IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yumiko Dunton
September 2021

Outline

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

Executive Summary

- Summary of methodologies
  - Gathered information about Space X and created dashboards identifying the success rate of each launch by site.
  - Public information was used to train a machine learning model to predict if Space X will reuse the first stage. This information can be used based on publicly available information about SpaceX launch cost to determine competitive viability.
- Summary of all results
  - Identified orbits with the highest success rates
  - Acquired names of the booster which have carried the maximum payload mass
  - Identified site that had the highest Launch Success Ratio and the payload affects on launch outcome

Introduction

- Project background and context
  - Space Y, founded by Billionaire industrialist Allon Musk, looks to compete with SpaceX with affordable, reusable first stage rockets.

- Problems you want to find answers
  - Determine affordability by calculating cost of each launch
  - Determine usability of first stage

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
  - Request to the SpaceX API
  - Web scraping related Wiki pages
- Perform data wrangling
  - Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- Data sets were collected via SpaceX REST API and webscraping public wiki pages related to launches.

Data Collection – SpaceX API calls flowchart

Launch data per url via requests

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

Request and parse the SpaceX launch data using the GET request

```
Response = Requets.get(spacex_url)
```

Apply get function method to get the booster version, Launch Site, Payload, and Core data

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

Data Collection – SpaceX API (cont)

Create dictionary launch_dict then dataframe launch_df from launch_dict

```
launch_dict = {'FlightNumber': ##etc
launch_df = pd.DataFrame.from_dict(launch_dict)
```

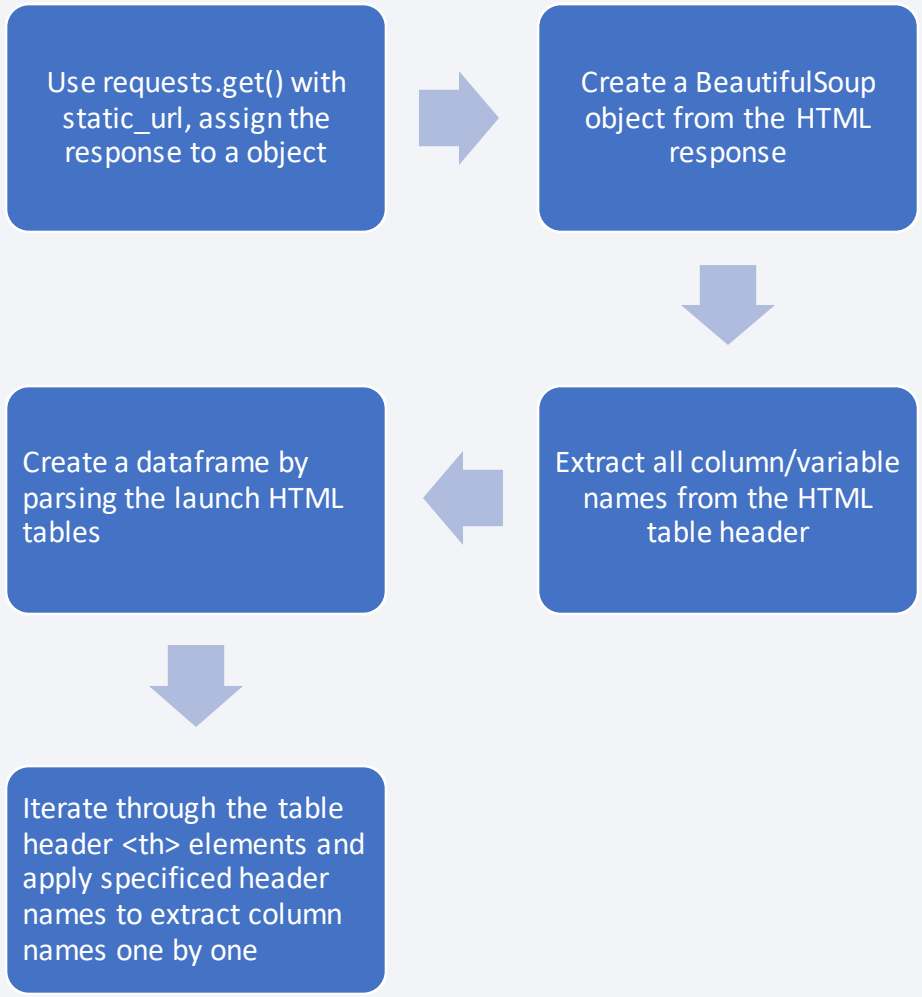
Filter dataframe to only include Falcon 9 launches

```
Click to add text
data_falcon9 = launch_df[launch_df.BoosterVersion == "Falcon 9"]
```

GitHub URL of the completed SpaceX API calls notebook

Data Collection – Scraping

- Using BeautifulSoup, performed web scraping to collect Falcon 9 historical launch records from a Wikipedia
- Extracted a Falcon 9 launch records HTML table from page titled "List of Falcon 9 and Falcon Heavy launches"
- Parsed the table and converted into a Pandas data frame



GitHub URL of the completed web scraping notebook

Data Wrangling

Performed Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

- Calculated the number of launch from each site
- Calculated the number and occurrence of each orbit achieved. [See Appendix 1 for definitions of various orbits]
- Calculated the number and occurrence of mission outcomes per orbit type
- Created a set of outcomes where the second stage did not land successfully.
- Calculated the success rate by calculating the mean of the successful landing outcomes.

GitHub URL of completed data wrangling related notebook

Apply value_counts() on column LaunchSite

```
df['LaunchSite'].value_counts()
```

Apply value_counts on Orbit column

```
df['Orbit'].value_counts()
```

Calculate the number & occurrence of mission outcome per orbit type

```
landing_outcomes=df['Outcome'].value_counts()
```

Create bad_outcomes set

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

Create a landing outcome label

```
landing_class=[0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
```

Calculate the success rate

```
Df["class'].mean()
```

EDA with Data Visualization

Summary of charts plotted

- Scatterplot to visualize the relationship between Flight Number and Launch Site
- Scatterplot to visualize the relationship between Payload and Launch Site
- Bar chart to visualize the Success Rate by Orbit Type
- Scatterplot to visualize the relationship between Flight Number vs Orbit Type
- Scatterplot to visualize the relationship between Payload vs Orbit Type
- Line chart to visualize the Launch Success Yearly Trend

GitHub URL of completed EDA with data visualization notebook

EDA with SQL

Summary of the SQL queries performed:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- 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 URL of completed EDA with SQL notebook

Build an Interactive Map with Folium

Summary of map objects such as markers, circles, lines, etc. added to a folium map:

- folium.Circle and folium.Marker for each launch site
- Markeres for all launch records, indicating success/failed launches for each site
- Polyline indicating the distances between a launch site to its proximities

Reasoning for use of added objects

- Identify locations of all launch sites, indicating success/failure and number of launches
- Determine launch sites' proximity to the Equator line, coast, railways, highways, cities
- Marker clusters placed to easily identify which launch sites have relatively high success rates

GitHub URL of completed interactive map with Folium notebook

Build a Dashboard with Plotly Dash

- Success Pie Chart indicates the proportion of successful launches by site or the ratio of success to failed for a selected site. This was indicated by the inclusion of a dropdown for site selection.
- A range slider was added so that the payload could be visualized on a scatter chart. The same dropdown as utilized for the pie chart selected the site for the scatter chart.

GitHub URL to completed Plotly Dash Lab

Predictive Analysis (Classification)

Built, evaluated, improved, and found the best performing classification model

- Performed exploratory Data Analysis and determined Training Labels
- Created a column for the class
- Standardize the data
- Split into training data and test data
- Found best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Found the best performing method using test data

Predictive Analysis (Classification)

Model development process flow

Create a NumPy array

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

Standardize the data in X then reassign it to the variable X using transform

```
transform = preprocessing.StandardScaler()
X = preprocessing.StandardScaler().fit(X).transform(X)
```

Split into training and test data

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,
Y, test_size=0.2, random_state=2)
```

*Create a logistic regression object then create a GridSearchCV object; fit to find the best parameters from the dictionary "parameters"

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'],
'solver':['lbfgs']}
lr=LogisticRegression()
grid_search=GridSearchCV(lr, parameters, cv=10)
logreg_cv=grid_search.fit(X_train, Y_train)
```

Calculate the accuracy using score

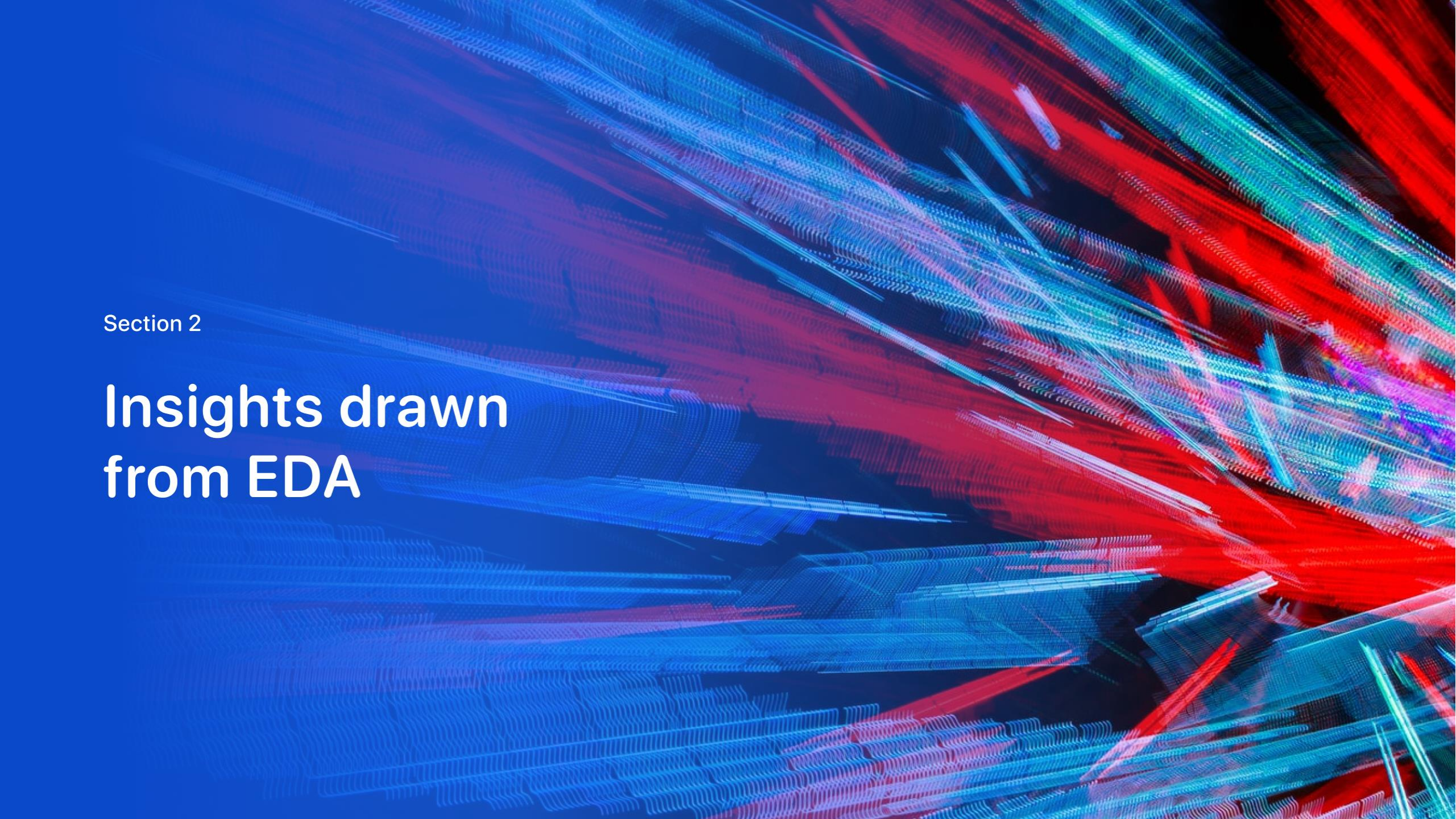
```
logreg_cv.score(X,Y)
```

Generate confusion matrix*

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

Repeat steps between ** for SVM, Decision Tree, KNN

[GitHub URL of completed predictive analysis lab]



Section 2

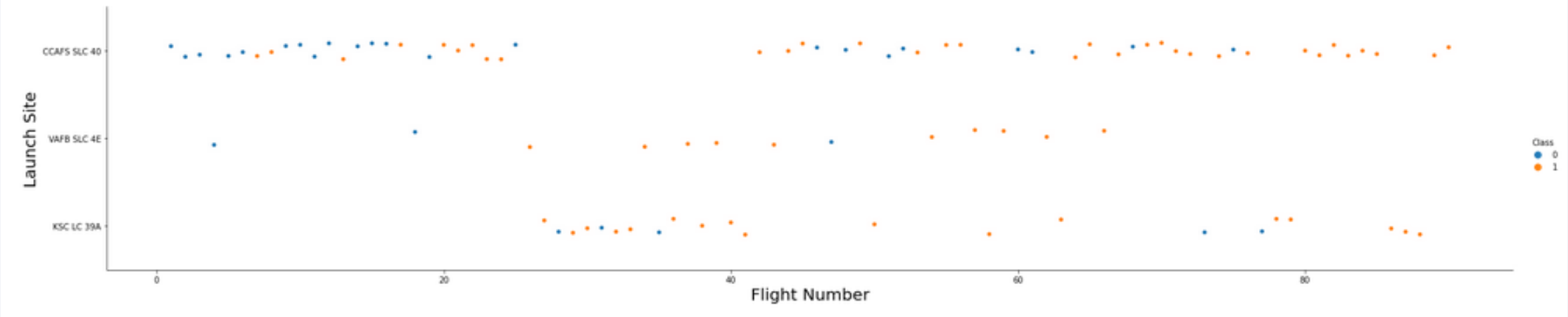
Insights drawn from EDA

Results

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

Flight Number vs. Launch Site

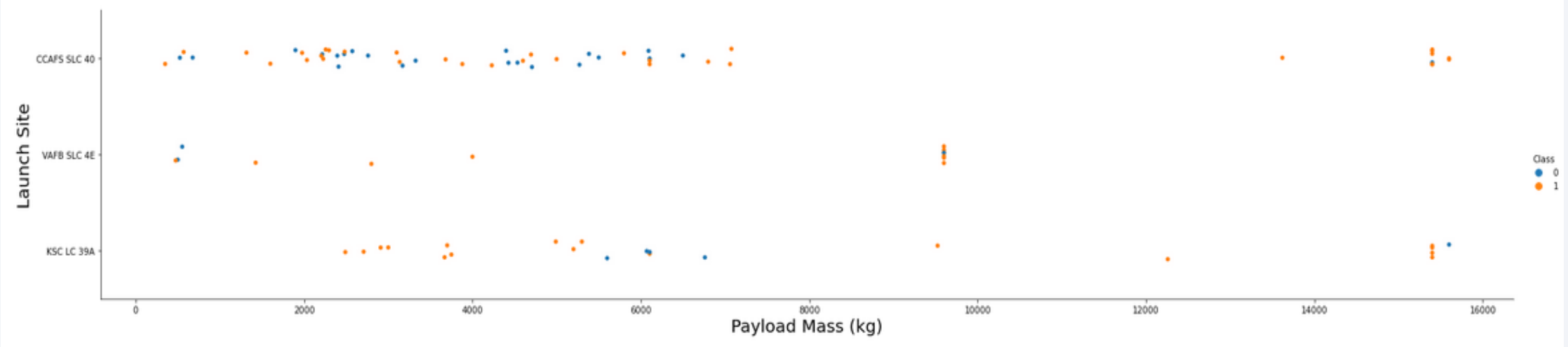
Scatter plot of Flight Number vs. Launch Site



Indicates the highest distribution of flights were from site CCAFS SLC 40

Payload vs. Launch Site

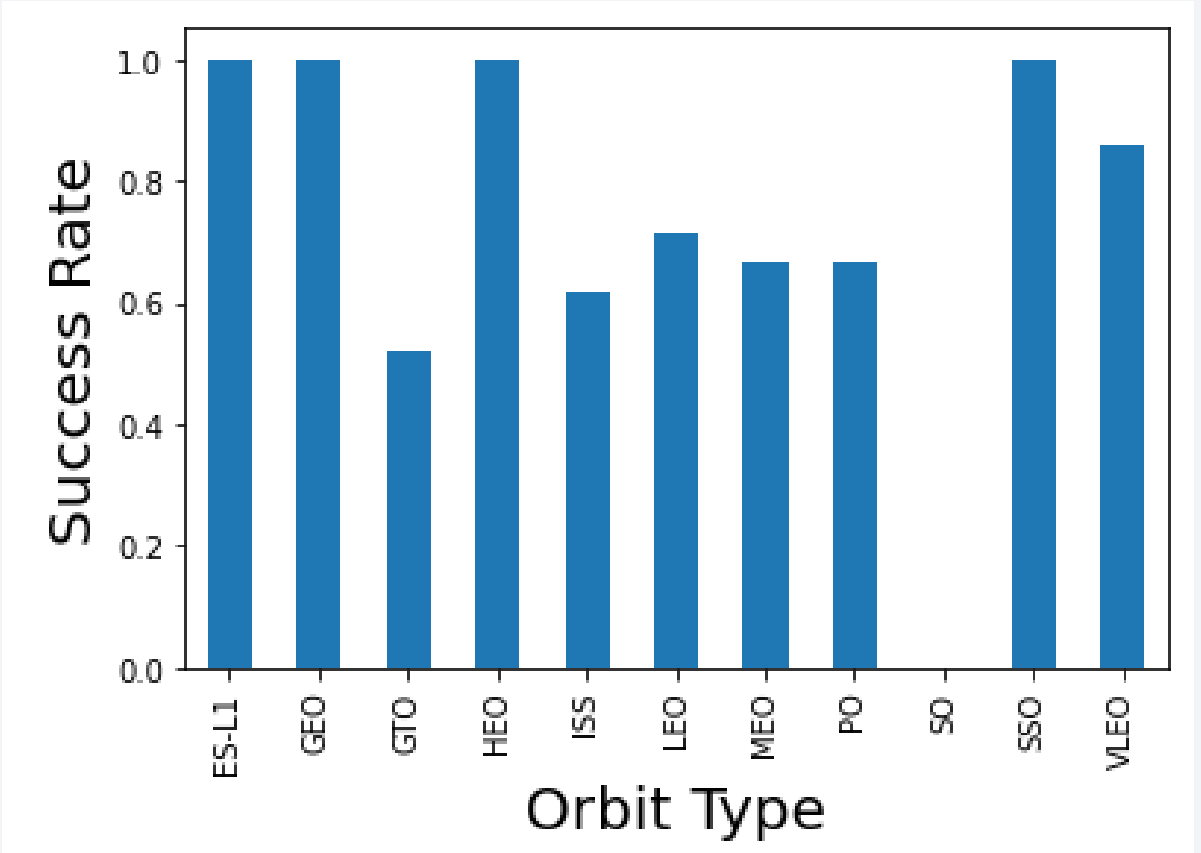
Scatter plot of Payload vs. Launch Site



- Launches with higher payloads were based out of CCAFS SLC40 and KSC LC39A, however most were unsuccessful
- There were more lighter payload launches out of CCAFS SLC40 up to about 7,000 kg

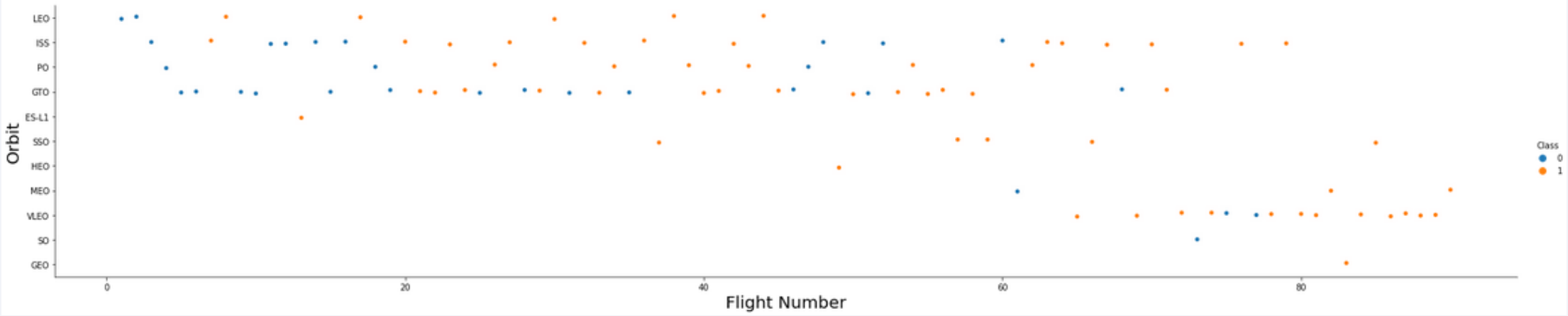
Success Rate vs. Orbit Type

- The orbits with the highest success rates are ES-L1, GEO, HEO, and SSO



Flight Number vs. Orbit Type

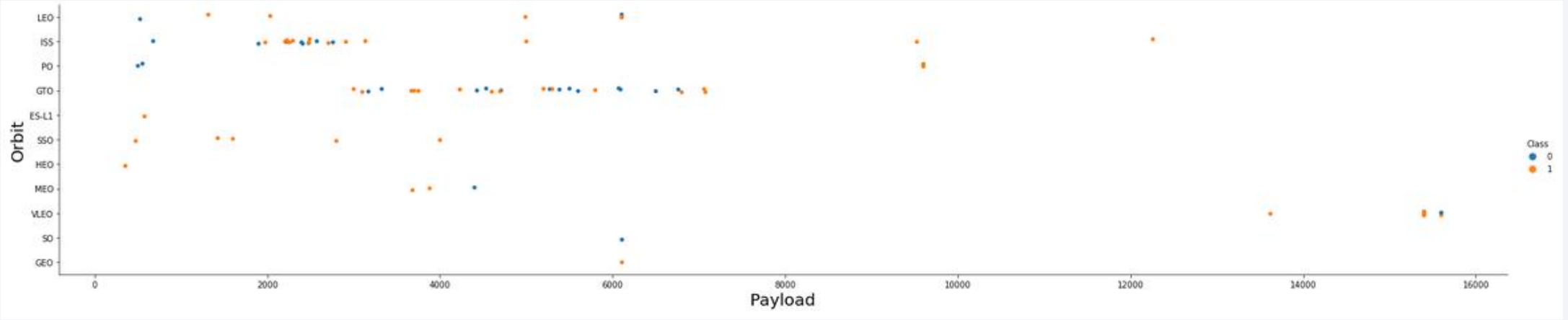
- Scatter plot of Flight number vs. Orbit type



- In the LEO orbit , success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

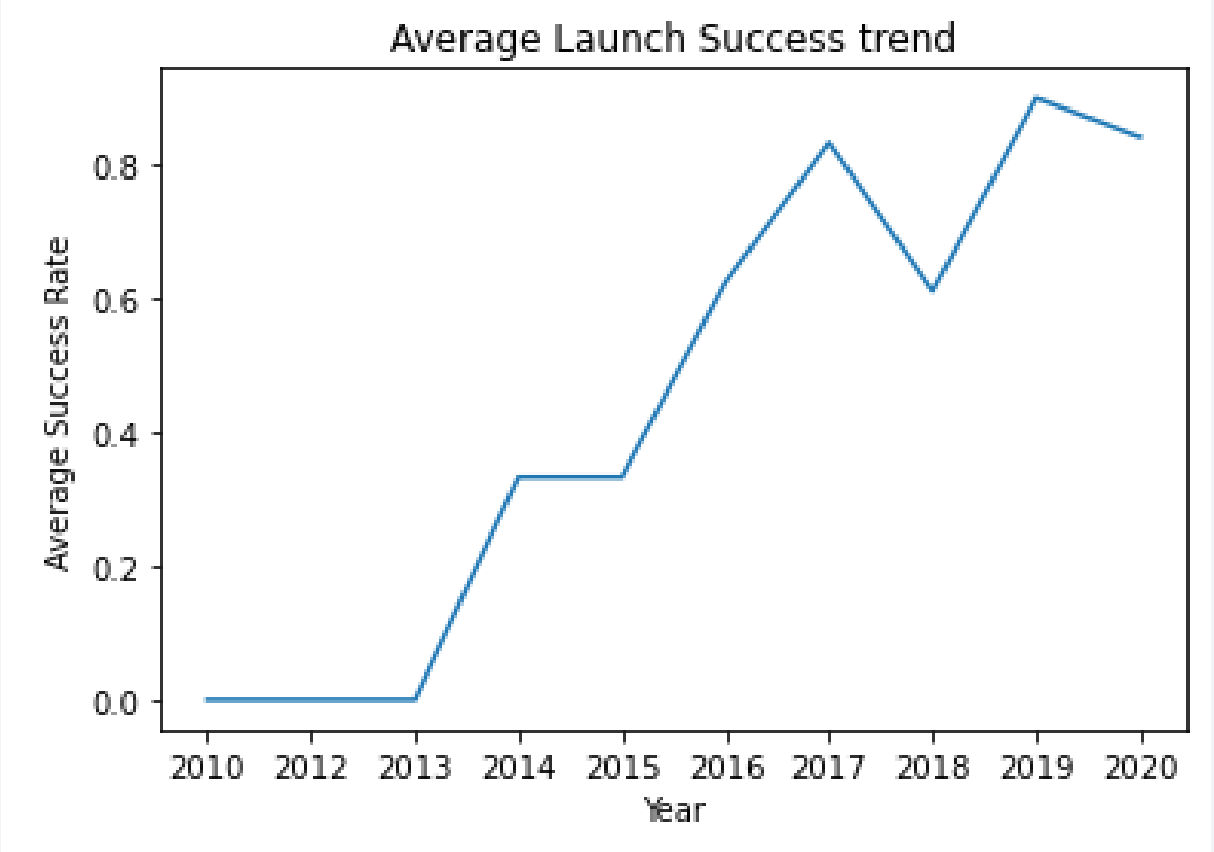
- Scatter plot of payload vs. orbit type



- Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend

- Line chart of yearly average success rate
- Success rate starting 2013 continued to increase until 2020



All Launch Site Names

- Queried the names of the unique launch sites to create a list of all launch sites

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA` via LIKE:

```
%sql select * from SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://hkr24117:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/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

Total Payload Mass

- Calculated the total payload carried using SUM and by boosters from NASA using WHERE

```
%sql select sum(payload_mass__kg_) FROM SPACEXTBL where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://hkr24117:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lc
Done.
```

1
45596

Average Payload Mass by F9 v1.1

- Calculated the average payload mass carried by booster version F9 v1.1 using AVG

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

```
%sql select avg(payload_mass__kg_) FROM SPACEXTBL where booster_version like 'F9 v1.1%'
```

```
 * ibm_db_sa://hkr24117:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.d
Done.
```

1
2534

First Successful Ground Landing Date

- Identified dates of the first successful landing outcome on ground pad using MIN based on landing outcomes containing "Success" using LIKE

```
%sql select min(DATE) FROM SPACEXTBL where landing__outcome like 'Success%'
```

```
 * ibm_db_sa://hkr24117:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lq
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Identified the names of boosters which have successfully landed on drone ship using where and had payload mass greater than 4000 but less than 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__kg_<=6000
```

```
 * ibm_db_sa://hkr24117:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculated the total number of successful and failure mission outcomes using GROUP BY

List the total number of successful and failure mission outcomes

```
%sql select mission_outcome, count(mission_outcome) FROM SPACEXTBL group by mission_outcome
```

```
 * ibm_db_sa://hkr24117:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

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

Boosters Carried Maximum Payload

- Listed the names of the booster which have carried the maximum payload mass using subquery

```
%sql select distinct booster_version FROM SPACEXTBL where payload_mass__kg_ =(select max(payload_mass__kg_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://hkr24117:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

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

2015 Launch Records

- Queried 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 YEAR(DATE)='2015'
```

Query result:

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

Both failed landing_outcomes in 2015 occurred at launch site CCAFS LC-40.

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

- 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 via following:

```
%sql select landing__outcome, count(*) nb FROM SPACEXTBL where DATE >='2010-06-04' and DATE <='2017-03-20'  group by landing__outcome order by 2 desc
```

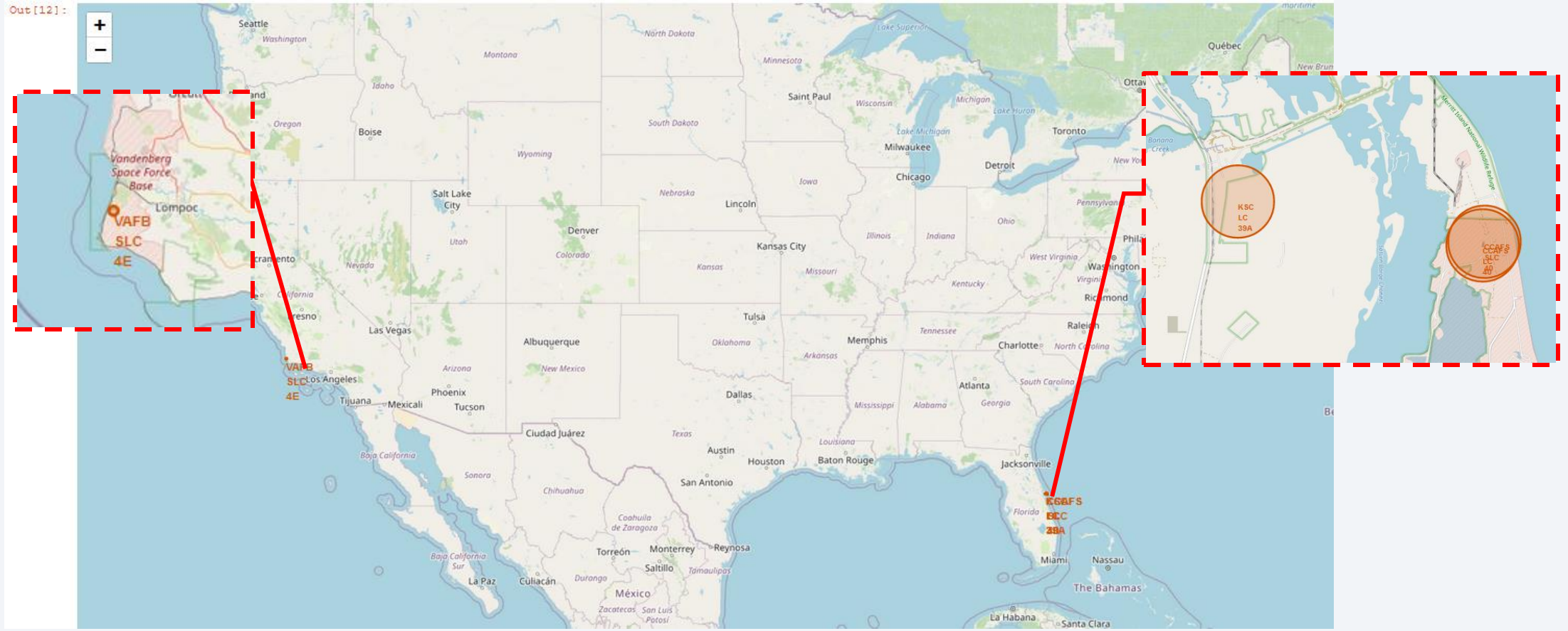
Query result :

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

Section 4

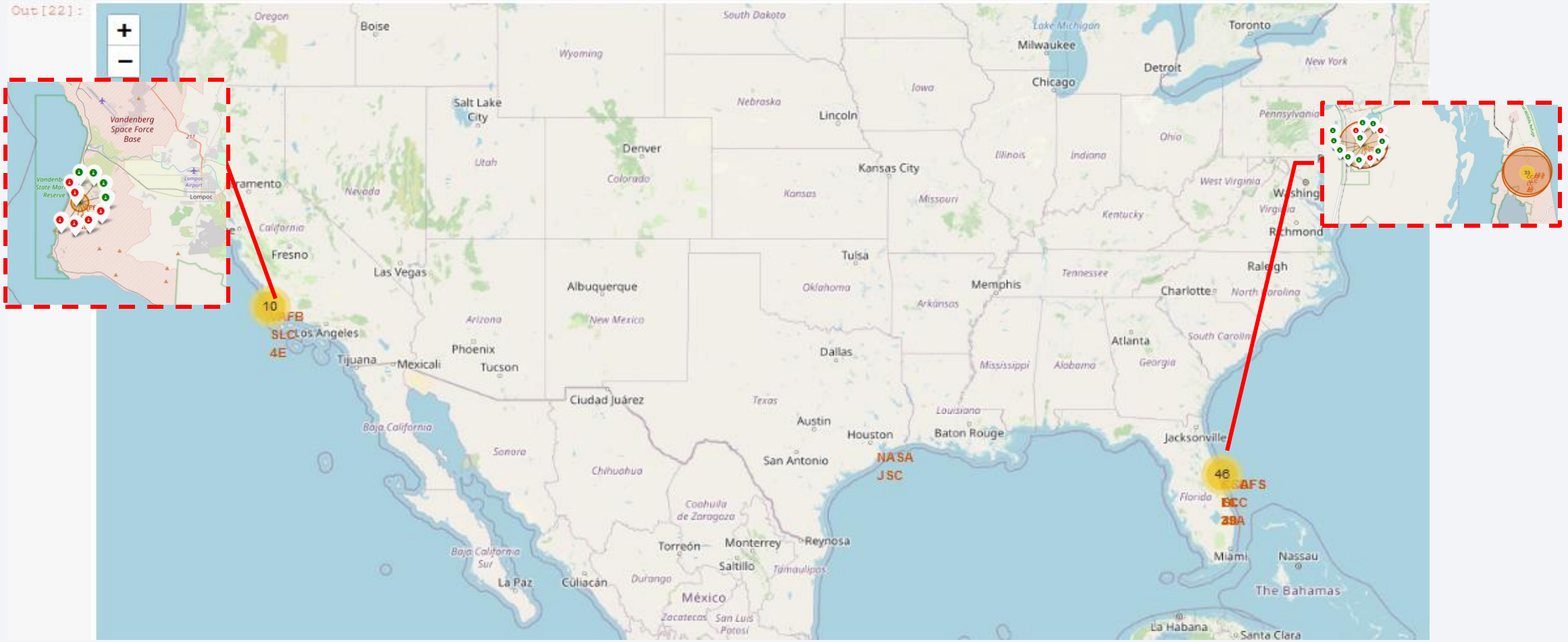
Launch Sites Proximities Analysis

Folium Map: All Launch Sites



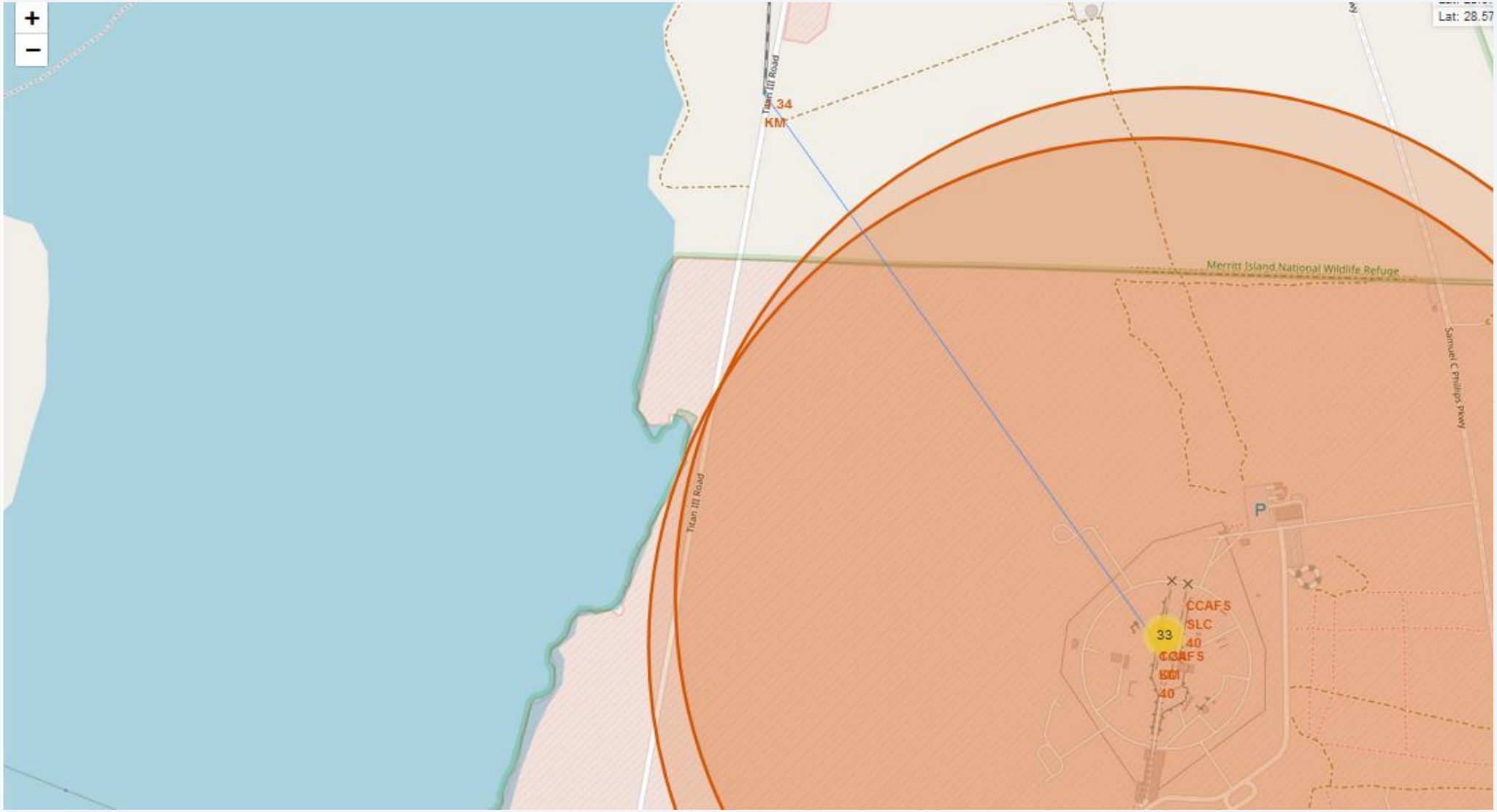
- Screenshot of all launch sites + detail

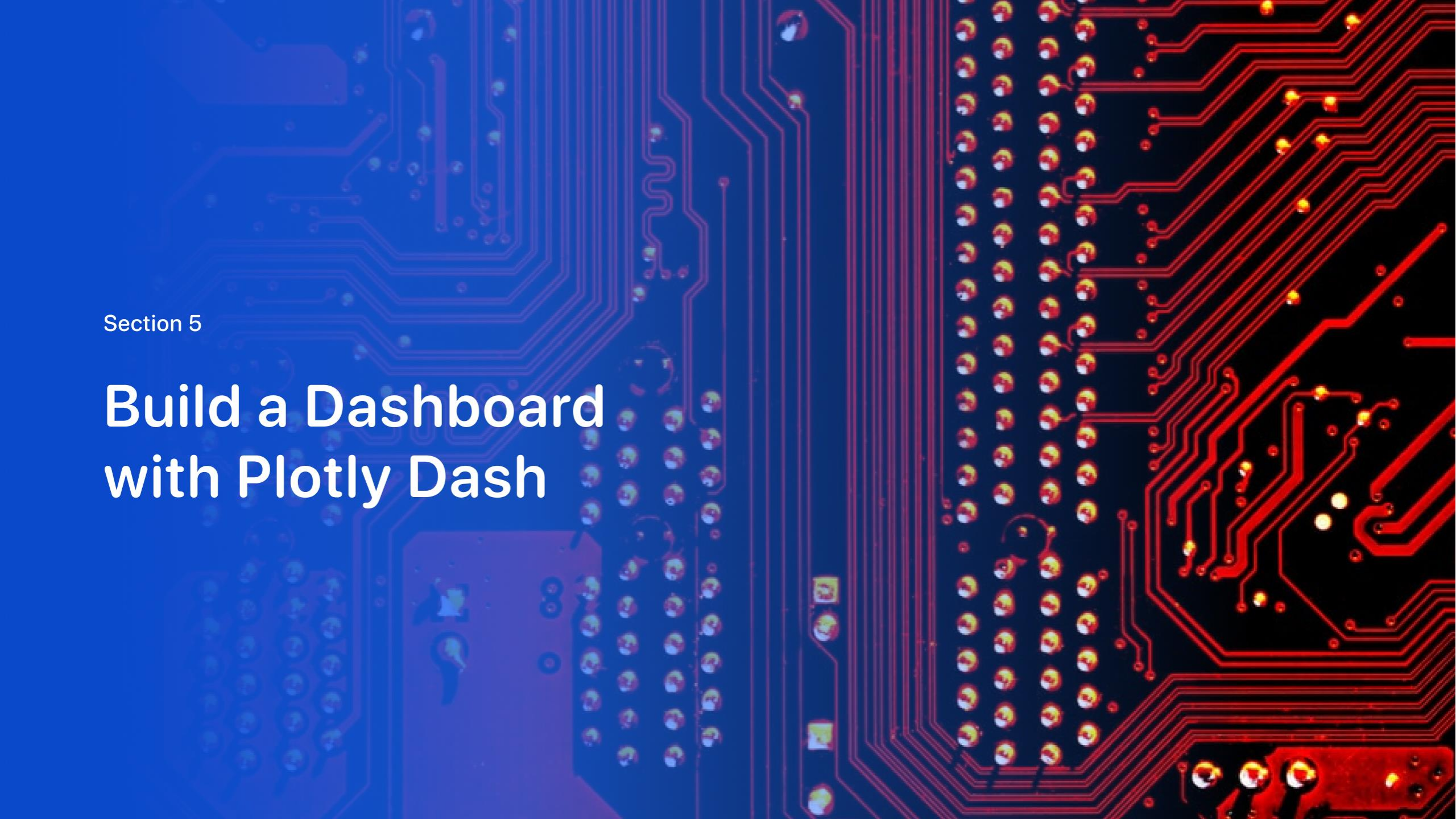
Success-Fail Marker-Clusters by Site Screenshot



Markers for all launch records were created. Green indicates successful (class=1) launches, Red Indicates failed (class=0) launches.

Proximity to nearest rail line screenshot





Section 5

Build a Dashboard with Plotly Dash

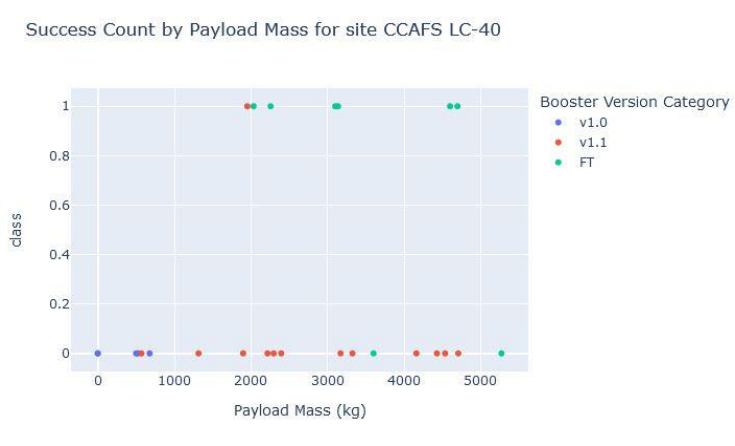
Success Rate for All Sites – Dashboard 1

Shows the portion of all successful launches by site

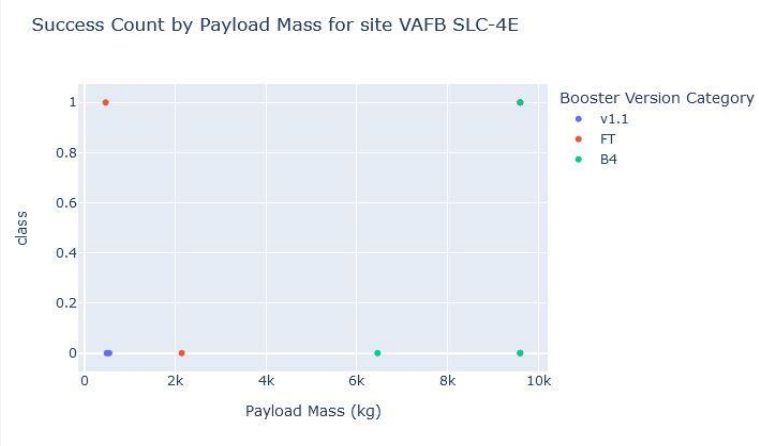
Highest Launch Success Ratio – Dashboard 2

Site KSC LC-39A had the highest Launch Success Ratio

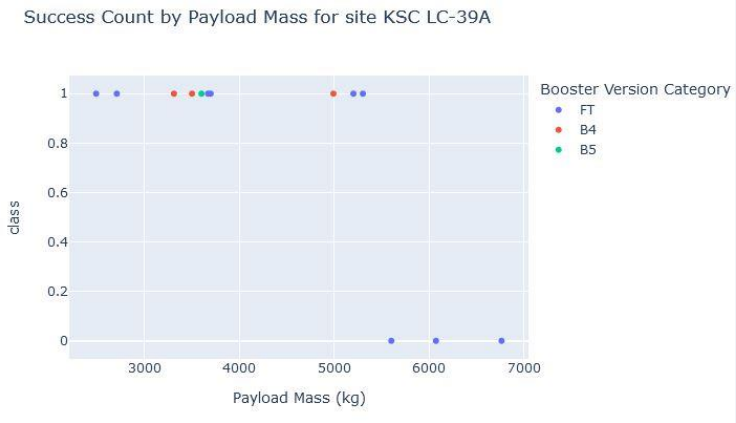
Payload vs. Launch Outcomes – All Sites Dashboard Screenshot 3



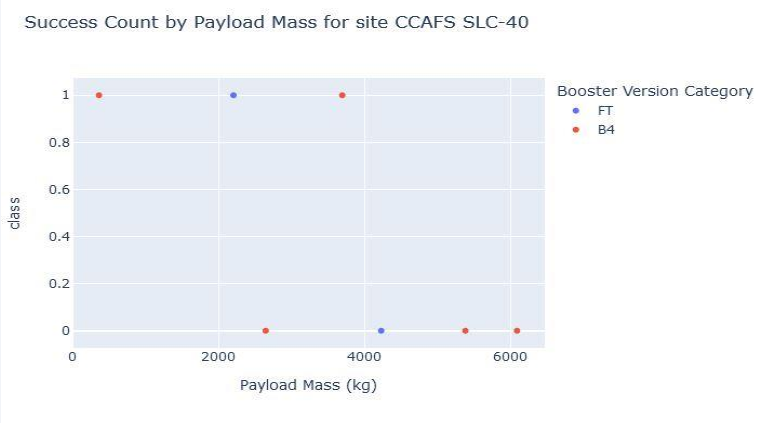
Max successful payload from this site is just over 5,000 (kg) with booster version FT



Mostly unsuccessful launches of varying payloads



Payloads over ~5,500 (kg) booster version FT were not successful out of site KSC LC39-A while all other launches and payloads were successful



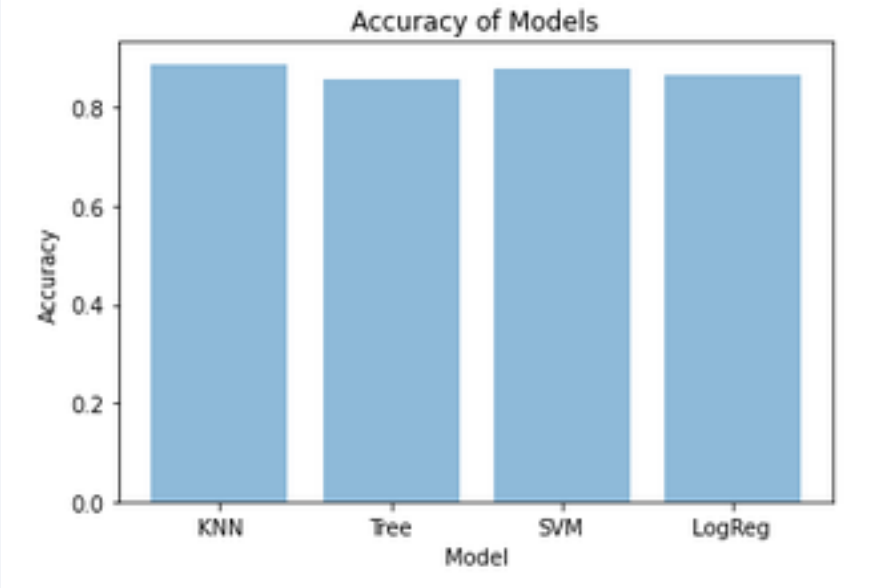
More successful lower payload launches on B4 versions boosters

Section 6

Predictive Analysis (Classification)

Classification Accuracy

- Model Accuracy score



- Jaccard, F1 and Log Loss by Model

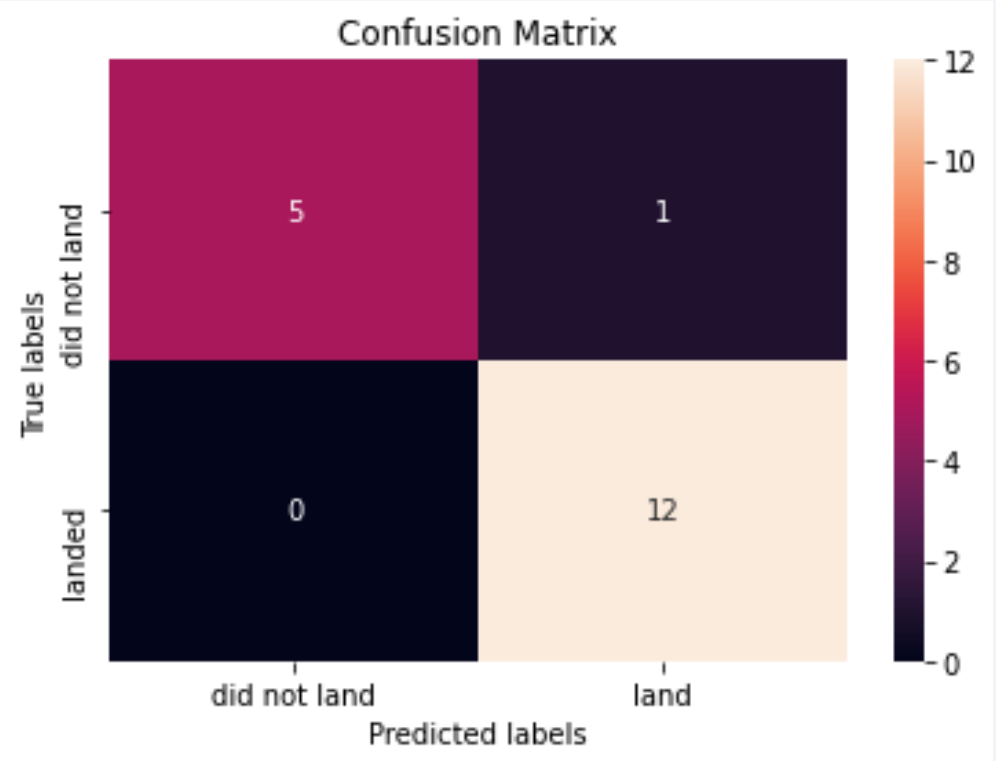
Algorithm	Jaccard	F1-Score	Log Loss
KNN	0.923077	0.943030	NA
Decision Tree	0.800000	0.814815	NA
SVM	0.800000	0.814815	NA
Logistic Regression	0.800000	0.814815	0.478667

Confusion Matrix for KNN

There were no launches that were predicted not to land that actually landed.

There were 5 that were predicted not to land out of a total of 6; one was predicted to land that did not actually land.

There were 12 launches that were predicted to land that actually landed and 1 that was predicted to land that did not actual land.



Conclusions

- The orbits with the highest success rates are ES-L1, GEO, HEO, and SSO
- Falcon 9 Booster category which carried the maximum payload mass is B5
- Site KSC LC-39A had the highest Launch Success Ratio
- Site KSC LC-39A has successfully launched booster version B5
- The best ML model for this data is KNN.

Appendix

- Link to github for SpaceX dataset

Thank you!