IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Daniella Santos
Nov 15, 2021

Outline

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

Executive Summary

The Capstone project for this Data Science course consists of analyzing and predicting whether the first stage of a Falcon9 rocket will land successfully so that the final cost of a launch can be properly estimated as the rocket costs are the most significant in any launch.

In order to accomplish that we collected data about past SpaceX launches utilizing the SpaceX API and webscraping from a table in Wikipedia. Then we proceeded to explore the collected data using data wrangling, advanced visualization techniques and SQL. In order to further the data analysis we also utilized Interactive Maps with Folium and dashboards with Plotly Dash.

Finally, we built predictive models utilizing four different techniques: K-nearest neighbors, Support Vector Machine, Decision Tree and Logistic Regression. Although all four models produced the same results, the Decision Tree model had the highest level of accuracy of all the models.

Introduction

With the race to space being a highly sought-after accomplishment by several companies today, SpaceX stands out as the leader in commercial space transportation. One of the reasons they are so successful rests in the fact that most of their rocket can be re-used several times, therefore lowering the costs of each launch, making them cheaper than traditional rockets technologies.

Our company, SpaceY, would like to enter in the commercial space travel market utilizing similar technology (re-usable first stage) and for that we need to predict if and at which rate the rocket's first stage will land successfully so we can determine the cost of each launch.

Section 1

Methodology

Methodology

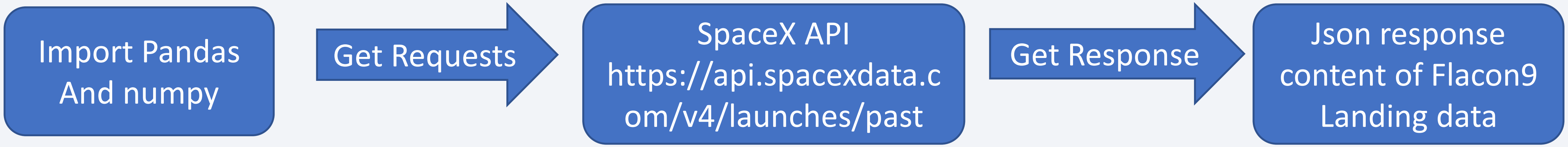
Executive Summary

- Data collection methodology:
  - SpaceX Launch data was collected utilizing the SpaceX REST API
  - Falcon9 launch data was collected utilizing Web Scraping methods (with BeautifulSoup)
- Perform data wrangling
  - Performed dataset filtering, one-hot encoding and replaced missing values as needed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Built and evaluated 4 predictive models: K-nearest neighbors, Decision Tree, Support Vector Machine and Logistic Regression.

Data Collection

Two data sets were collected as described below:

- SpaceX Falcon 9 first stage Landing data was collected utilizing the SpaceX API in conjunction with pandas and numpy libraries of functions to collect, clean and shape the data.



Import Pandas And numpy

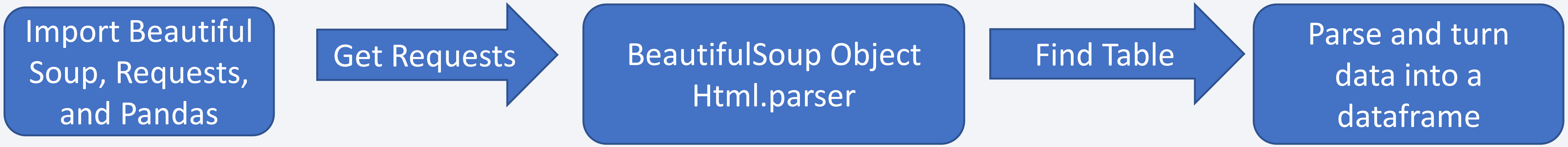
Get Requests

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

Get Response

Json response content of Flacon9 Landing data

- Falcon 9 and Falcon Heavy Launches Records from Wikipedia (HTML table) dataset was collected utilizing Web Scraping methodology with the BeautifulSoup library.



Import Beautiful Soup, Requests, and Pandas

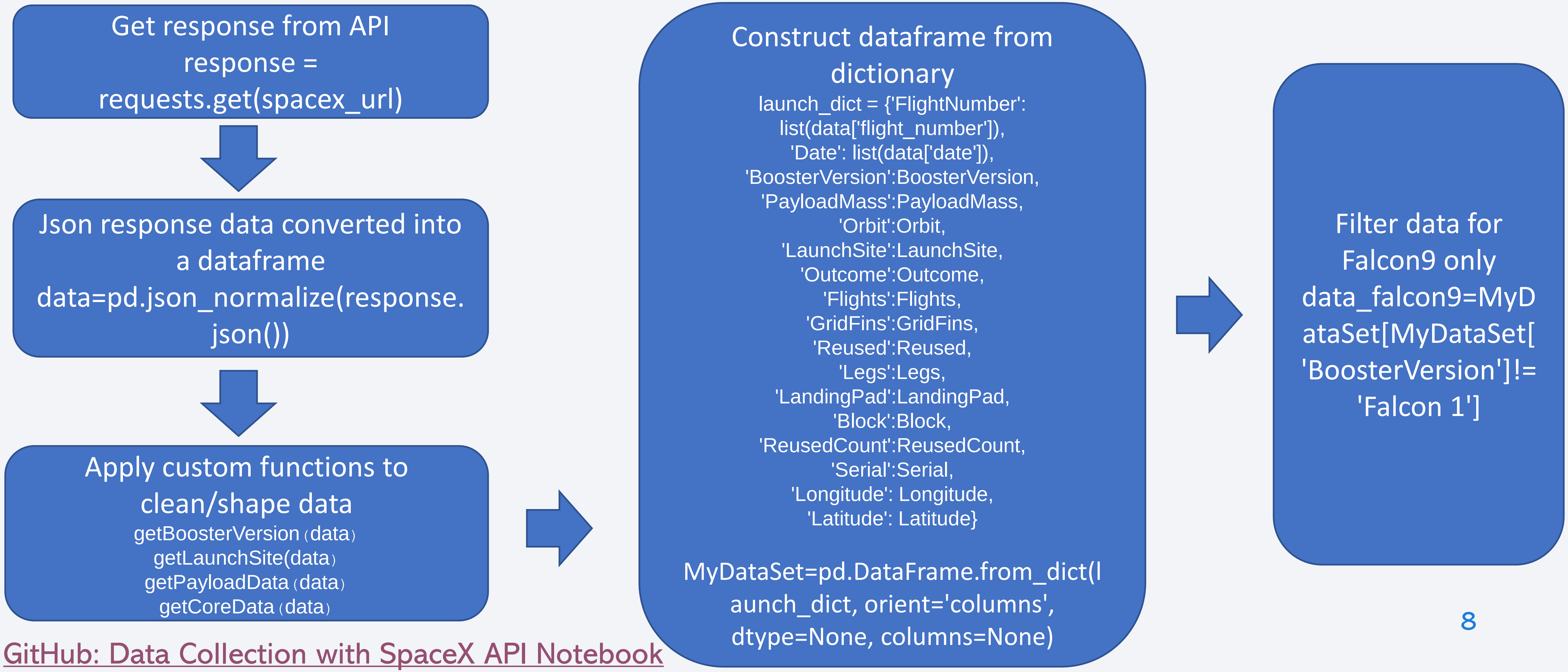
Get Requests

BeautifulSoup Object Html.parser

Find Table

Parse and turn data into a dataframe

Data Collection – SpaceX API



Get response from API
response = requests.get(spacex_url)

Json response data converted into a dataframe
data=pd.json_normalize(response.json())

Apply custom functions to clean/shape data
getBoosterVersion (data)
getLaunchSite(data)
getPayloadData (data)
getCoreData (data)

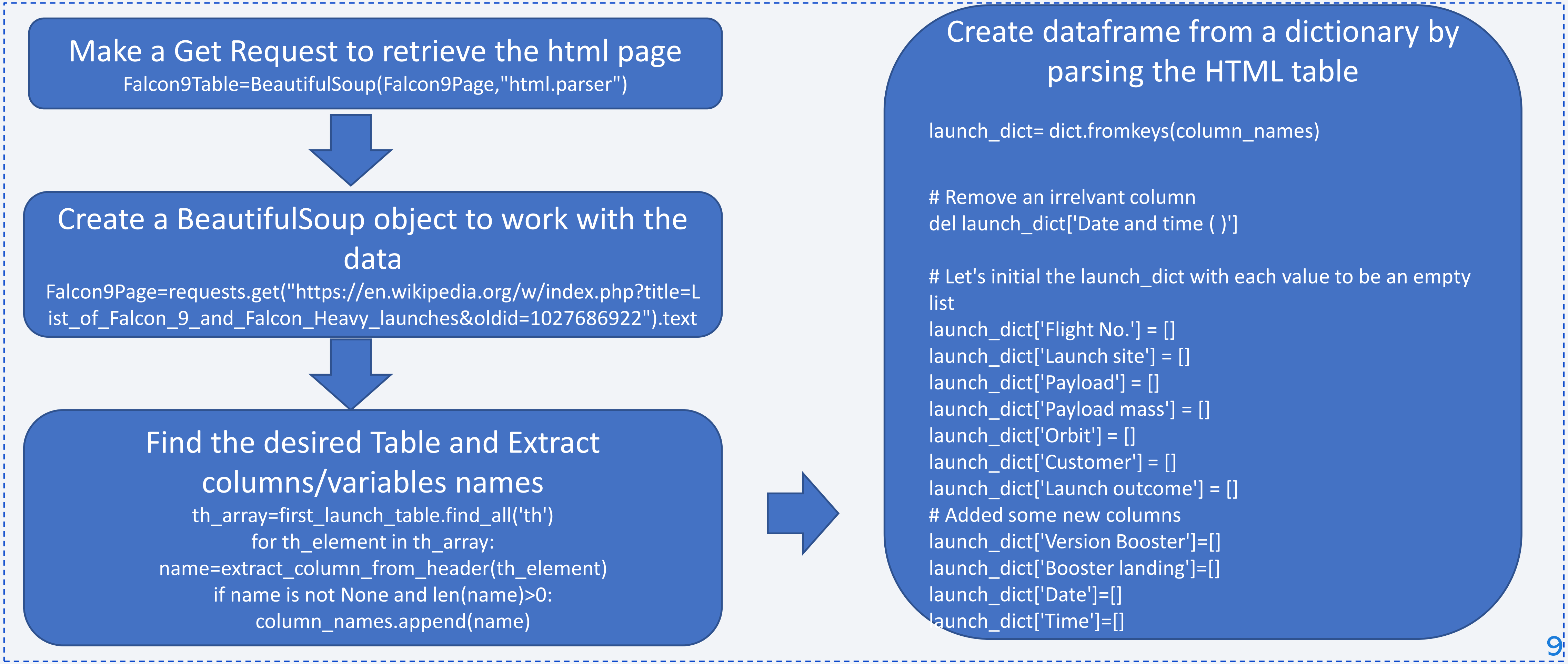
Construct dataframe from dictionary

```
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}

MyDataSet=pd.DataFrame.from_dict(launch_dict, orient='columns', dtype=None, columns=None)
```

Filter data for Falcon9 only
data_falcon9=MyDataSet[MyDataSet['BoosterVersion']!='Falcon 1']

GitHub: Data Collection with SpaceX API Notebook

Data Collection - Scraping



Make a Get Request to retrieve the html page

```
Falcon9Table=BeautifulSoup(Falcon9Page,"html.parser")
```

Create a BeautifulSoup object to work with the data

```
Falcon9Page=requests.get("https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922").text
```

Find the desired Table and Extract columns/variables names

```
th_array=first_launch_table.find_all('th')
for th_element in th_array:
name=extract_column_from_header(th_element)
if name is not None and len(name)>0:
column_names.append(name)
```

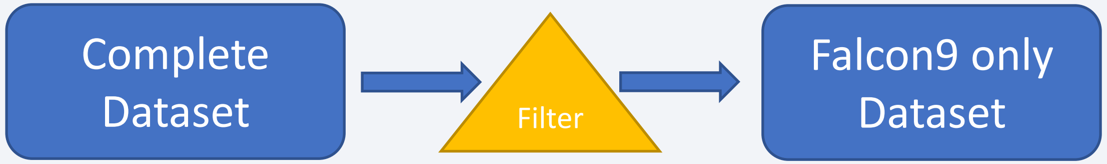
Create dataframe from a dictionary by parsing the HTML table

```
launch_dict= dict.fromkeys(column_names)


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

# Let's initial the launch_dict with each value to be an empty list
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']=[]
```

GitHub: Data Collection with Web Scraping Notebook

Data Wrangling

- After data was collected we filtered the dataset to show only Falcon9 related data
- Performed one-hot encoding
- Replaced Payload_Mass missing values with the Mean



Complete Dataset

Filter

Falcon9 only Dataset

GitHub URL: Data Wrangling Notebook

EDA with Data Visualization

To draw insights from the data collected we created the following charts:

- Scatter Plot:
  - To see how the FlightNumber (number of attempts) and Payload variables would affect the launch outcome.
  - To visualize the relationship between Flight Number and Launch Site
  - To visualize the relationship between Payload and Launch Site
  - To visualize the relationship between Flight Number and Orbit Type
  - To visualize the relationship between Payload and Orbit Type

- Bar Chart
  - To visualize the relationship between Success Rate and Orbit Type

- Line Chart
  - To visualize the Launch Success yearly trend

GitHub: EDA with Data Visualization Notebook

EDA with SQL

We have performed the following SQL queries to gain insight into the SpaceX data:

- Find and display the unique Launch Site names
- 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

GitHub: EDA with SQL Notebook

Build an Interactive Map with Folium

We would like to have a visual representation of the Launch Sites and their characteristics on a map and for that we used Folium to:

- Mark all Launch Sites on a Map utilizing “Markers” and “Circles”
- Visualize all success/failed launches for each site on a map utilizing “Markers” and “Marker Clusters”
- Calculate and mark distances between a Launch Site and its proximities utilizing “Mouse Position”, “Markers”, the custom function “calculate_distance” and PolyLine

GitHub: Interactive Visual Analytics with Folium

Build a Dashboard with Plotly Dash

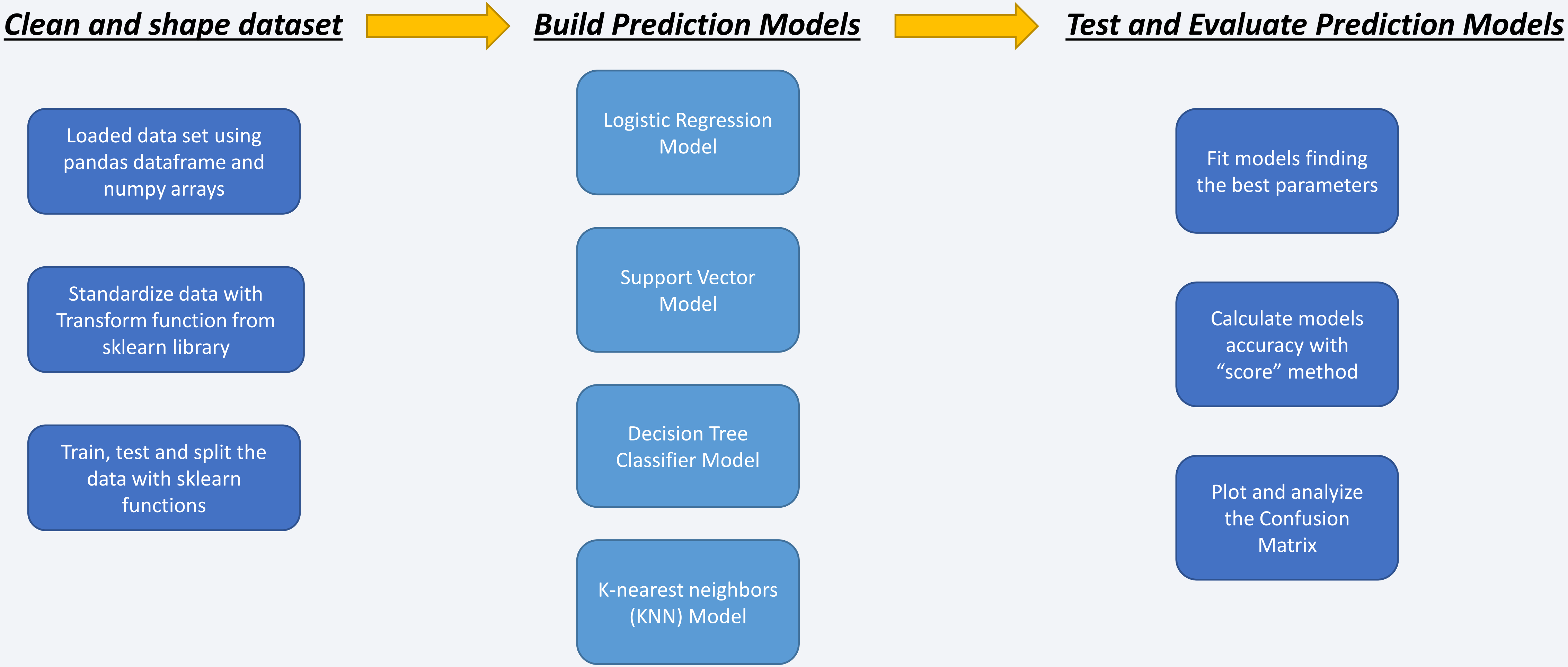
We built a dashboard to visualize and gain insights as to the Success Rates of all Launch Sites

In the dashboard we utilized Pie Charts and Scatter Plots to:

- Visualize Success Launches by Site and All Sites and answer the following questions:
  - Which site has the largest successful launches?
  - Which site has the highest launch success rate?
- Find and display the correlation between Payload Mass and Success Rate by Site and All Sites, by Booster Version
  - Which payload range(s) has the highest launch success rate?
  - Which payload range(s) has the lowest launch success rate?
  - Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate?

GitHub: Plotly Dash Dashboard python file

Predictive Analysis (Classification)



Clean and shape dataset

- Loaded data set using pandas dataframe and numpy arrays
- Standardize data with Transform function from sklearn library
- Train, test and split the data with sklearn functions

Build Prediction Models

- Logistic Regression Model
- Support Vector Model
- Decision Tree Classifier Model
- K-nearest neighbors (KNN) Model

Test and Evaluate Prediction Models

- Fit models finding the best parameters
- Calculate models accuracy with “score” method
- Plot and analyze the Confusion Matrix

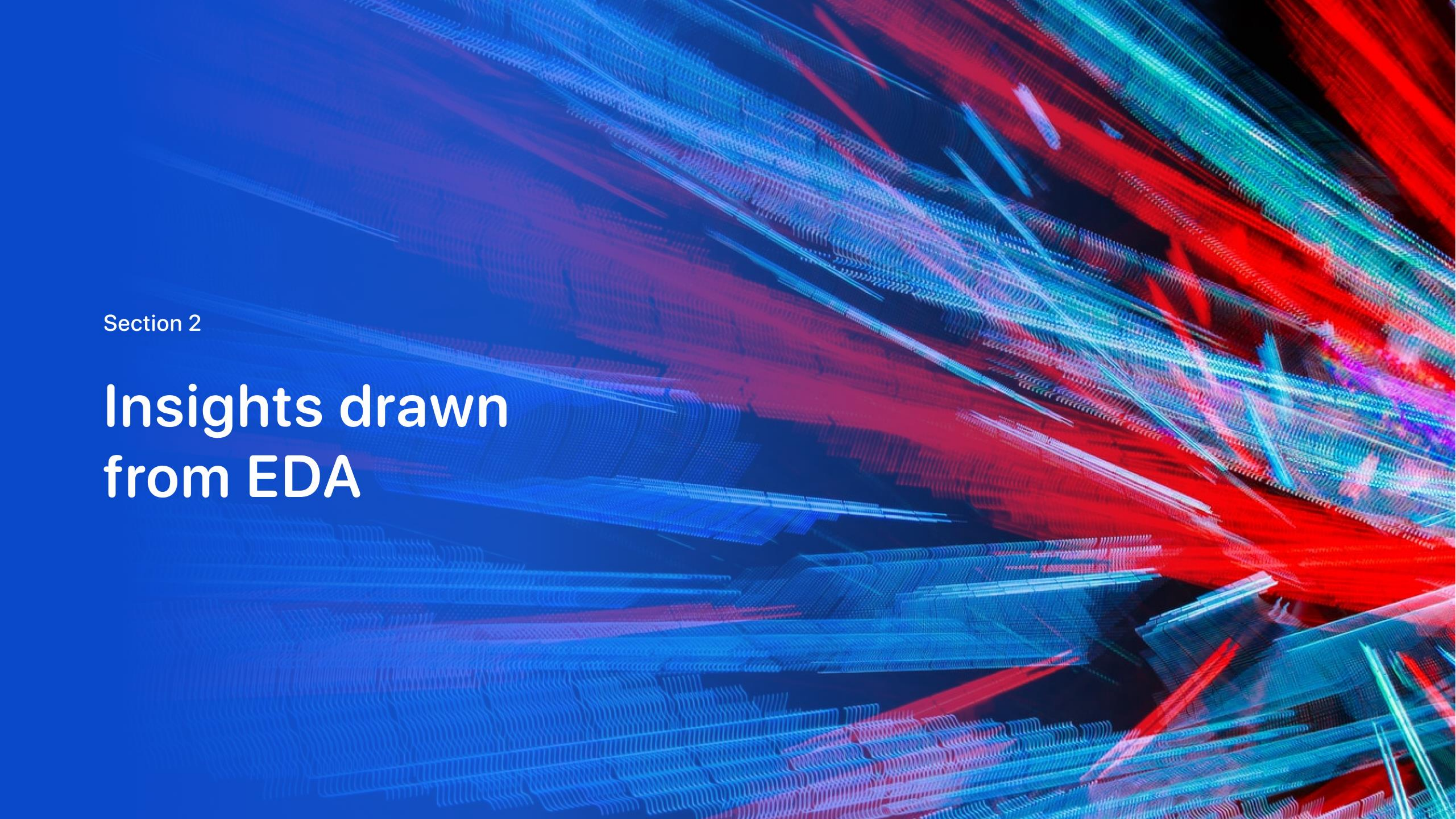
GitHub: Machine Learning Prediction Notebook

Results

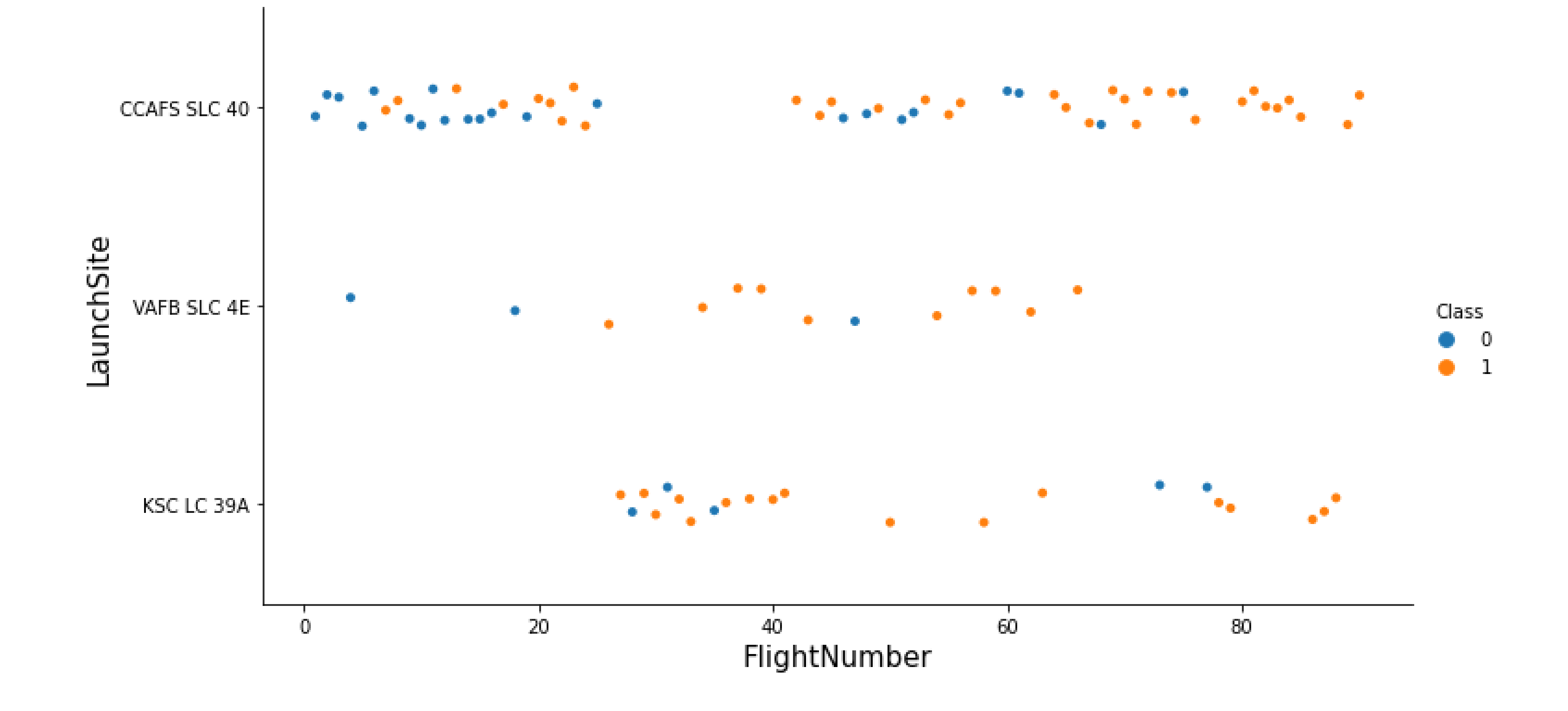
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

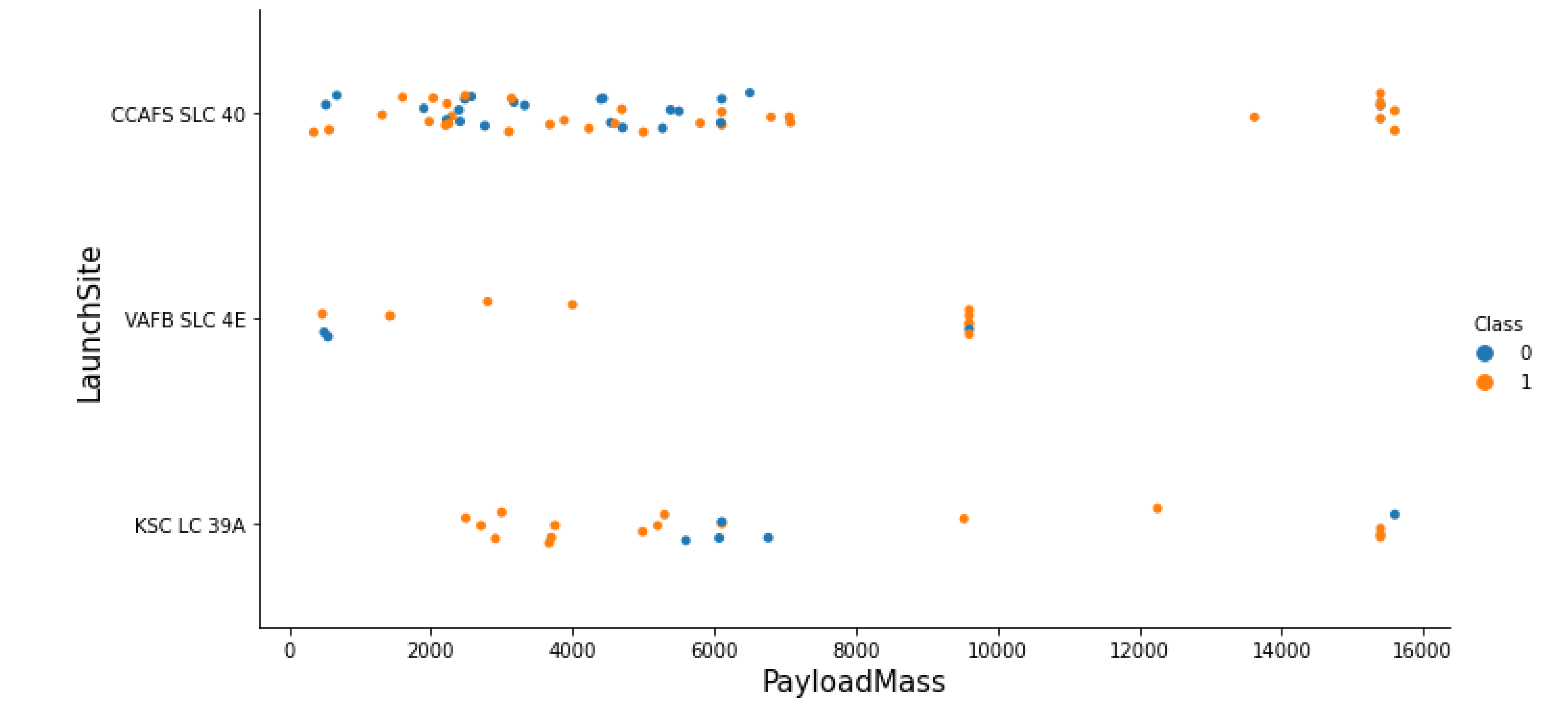


Flight Number vs. Launch Site



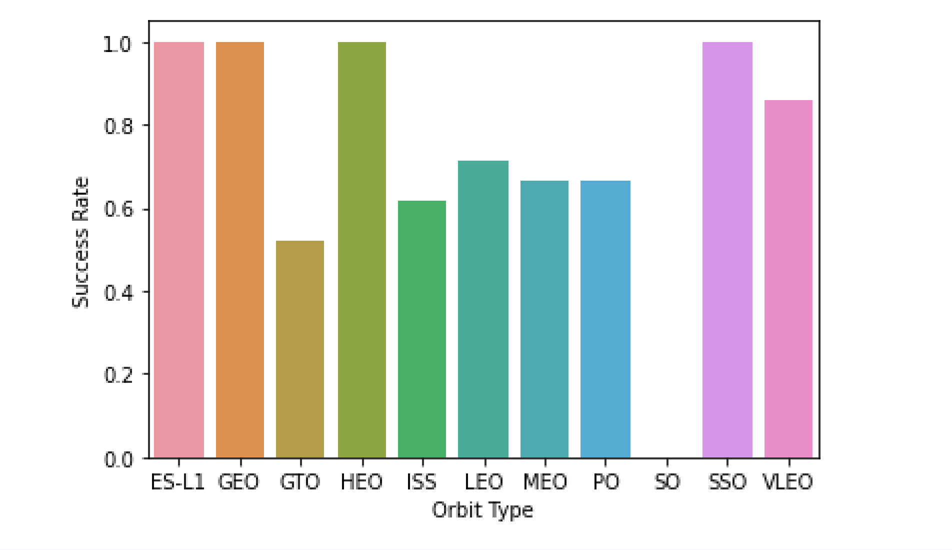
- The majority of earlier flights were unsuccessful
- As experience and testing progressed we can see that after the first ~20 flights the success rate increased significantly
- Most flights are successful regardless of the Launch Site

Payload vs. Launch Site



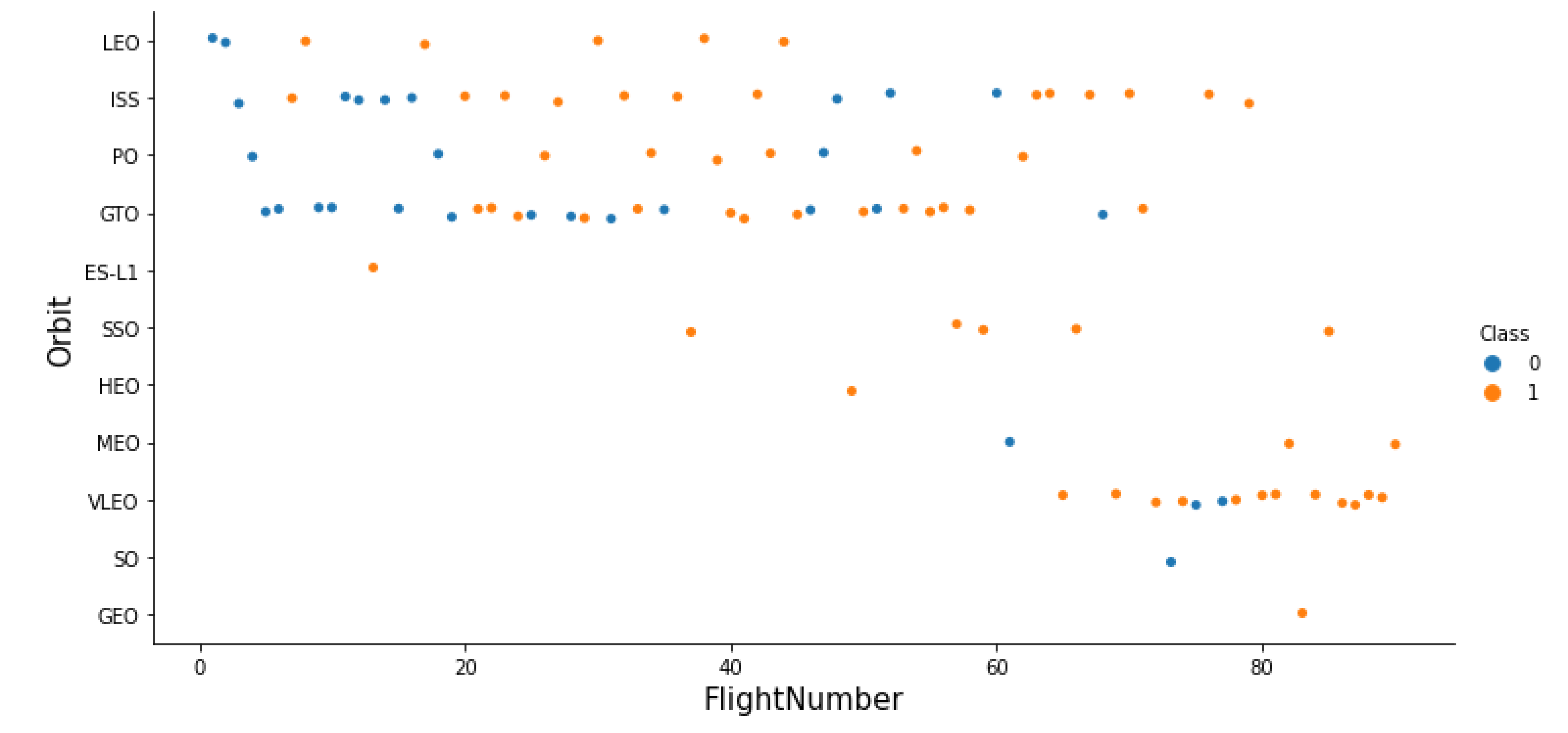
- The majority of flights with Payload between 0 kg and 8000 kg have been successful
- There is not enough data to draft insights as to the success rate and payload mass between 8000kg and 14000 kg
- Heavier payload flights have a high success rate
- There is no indication that success rate is tied to specific Launch Sites. They all show the same level of success rate.
- Flights heavier than 10000 kg are not launched from site VAFB-SLC 4E

Success Rate vs. Orbit Type



- Orbits: ES-L1, GEO, HEO and SSO have 100% success rate

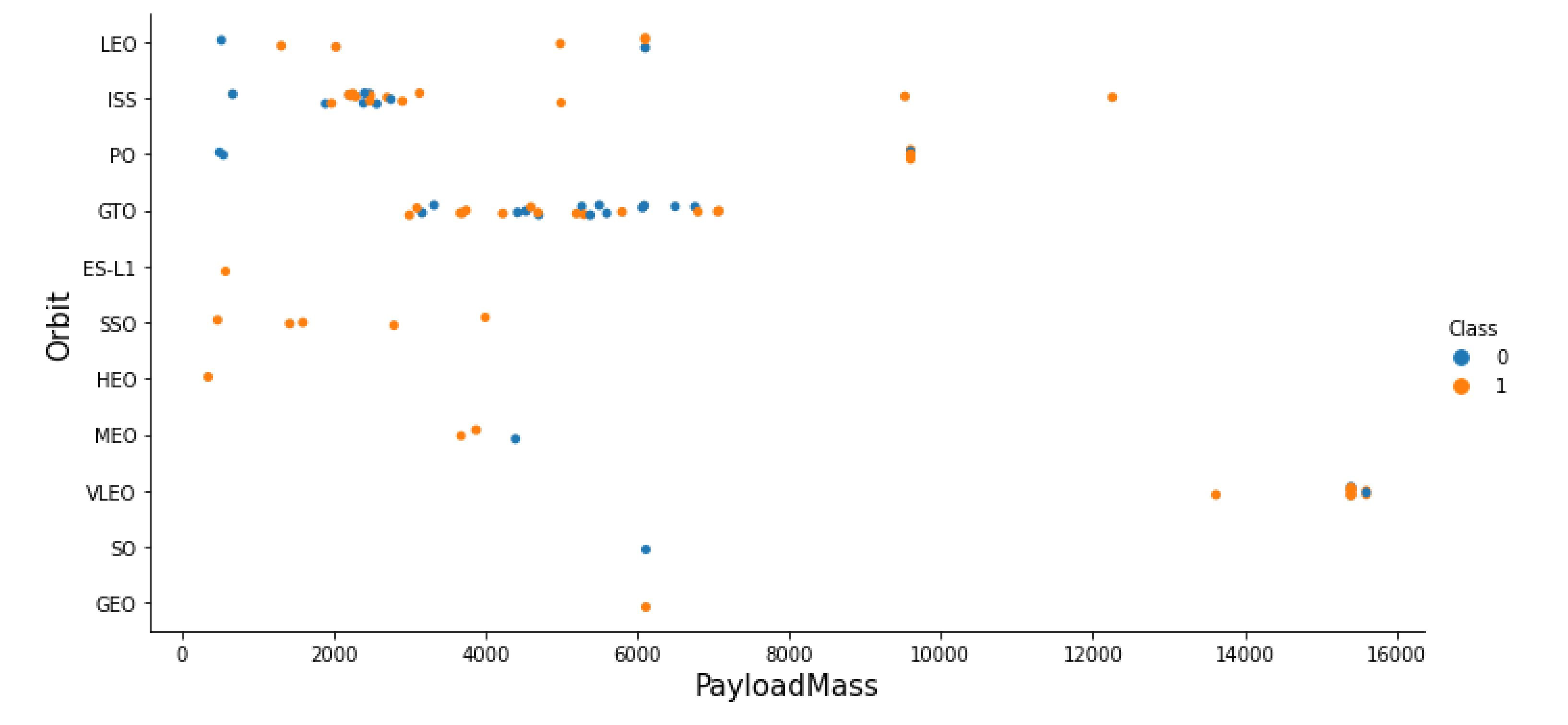
Flight Number vs. Orbit Type



- Flights to LEO orbits have shown a relationship between number of flights and the increase in success rate

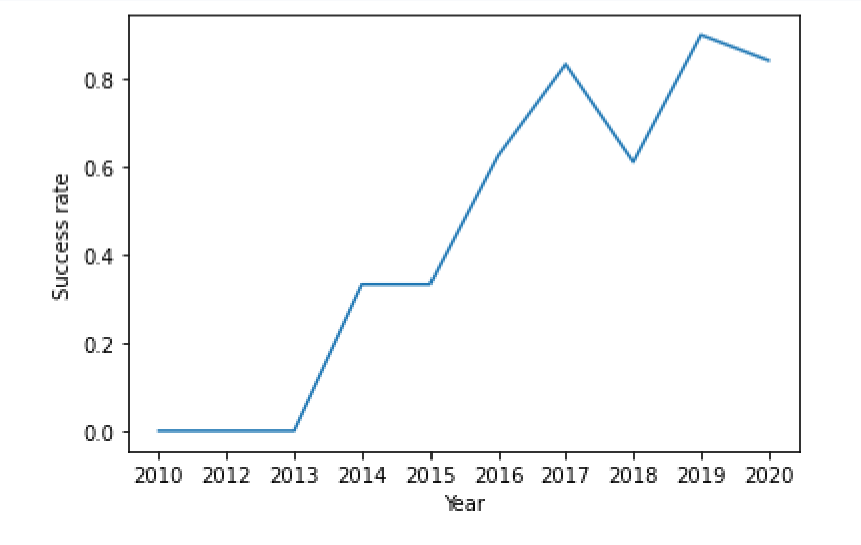
- Flights to GTO have not shown much increase in success rate with the number of flights

Payload vs. Orbit Type



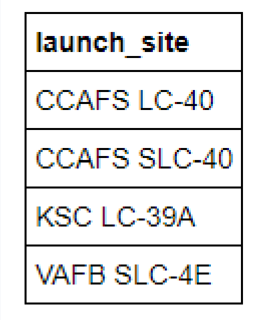
- Heavier loads have high success rates in the Orbits where they have been launched (ISS, PO, VLEO)
- For the GTO orbit there is no clear pattern as to a relationship between PayloadMass and success rate

Launch Success Yearly Trend



- The success rate started increasing constantly and rapidly since 2013

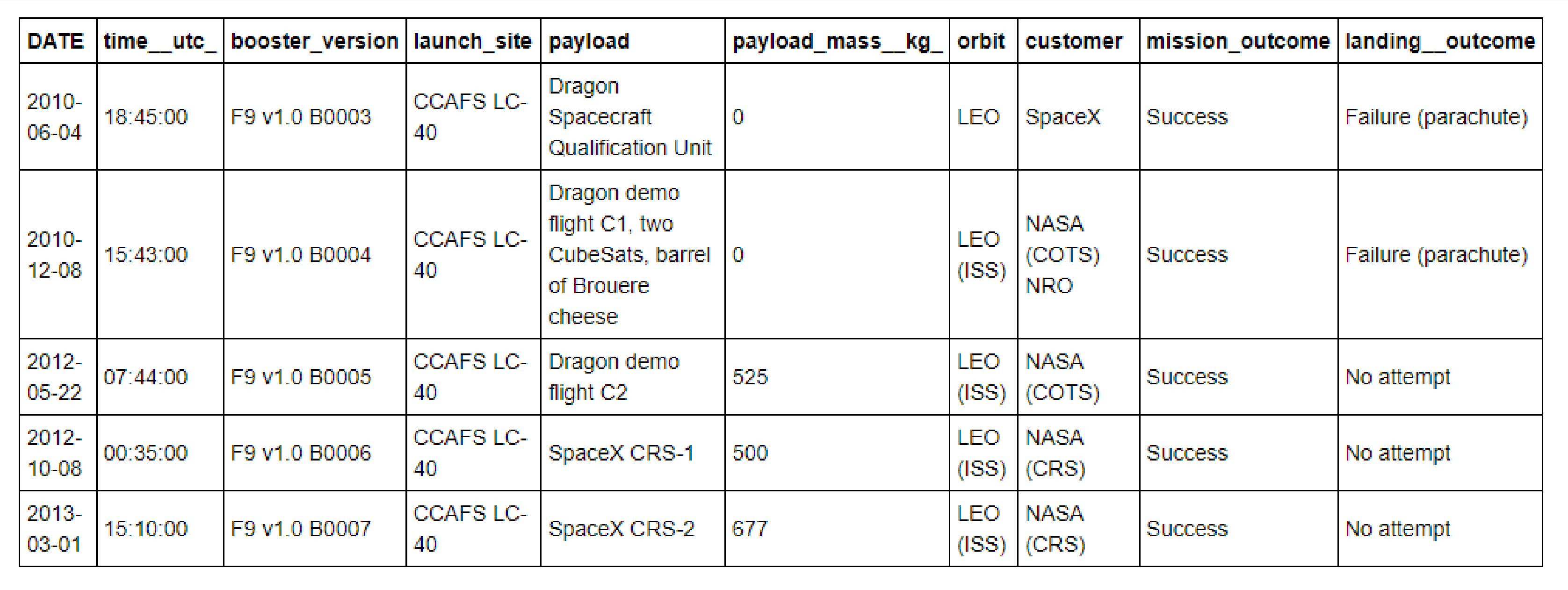
All Launch Site Names



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

- SQL query: %sql select DISTINCT LAUNCH_SITE from SPACEXTBL
- The DISTINCT function helps return only unique names from the specified field/column. In this case : Launch_Site.

Launch Site Names Begin with 'CCA'



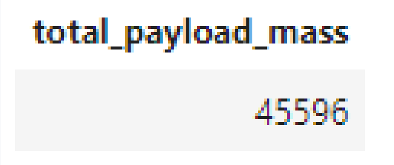
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

- SQL Query: %sql select * from SPACEXTBL where LAUNCH_SITE Like 'CCA%' LIMIT 5

- Utilizing the "Like" function and "%" wildcard character allows the query to find only records that start with the string "CCA".

- The LIMIT function allows us to limit the query result to a determined number of records (in this case 5)

Total Payload Mass

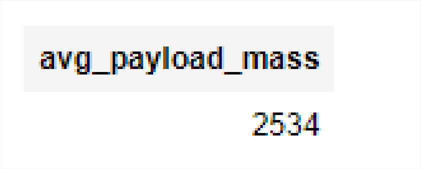


total_payload_mass
45596

- SQL query: %sql select sum(PAYLOAD_MASS__KG_) as Total_Payload_Mass from SPACEXTBL where CUSTOMER = 'NASA (CRS)’;

- SUM aggregates the values of all records for Customer “NASA (CRS)” (with WHERE filter).

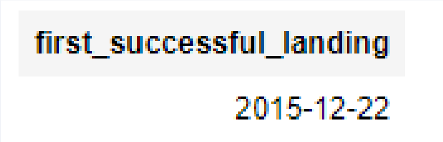
Average Payload Mass by F9 v1.1



avg_payload_mass
2534

- SQL Query: %sql select AVG(PAYLOAD_MASS__KG_) as Avg_Payload_Mass from SPACEXTBL where booster_version Like 'F9 v1.1%’
- The AVG function calculates the Average Value of Payload Mass for all records of Booster “F9 v1.1” – the filtering of data is done with the “Like” function and wildcard “%” to pick up all records that start with string “F9 v1.1”

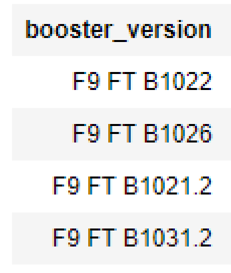
First Successful Ground Landing Date



first_successful_landing
2015-12-22

- SQL query:%sql select MIN(DATE) as First_Successful_Landing from SPACEXTBL where landing__outcome ='Success (ground pad)’
- Find the earliest Date of a successful landing using the MIN function to find that date and the “where” function to filter data landing outcomes with value of “Success (ground pad)”

Successful Drone Ship Landing with Payload between 4000 and 6000

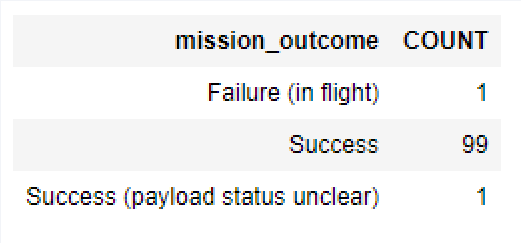


booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- %sql select booster_version,payload,PAYLOAD_MASS__KG_ from SPACEXTBL where landing__outcome ='Success (drone ship)' and payload_mass__kg_ between 4000 and 6000

- The “between” function will allow for data filtering and returning results between 2 specified values from all records that have a landing outcome value of “Success (drone ship)

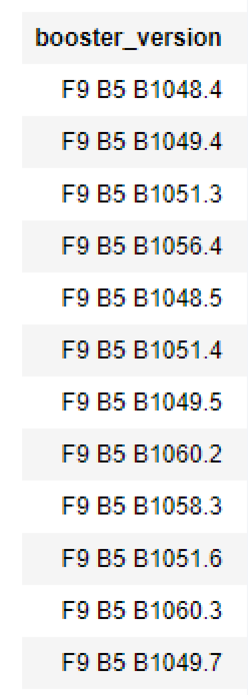
Total Number of Successful and Failure Mission Outcomes



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

- SQL query: %sql select mission_outcome, count(mission_outcome) as count from SPACEXTBL group by mission_outcome
- With the “Count” and “Group By” functions we can retrieve the total number of records of unique mission outcome values grouped by each unique value.

Boosters Carried Maximum Payload

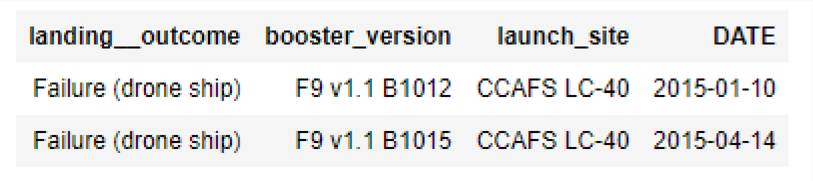


booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- SQL query: %sql select booster_version,payload_mass__kg_ from SPACEXTBL where payload_mass__kg_ = (select MAX(payload_mass__kg_) from SPACEXTBL)

- With the “MAX” function we can retrieve the highest value of Payload Mass and then use that value to filter the data and retrieve only the booster versions that carried that highest value of Payload Mass (with “where” function).

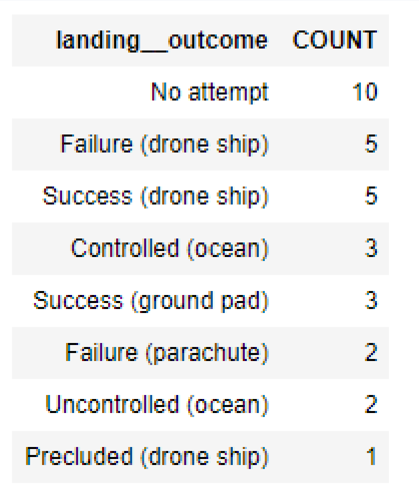
2015 Launch Records



landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

- SQL Query: %sql select landing__outcome, booster_version, launch_site, DATE from SPACEXTBL where landing__outcome Like 'Failure (drone ship)' and DATE Like '2015%’
- The “And” function allows to use 2 conditions to filter the data and return records where landing outcome is “Failure (drone ship)” and Date is in “2015”.

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



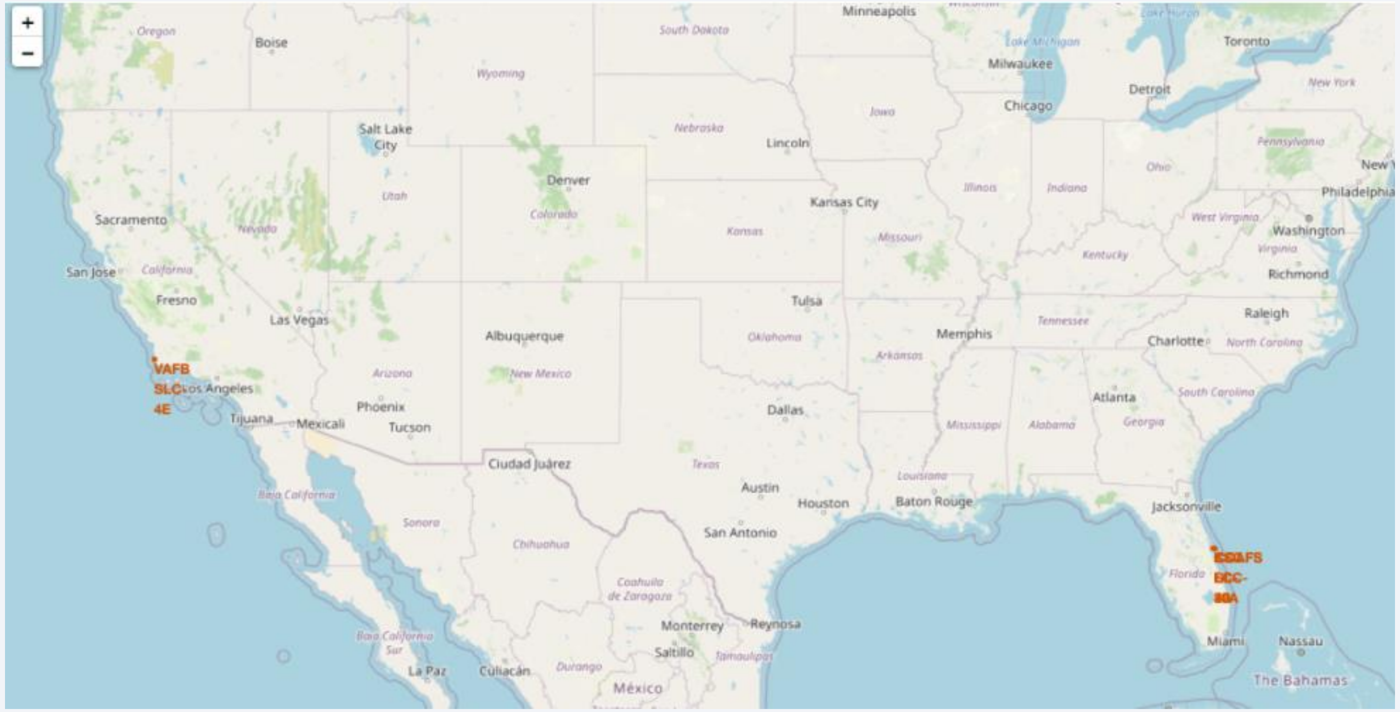
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- %sql select LANDING__OUTCOME, COUNT(*) as Count from SPACEXTBL WHERE (DATE BETWEEN '2010-06-04' AND '2017-03-20')GROUP BY LANDING__OUTCOME ORDER BY COUNT(*) DESC;
- The "Rank" function allows to order the results of the query in ascending or descending order.
- "Count" and "Between" functions allow to retrieve the number of records that have landing outcomes between the 2 specified dates.

Section 4

Launch Sites Proximities Analysis

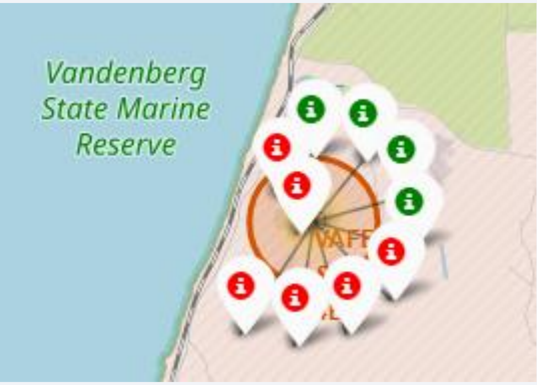
All Launch Sites



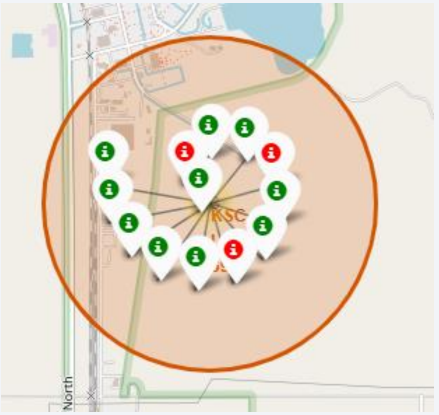
- All Launch sites in the US: California and Florida
- The sites are very close to the coastlines of their states and in relation to the rest of the USA they are closer to the Equator line.

Success and Failure Map – color coded outcomes

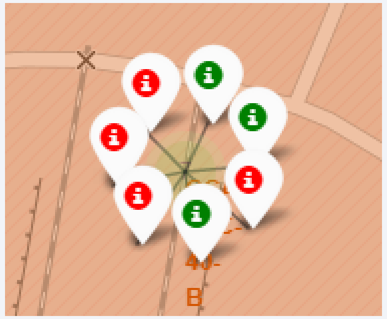
VAFB-SLC 4E (CA)



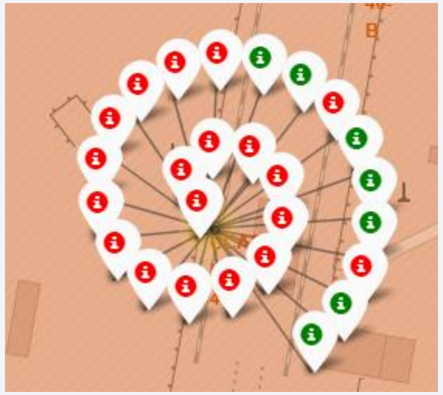
KSC LC-39A (FL)



CCAFS SLC-40-B (FL)

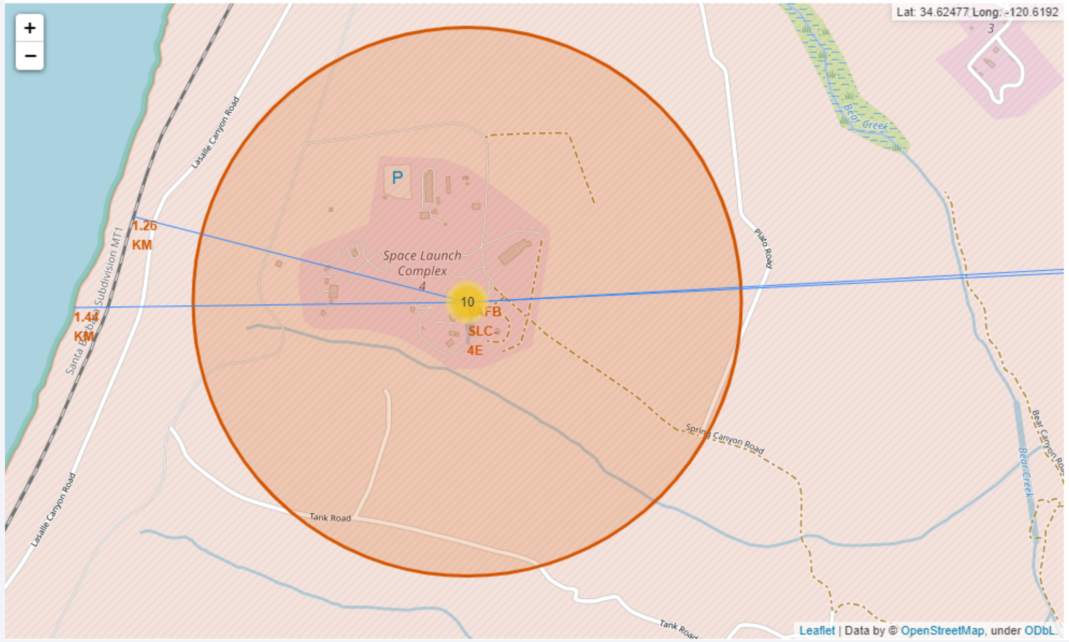


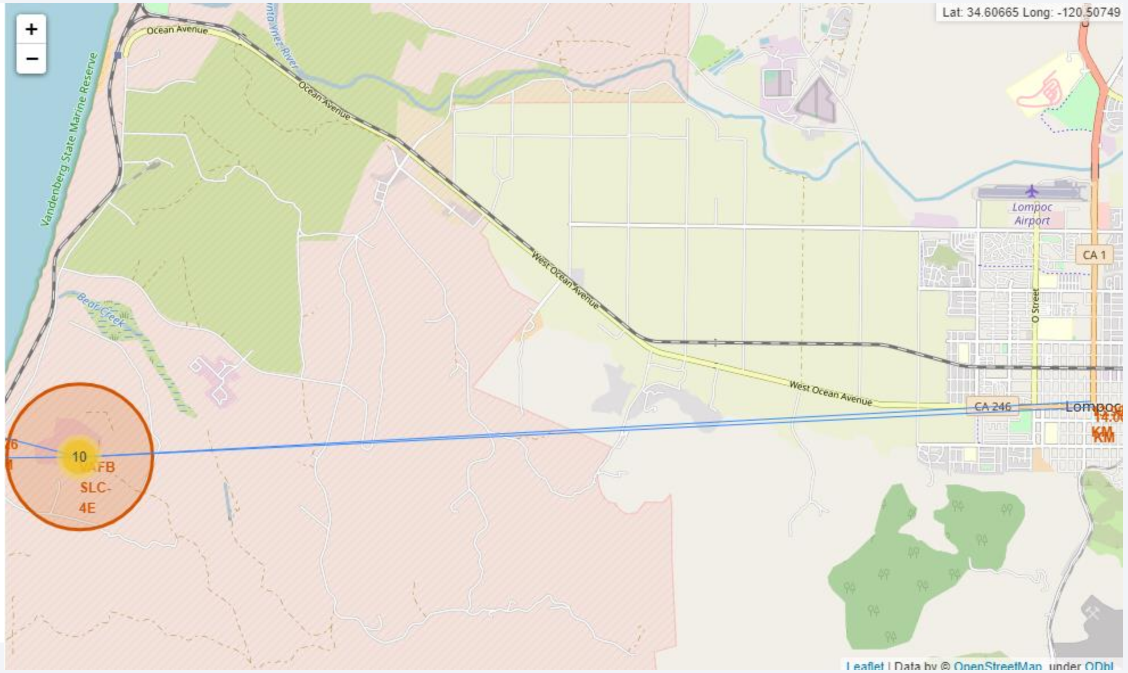
CCAFS SLC-40 (FL)



- GREEN markers depict the Successful Launches and RED markers depict the Failed Launches
- Site KSC LC-39A has the highest success rate of all 4 sites

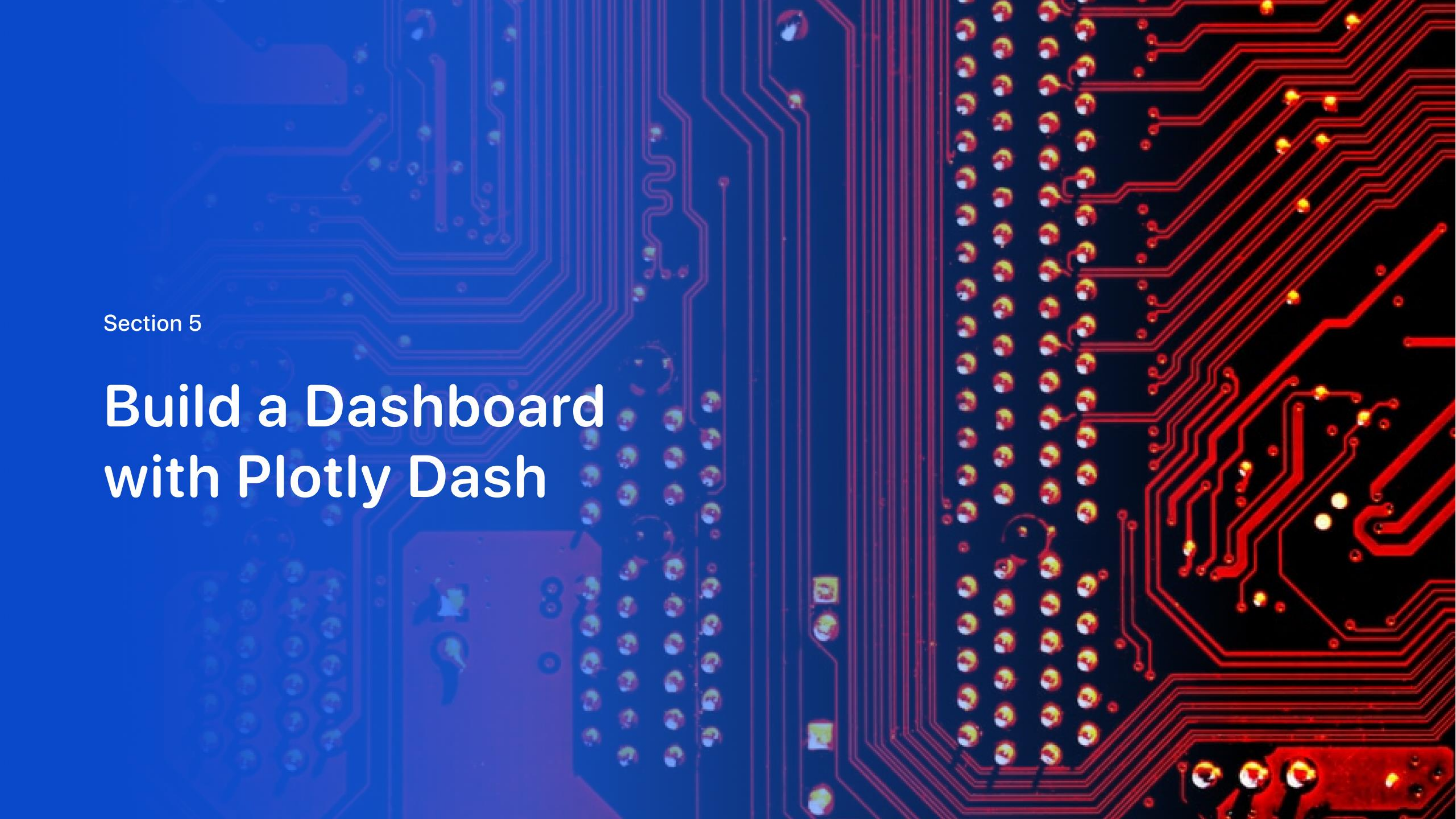
Launch Site distances to its proximities





+
−
Lat: 34.60665 Long: -120.50749
Ocean Avenue
West Ocean Avenue
Vandenberg State Marine Reserve
Bear Creek
Lompoc Airport
CA 1
CA 246
O Street
Lompoc
14.00 KM
10
VAFB SLC-4E
Leaflet | Data by © OpenStreetMap, under ODbL

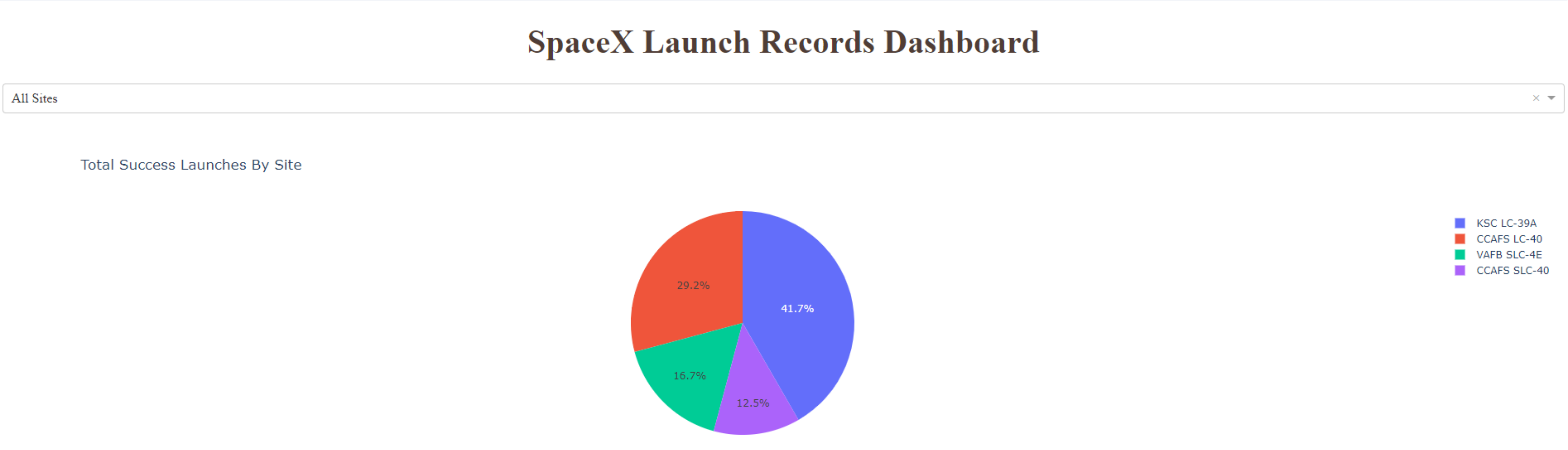
- Site VAFB SLC-4E distance from the following proximities are:
  - Coastline – 1.44Km
  - Railway – 1.26Km
  - Highway: 14 km
  - City: 14 km



Section 5

Build a Dashboard with Plotly Dash

Total Success Launches by Site



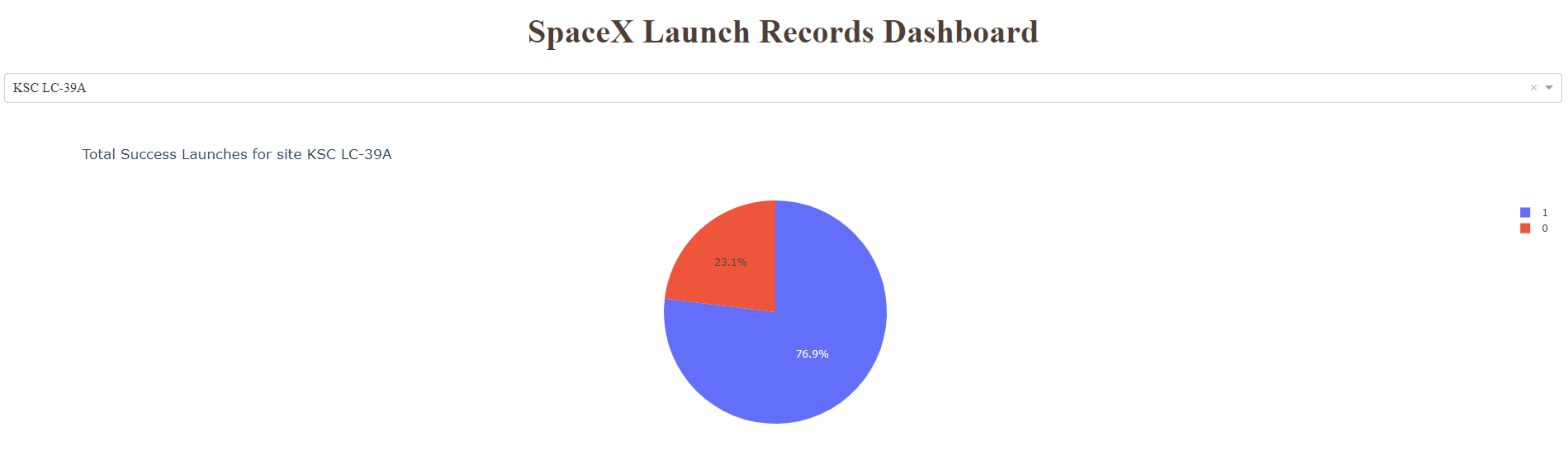
SpaceX Launch Records Dashboard

All Sites

Total Success Launches By Site

- KSC LC-39A (FL) is the site with the most success launches and highest success rate.

Analysis of Launch Site with Highest Success Rate



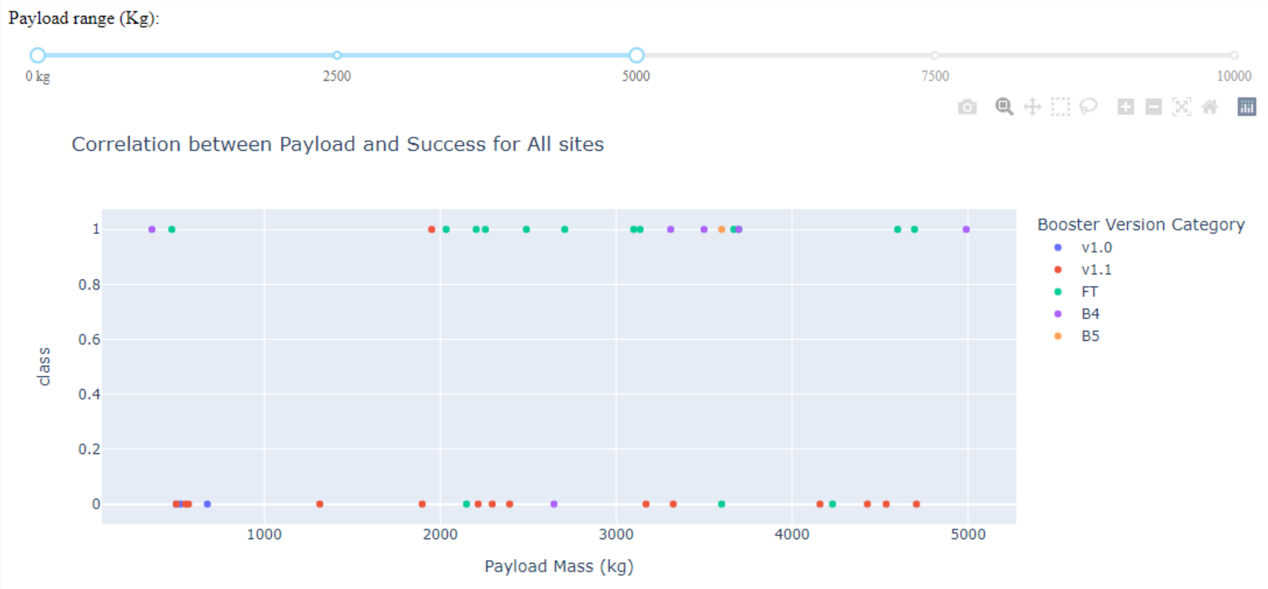
SpaceX Launch Records Dashboard

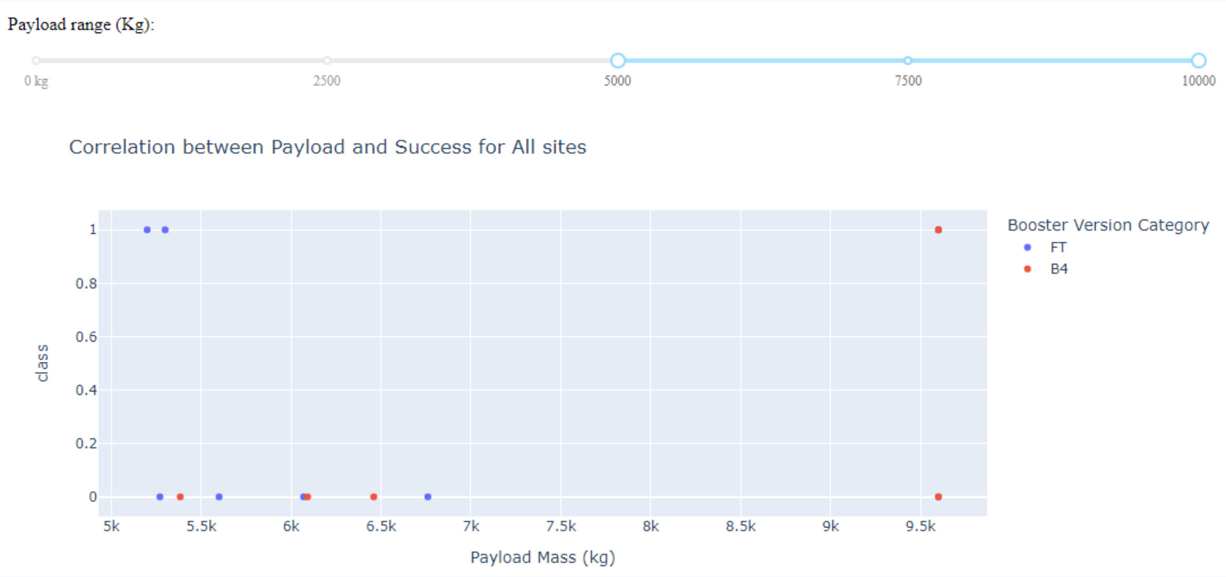
KSC LC-39A

Total Success Launches for site KSC LC-39A

- 76.9% of all launches from KSC LC-39A are successful.

Payload vs Launch Outcome for All Sites



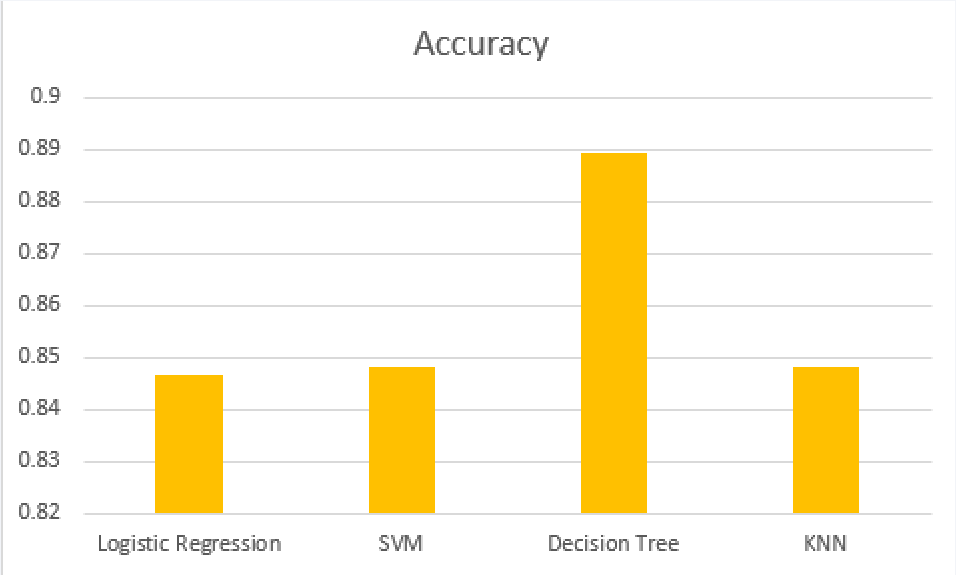


- Lighter payloads (between 0-5000) have higher success rates than the heavier ones
- The FT booster has the higher success rate of all boosters regardless of payload mass

Section 6

Predictive Analysis (Classification)

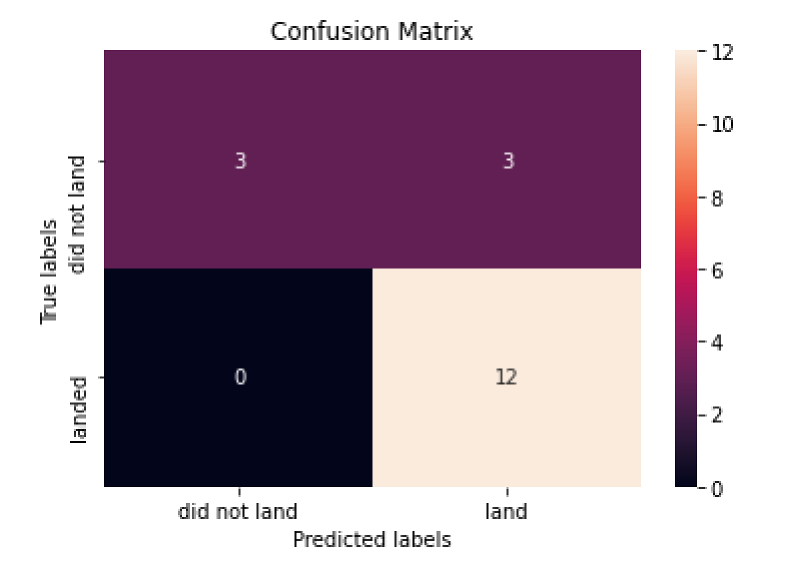
Classification Accuracy



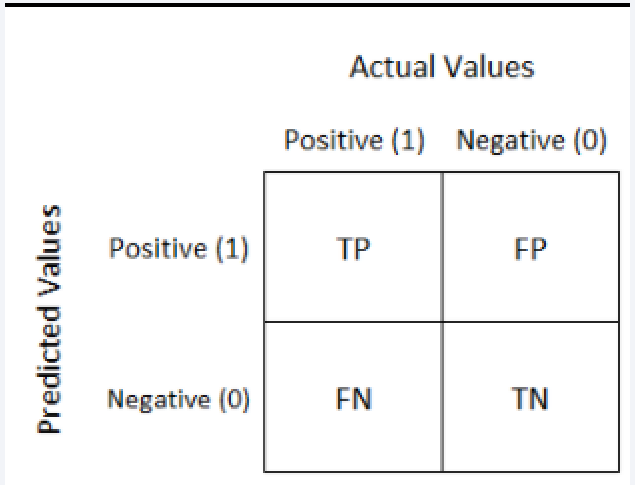
- While all models perform relatively the same the Decision Tree model has a slightly higher Accuracy rate at ~88%

Confusion Matrix

Decision Tree Confusion Matrix



Confusion Matrix Definition



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

- The model performs well, however we can see that it has a problem with the False Positives predictions.

Conclusions

- The best model to predict successful landings is the Decision Tree
- There is a positive correlation between the increased number of launches and increase in success rates.
- Launches with lower Payload Mass have higher success rates
- KSC LC-39A site has the highest success rates among all launch sites
- All launch sites are close to coastlines
- Orbits: ES-L1, GEO, HEO and SSO have 100% success rate

Appendix

- My GitHub project repository

Thank you!