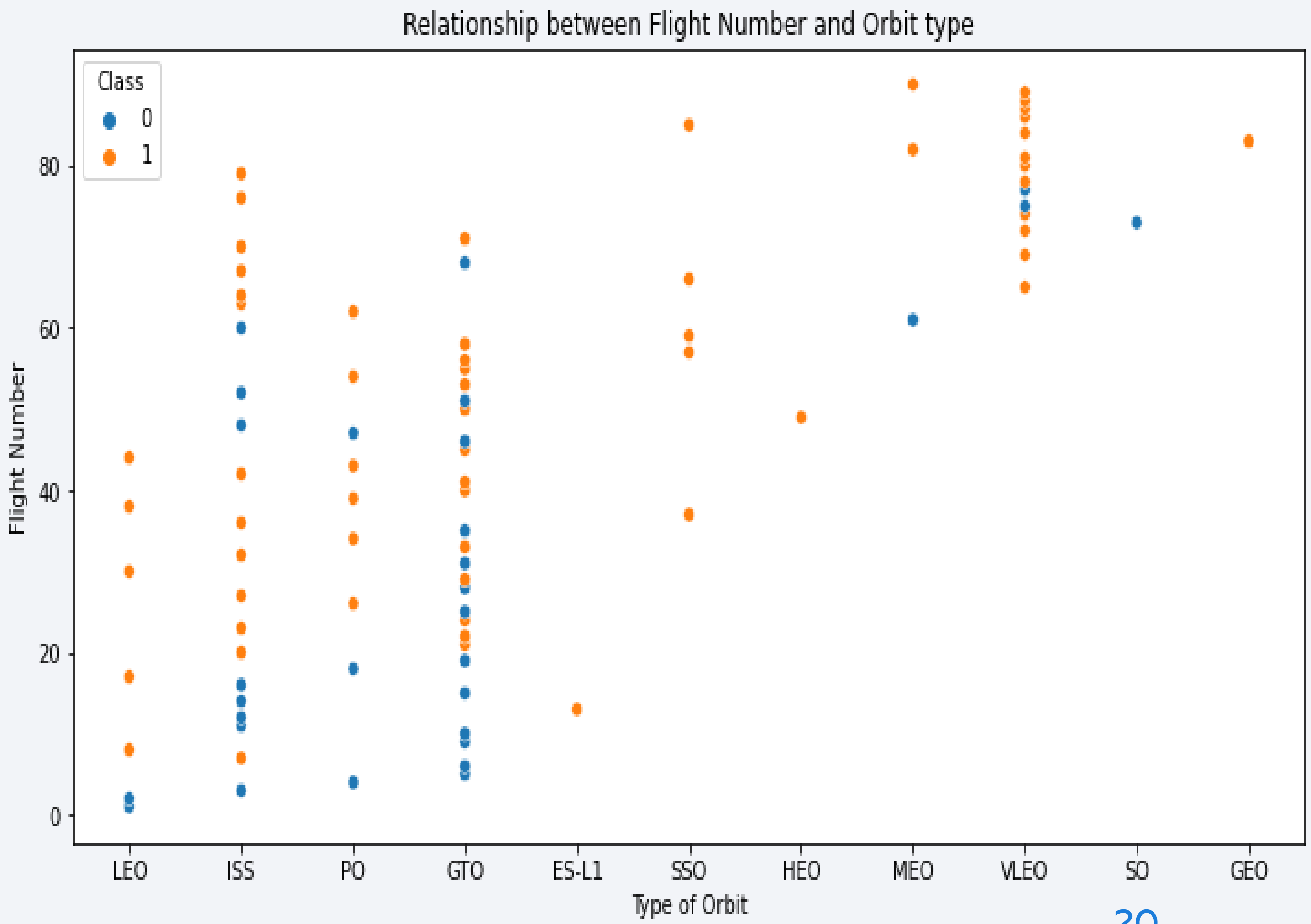
  have the best success rate





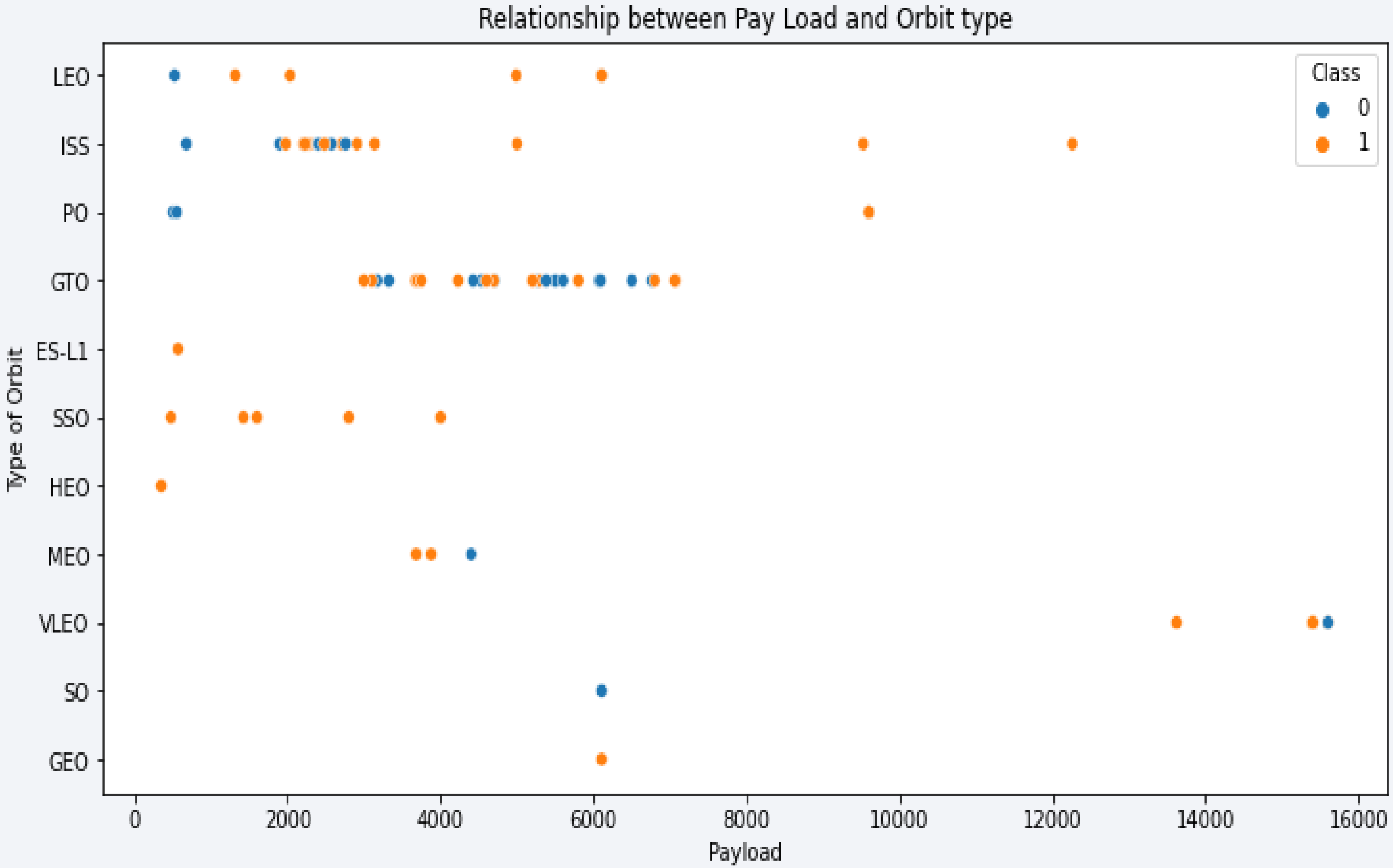
Flight Number vs. Orbit Type

- This graph is giving additional information to the previous graph.
- We saw that launches into ES-L1, GEO, HEO and SSO were highly successful, however here we see that there were only a few launches made.
- Only one launch in SO, unsuccessful
- Multiple successful launches into VLEO, the rest of orbits have a relative equal success rate



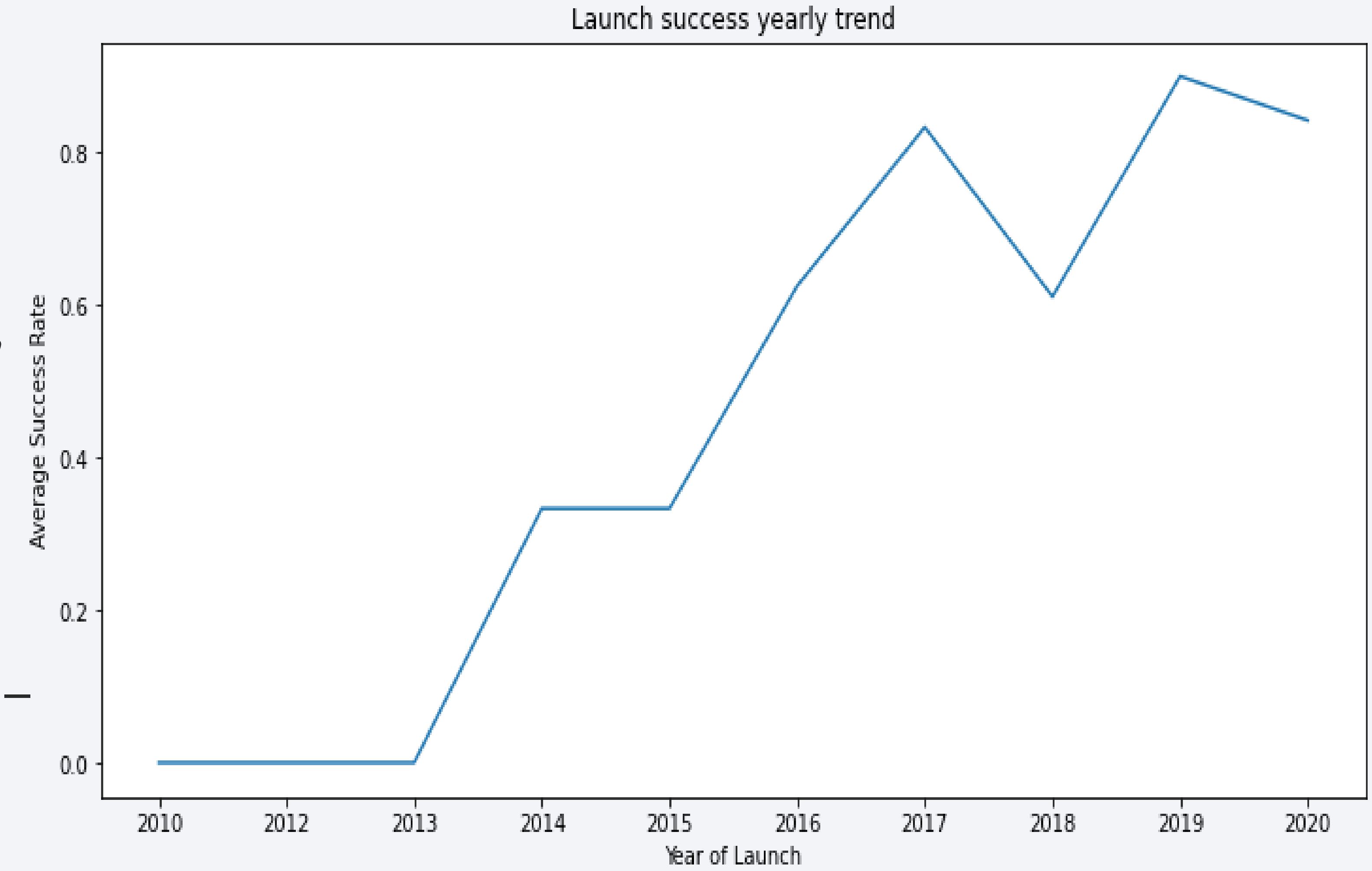
Payload vs. Orbit Type

With heavier payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.



Launch Success Yearly Trend

- As we can see the success rate of the launches increased every year from the start in 2013 till 2020, with a small dip in 2018.
- This dip can be explained due to the increased ‘no attempt to land’ the first stage (8x in 2018), which influences the success rate – Source: https://en.wikipedia.org/wiki/Falcon_9 - Booster landings graph.



All Launch Site Names

- Find the names of the unique launch sites
- By using SELECT DISTINCT only unique values will be displayed

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

In [6]:
```
%sql SELECT DISTINCT launch_site FROM SPACEXTBL
```

```
 * ibm_db_sa://tfp70922:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[6]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Using LIKE 'CCA' selects the records, LIMIT 5 to show only first 5 records

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

```
In [7]: %sql SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://tfp70922:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

Out[7]:

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

- Calculate the total payload carried by boosters from NASA
- First select records where customer = NASA (CRS), then calculate the SUM of all payloads

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

```
%sql SELECT SUM(payload_mass__kg_) AS Total_Payload_Mass FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://tfp70922:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Select records with booster version F9 v1.1, then calculate the average payload.

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

```
# Select only pure booster version F9 v1.1 for outcome task 4
%sql SELECT AVG(payload_mass__kg_) AS Average_Payload_Mass FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://tfp70922:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

average_payload_mass
2928

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Select records with ‘success (ground pad)’ , then calculate the minimum date

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql SELECT MIN(DATE) AS Date_First_Success FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://tfp70922:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

date_first_success
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Select records with ‘Success (drone ship)’ AND payload BETWEEN 4000-6000 kgs

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

```
%%sql SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_ FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
 * ibm_db_sa://tfp70922:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

booster_version	payload_mass__kg_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Group records by mission outcome and Count the mission outcomes.
  A total of 100 Successful mission outcomes and 1 Failure mission.

List the total number of successful and failure mission outcomes

```
%%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

```
 * ibm_db_sa://tfp70922:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- First find the max payload (sub-query) and use this to select the booster versions that carried this payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%%sql SELECT BOOSTER_VERSION FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://tfp70922:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Select records with YEAR(DATE)=2015 AND landing outcome = 'Failure (drone ship)'

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

```
%%sql SELECT BOOSTER_VERSION, LAUNCH_SITE, DATE FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = 2015
```

```
 * ibm_db_sa://tfp70922:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

booster_version	launch_site	DATE
F9 v1.1 B1012	CCAFS LC-40	2015-01-10
F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

- Select records between the two dates mentioned above, group them by Landing_outcome and sort them by the count of landing_outcome (descending).

```
%%sql SELECT LANDING_OUTCOME,COUNT(LANDING_OUTCOME) FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME ORDER BY COUNT(LANDING_OUTCOME)DESC
```

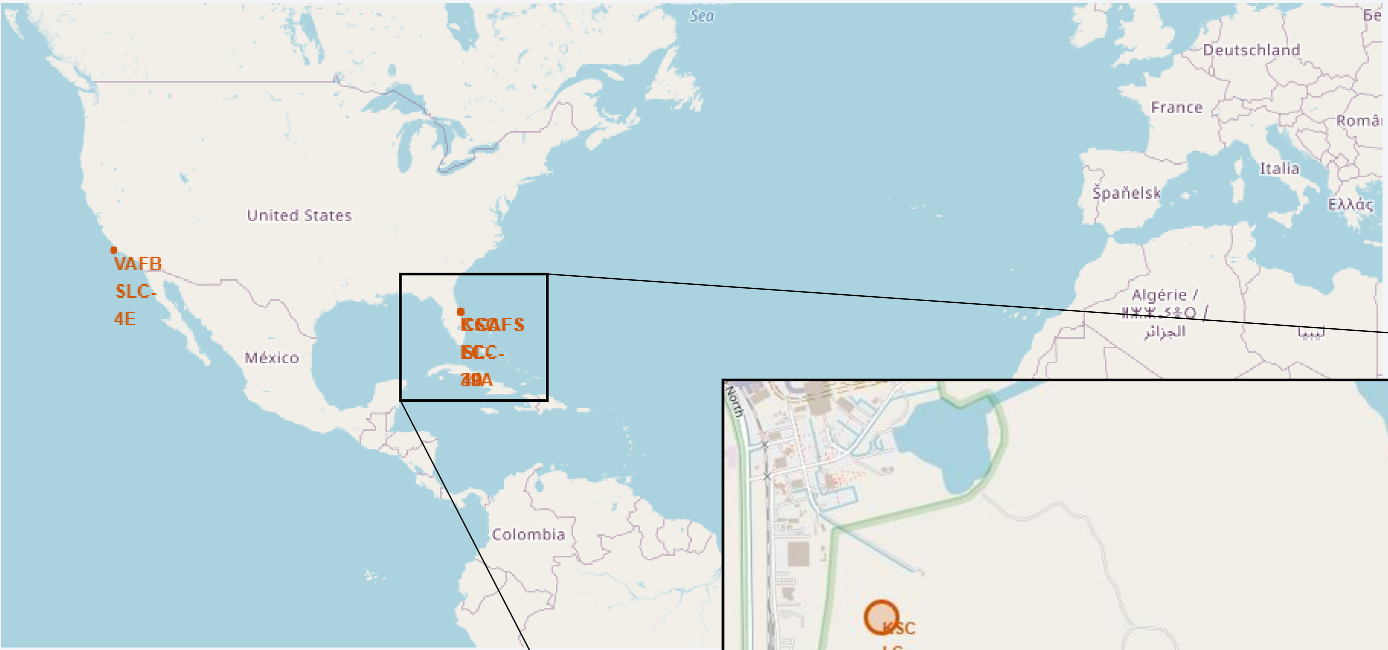
```
 * ibm_db_sa://tfp70922:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108k
Done.
```

landing_outcome	2
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

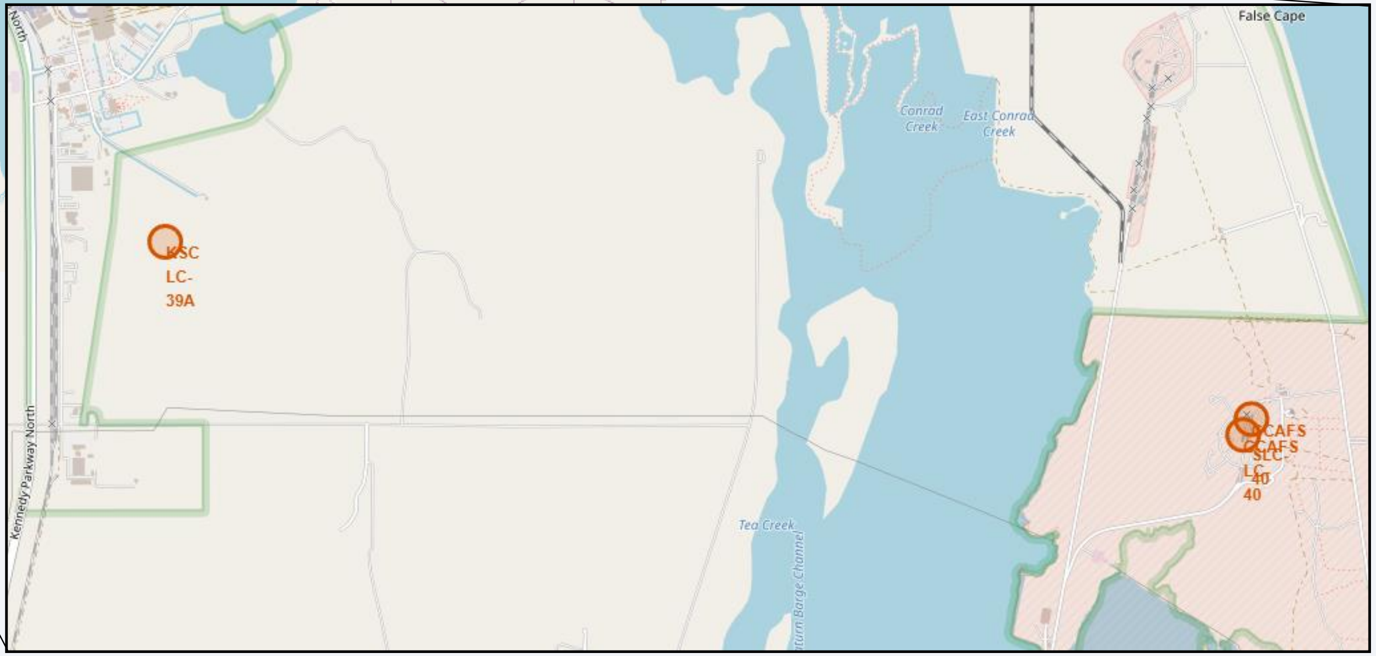
Section 3

Launch Sites Proximities Analysis

All launch sites’ location markers

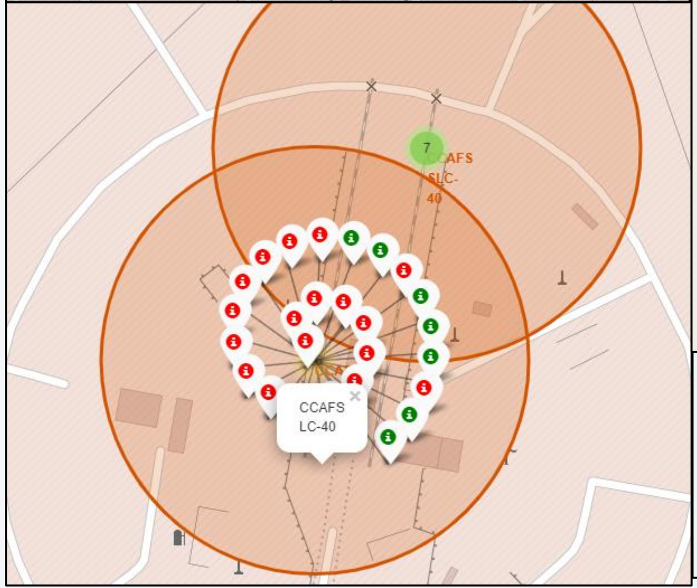


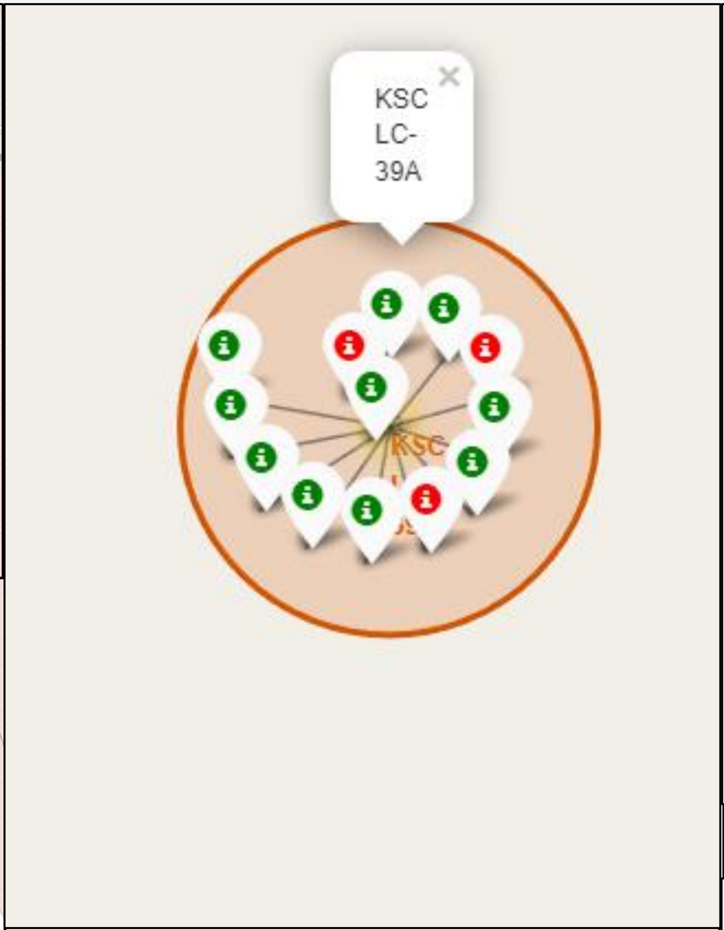
World Map with the launch sites located at the east- and west coast of the USA

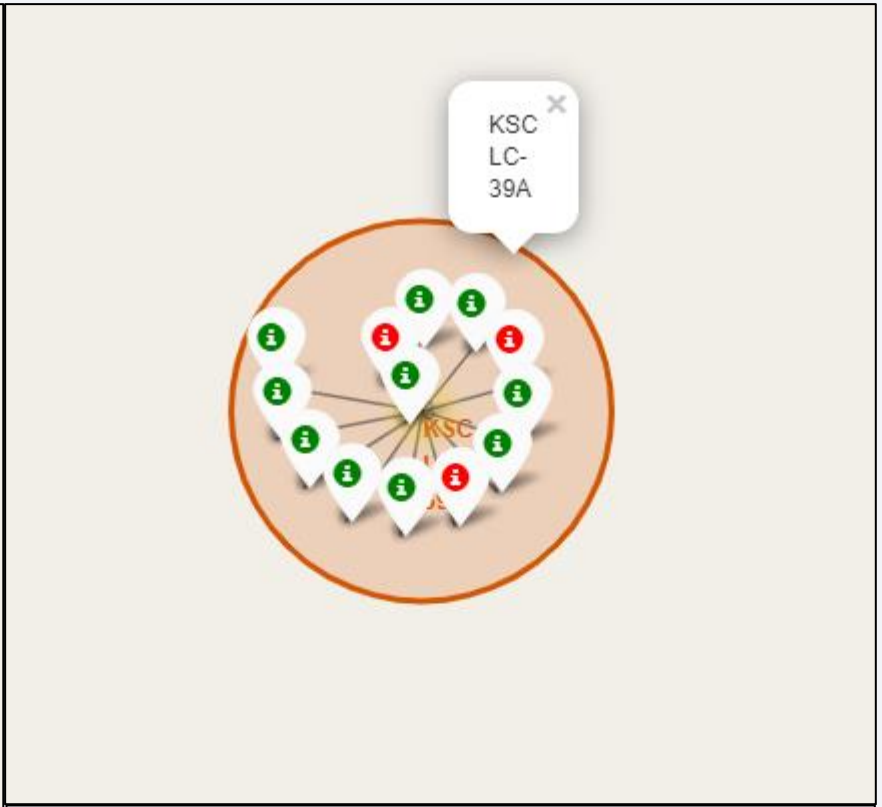


Detailed zoomed map of the three launch locations at the East Coast

Launch outcomes map







<u>Florida</u>

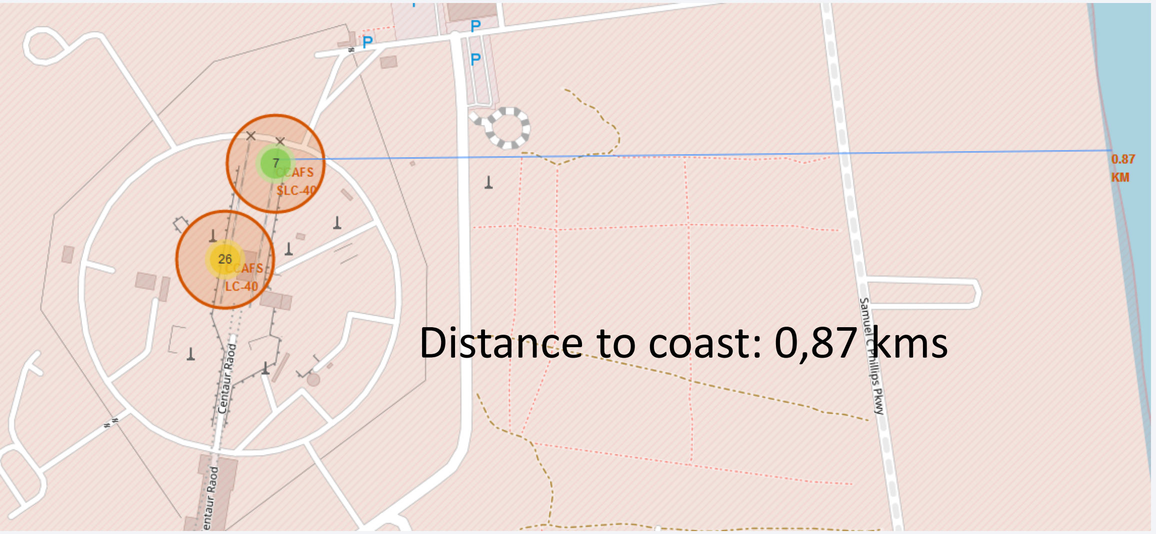
Launch outcomes CCAFS SLC-40
CCAFS LC-40
KSC LC-39A

Launch outcomes KSC LC-39A (California)

Green Icon shows success,
Red Icon shows failure

Launch site distances to landmarks

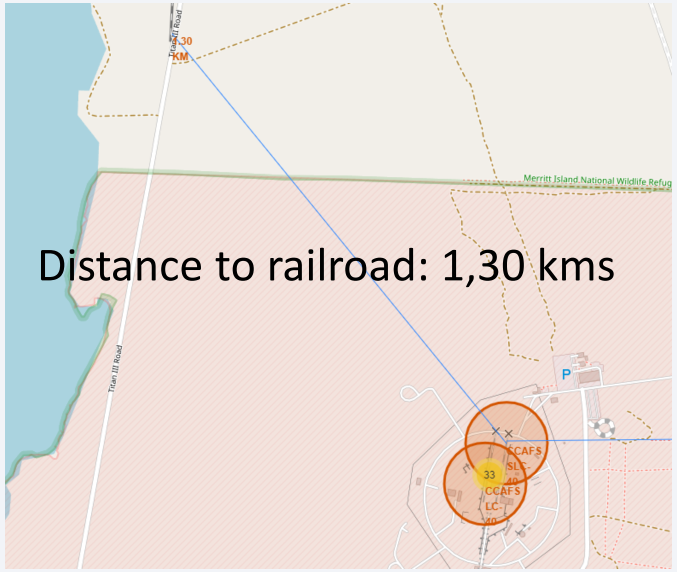
All distances measured from launch site CCAFS SLC-40



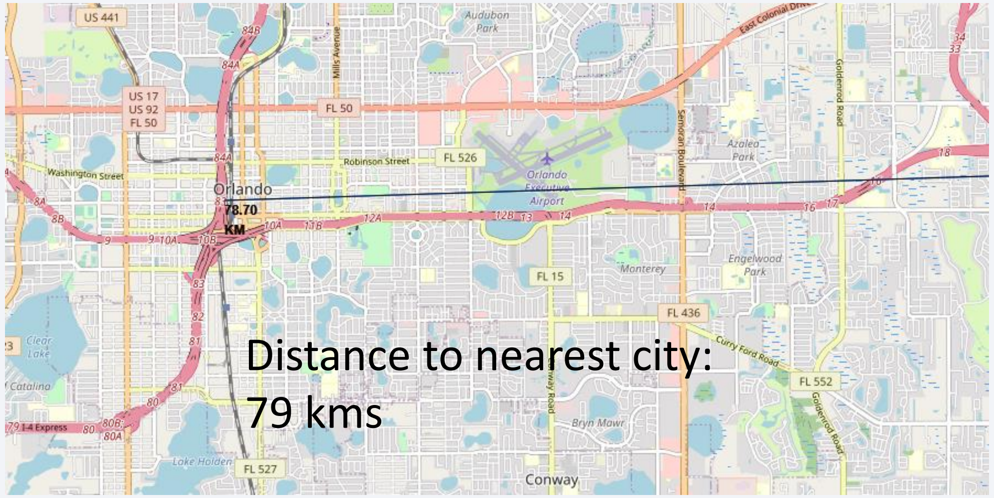
Distance to coast: 0,87 kms

Questions:

- Are launch sites in close proximity to railways? Yes for 'duty railroad', no for civilian railroads
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes, very close
- Do launch sites keep certain distance away from cities? Yes, besides some small villages the nearest city is 79 Kms away.



Distance to railroad: 1,30 kms



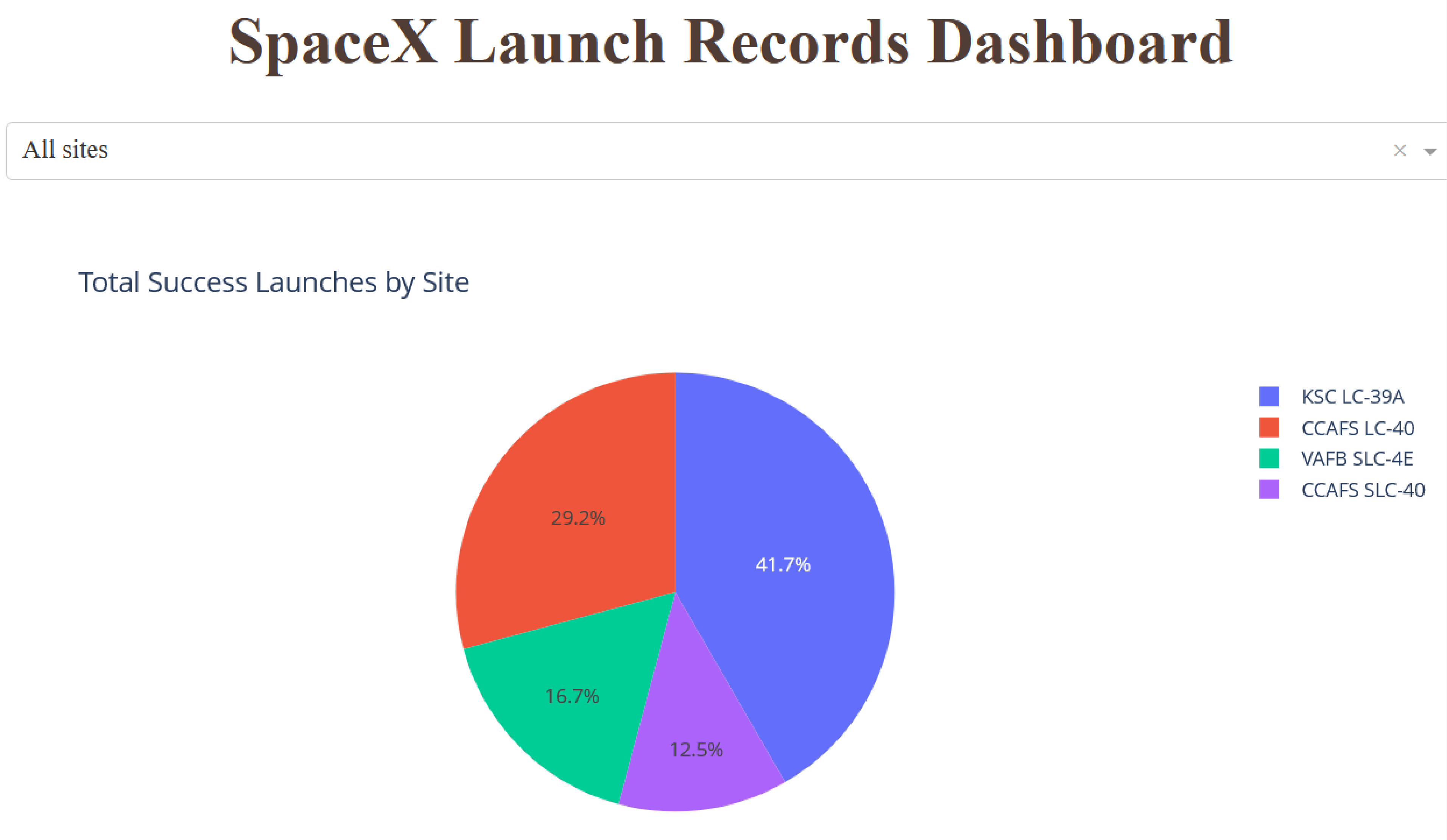
Distance to nearest city: 79 kms

Distance to highway junction: 30 kms

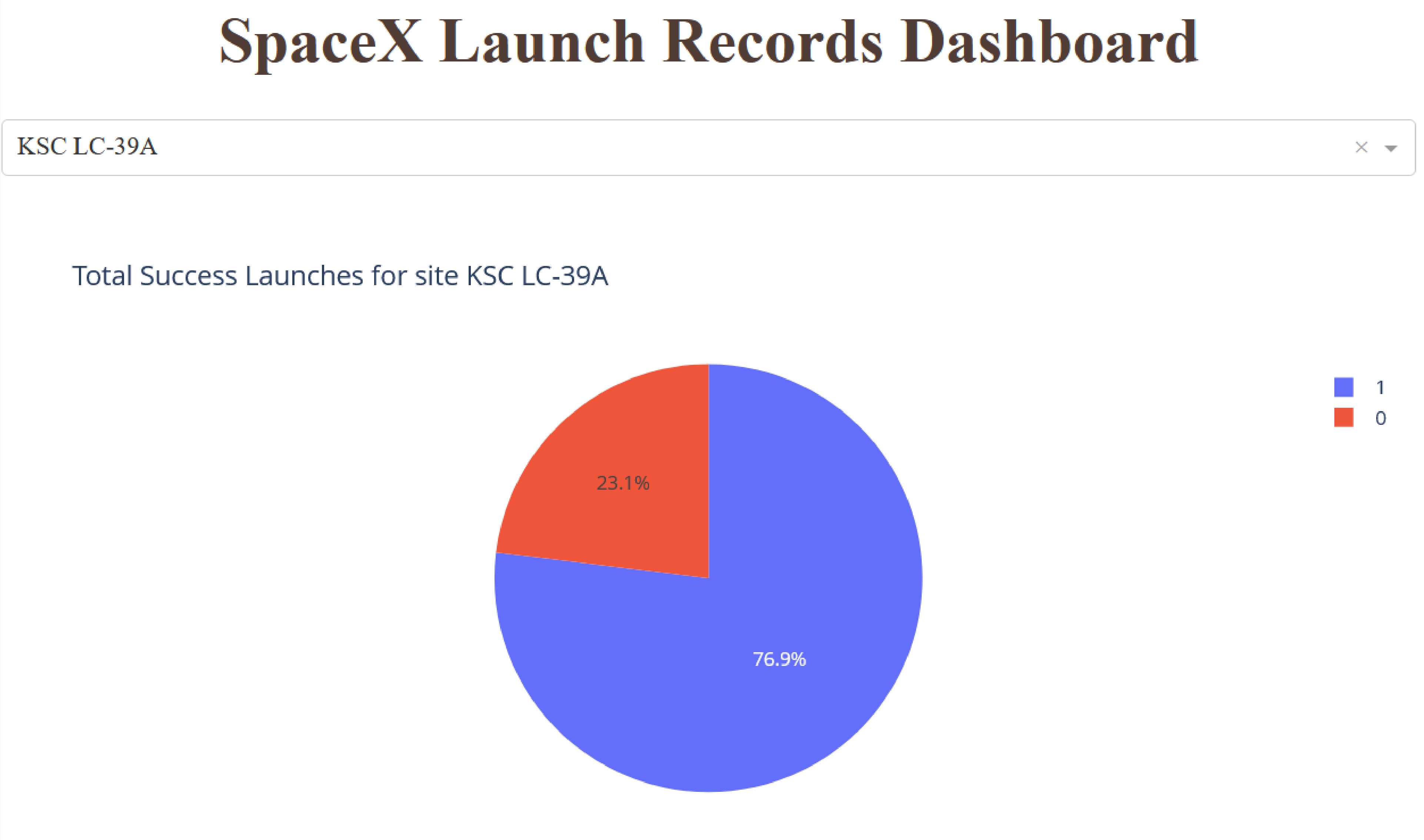
Section 4

Build a Dashboard with Plotly Dash

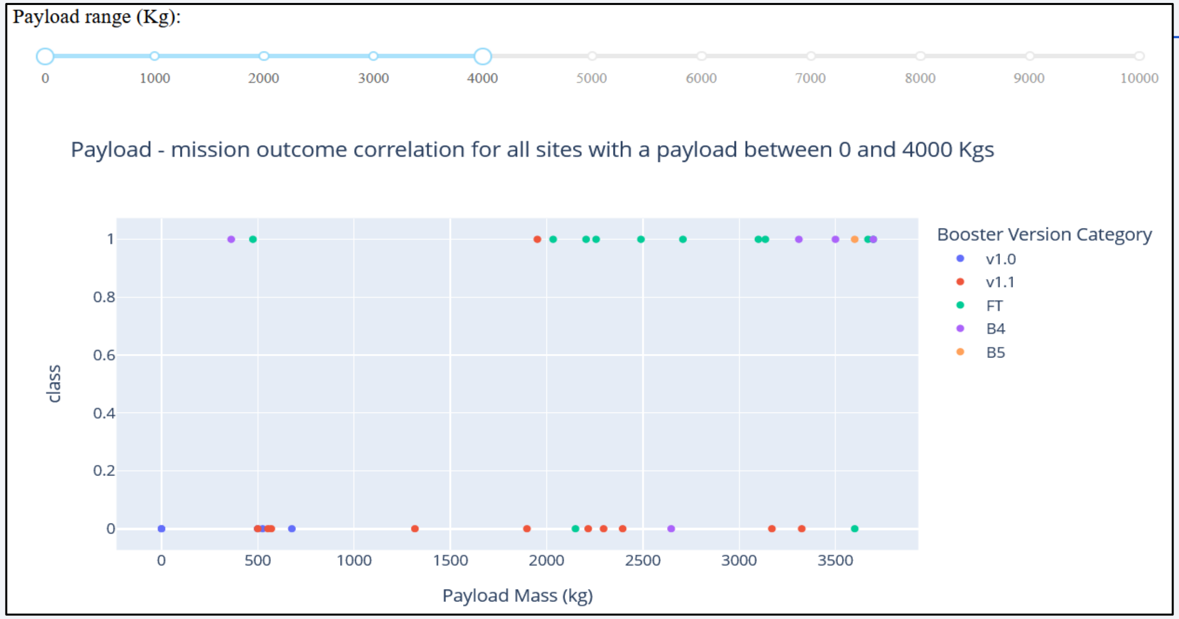
Pie chart showing total success launches per site



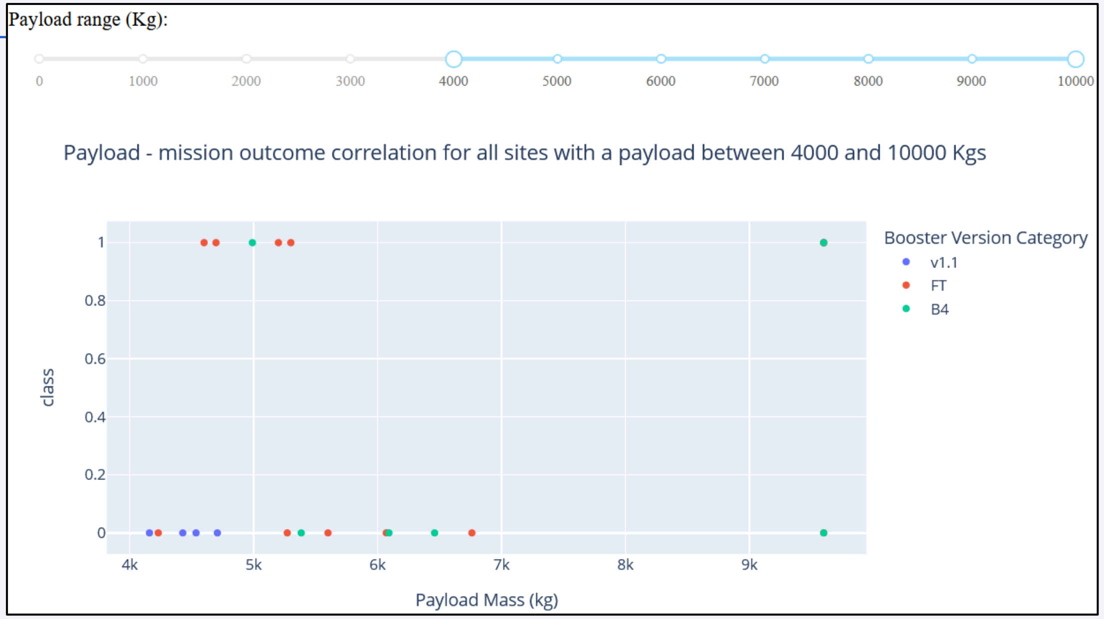
Launch Site with highest launch success ratio



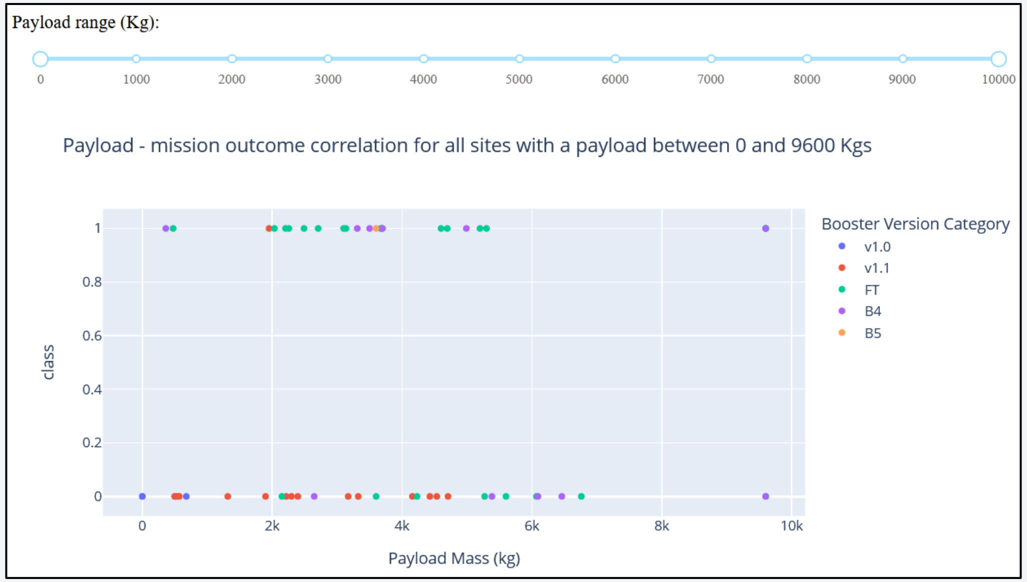
Payload vs Launch Outcome – all sites, different Payloads



Low Payload



Heavy Payload



All Payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Logistic Regression:

```
GridSearchCV(cv=10, estimator=LogisticRegression(),
             param_grid={'C': [0.01, 0.1, 1], 'penalty': ['l2'],
                         'solver': ['lbfgs']})
```

```
tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

- Support Vector Machine

```
GridSearchCV(cv=10, estimator=SVC(),
             param_grid={'C': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
       1.00000000e+03]),
                         'gamma': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
       1.00000000e+03]),
                         'kernel': ('linear', 'rbf', 'poly', 'rbf', 'sigmoid')})
```

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

- Decision Tree Classifier

```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']},
             scoring='accuracy')
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 6, 'max_features': 'auto', 'min_samples_lea
f': 1, 'min_samples_split': 10, 'splitter': 'random'}
accuracy : 0.8892857142857142
```

Classification Accuracy – Best performing model

Although all models perform nearly equal, the Tree classification model has a slightly higher score than the rest.

- K Nearest neighbors

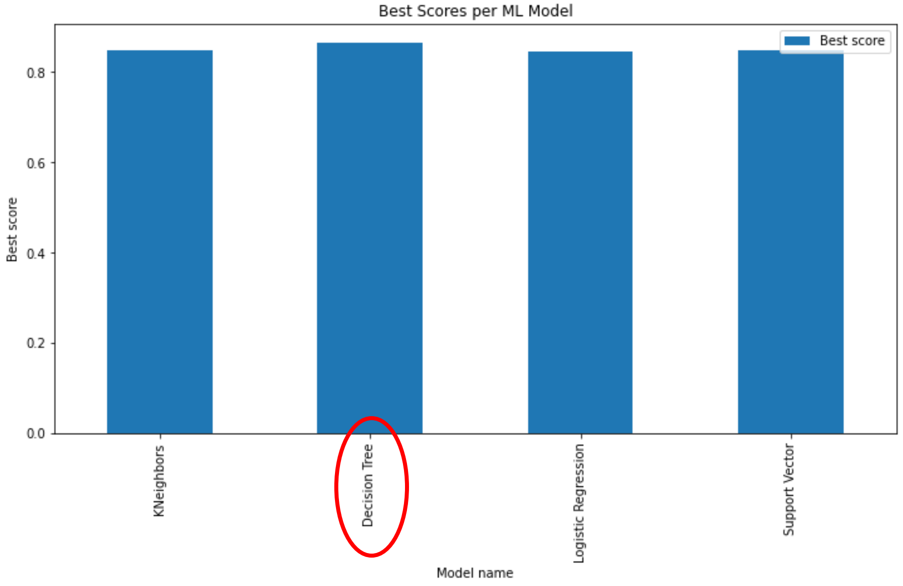
```
GridSearchCV(cv=10, estimator=KNeighborsClassifier(),
             param_grid={'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                         'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
                         'p': [1, 2]},
             scoring='accuracy')

tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858
```

- Find the best performing model:

Find the method performs best:

```
print(f'score knn: {knn_cv.best_score_:.4f}' )
print(f'score tree: {tree_cv.best_score_:.4f}' )
print(f'score logreg: {logreg_cv.best_score_:.4f}' )
print(f'score SVM: {svm_cv.best_score_:.4f}' )
```

```
score knn: 0.8482
score tree: 0.8893
score logreg: 0.8464
score SVM: 0.8482
```




```
In [32]: print(f'The best model is Decision Tree with score: {tree_cv.best_score_:.4f}, Best parameters are: {tree_cv.best_params_}')
```

```
The best model is Decision Tree with score: 0.8893, Best parameters are: {'criterion': 'entropy', 'max_depth': 6, 'max_feat
ures': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random'}
```

Confusion Matrix

The confusion matrix shows the ways in which the classification model is confused when it makes predictions.

1. From the 12 predicted true positive landings, 12 did land.
2. There are 3 false positive (predicted to fail, but landed) and 3 false negative (predicted to land, but failed) landings

There were only 18 test cases. In order to thoroughly train and test the model, we will need more data. First of all more training data, in order to finetune the hyperparameters, as well as more test data in order to find out the model works properly.

Conclusions

Based on our analysis we can say that:

- Earlier launches were less successful than recent launches
- Orbits into which rockets are launched and payloads have an influence on success rate
    - Launches into ES-L1, GEO, HEO and SSO were highly successful, however only a few launches were made.
    - Multiple successful launches into VLEO.
    - Higher payloads give more successful launches
- Launch site KSC LC-39A has the highest success rate.

With regards to our Machine Learning Model we can conclude that:

- The Decision Tree Classifier Model gives the highest score
- The confusion matrix shows us that we need more training data to finetune the model and more test data to validate its performance

Thank you!