IBM Data Science Capstone – SpaceX

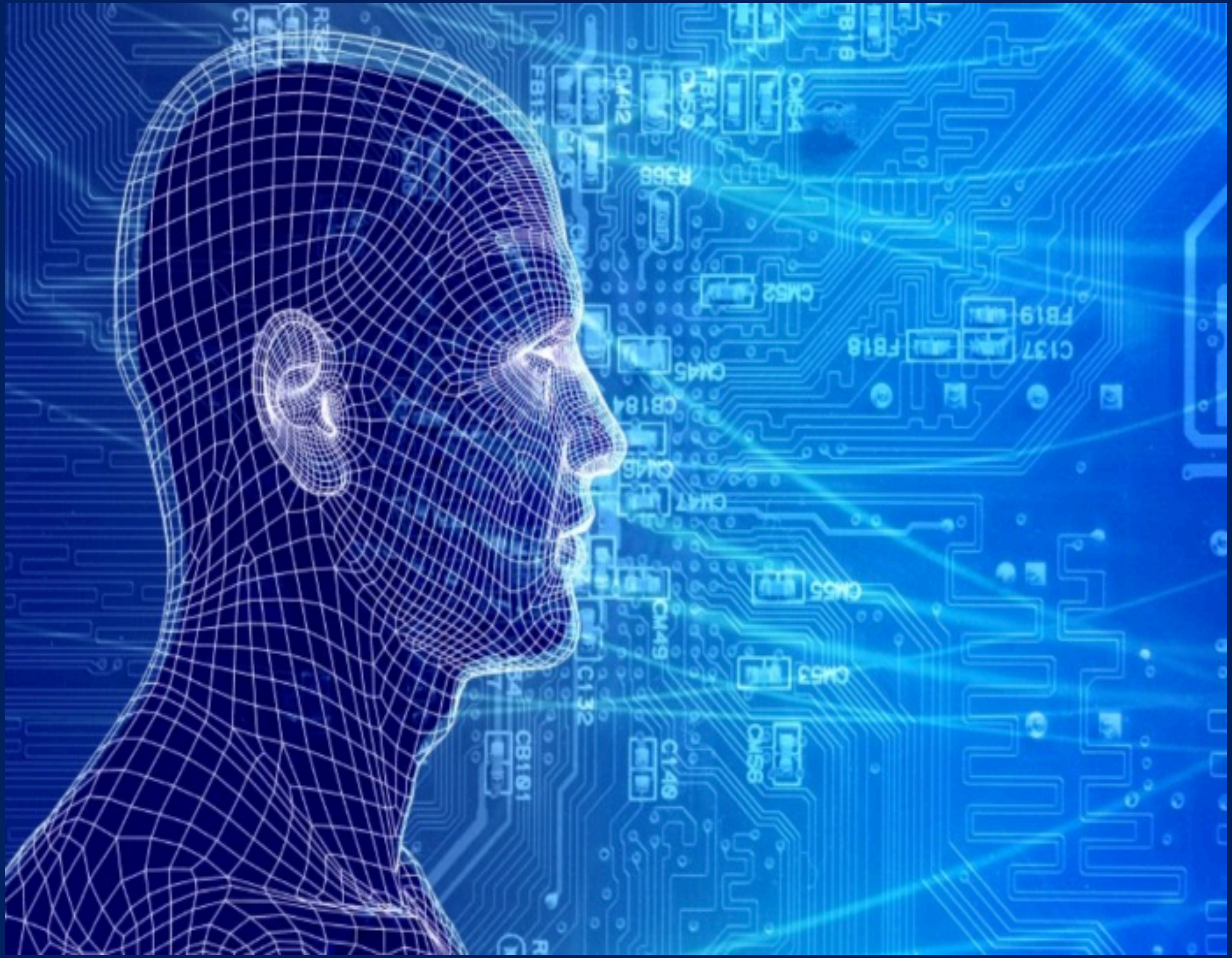
By: Jeff Paynter, MMCI

Date: 01-09-2022

SpaceX - Outline

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

SpaceX - Executive Summary



- **Summary of Methodologies**
  - Data Collection
  - Data Wrangling for Normalization
  - Exploratory Data Analysis (EDA) with
    - *Data Visualization*
    - *SQL*
  - Interactive Map with Folium
  - Dashboard with Plotly Dash
  - Predictive Analysis (Classification)
- **Summary of all Results**
  - Exploratory Data Analysis (EDA) Results
  - Interactive Analytic Demo in Screenshots
  - Predictive Analysis Results

SpaceX - Introduction

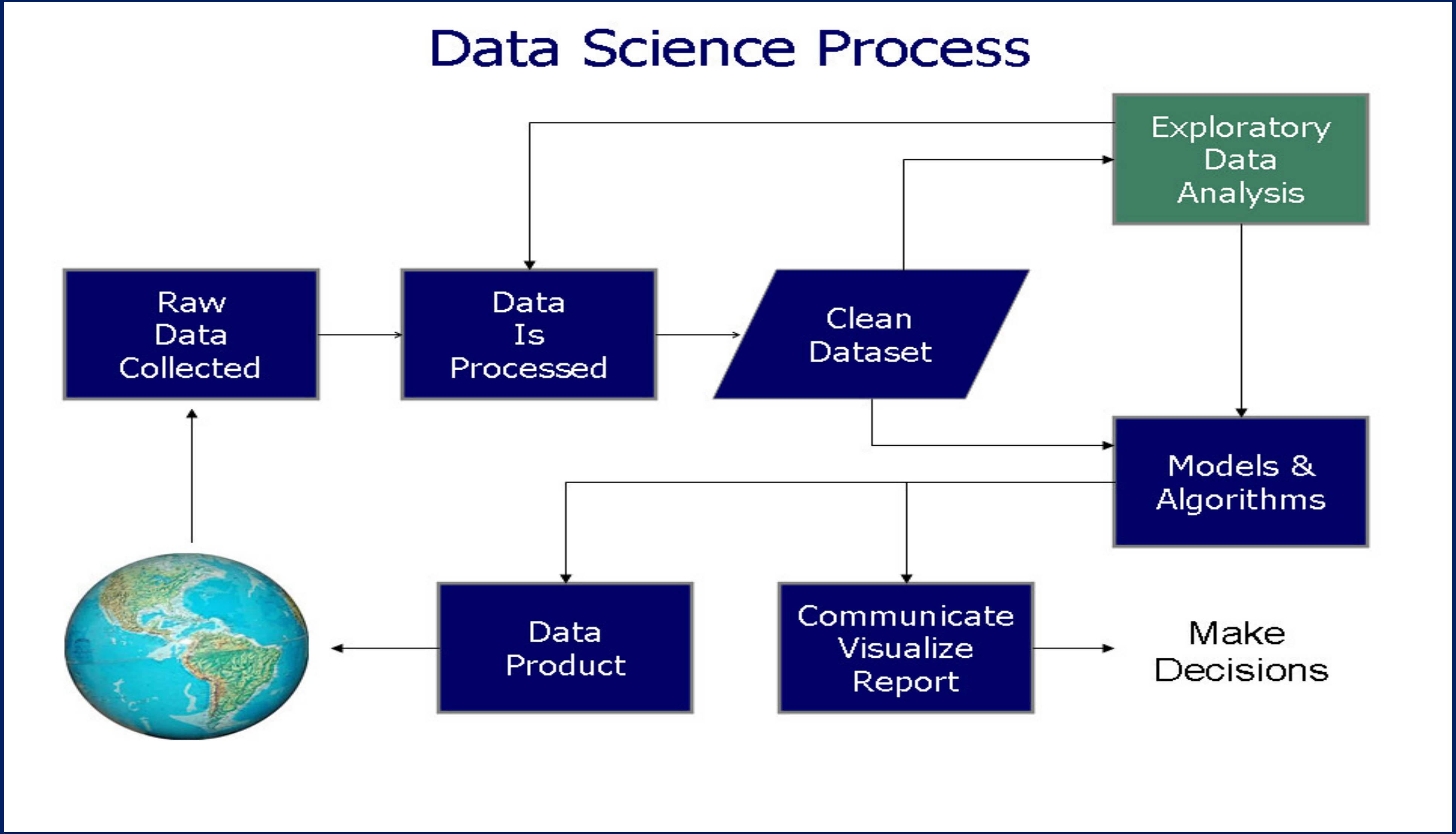
Project Background and Context Summary

- We are tasked to predict if the first stage of the SpaceX Falcon 9 rocket will land successfully for reuse in other rocket launches.
    - SpaceX advertises on their website that a Falcon 9 rocket launch costs 62 million dollars due to the fact of reusing the first stage rocket.
    - While their competitors rocket launches cost upward of 165 million dollars.
    - Therefore, if we can accurately predict if the first stage rocket will land successfully, then we can determine the cost of a launch.
    - Note this information can be used by SpaceX competitors to bid against them for launches.

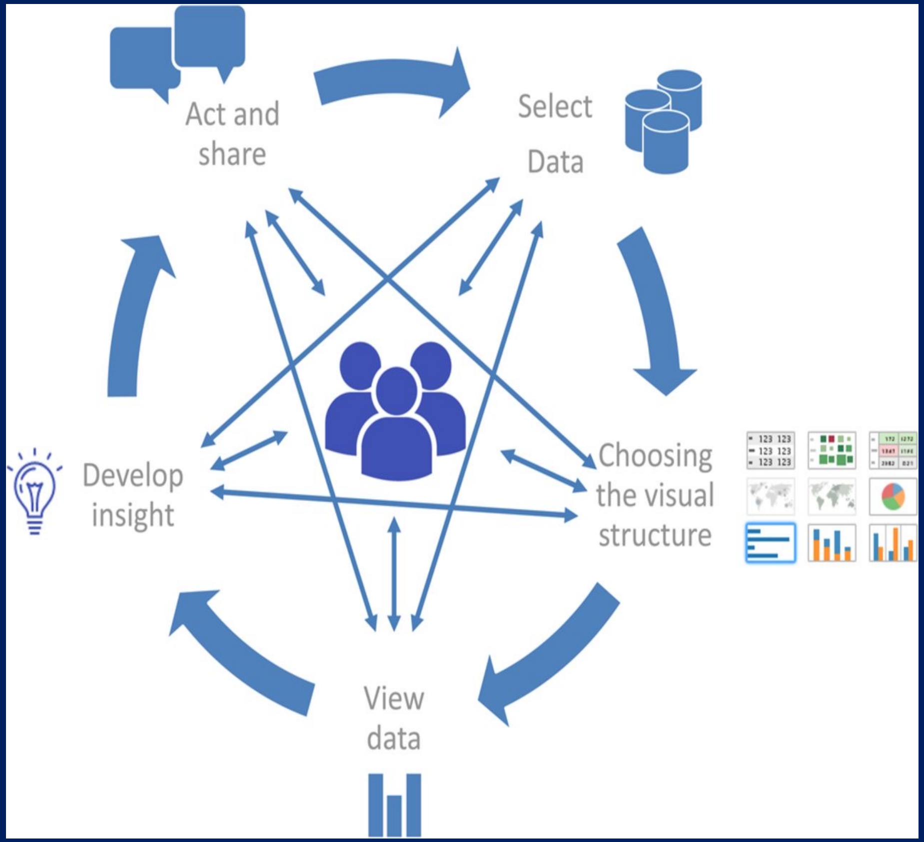
Problems That Need To Be Solved

- What factors influences if a rocket will land successfully?
- What relationships and effects between certain rocket variables will impact the success rate for rocket landings?
- What conditions can SpaceX control to ensure a successful rocket landing?

SpaceX - Methodology



SpaceX – Methodology Executive Summary

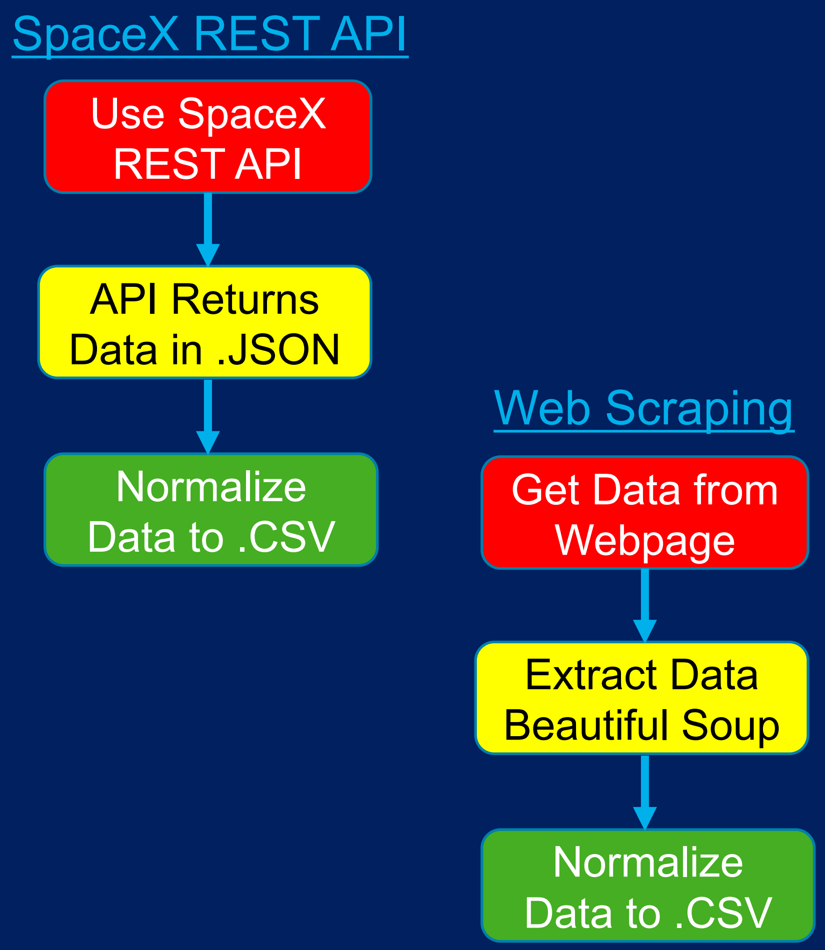


- **Data Collection Methodologies**
  - SpaceX Rest API from Postman
  - Falcon Launch Web Scraping from Wikipedia
- **Performed Data Wrangling to Normalize the Data**
  - Identify Missing Data, Types and to Label Data
    - Total Launches by Site per Orbit Types and Landing Outcomes
- **Perform Exploratory Data Analysis (EDA) using Visualization & SQL**
  - Plotting Data using Scatter & Bar Graphs to Show the Relationships between Variables to Highlight Data Patterns
- **Perform Interactive Visual Analytics using Folium and Plotly Dash**
- **Perform Predictive Analysis using Classification Models**
  - How to Build, Tune, Evaluate Classification Models

SpaceX – Data Collection

We Collected the Data by

- Using the SpaceX REST API to Get the publicly available Launch Data
- Using the Python Open Source package Beautiful Soup to perform Web Scraping to Parse the HTML and XML Data for the SpaceX Falcon 9 Launch Data.
- This allows us to Get Data for each Launch which includes Date, Booster Version, Payload Mass, Orbit, Launch Site, Outcome.
- Our goal is to use this Data to Predict if SpaceX will be able to Land the Rocket for reuse or if the Rocket will crash based on the historical data combinations of these variables.



SpaceX – Data Collection - API

Step 1 SpaceX API URL → Step 2 Convert to .json → Step 3 Normalize Data → Step 4 Create Dataframe → Step 5 Create .csv → GitHub URL to API Notebook

1 - Get Data Response from SpaceX API with URL

```
spacex_url=https://api.spacexdata.com/v4/launches/past
response = requests.get(spacex_url)
```

2 - Convert Response to .json

```
# 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 Normalize Data

```
# Call getBoosterVersion
getBoosterVersion(data)

# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4 - Assign List to Dictionary create 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, Mean for missing Data and create .csv

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

data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9

# Calculate the mean value of PayloadMass column
mean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].fillna(mean)

data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

[GitHub URL to API Notebook]

SpaceX – Data Collection – Web Scraping

Step 1 Wikipedia HTML → Step 2 BeautifulSoup Object → Step 3 Functions Create Table → Step 4 Get Column Names → Step 5 Create Dictionary → Step 6 Dictionary Data from Table → Step 7 Dictionary to Dataframe → Step 8 Create .csv → [GitHub URL to Web Scraping Notebook]

1 - Get Data Response from Wikipedia HTML

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"

# use requests.get() method with the provided static_url
page = requests.get(static_url)

# assign the response to a object
page.status_code
```

2 - Create BeautifulSoup Object from HTML

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(page.text, 'html.parser')
```

3 - Find Functions Create Table

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

4 - Get Column Names

```
column_names = []
# Apply find_all() function with `th` element on first_launch_table
temp = soup.find_all('th')

# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
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 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']=[]
```

6 - Append Dictionary Data from Table

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
         (see Notebook for the rest of the code)
```

7 - Convert Dictionary to Dataframe

```
df = pd.DataFrame.from_dict(launch_dict)
 (see Notebook for the rest of the code)
```

8 - Create .csv from Dataframe

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

[GitHub URL to Web Scraping Notebook]

SpaceX – Data Wrangling (EDA)

Step 1
Number Launches per Site

Step 2
Number and Occurrence per Orbit

Step 3
Number and Occurrence of Outcomes per Orbit

Step 4
Create Landing Outcome Labels

Step 5
Create .csv

GitHub URL to EDA Notebook

Introduction

We will perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models. In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship. We converted the outcomes into Training Labels, 0 for failed landing and 1 for successful landing.

Space X Launch Sites

Cape Canaveral Space
Launch Complex 40
CCAFS SLC 40

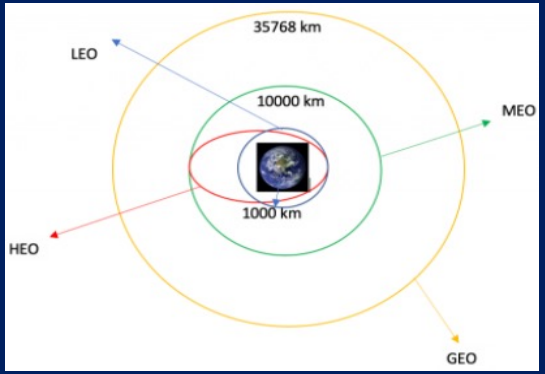
Vandenberg Air Force Base Space
Launch Complex 4E (SLC-4E)
VAFB SLC 4E

Kennedy Space Center
Launch Complex 39A
KSC LC 39A

Process

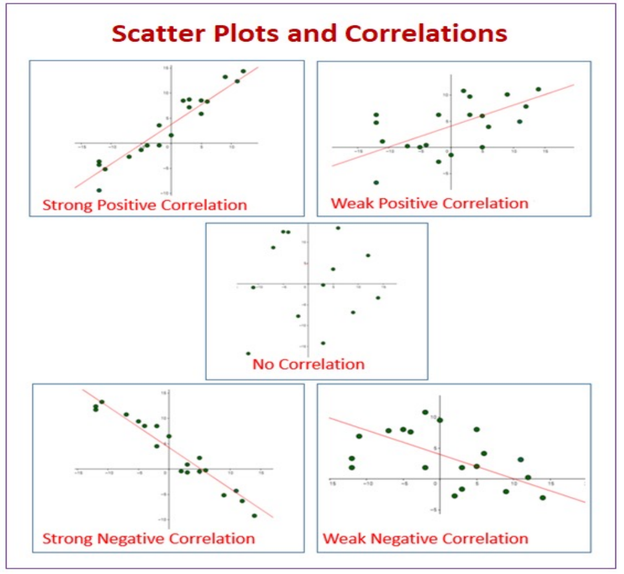
1. Calculate the number of launches on each site
2. Calculate the number and occurrence of each orbit
3. Calculate the number and occurrence of mission outcome per orbit type
4. Create a landing outcome label from Outcome column
5. Export Data to .csv

GitHub URL to EDA Notebook



Each launch used a dedicated orbit type
Low Earth Orbit (LEO)
Geosynchronous orbit (GEO)
Highly Elliptical Orbit (HEO)
Geocentric Orbit (MEO)

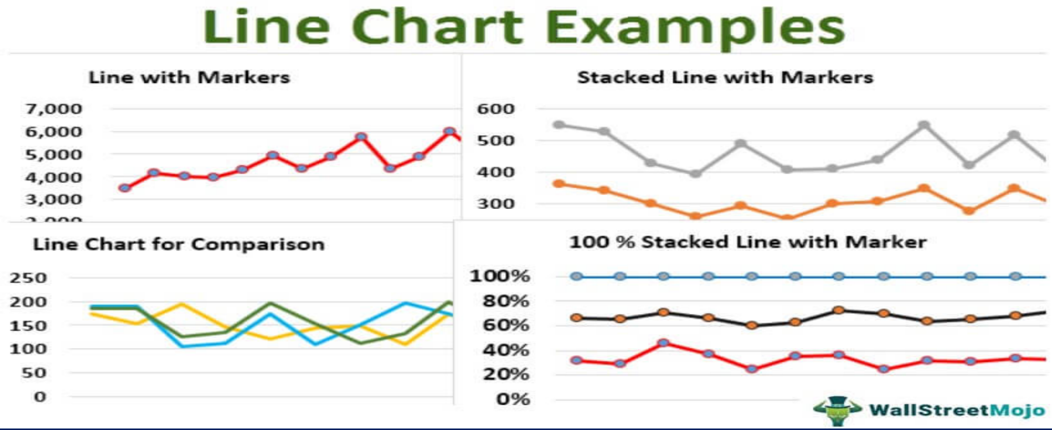
SpaceX – EDA with Data Visualization



Scatter Plot - Wikipedia
A scatter plot is a type of plot that uses coordinates to display values for typically two variables for a set of data. A graph in which the values of two variables are plotted along two axes, typically x-axis & y-axis. The pattern of the resulting points might reveal any correlation between the two variables.

Scatter Plots used for Analysis

- Flight Number vs Payload Mass
- Flight Number and Launch Site
- Payload and Launch Site
- Flight Number and Orbit type
- Payload and Orbit type



Line Chart - Wikipedia
A line chart or line plot or line graph or curve chart is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments. It is a basic type of chart common in many fields.

A line graph is a graphical display of information that changes continuously over time. A **line graph** may also be referred to as a **line chart**.

Line Chart used for Analysis

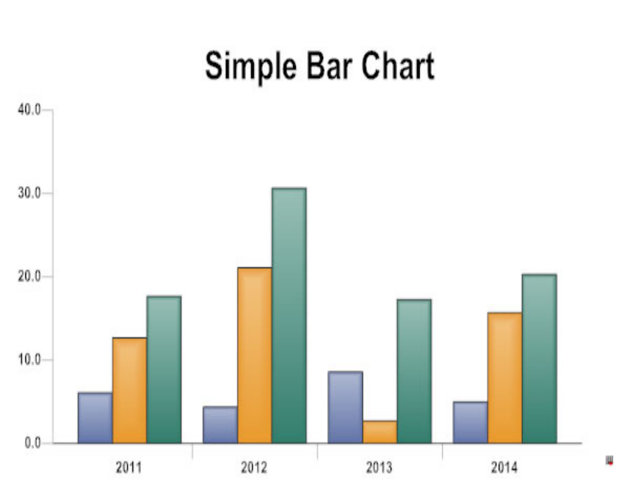
- Success Rate vs Year

Bar Chart - Wikipedia
A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally.

Bar graphs are used to compare and contrast numbers, frequencies or other measures of distinct categories of data.

Bar Chart used for Analysis

- Mean (Success Rate) vs Orbit

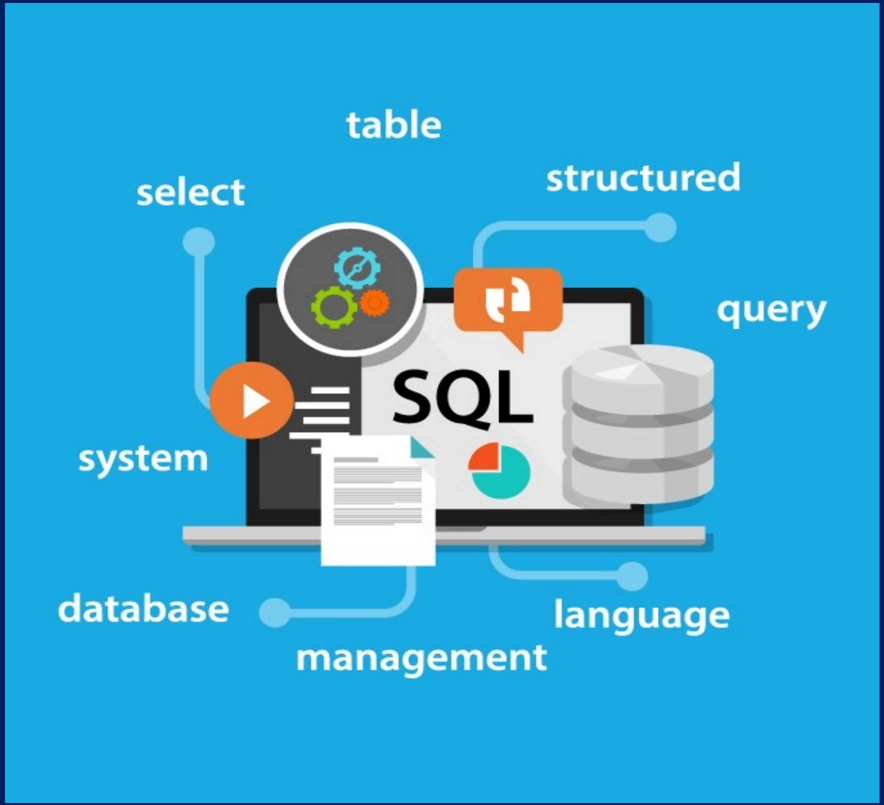


GitHub URL to Data Visualization Notebook

SpaceX – EDA with SQL

Performed the following SQL Queries to Analyze the Data

- Displayed the names of the unique launch sites in the space mission
- Displayed 5 records where launch sites begin with the string 'CCA’
- Displayed the total payload mass carried by boosters launched by NASA (CRS)
- Displayed average payload mass carried by booster version F9 v1.1
- Listed the date when the first successful landing outcome in ground pad was achieved
- Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listed the total number of successful and failure mission outcomes
- Listed the names of the booster_versions which have carried the maximum payload mass, using a subquery
- Listed the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Ranked 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 to SQL Notebook]

SpaceX – Building an Interactive Map with Folium

Created an Interactive Map to Visualize the Launch Site Locations for Data Analysis
For each Launch Site, added a Circle object based with Name as a popup label based on its Latitude and Longitude Coordinates.

Created a new column in launch_sites dataframe called marker_color to store the Marker colors based on the launch_outcomes for Failures and Successes with class values 0 and 1 with Red & Green color-labeled markers in marker clusters, to easily visually identify each launch sites failure & success rates.

Added Interactive capabilities to be able to Calculate the Distance between a Launch Site and its proximities to Coastline, Railway, Highway, City on the map based on their Latitude and Longitude Coordinates with a Line between the two locations with Distance as a popup label.

After plotting the distance lines to the proximities, it is easy to answer the following questions:
Are launch sites in close proximity to railways? No
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 to Folium Map Notebook

SpaceX – Build a Dashboard with Plotly Dash

Visualizing the Launch Data in an Interactive Map

- ➢ Added a dropdown list to enable Launch Site selection with default select value is for ALL sites.

Pie Chart

- ➢ Added a pie chart to show the total successful launches count for all sites.
  - ✓ If a specific launch site was selected, show the Success vs. Failed counts for the site.
- ➢ Added a slider to select payload range
  - ✓ Selectable dynamic payload range
- ➢ Added a callback function for `site-dropdown` as input, `success-pie-chart` as output
  - ✓ Selectable Launch Site

Scatter Plot

- ➢ Added a callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output
  - ✓ Shows relationship between two variables
  - ✓ Shows non-linear patterns
  - ✓ Minimum and Maximum range is determined
  - ✓ Visually easy to interpret data

GitHub URL to Code Notebook

SpaceX – Predictive Analysis (Classification)

Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8	Step 9	Step 10	Step 11	Step 12	
NumPy Array	Standardize Data	Split Data into Training & Test	Logistic Regression Object	Test Data Confusion Matrix	Vector Machine Best Fit	Test Data Confusion Matrix	Decision Tree Classifier	Decision Tree Classifier	K Nearest Neighbor	Test Data Confusion Matrix	Best Fit	GitHub URL to Learning Prediction Notebook

1 - Create a NumPy array from the column Class in data, by applying the method to_numpy() then assign it to the variable Y

2 - Standardize the data in X then reassign it to the variable X

3 - Use the function train_test_split to split the data X and Y into training and test data. Set the parameter test_size to 0.2 and random_state to 2. The training data and test data should be assigned to the following labels, X_train, X_test, Y_train, Y_test.

4 - Create a logistic regression object then create a GridSearchCV object logreg_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.

5 - Calculate the accuracy on the test data using the method score and confusion matrix

6 - Create a support vector machine object then create a GridSearchCV object svm_cv with cv - 10. Fit the object to find the best parameters from the dictionary parameters

7 - Calculate the accuracy on the test data using the method score and Confusion Matrix

8 - Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.

9 - Calculate the accuracy of tree_cv on the test data using the method score and confusion matrix

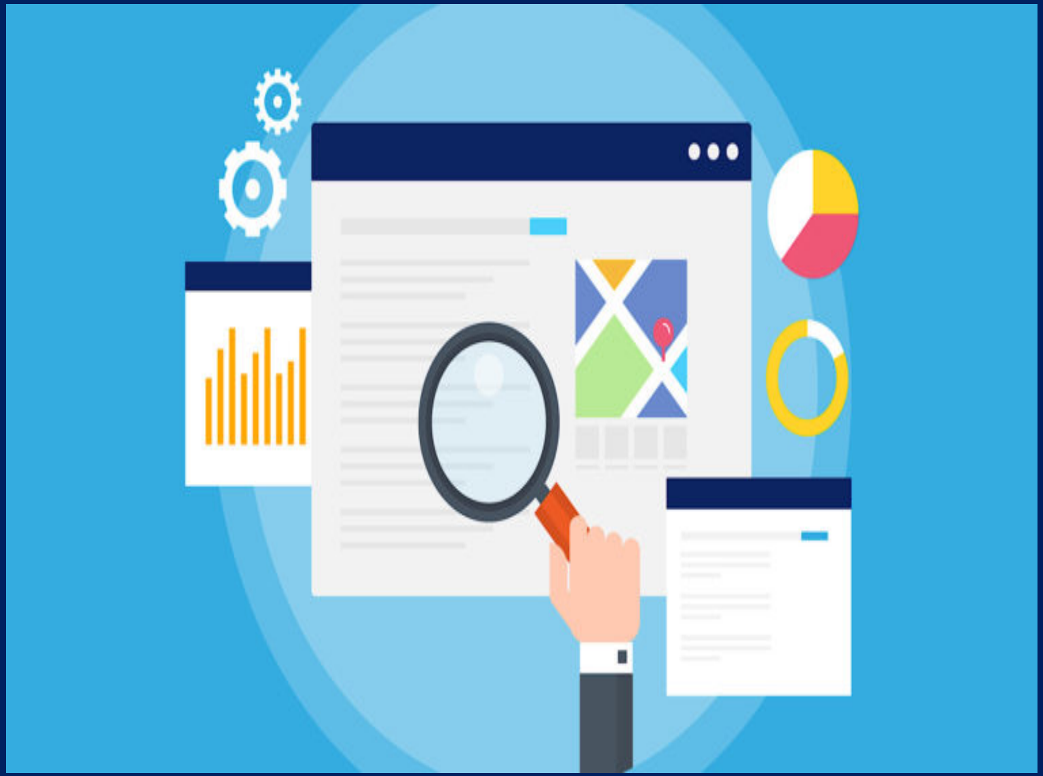
10 - Create a k nearest neighbor object then create a GridSearchCV object knn_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.

11 - Calculate the accuracy of tree_cv on the test data using the method score and confusion matrix

12 - Find the method performs best

GitHub URL to Learning Prediction Notebook

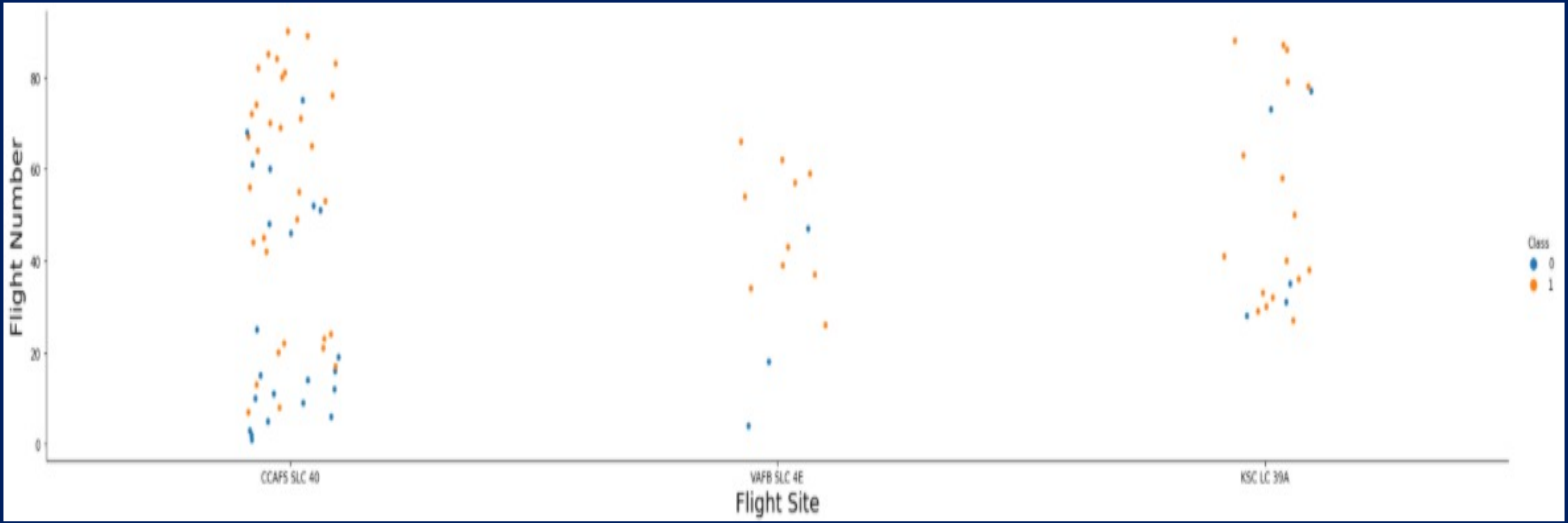
SpaceX – Results



- ❖Exploratory Data Analysis Results
- ❖Interactive Analytics Demo in Screenshots
- ❖Predictive Analysis Results

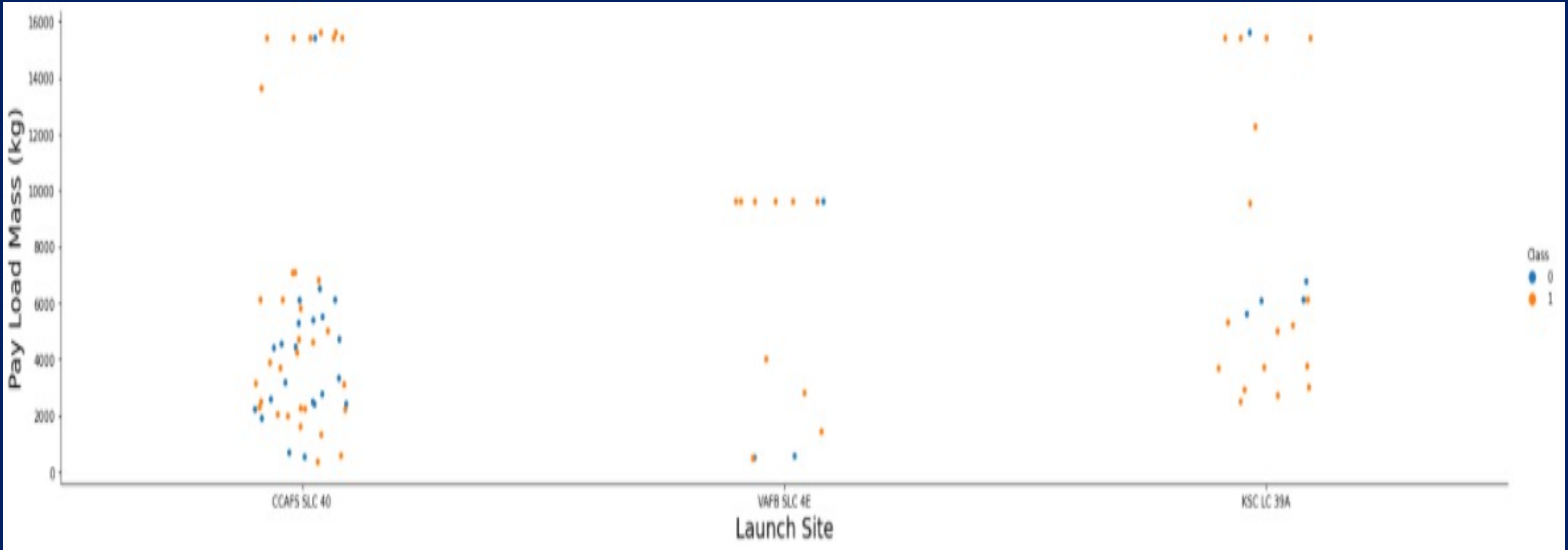
SpaceX – Insights Drawn from EDA

SpaceX – Flight Number vs. Launch Site



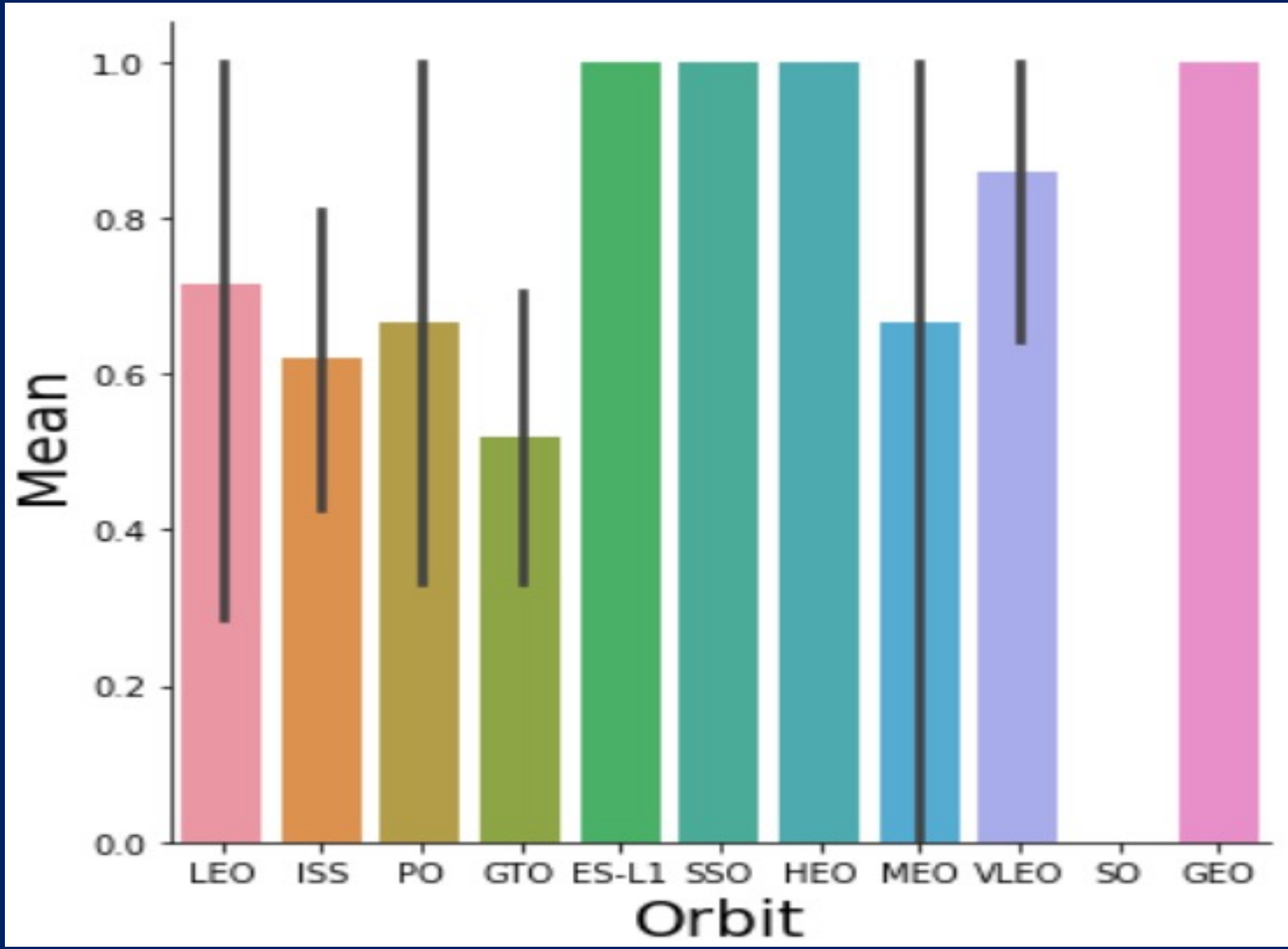
The more flights at Launch Sites the greater the Success Rate at the Launch Site

SpaceX – Payload vs. Launch Site



The more Payload for each Launch Site the Higher the Success Rate at the Launch Site

SpaceX – Success Rate vs. Orbit Type



Success Rate vs. Orbit Type

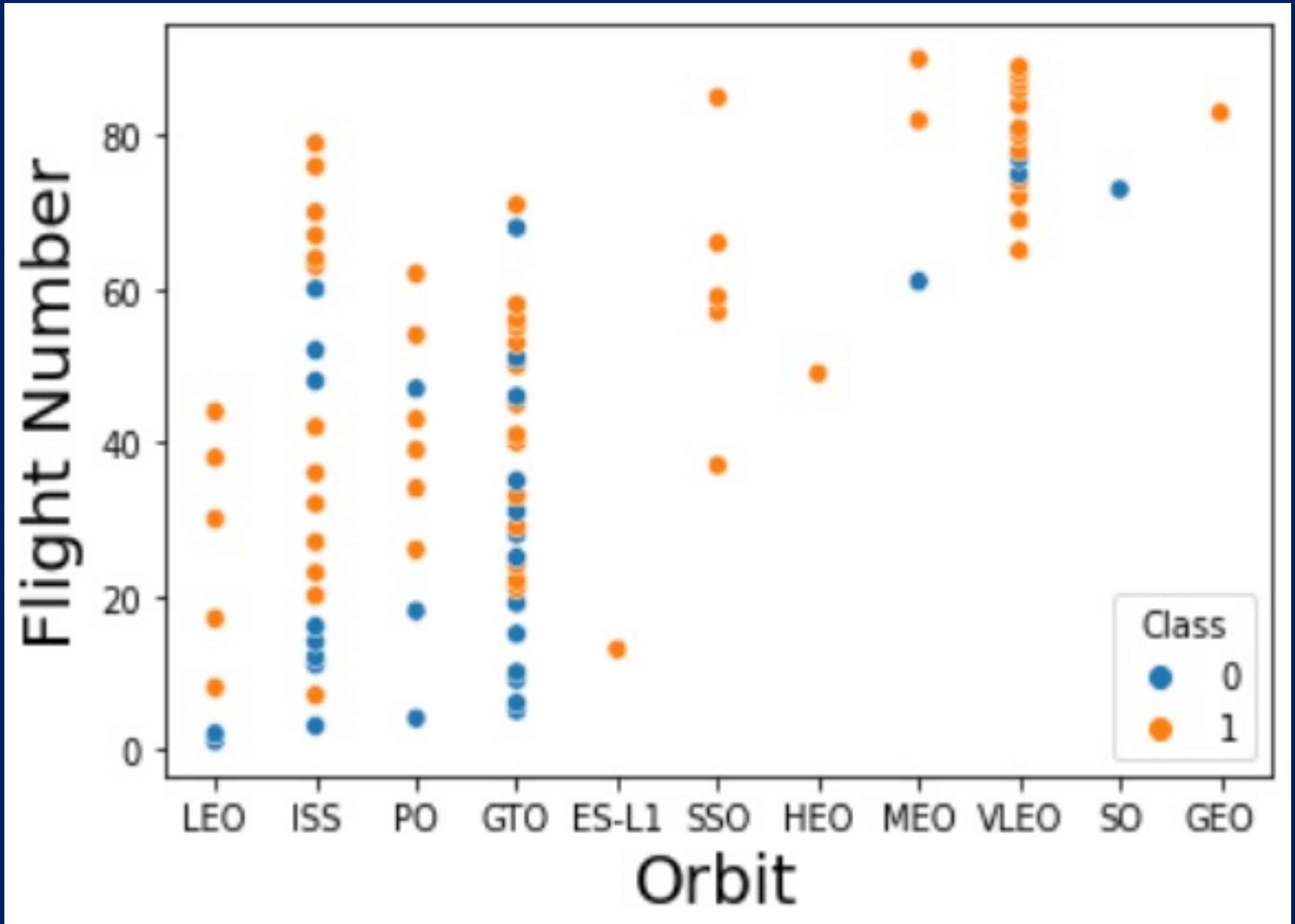
The orbits with the best Success Rate are:

- ES-L1
- GEO
- HEO
- SSO

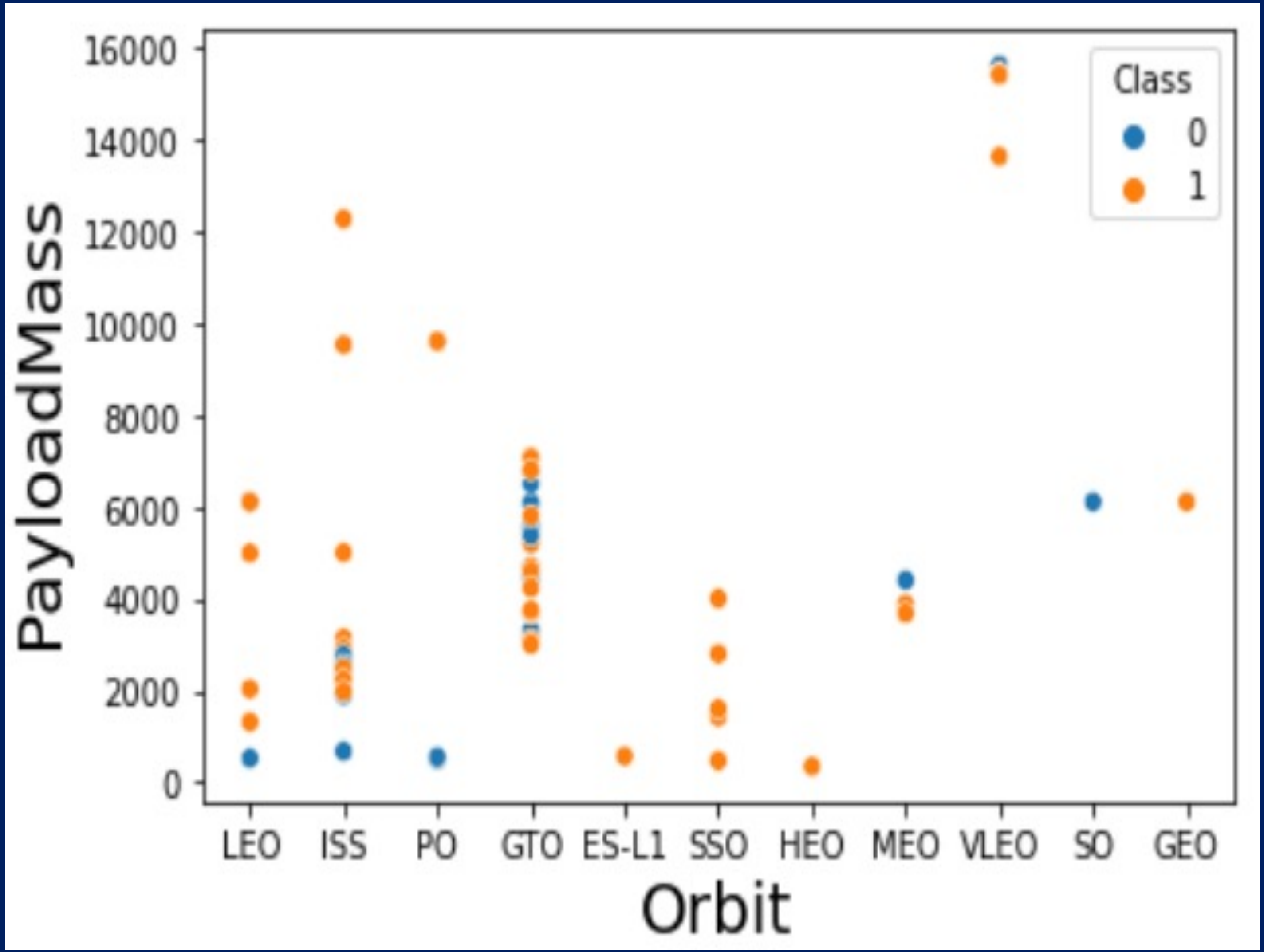
SpaceX – Flight Number vs. Orbit Type

Flight Number vs. Orbit Type

- More Flight Numbers = Higher Success Rate
  - ✓ LEO, PO, MEO
- More Flight Numbers No impact on Success Rate
  - ✓ ISS, GTO, ES-L1, SSO, HEO, VLEO, SO, GEO



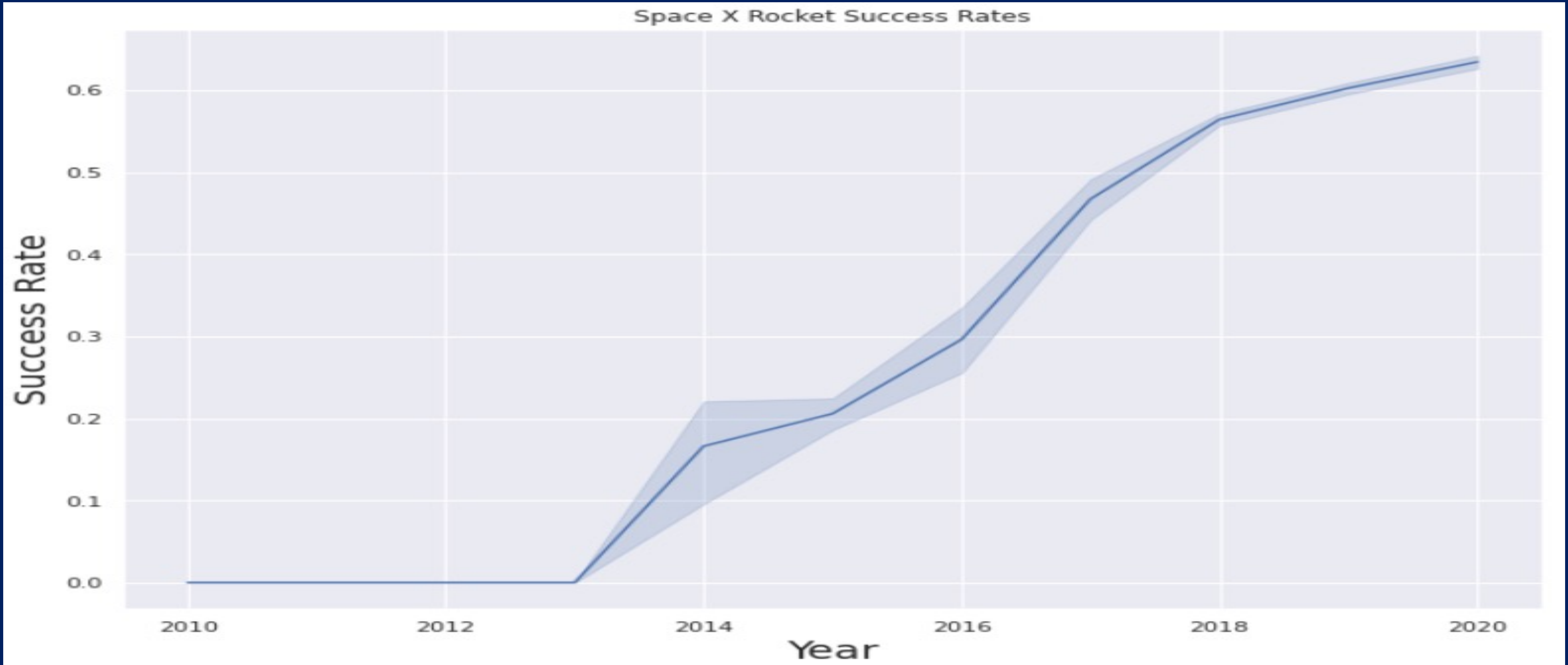
SpaceX – Payload vs. Orbit Type



Payload vs. Orbit Type

- ➢ More Payload = Higher Success Rate
  - ✓ LEO, PO
- ➢ More Payload No impact on Success Rate
  - ✓ ISS, GTO, ES-L1, SSO, HEO, MEO, VLEO, SO, GEO

SpaceX – Launch Success Yearly Trend



The Success Rate since 2013 keeps increasing

SpaceX – All Launch Site Names

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

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

```
* ibm_db_sa://mtb77289:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/BLUDB
Done.
```

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

The SQL Query searches the SPACXEXTBL for distinct(LAUNCH_SITE) to return a list of Unique Launch Sites.

SpaceX – Launch Site Names Begin with 'CCA'

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

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

```
 * ibm_db_sa://mtb77289:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/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

The SQL Query searches the SPACXEXTBL for LAUNCH_SITE with like 'CCA%' and limits the resulting output to first 5 records. The percentage % is used after CCA meaning the Launch_Site name must start with CCA.

SpaceX – Total Payload Mass from NASA

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://mtb77289:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/BLUDB
Done.
```

1
45596

SQL Query Explanation

- Uses the SUM function to summate the total of values in the PAYLOAD_MASS_KG_ column from the SPACEXTBL
- And uses the WHERE clause to filter the data to only perform calculations when the CUSTOMER column from the SPACEXTBL is equal to 'NASA (CRS)'
- The result is 45,596 KG Total Payload Mass carried by boosters launched from NASA (CRS)

SpaceX – Average Payload Mass by F9 v1.1

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://mtb77289:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/BLUDB
Done.
```

1
2928

SQL Query Explanation

- Uses the AVG function to average the total of values in the PAYLOAD_MASS_KG_ column from the SPACEXTBL
- And uses the WHERE clause to filter the data to only perform calculations when the BOOSTER_VERSION column from the SPACEXTBL is equal to ‘F9 v1.1’
- The result is 2,928 KG Total Payload Mass carried by booster version F9 v1.1

SpaceX – First Successful Ground Landing Date

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

Hint:Use min function

```
%sql select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://mtb77289:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/BLUDB
Done.
```

1
2015-12-22

SQL Query Explanation

- Uses the MIN function on the DATE column from the SPACEXTBL to determine the earliest date
- And uses the WHERE clause to filter the data to only include when the Landing_Outcome column from the SPACEXTBL is equal to ‘Success (ground pad)’
- The result is the date 2015-12-22 for the first successful landing outcome in ground pad

SpaceX – Successful Drone Ship Landing with Payload between 4000 and 6000

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 from SPACEXTBL where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__1
```

```
 * ibm_db_sa://mtb77289:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/BLUDB
Done.
```

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

%sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome = ‘Success (drone ship)’ and PAYLOAD_MASS_KG_ > 4000 and PAYLOAD_MASS_KG < 6000

SQL Query Explanation

- Selecting only BOOSTER_VERSION column from the SPACEXTBL
- And uses the WHERE clause to filter the data to only include when the Landing_Outcome column from the SPACEXTBL is equal to ‘Success (drone ship)’
- The AND clause specifies the additional filter conditions on PAYLOAD_MASS_KG_ column from the SPACEXTBL when the payload mass is > 4000 & < 6000
- The results are booster_version: F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2

SpaceX – Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'
```

```
 * ibm_db_sa://mtb77289:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/BLUDB
Done.
```

1
100

SQL Query Explanation

- Uses the COUNT function on the MISSION_OUTCOME column from the SPACEXTBL
- And uses the WHERE clause to filter the data to only include when the MISSION_OUTCOME column from the SPACEXTBL is equal to ‘Success’ or ‘Failure (in flight)’
- The results are Failures = 1 & Success = 100 for Mission Outcomes

SpaceX – Boosters Carried Maximum Payload

SQL Query Explanation

- Selects the BOOSTER_VERSION column from the SPACEXTBL
- And uses the WHERE clause to filter the data to only include when the PAYLOAD_MASS_KG_ column from the SPACEXTBL is equal to using max function on the PAYLOAD_MASS_KG_ column from the SPACEXTBL
- The result are the listed booster versions

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://mtb77289:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/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

SpaceX – 2015 Launch Records

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

```
%sql select DATE, BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where Landing__Outcome = 'Failure (drone ship)' and (DATE between '01-01-201!
```

```
 * ibm_db_sa://mtb77289:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/BLUDB
Done.
```

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

%sql select DATE, BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where Landing_Outcome = 'Failure (drone ship)' and (DATE between '01-01-2015 and '12-31-2015')

SQL Query Explanation

- Selects the DATE, BOOSTER_VERSION, LAUNCH_SITE columns from the SPACEXTBL
- And uses the WHERE clause to filter the data to only include when the Landing_Outcome column from the SPACEXTBL is equal to 'Failure (drone ship)' and when the DATE column from the SPACEXTBL is between 01-01-2015 and -12-31-2015 (i.e. for all of 2015)
- The results are two failed landing drone ship outcomes for 2015:

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

SpaceX – 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 = 'Success (ground pad)' or Landing__Outcome = 'Failure (drone ship)') and (DATE betw
```

```
* ibm_db_sa://mtb77289:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/BLUDB
Done.
```

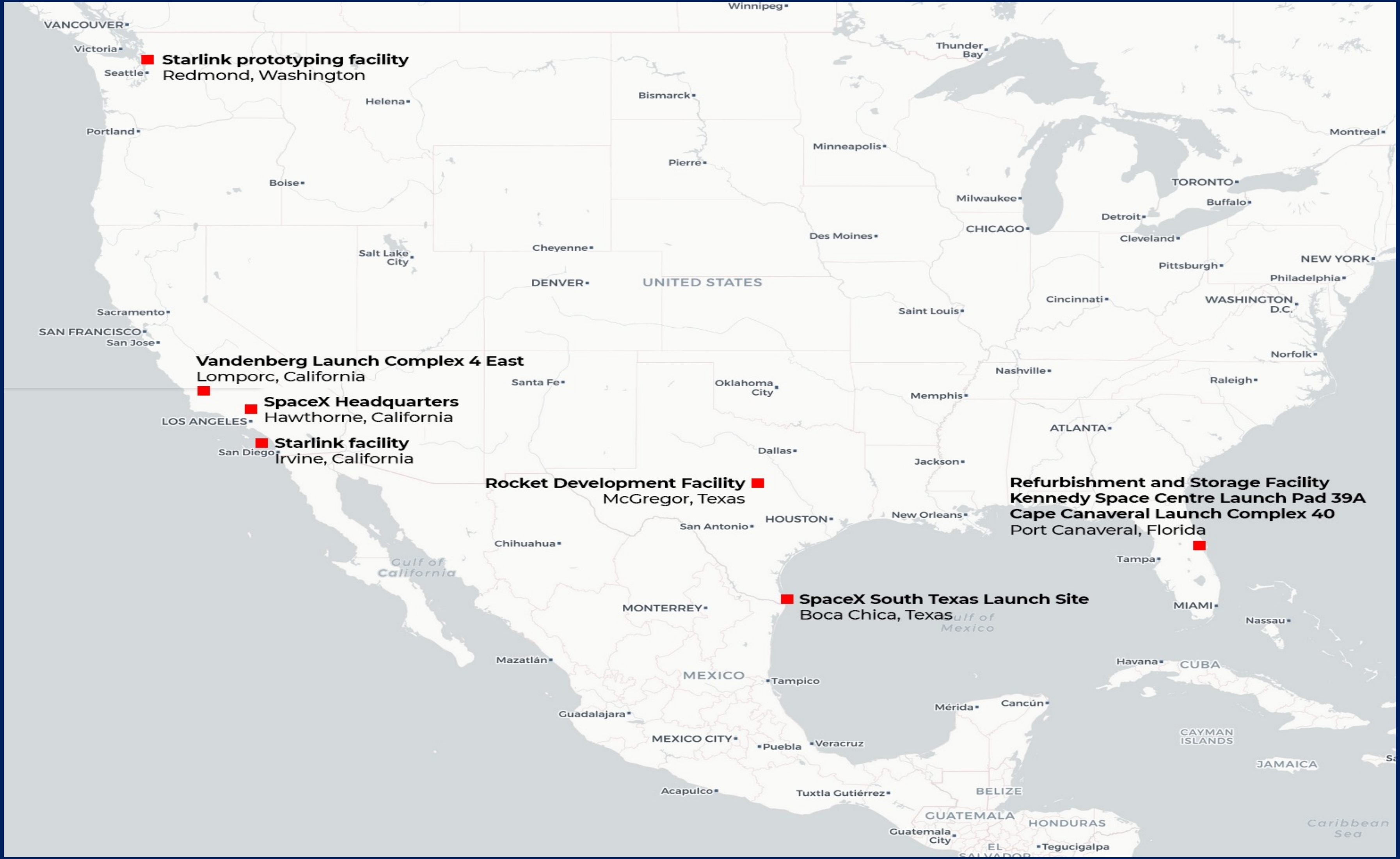
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-06-15	14:29:00	F9 FT B1024	CCAFS LC-40	ABS-2A Eutelsat 117 West B	3600	GTO	ABS Eutelsat	Success	Failure (drone ship)
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-01-17	18:42:00	F9 v1.1 B1017	VAFB SLC-4E	Jason-3	553	LEO	NASA (LSP) NOAA CNES	Success	Failure (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)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-01-10	09:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)

%sql select * from SPACEXTBL where (Landing_Outcome = 'Success (ground pad)' or Landing_Outcome = 'Failure (drone ship)' and (DATE between '2010-06-04' and '2017-03-20') order by date desc

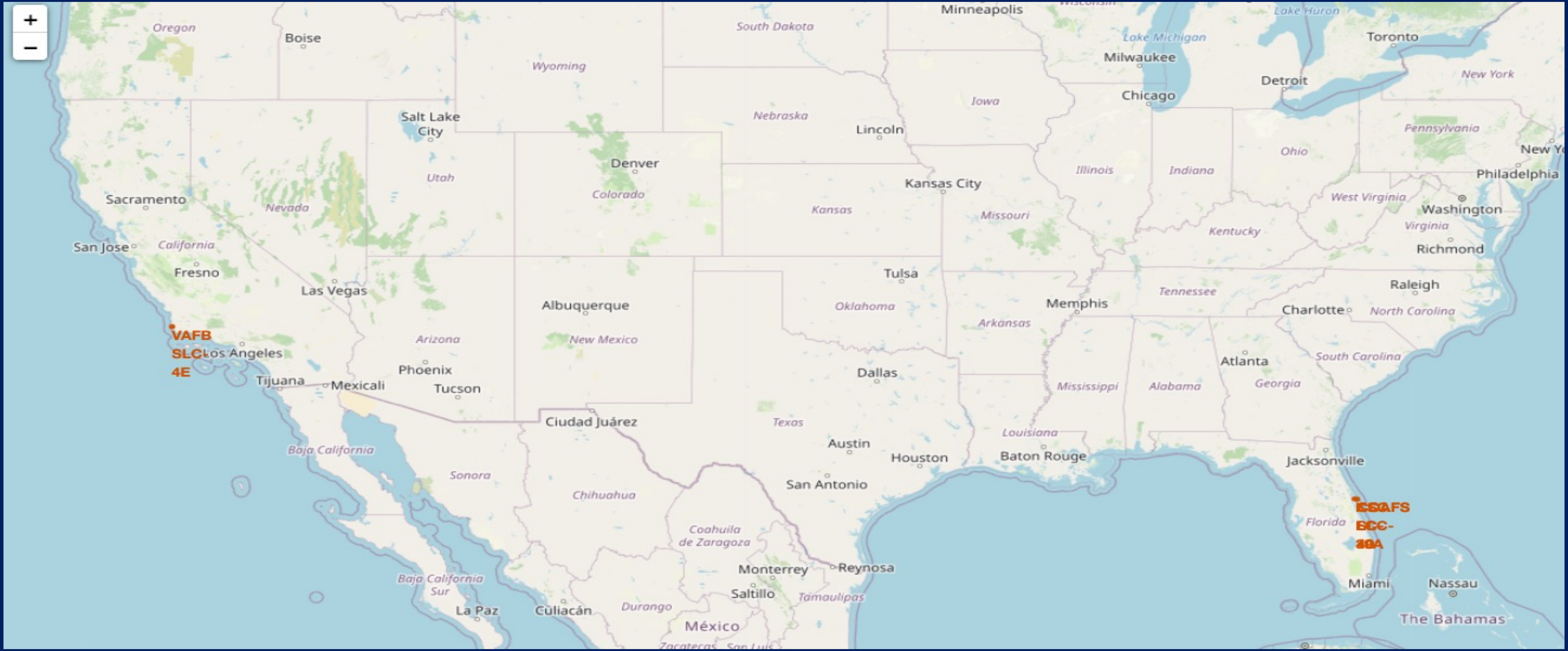
SQL Query Explanation

- Selects using * (wildcard) all columns from the SPACEXTBL
- And uses the WHERE clause to filter the data to only include when the Landing_Outcome column from the SPACEXTBL is equal to 'Success (ground pad)' or Landing_Outcome column from the SPACEXTBL is equal to 'Failure (drone ship)' and Date between 2010-06-04 and 2017-03-20 use DESC function to sort results in descending order
- There are eight listed results, see screen shot to the left

SpaceX – Launch Sites Proximities Analysis



SpaceX – All Launch Sites Interactive Map

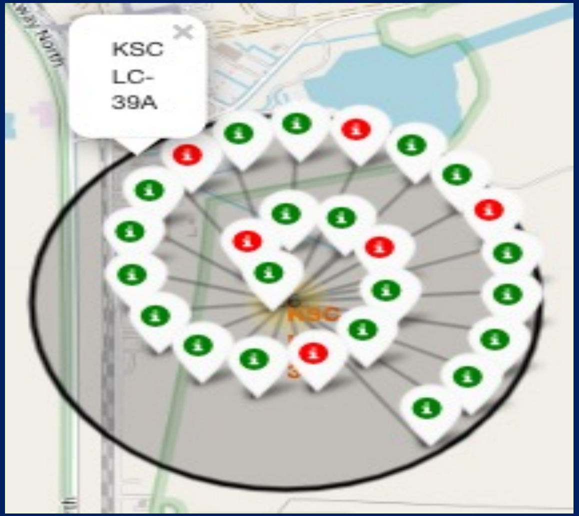


We can see the SpaceX Launch Sites are located near the East and West Coasts in Florida and California of the United states of America

SpaceX – Interactive Map for Launch Site Outcomes Using Colored Label Markers and Marker Clusters

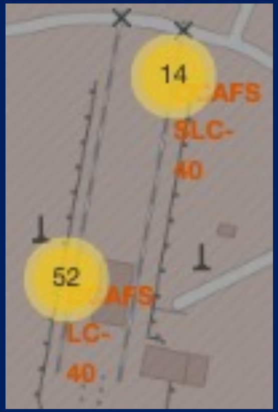
Florida Launch Sites

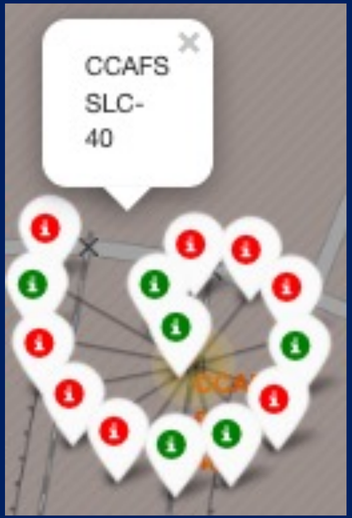
- Shows Total Launches per Site
  - ✓ 26 Launches at KSC LC-39A
  - ✓ 66 Launches at CCAFS SLC-40
- Drill down on Site KSC LC-39A
  - ✓ Shows 6 Failed Launches
  - ✓ Shows 20 Successful Launches
  - ✓ Shows Launch Site Name



CCAFS SLC-40 Florida Launch Sites

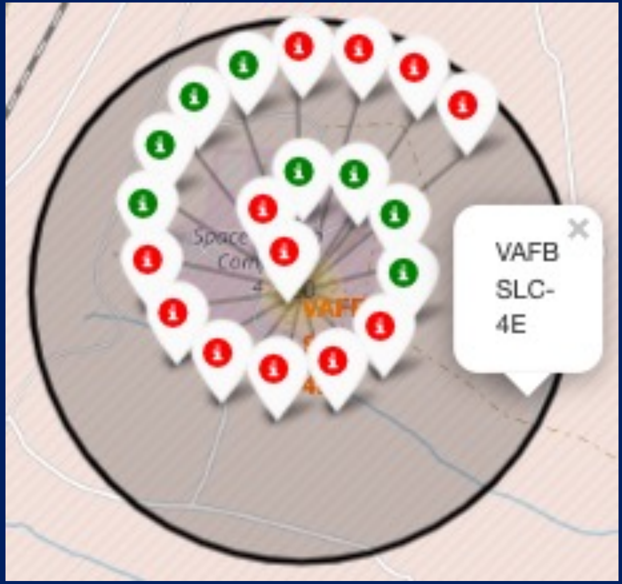
- Shows Total Launches per Coordinates
  - ✓ 14 Launches at CCAFS SLC-40
  - ✓ 52 Launches at CCAFS LC-40
- Drill down on Site CCAFS SLC-40
  - ✓ Shows 8 Failed Launches
  - ✓ Shows 6 Successful Launches
  - ✓ Shows Launch Site Name
- Drill down on Site CCAFS LC-40
  - ✓ Shows 38 Failed Launches
  - ✓ Shows 14 Successful Launches
  - ✓ Shows Launch Site Name



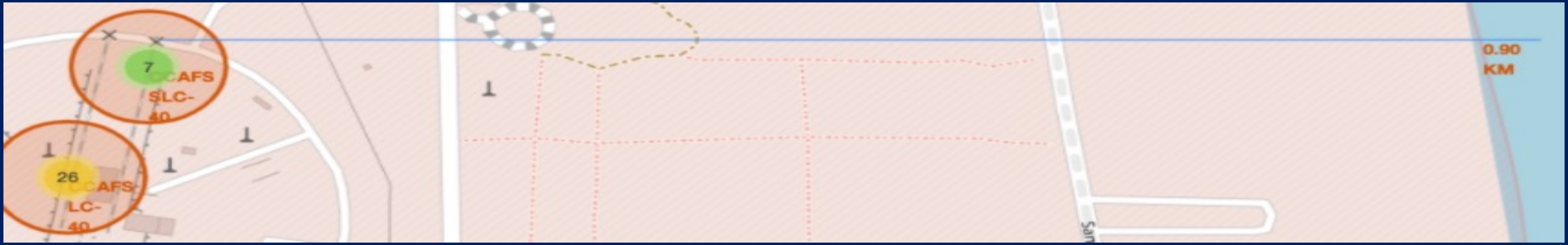


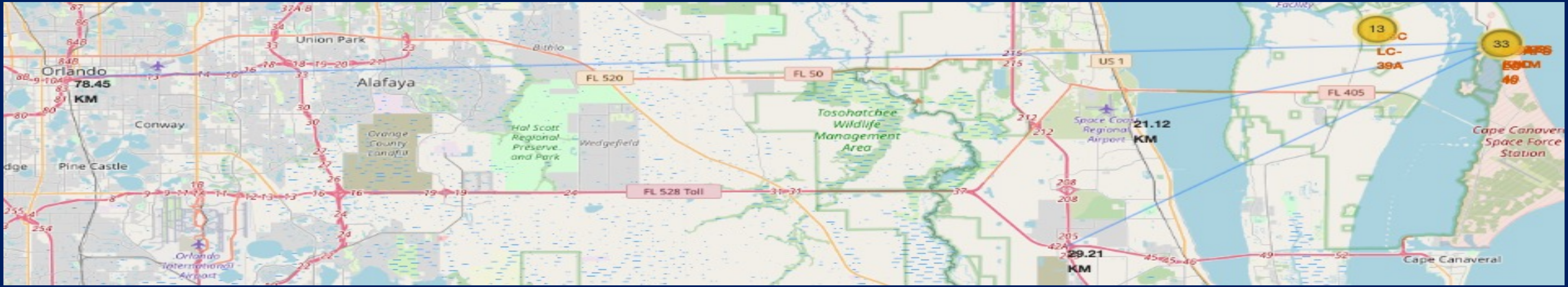
California Launch Site

- Shows 20 Total Launches
- Drill down on Site VAFB SLC-4E
  - ✓ Shows 12 Failed Launches
  - ✓ Shows 8 Successful Launches
  - ✓ Shows Launch Site Name



SpaceX – Interactive Launch Site Map to Calculate Distances between CCAFS SLC-40 Launch Site

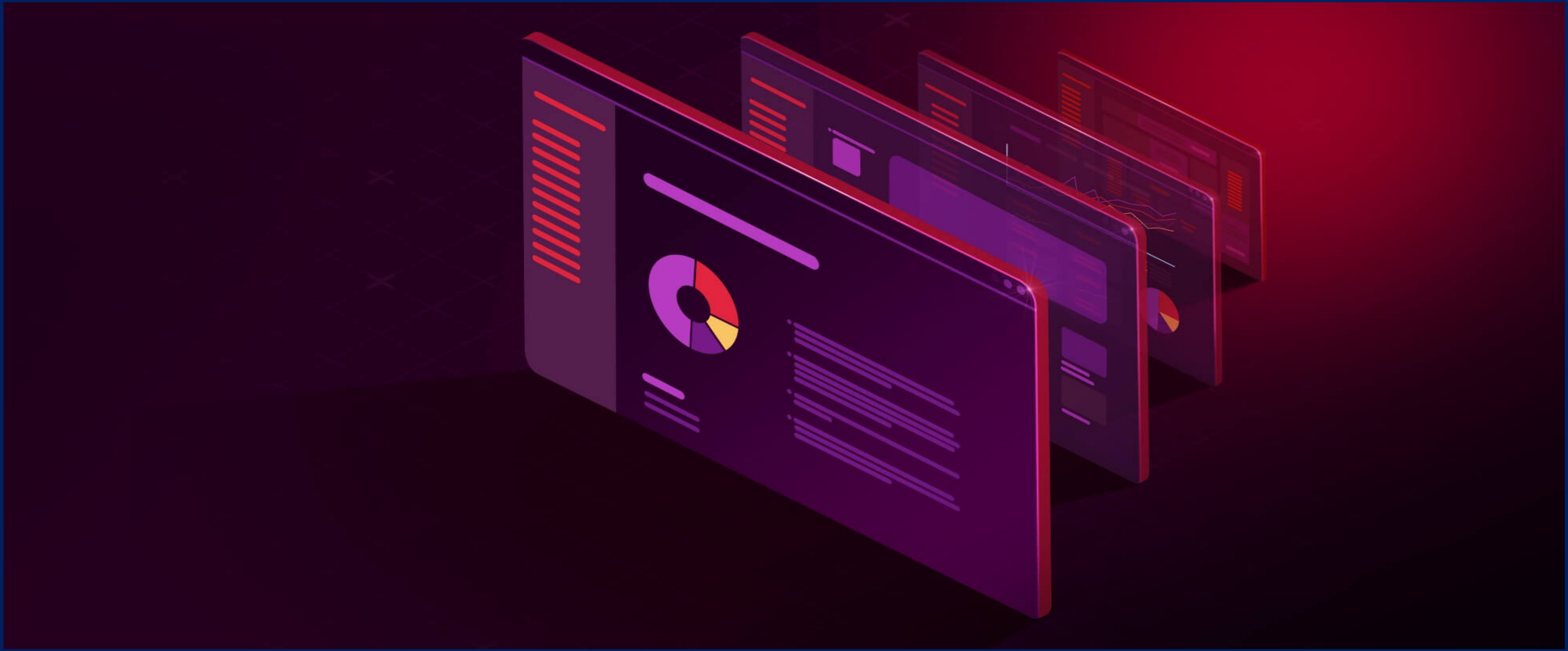




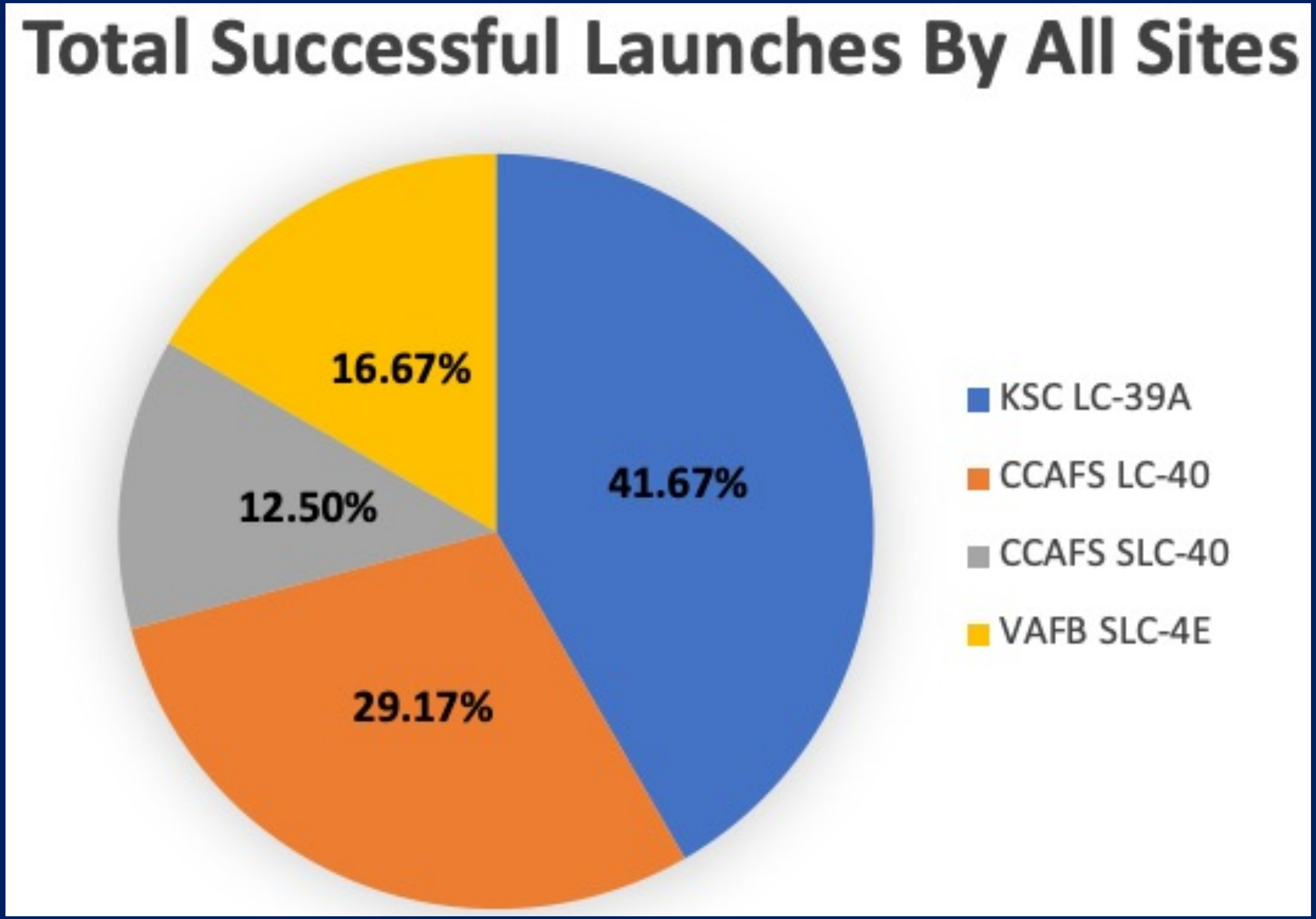
After plotting the distance lines between CCAFS SLC-40 to the proximities, it is easy to answer the following questions:

- Are launch sites in close proximity to railways? No, 21.12 KM
- Are launch sites in close proximity to highways? No, 29.21 KM
- Are launch sites in close proximity to coastline? Yes, 0.90 KM
- Do launch sites keep certain distance away from cities? Yes, 78.45 KM

SpaceX – Build a Dashboard with Plotly Dash



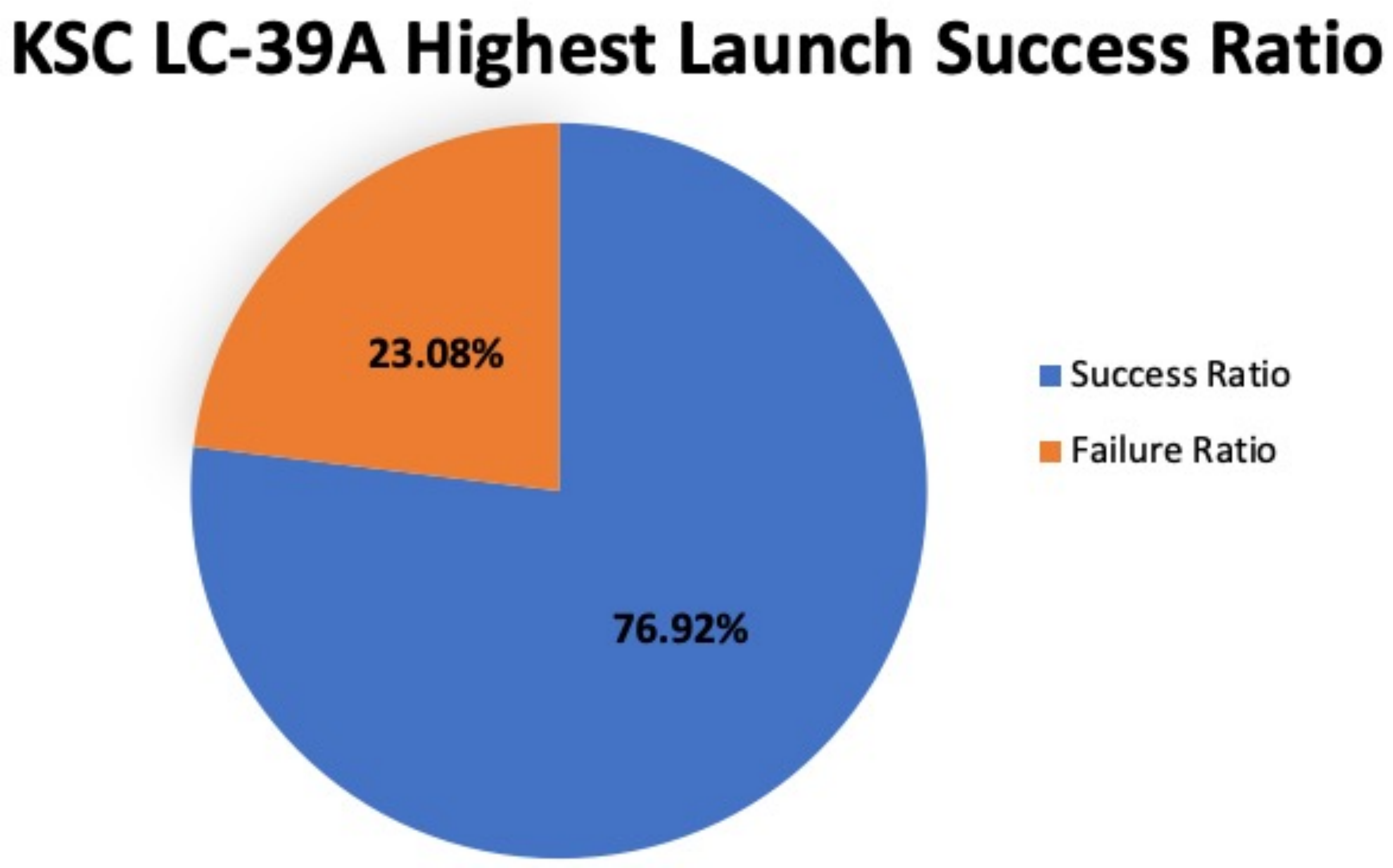
SpaceX – Pie Chart Launch Success Rate by Site



Total Successful Launches by All Sites

- KSC LC-39A has the first highest success rate overall with 41.67%
- CCAFS LC-40 has the second highest success rate overall with 29.17%
- VAFB SLC-4E has the third highest success rate overall with 16.67%
- CCAFS SLC-40 has the fourth highest success rate overall with 12.50%

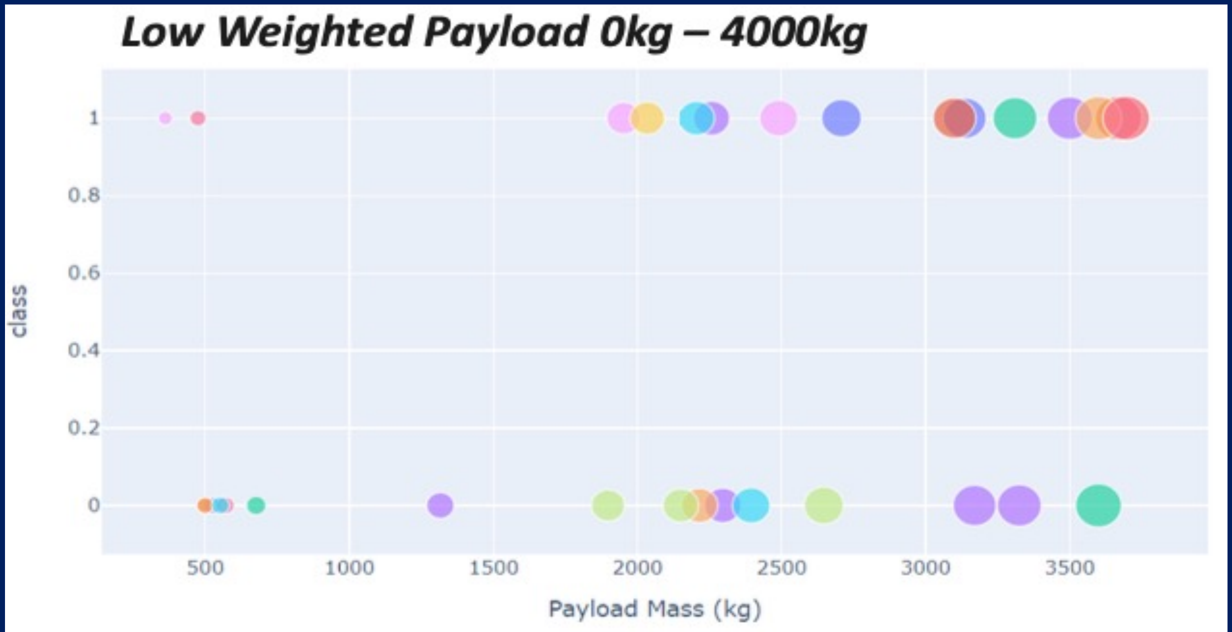
SpaceX – Pie Chart Highest Launch Site Success Ratio

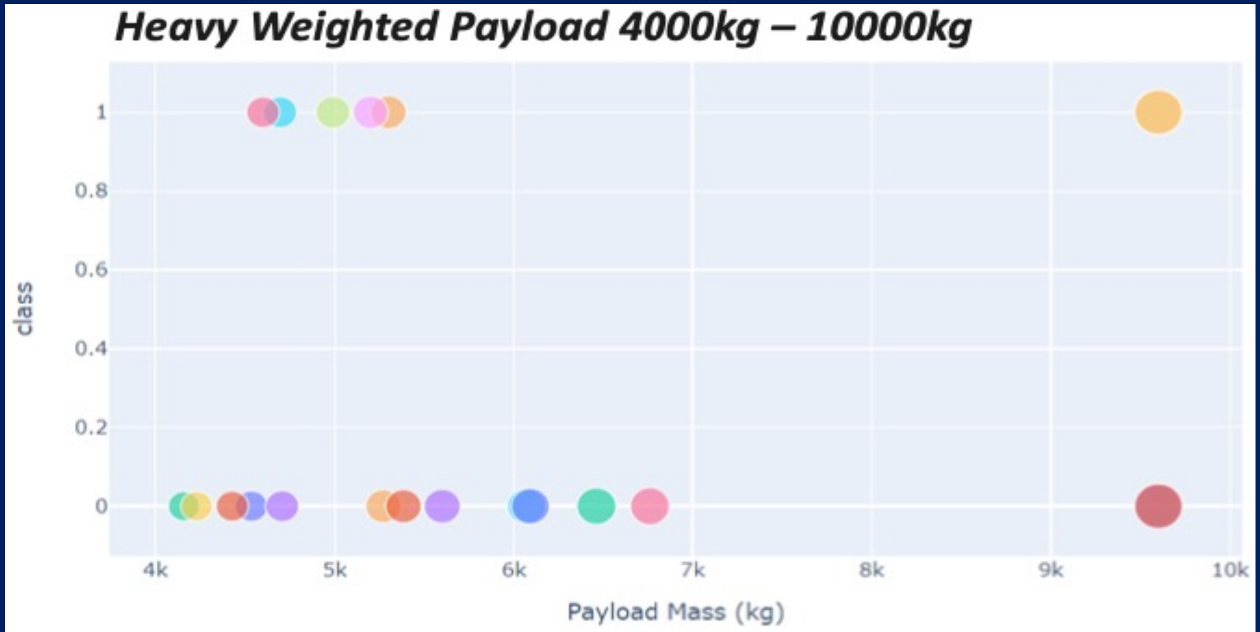


Highest Launch Site Success Ratio

- KSC LC-39A has the highest success ratio overall with 76.92%

SpaceX – Scatter Plots for all Sites, with different Payload Ranges Selected

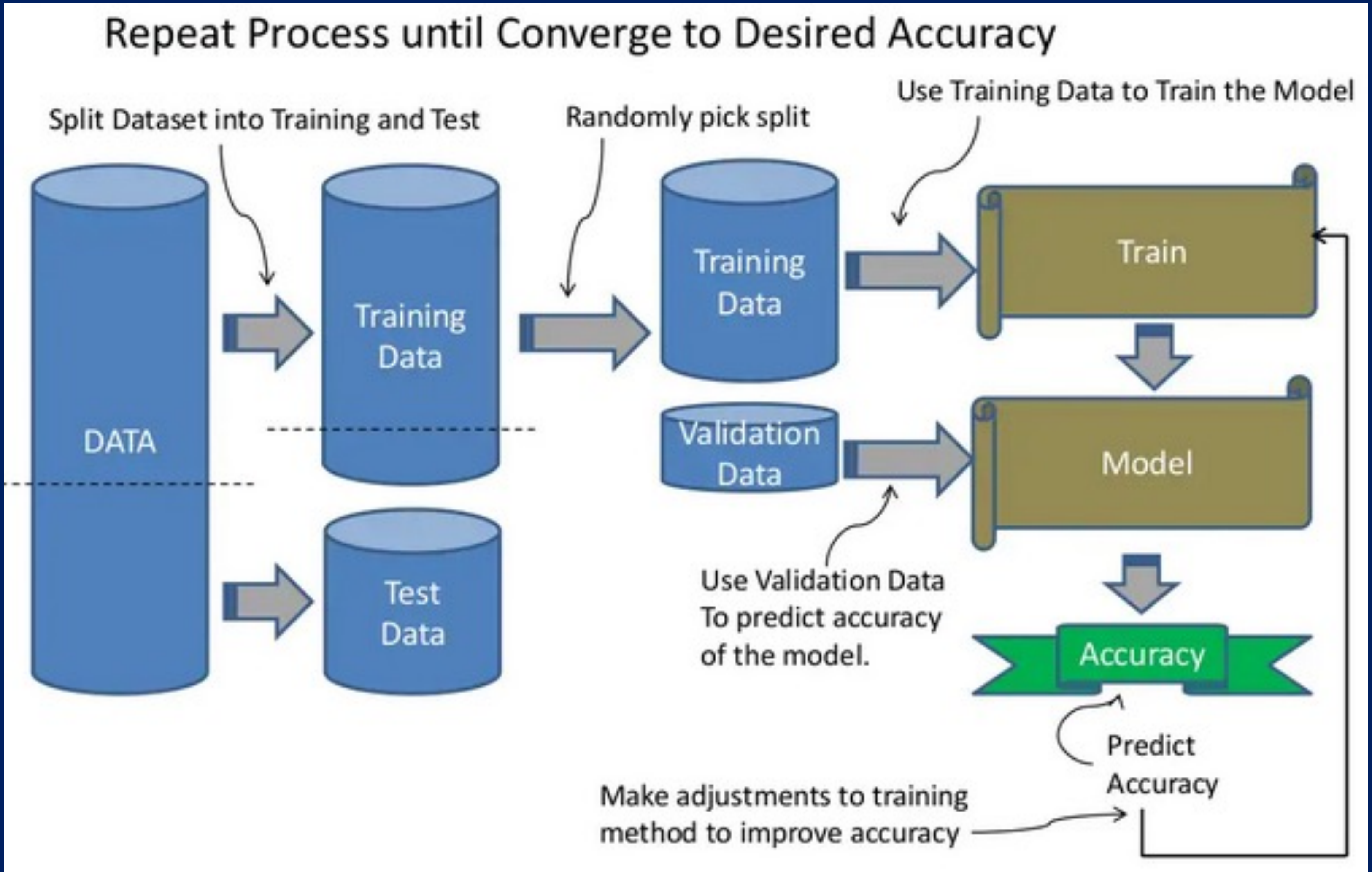




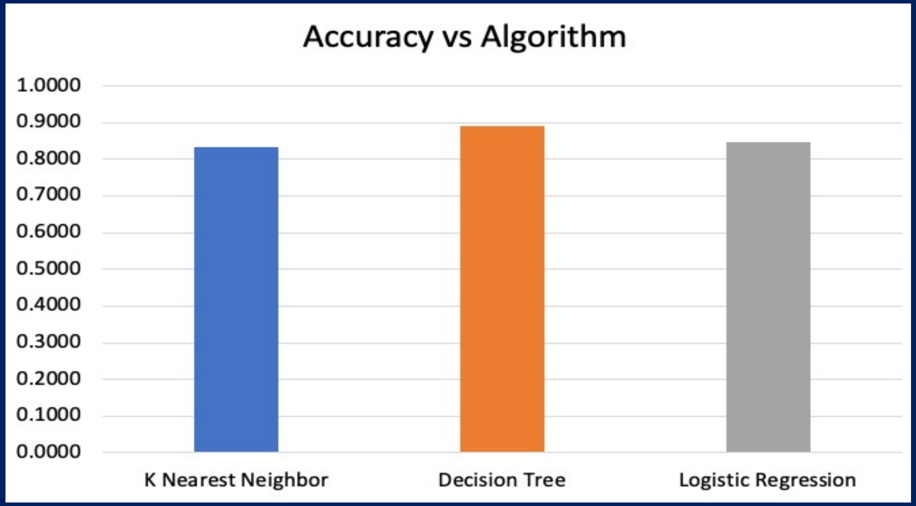
Scatter Plots for all Sites, with different Payload Ranges Selected

- By analyzing the scatter plots
  - ✓ Low Weighted Payloads have a higher success rate
  - ✓ Heavy Weighted Payloads have a higher failure rate

SpaceX – Predictive Analysis (Classification)



SpaceX – Classification Accuracy



Highest Classification Accuracy

- Decision Tree has the highest accuracy
- Using the best hyperparameters for each algorithm and using the validation data, the decision tree classifier achieved 88.93% accuracy on the test data
- The other Algorithms were close
  - ✓ K Nearest Neighbor (KNN) achieved 83.34% accuracy on the test data
  - ✓ Logistic Regression achieved 84.64% accuracy on the test data

Find the method performs best:

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

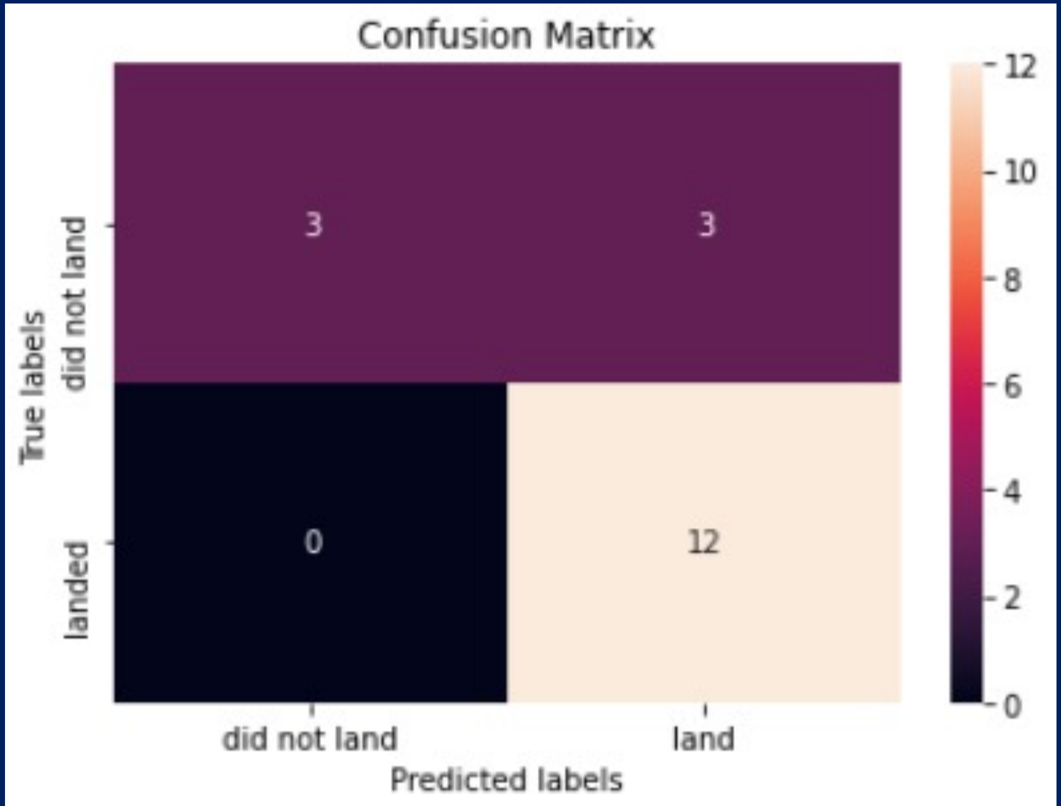
```
Best Algorithm is Tree with a score of 0.8892857142857145
Best Params is : {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitte
r': 'random'}
```

SpaceX – Confusion Matrix for Best Algorithm

Confusion Matrix Actual vs Predicted

- The sweet spot is True Positives (TP)

		Predicted	
		Negative (**N**) -	Positive (**P**) +
Actual	Negative -	True Negatives (**TN**)	False Positives (**FP**) **Type I error**
	Positive +	False Negatives (**FN**) **Type II error**	True Positives (**TP**)



Confusion Matrix Actual vs Predicted

- Decision Tree Algorithm Confusion Matrix

SpaceX – Conclusions

SpaceX Conclusions

- The Orbits having the best Success Rate are ES-L1, GEO, HEO, SSO
- Launch Site KSC LC-39A had the most successful launches of all sites
- The Success Rate is proportional with time in years, overtime the Success Rate increases
- The Payload Mass affects the Success Rate
  - ✓ Lower the Payload Mass the Higher the Success Rate
- Date, Booster Version, Orbit, Launch Site, and Payload Mass individually and combined will contribute to the:
  - ✓ Overall Success Rate
  - ✓ Overall Failure Rate
- The Best Machine Learning Algorithm for the Data was Decision Tree

SpaceX – Appendix-1

SpaceX Appendix-1

IBM Data Science Professional Certificate

[IBM Data Science Professional Certificate Program on Coursera]

- ➢ Data Collection API Lab Notebook
- ➢ Data Collection with Web Scrapping Lab Notebook
- ➢ EDA Lab Notebook
- ➢ EDA with Data Visualization Lab Notebook
- ➢ EDA with SQL Lab Notebook
- ➢ Interactive Dashboard App with Plotly Dash Lab Notebook
- ➢ Interactive Visual Analytics with Folium Lab Notebook
- ➢ Machine Learning Prediction Lab Notebook
- ➢ Applied Data Science Capstone SpaceX
  - ✓ SpaceX.csv
  - ✓ SpaceX-Web-Scraped.csv
  - ✓ Dataset-Part-1.csv
  - ✓ Dataset-Part-2.csv
  - ✓ Dataset-Part-3.csv

coursera
education for everyone

FALCON 9

SpaceX – Appendix-2

SpaceX Appendix-2

SpaceX

Coursera

IBM

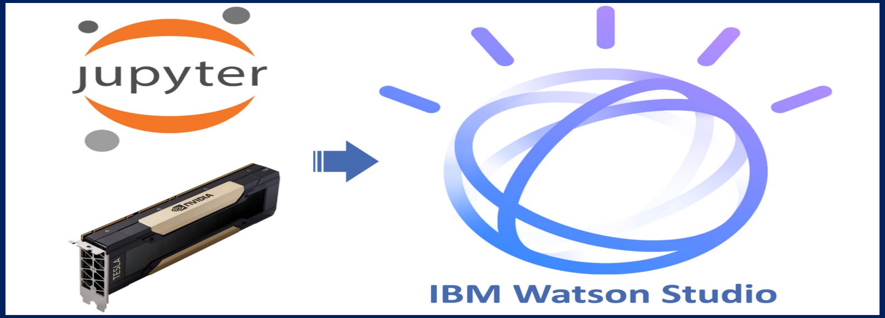
GitHub

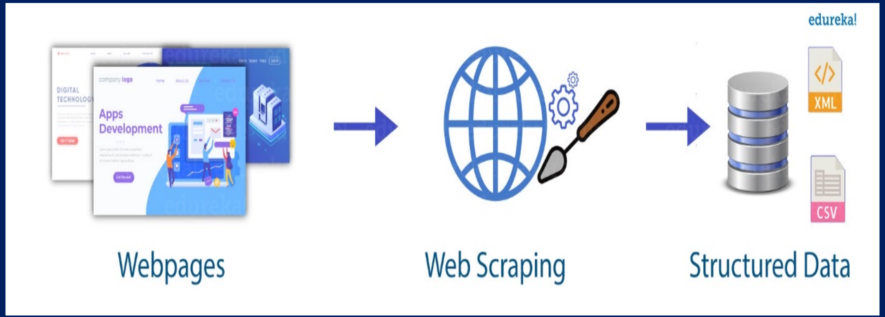
Python

Wikipedia

Jupyter Notebook

SpaceX – Appendix-3





SpaceX Appendix-3

Watson Studio

Postman

Web Scraping

Beautiful Soup

SQL

Plotly Dash

Folium

SpaceX – Thank You & Questions

Thank You

- Teachers for each Course
- IBM
- Coursera