IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Angelo Giardini
August 2023

Outline

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

Executive Summary

- Space X is One of the most successful is the rocket launch program of SpaceX. One reason for this is that SpaceX's rocket launches are relatively inexpensive as SpaceX can reuse the first rocket stage
- Space Y that would like to compete with SpaceX. Using publicly available data on SpaceX launches (API, Wikipedia), we determine if the first stage will land, and thus are able to determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch
- We gathered information using explorative data analyses, including static and interactive graphs (dashboards) to determine variables that are useful to predict the launch outcome (failure vs success). Among these variables were, for example, Launch Site and Payload Mass.
- We trained four different machine learning (ML) models to predict if SpaceX will reuse the first stage to be able to choose the best model
- Three ML models / methods (i.e., Logistic Regression, Support Vector Machines, Decision Tree Classifier) yielded identical results with an accuracy score of 83.3% correct predictions
- This underlines that it is possible to make valid predictions for a technical complex activity such as rocket launches based on publicly available data and established data science methods

Introduction

- The commercial space age is here, companies are making space travel affordable for everyone
- One of the most successful is the rocket launch program of SpaceX. One reason for this is that SpaceX's rocket launches are relatively inexpensive. 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.
- Space Y that would like to compete with SpaceX. 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.
- In the following, we will predict if the Falcon 9 first stage will land successfully.
- We will do this by gathering information about Space X, using explorative data analyses, including the creation of static and interactive graphs (dashboards).
- Further, instead of using rocket science to determine if the first stage will land successfully, we will train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

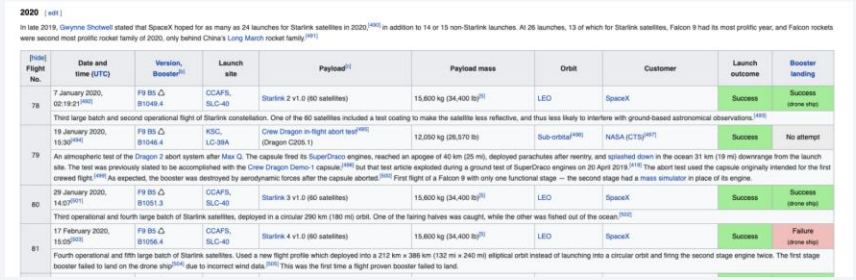
Executive Summary

- Data collection methodology: Two main sources for Space X launch data:
  a) Official SpaceX Rest API and
  b) Wikipedia (webscraping methodology)
- Perform data wrangling: Transform data into csv and Dataframe (Pandas) format
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models:
  - Split predictors X (standardized) and outcome Y datasets into train and test sets
  - Run four different ML classification models (incl. GridSearch): Logistic Regression, SVM, Decision Tree Classification, K-neares Neighbor
  - Choose the best model(s) based on accuracy score / confusion matrix

Data Collection

We used two main sources for Space X launch data:

- **Space X Rest API:** data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome
- **Wikipedia:** We were then performing web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled "List of Falcon 9 and Falcon Heavy launches". The launch records were stored in a HTML table as shown below:



2020 [edit]

In late 2019, Gwynne Shotwell stated that SpaceX hoped for as many as 24 launches for Starlink satellites in 2020,[490] in addition to 14 or 15 non-Starlink launches. At 26 launches, 13 of which for Starlink satellites, Falcon 9 had its most prolific year, and Falcon rockets were second most prolific rocket family of 2020, only behind China's Long March rocket family.[491]

[hide] Flight No.	Date and time (UTC)	Version, Booster[b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
78	7 January 2020, 02:19:21[492]	F9 B5 B1049.4	CCAFS, SLC-40	Starlink 2 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third large batch and second operational flight of Starlink constellation. One of the 60 satellites included a test coating to make the satellite less reflective, and thus less likely to interfere with ground-based astronomical observations.[493]								
79	19 January 2020, 15:30[494]	F9 B5 B1046.4	KSC, LC-39A	Crew Dragon in-flight abort test[495] (Dragon C205.1)	12,050 kg (26,570 lb)	Sub-orbital[496]	NASA (CTS)[497]	Success	No attempt
	An atmospheric test of the Dragon 2 abort system after Max Q. The capsule fired its SuperDraco engines, reached an apogee of 40 km (25 mi), deployed parachutes after reentry, and splashed down in the ocean 31 km (19 mi) downrange from the launch site. The test was previously slated to be accomplished with the Crew Dragon Demo-1 capsule,[498] but that test article exploded during a ground test of SuperDraco engines on 20 April 2019.[418] The abort test used the capsule originally intended for the first crewed flight.[499] As expected, the booster was destroyed by aerodynamic forces after the capsule aborted.[500] First flight of a Falcon 9 with only one functional stage — the second stage had a mass simulator in place of its engine.								
80	29 January 2020, 14:07[501]	F9 B5 B1051.3	CCAFS, SLC-40	Starlink 3 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third operational and fourth large batch of Starlink satellites, deployed in a circular 290 km (180 mi) orbit. One of the fairing halves was caught, while the other was fished out of the ocean.[502]								
81	17 February 2020, 15:05[503]	F9 B5 B1056.4	CCAFS, SLC-40	Starlink 4 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Failure (drone ship)
	Fourth operational and fifth large batch of Starlink satellites. Used a new flight profile which deployed into a 212 km × 386 km (132 mi × 240 mi) elliptical orbit instead of launching into a circular orbit and firing the second stage engine twice. The first stage booster failed to land on the drone ship[504] due to incorrect wind data.[505] This was the first time a flight proven booster failed to land.								

Data Collection – SpaceX API

- SpaceX REST API endpoint:

https://api.spacexdata.com/v4/launches/past

- GitHub URL:

Coursera_10-spacex-data-collection-api.ipynb

Flowchart of API calls

1. Getting Response from API

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

response = requests.get(spacex_url)
```

2. Converting Response to .json file

```
data_all=pd.json_normalize(response.json())
```

3. Apply custom functions to clean data

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
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}
```

5. Filter dataframe

```
data_falcon9=df[df['BoosterVersion']!='Falcon 1']
```

Data Collection - Scraping

- Falcon 9 historical launch records were taken from a Wikipedia page:

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

- GitHub URL of the web scraping notebook:

https://github.com/giardini/Coursera_DS_Capstone_Project/blob/main/jupyter-labs-spacex_webscraping.ipynb

Flowchart of web scraping:

Request the Falcon9 Launch Wiki page and create a BeautifulSoup object to for parsing

```
res = requests.get(static_url)
Falcon9Soup=BeautifulSoup(res.text, 'html.parser')
```

Select the right table

```
html_tables=Falcon9Soup.find_all('table')
first_launch_table = html_tables[2]
print(first_launch_table)
```

Extract all column/variable names from the HTML table header

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

Create a data frame by parsing the launch HTML tables

```
extracted_row = 0
for table_number,table in enumerate(Falcon9Soup.find_all('table',"wikitable plainrowhe
    for rows in table.find_all("tr"):
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        row=rows.find_all('td')
        if flag:
            extracted_row += 1
            launch_dict["Flight No."].append(flight_number)
            datatimelist=date_time(row[0])
            print(flight_number)

            date = datatimelist[0].strip(',')
            launch_dict["Date"].append(date)
            date = datatimelist[0].strip(',')
```

Etc.

Data Wrangling

- We performed quantitative exploratory data analysis to get an overview of our variables (e.g., number auf launches per site) and in order to determine Training Labels (e.g., Launch Outcome: Success / Failure)
- We also used SQL to gain some more quantitative insights into, for example, Launch Sites, and Payloads
- We used graphics (scatterplots, line plots, bar charts) to obtain some preliminary insights about if and how each variable would affect the Launch Outcome
- Based on this, we selected the features that will be used in success prediction in the future module: Flight Number, Payload Mass, Orbit type, Launch Site, etc.

Github URLs for notebooks:

- Quantitative data analysis: https://github.com/giardini/Coursera_DS_Capstone_Project/blob/main/jupyter_labs_week_1_spacex-data_wrangling.jupyterl.ipynb
- Visual data analysis: https://github.com/giardini/Coursera_DS_Capstone_Project/blob/main/jupyter-labs-eda-sql-coursera_exploration_Week_2(1).ipynb
- SQL analysis: https://github.com/giardini/Coursera_DS_Capstone_Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with Data Visualization

The Following charts were plotted:

- Scatterplot: How launch outcome changes with number of flights and Payload Mass
- Scatterplot: How launch outcome changes with number of flights and Launch site
- Barplot: Relationship between success rate and orbit type
- Scatterplot: Launch outcome by Flight number and Orbit type
- Line plot: Average success rate by year (2010 – 2020)

- GitHub URL of data visualization notebook:
  https://github.com/giardini/Coursera_DS_Capstone_Project/blob/main/jupyter-labs-eda-sql-coursera_exploration_Week_2(1).ipynb

EDA with SQL

- The following SQL queries were performed:
  - Display names of 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
  - Display the average payload mass carried by booster version F9 v1.1
  - List the data when the first successful landing outcome in ground pad was achieved
  - List the names of the boosters which have a 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 max 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 (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.
- GitHub URL for notebook: https://github.com/giardini/Coursera_DS_Capstone_Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- In the previous exploratory data analysis labs, we have visualized the SpaceX launch dataset using matplotlib and seaborn and discovered some preliminary correlations between the launch site and success rates. We then performed more interactive visual analytics using Folium.
- Process:
  - Create a site map (Folium Map object)
  - Add a marker (folium.map.Marker) and a highlighted circle with a text label (folium.Circle) for each launch site on the site map
  - Add the launch outcomes for each site by creating Marker clusters first
  - Calculate the distances between a launch site to its proximities: adding MousePosition on the map (get coordinate for a mouse over a point on the map)
- GitHub URL of interactive Folium map: https://github.com/giardini/Coursera_DS_Capstone_Project/blob/main/Capstone_Folium_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Next, we were building a Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time. This dashboard application contains input components such as a:
  - a Launch Site drop-down menu
  - a callback function to render success-pie-chart based on selected site dropdown
  - a range slider to select Payload
  - callback function to render the success-payload-scatter-chart scatter plot
- With this, we were able to address the following questions:
  - 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 aunch success rate?
- GitHub URL of Python syntax file: https://github.com/giardini/Coursera_DS_Capstone_Project/blob/main/spacex_dash_app(11).py

Predictive Analysis (Classification)

- To predict if the first stage will land given the data from the preceding labs, we created a machine learning pipeline
- Process flow:
  1. Created a column for the Launch outcome (class)

```
array = data['Class'].to_numpy()
Y = pd.Series(array)
```

  2. Standardized the data

```
transform = preprocessing.StandardScaler()
X_standardized = transform.fit_transform(X)
X_new = pd.DataFrame(X_standardized, columns=X.columns)
```

  3. Split into training data and test data

```
X_train, X_test, Y_train, Y_test = train_test_split(X_new, Y, test_size=0.2, random_state=2)
```

  4. Applied four different methods separately: Logistic Regression, Support Vector Machines (SVM), Decisio Tree Classifier, and K-nearest neighbors.
  5. For each method, we trained the model and performed Grid Search, allowing us to find the hyperparameters that allow a given algorithm to perform best in terms of accuracy. We also computed a confusion matrix.
  6. Finally, we compared the accuracy scores / confusion matrices of the four methods to find the one that performs best using test data

Example SVM:

```
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
```

```
svm.fit(X_train, Y_train)
svm_cv = GridSearchCV(svm, parameters, cv=10)
svm_cv.fit(X_train,Y_train)
```

```
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 0.03162277660168379, 'gamma': 0.001, 'kernel': 'linear'}
accuracy : 0.8214285714285714
```

```
SVM_test_accuracy = svm_cv.score(X_test, Y_test)
print("Accuracy on test data:", SVM_test_accuracy)
```

```
Accuracy on test data: 0.8333333333333334
```

```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

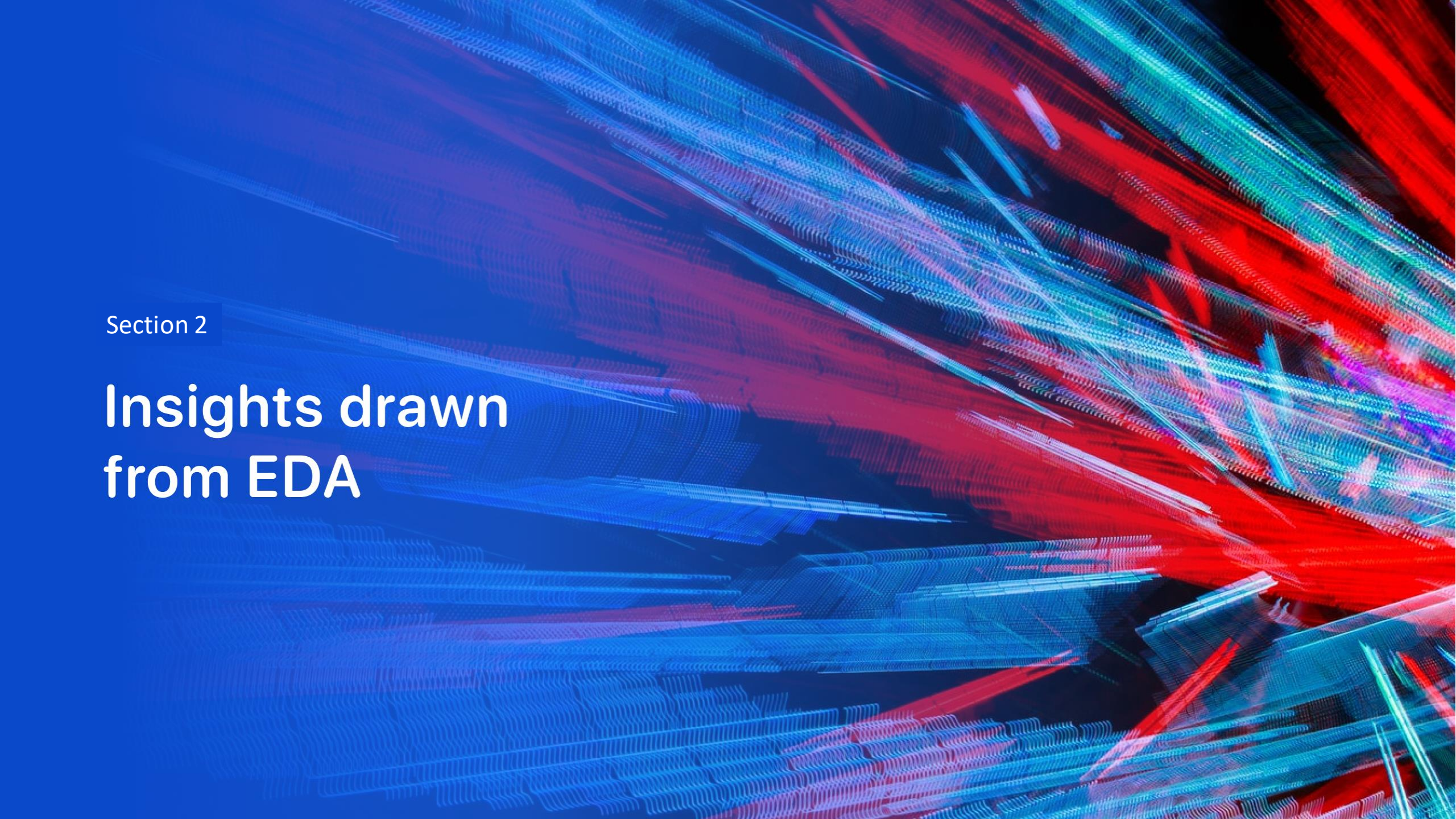
- GitHub URL of the predictive analysis lab: https://github.com/giardini/Coursera_DS_Capstone_Project/blob/main/SpaceX_Machine_Learning_Prediction_Part_5(2).ipynb

Results

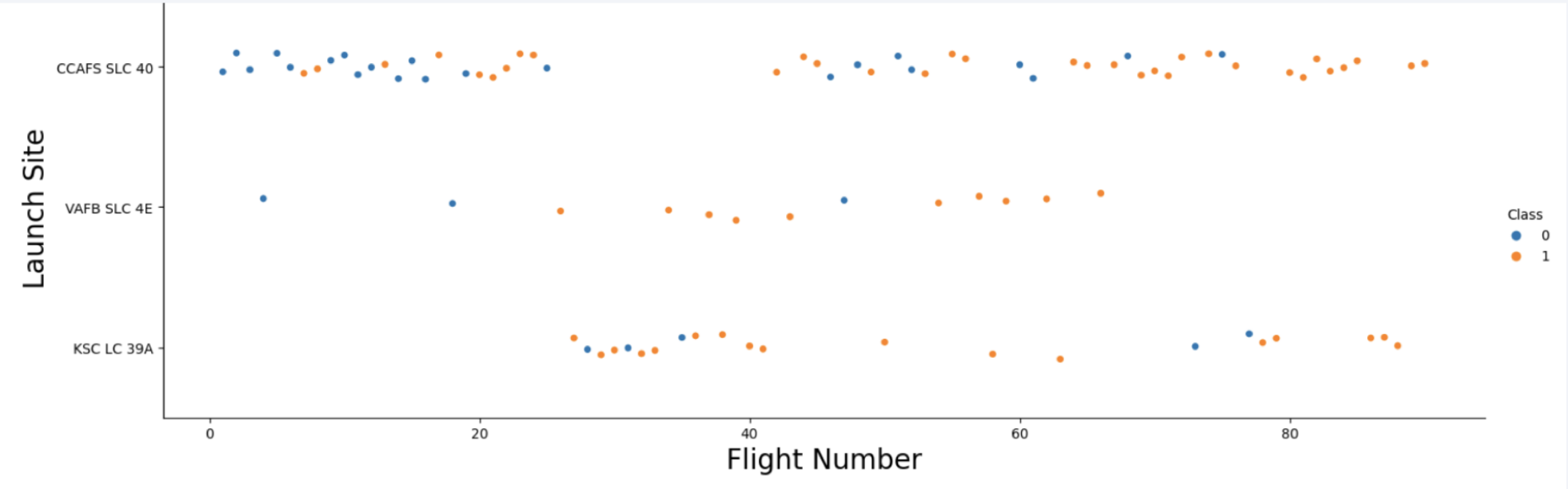
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

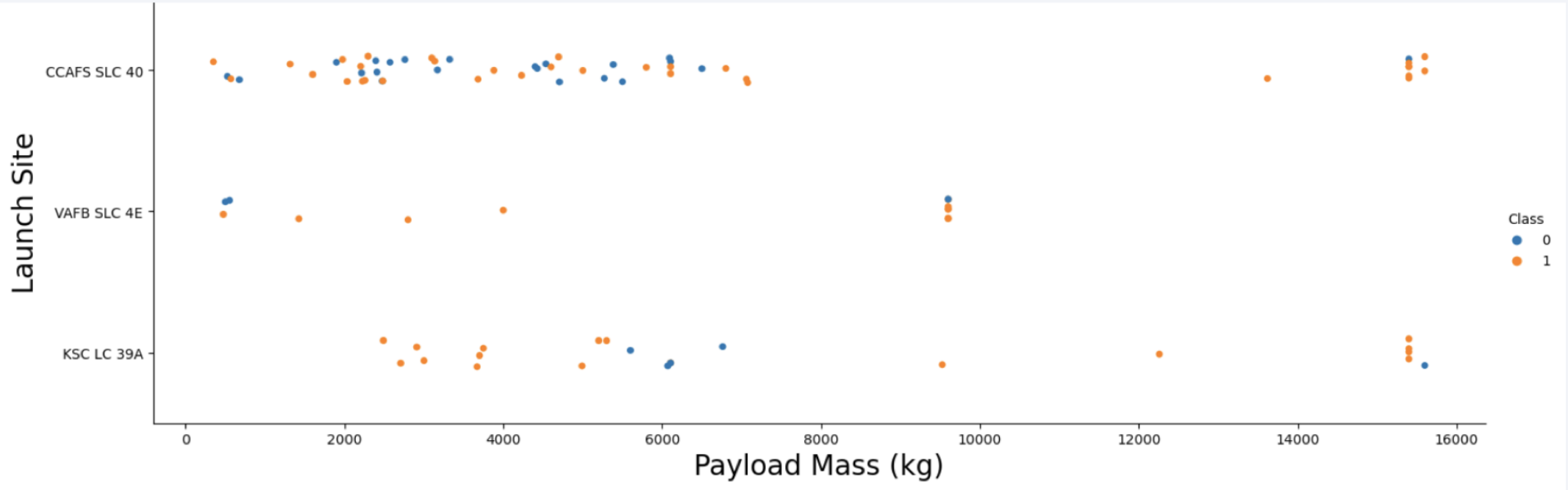


Flight Number vs. Launch Site



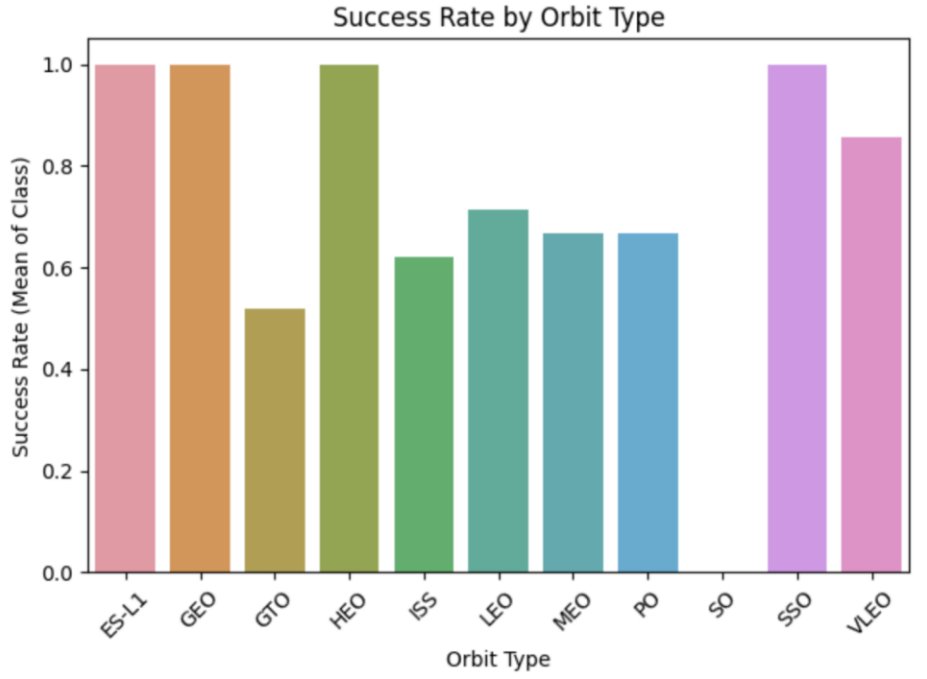
- Overall, the later flights have a higher success rates (0 / blue = failure, 1 / orange = success)
- Different launch sites have different success rates: CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC-4E have a success rate of 77%.

Payload vs. Launch Site



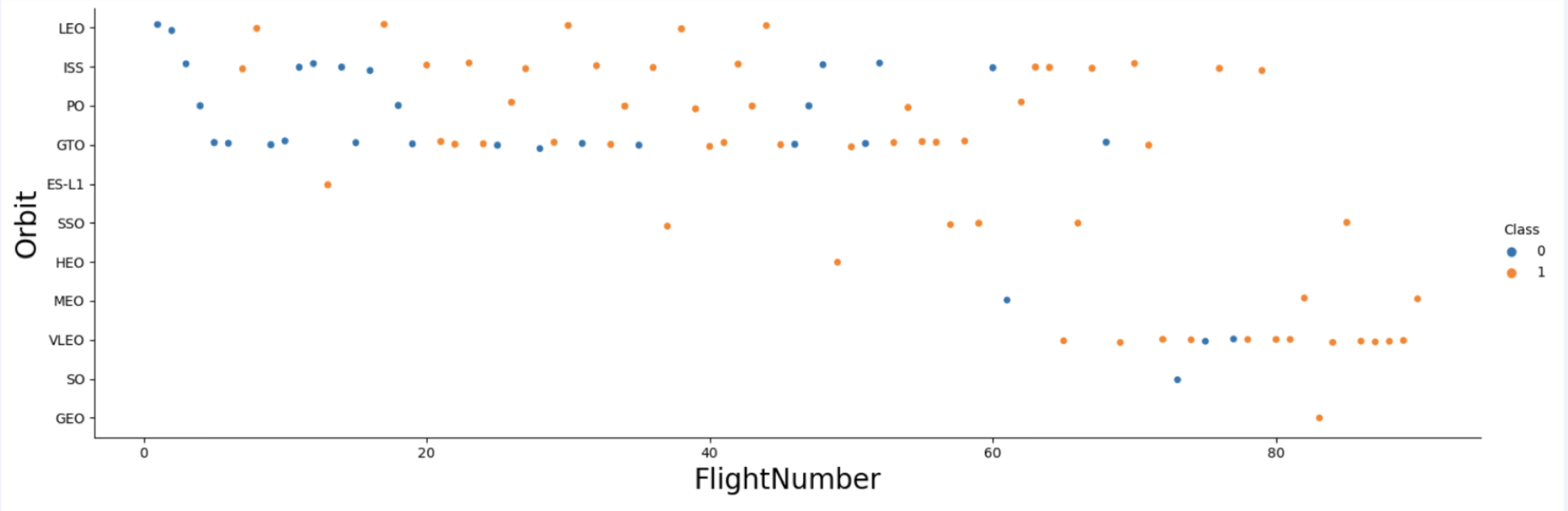
- Most of the launches have a payload mass between 0 and 7k
- For the VAFB-SLC Launch Site there are no rockets launched for heavy payload mass (greater than 10000).

Success Rate vs. Orbit Type



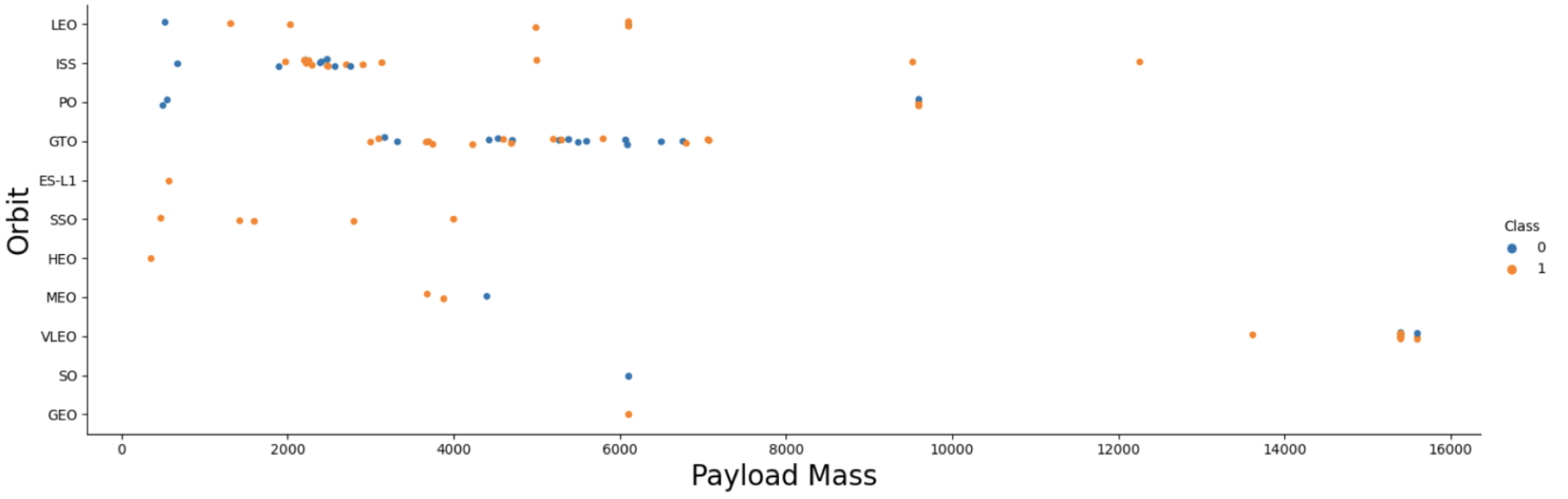
- ES-L1, GEO, HEO, and SSO orbits show a 100% success rate
- VLEO success rate is slightly above 80%
- GTO, ISS, LEO, MEO and PO have a success rate between 50% and 70%
- In the SO orbit there is a 0% success rate

Flight Number vs. Orbit Type



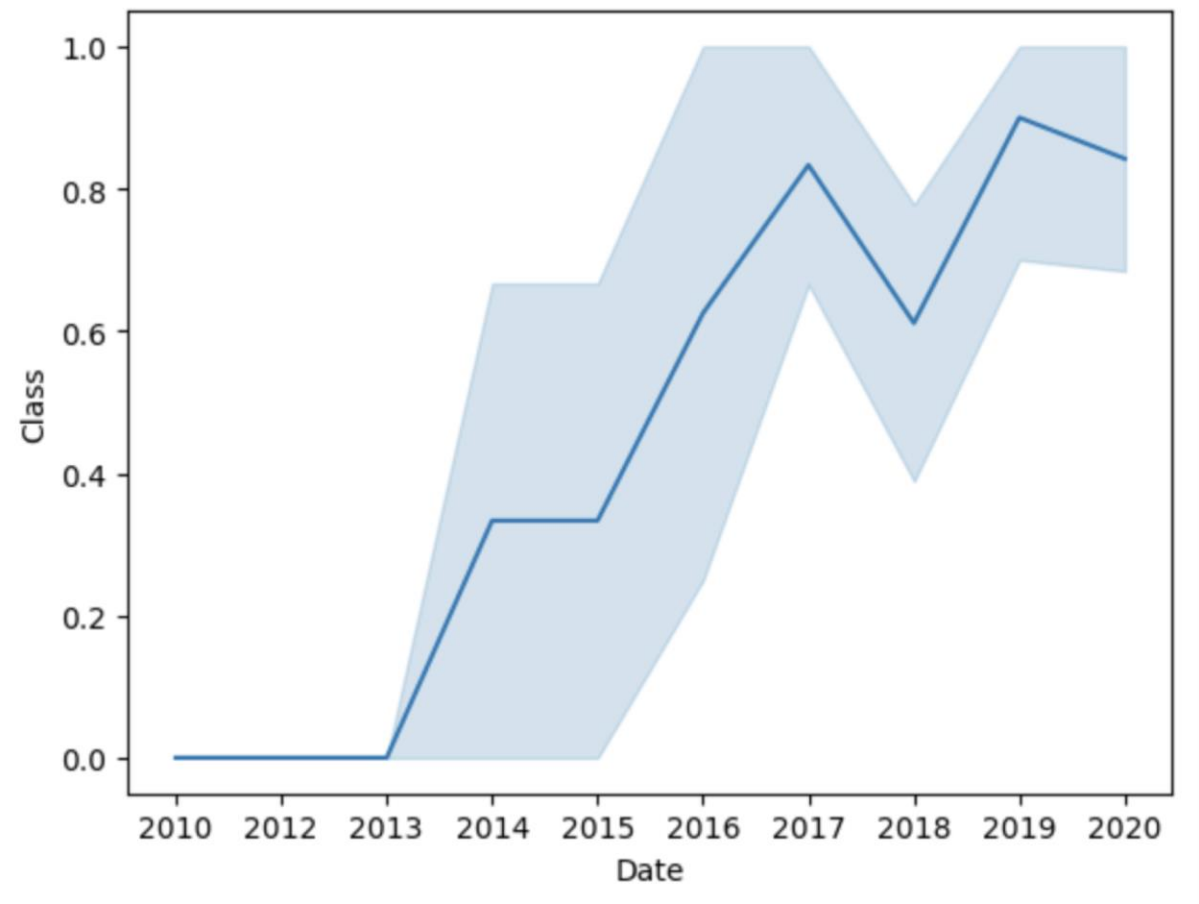
- In the LEO orbit the Success appears related to the number of flights
- There seems to be no relationship between flight number when in GTO orbit

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are higher for Polar, LEO and ISS
- However, for GTO we cannot distinguish well as we see both successful and unsuccessful missions

Launch Success Yearly Trend



- The success rate since 2013 kept increasing till 2020
- Since 2017, we see an annual success rate between 60% and 90% percent

All Launch Site Names

- Names of the unique launch sites

%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

SQL Command:

*%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE "CCA%" LIMIT 5*

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

Total Payload Mass

- Total payload carried by boosters from NASA: 45596
- SQL command:

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Customer" = "NASA (CRS)"
```

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1: 2534.7 kg
- SQL Command:

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE WHERE "Booster_Version" LIKE "F9 v1.1%"
```

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad: Dec 22, 2015
- SQL command:

%sql SELECT "Date" FROM SPACEXTABLE WHERE "Landing_Outcome" = "Success (ground pad)" LIMIT 1

Date
2015-12-22

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
- SQL command:

%sql SELECT "Booster_Version", "Landing_Outcome" from spacextable where "Landing_Outcome" = "Success (drone ship)" AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000

Booster_Version	Landing_Outcome
F9 FT B1022	Success (drone ship)
F9 FT B1026	Success (drone ship)
F9 FT B1021.2	Success (drone ship)
F9 FT B1031.2	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes
- SQL command:

%sql select "Mission_Outcome", count(*) from spacextable group by "Mission_Outcome"

Mission_Outcome	count(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass
- SQL command:

%sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" from spacextable where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_") from spacextable)

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

2015 Launch Records

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- SQL command:

%sql select Date, substr(Date, 6,2) as "Month", "Landing_Outcome", "Booster_Version", "Launch_Site", substr(Date,1,4)as "Year" from spacextable where "Landing_Outcome" = "Failure (drone ship)" AND substr(Date,1,4)='2015'

Date	Month	Landing_Outcome	Booster_Version	Launch_Site	Year
2015-10-01	10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015
2015-04-14	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015

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

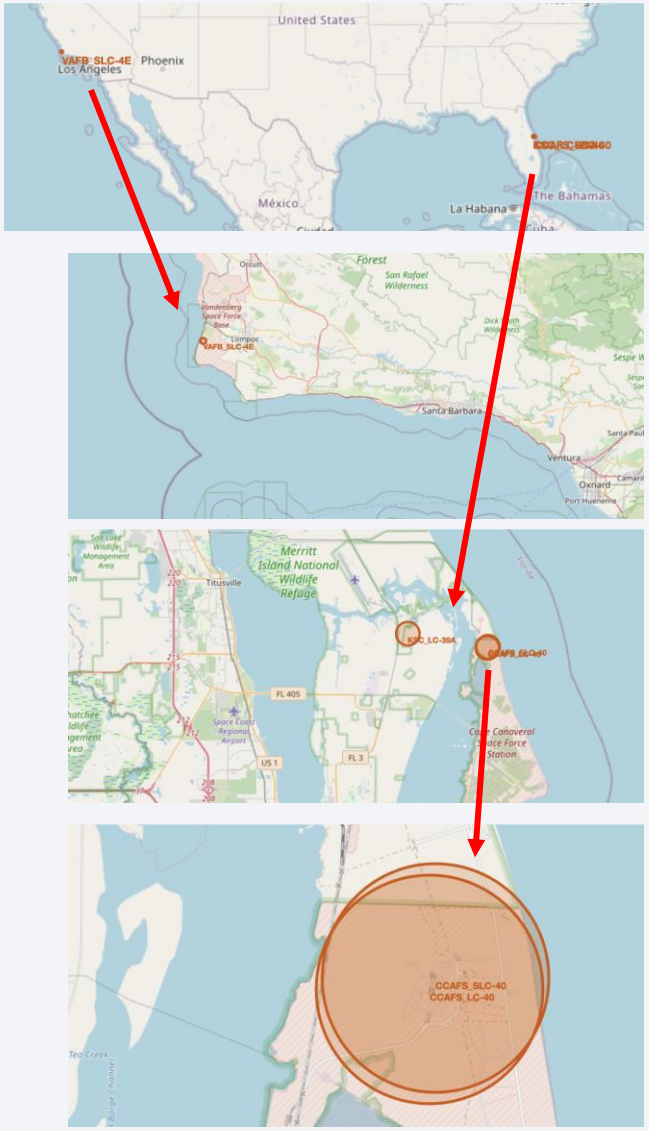
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- SQL command:

%sql select "Date", "Landing_Outcome", count("Landing_Outcome") as "Count" from spacextable where "Date" between "2010-06-04" and "2017-03-20" group by "Landing_Outcome" order by "Count" desc

Date	Landing_Outcome	Count
2012-05-22	No attempt	10
2015-12-22	Success (ground pad)	5
2016-08-04	Success (drone ship)	5
2015-10-01	Failure (drone ship)	5
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2015-06-28	Precluded (drone ship)	1
2010-08-12	Failure (parachute)	1


Section 3
Launch Sites
Proximities Analysis

Locate all Launch Sites on a map



- Locate the four Launch Sites on the US map: one site in California and three sites in Florida
- One Launch Site in California: VAFB SLC-4E
- Three Launch Sites in Florida: KSC LC-39A, CCAFS SLC-40 and CCAFS LC-40
- Coordinates for CCAFS SLC-40 and CCAFS LC-40 are almost identical

Visual representation of Launch Outcome for Launch Site

- VAFB SLC-4E (California) - 6 Failures (red) and 4 Successes (green)

- KSC LC-39A (Florida) - 3 Failures (red) and 10 Successes (green)

- CCAFS SLC-40 (Florida) - 4 Failures (red) and 3 Successes (green)

- CCAFS LC-40 (Florida) - 19 Failures (red) and 7 Successes (green)

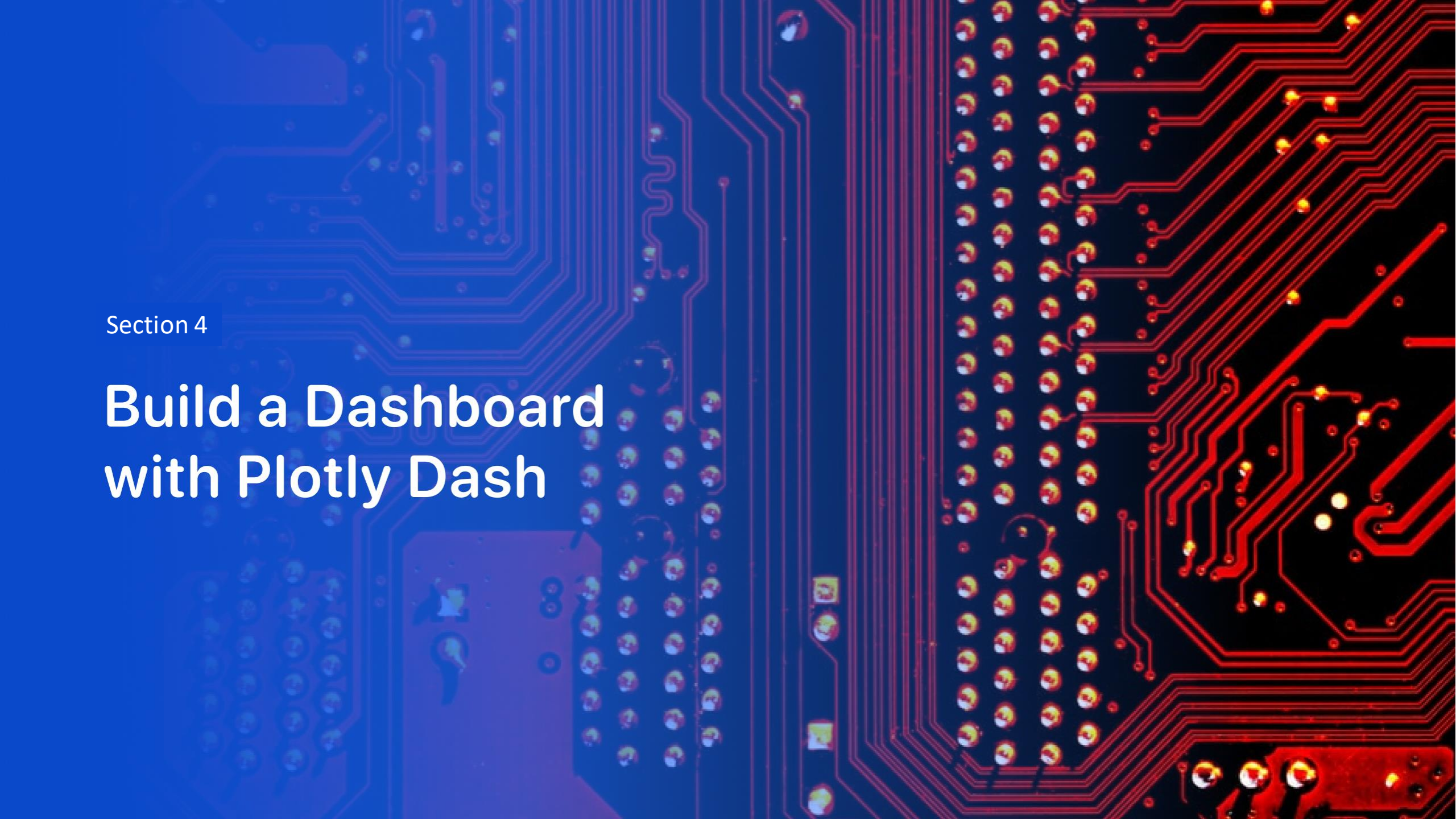
Explore proximities to Launch Sites - Example

- Display and calculate the distance between Launch Site KSC LC-39A (Florida) to the coastline: 6.58

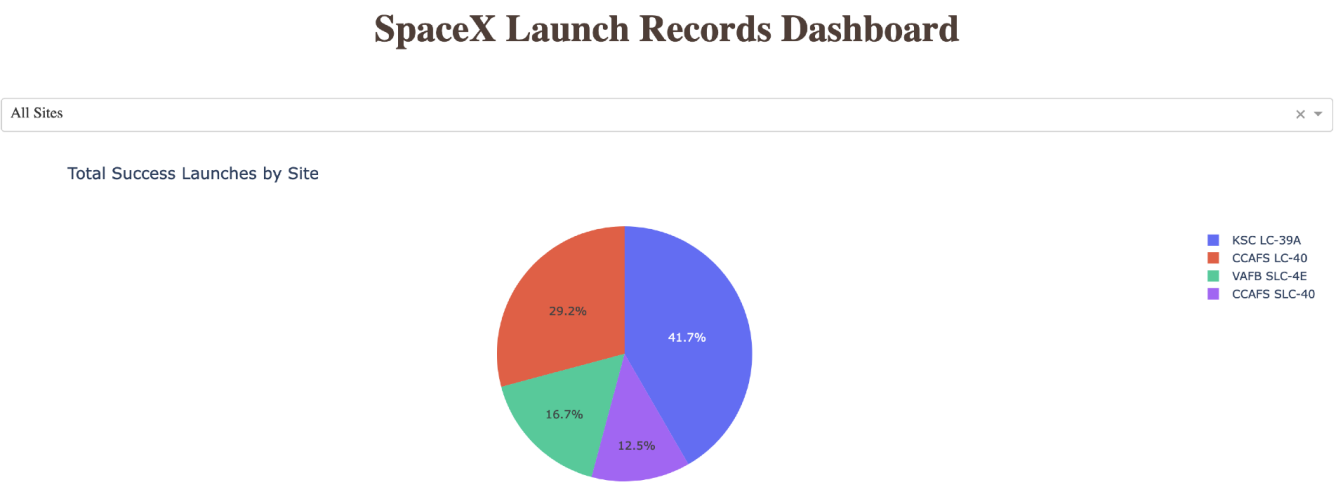
```
def calculate_distance(lat1, lon1, lat2, lon2):
    # approximate radius of earth in km
    R = 6373.0

    lat1 = radians(lat1)
    lon1 = radians(lon1)
    lat2 = radians(lat2)
    lon2 = radians(lon2)

    dlon = lon2 - lon1
    dlat = lat2 - lat1

    a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))

    distance = R * c
    return distance
```

```
KSC_LC_39A_coordinate= [28.573255, -80.646895]
coastline_lat=28.59748
coastline_lon=-80.5854
distance=calculate_distance(coastline_lat, coastline_lon, KSC_LC_39A_coordinate[0], KSC_LC_39A_coordinate[1])
distance
```

```
6.583031307789318
```

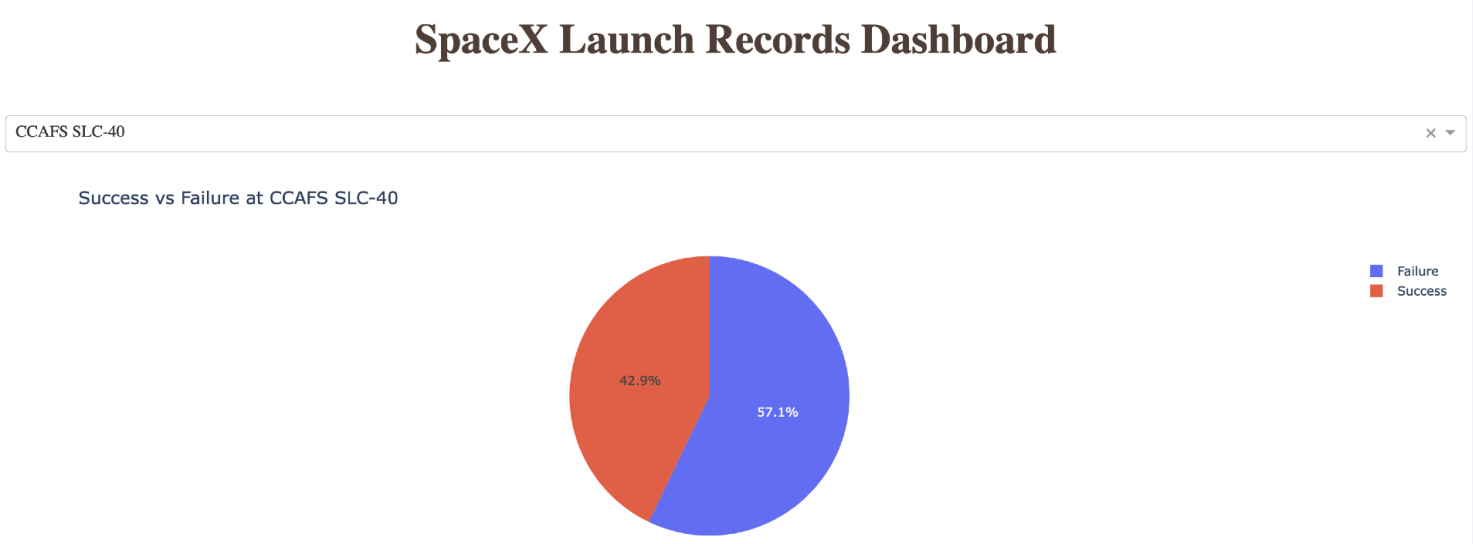

Section 4
Build a Dashboard with Plotly Dash

Launch success count for all sites (Piechart)



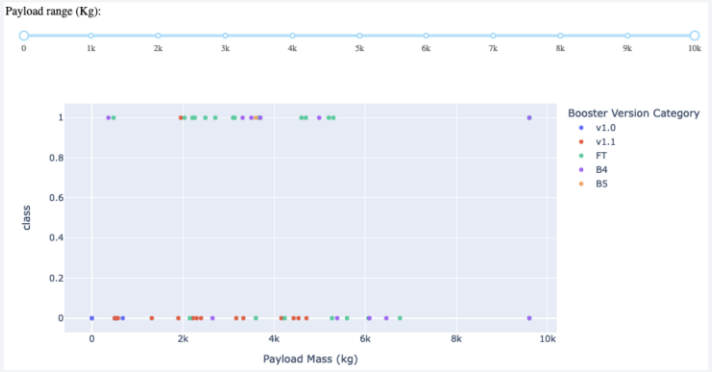
- Graph shows the Total of successful launches (in %) by Launch Site
- KSC LC-39A is the Launch Site with the highest number of successful launches
- The dropdown menu allows to choose a specific Launch Site to see the ratio between successful and unsuccessful launches for the selected site

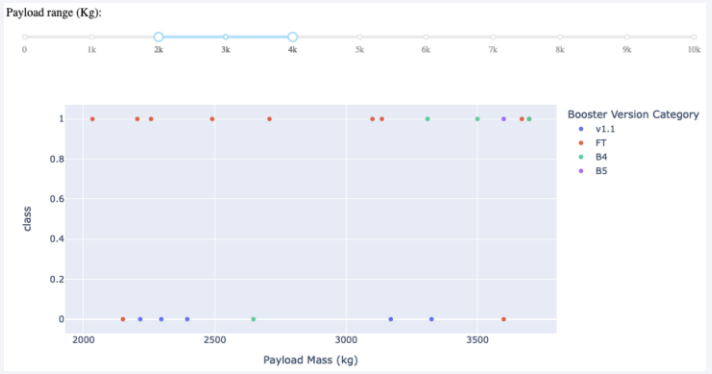
Launch Site with the highest Launch Success ratio

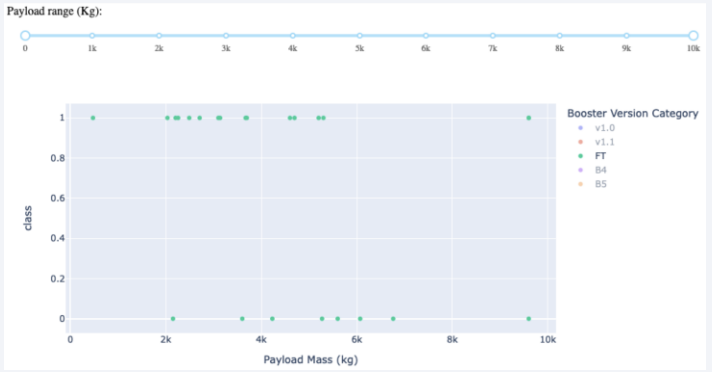


- CCAFS SLC-40 is the site with the best success to failure ratio, with 42.9% of launches being a success

Payload vs Launch Outcome for all sites







- The Payload vs. Launch Outcome scatter plot shows each successful (1) and unsuccessful (0) launch for all sites over the complete payload spectrum, and for different booster versions (colored dots)

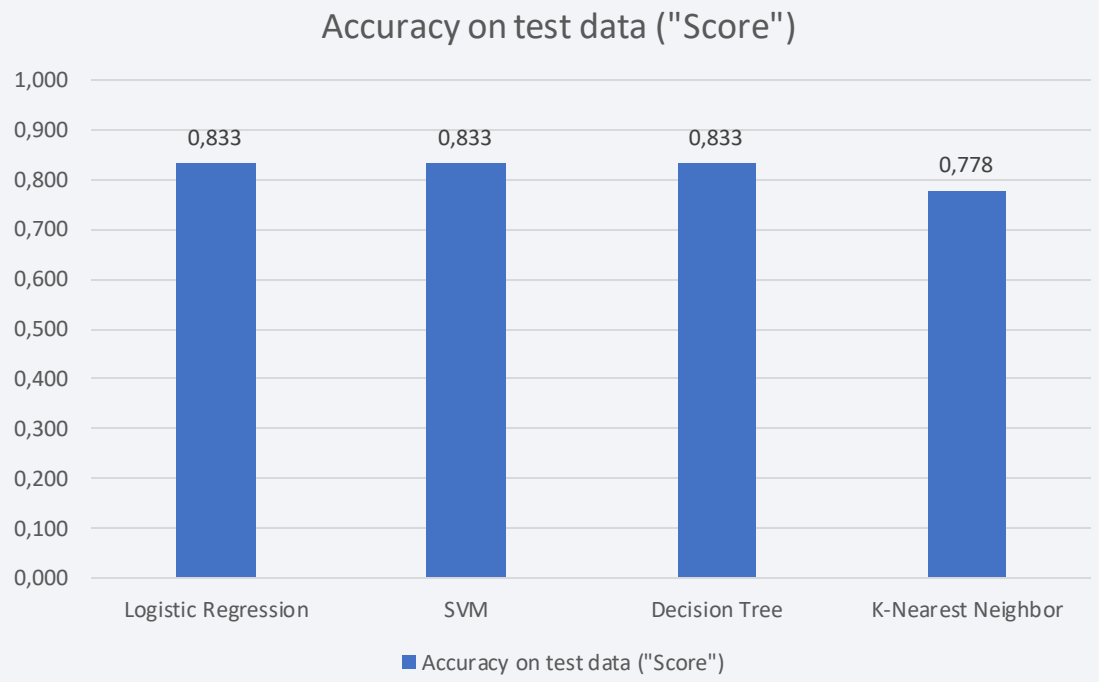
- Zooming in on payload, it becomes evident that there are more successful than unsuccessful launches when the payload is between 2k and 4k.
- Particularly unsuccessful are the payload ranges below 2k and above 6k (see first graph).

- Over the complete payload spectrum, the booster version "FT" is the most successful (see graphs left and above), followed by B4. The versions "v1.0" and "v1.1" have almost completely unsuccessful outcomes.

Section 5

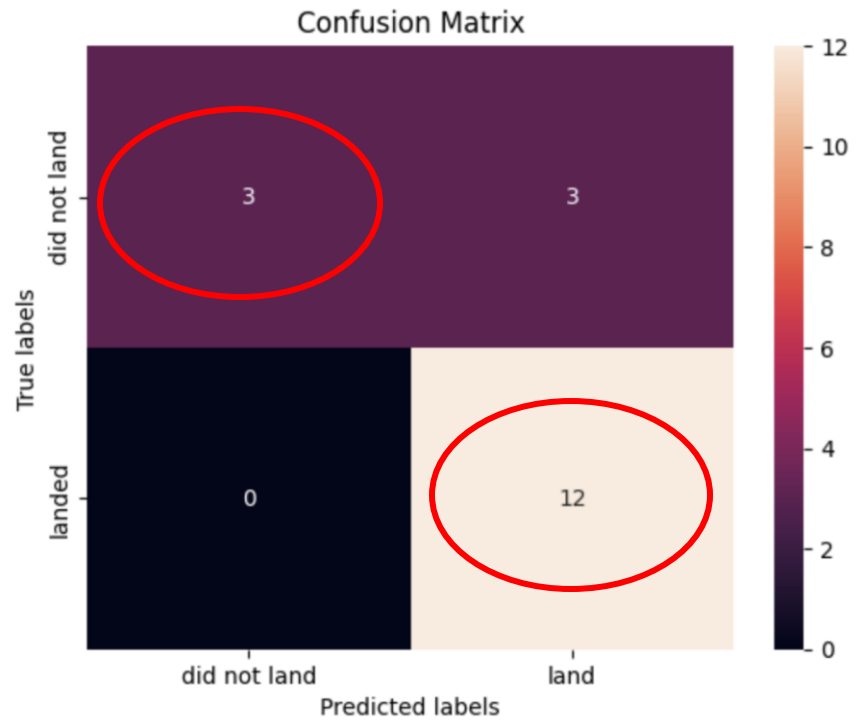
Predictive Analysis (Classification)

Classification Accuracy



- Using the "Score" method to assess classification accuracy of the different models on test data, Logistic Regression, SVM and Decision Tree show the exact same accuracy of 0.833

Confusion Matrix for best performing models



- The Confusion Matrix is a 2x2 matrix that compares the predicted outcomes (x-axis) and the actual outcomes (y-axis). Ultimate goal is to have a perfect match between predicted and actual outcomes.
- The Confusion Matrices for the best performing models (Logistic Regression, SVM, Decision Tree) show the exact same pattern:
  - 100% (i.e., 12 out of 12) successful landings were predicted correctly
  - 50% (3 out of 6) unsuccessful landings were predicted correctly

Conclusions

- Explorative data analyses helped to determine variables that are useful to predict the launch outcome (failure vs success). Among these variables are:
  - Launch sites
  - Orbit type
  - Payload Mass
- Applying different Machine Learning Methods help to choose the best model to predict launch success
- In our case, three of the four applied methods (i.e., Logistic Regression, Support Vector Machines, Decision Tree Classifier) yielded identical results with an accuracy score of 83.3% correct predictions
- This underlines that it is possible to make valid predictions for a technical complex activity such as rocket launches based on publicly available data and established data science methods

Appendix

- Additional sources used / consulted:
  - Al Sweigart: Automate the Boring Stuff with Python (2nd edition)
  - Official documentation for Plotly Dash: https://plotly.com/dash/

Thank you!