IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Darrell Larsen
Feb. 12, 2023

Outline

Executive Summary

- Methodologies
  - Data collection via SpaceX API and webscraping
  - Data wrangling with pandas
  - Data analysis with SQL
  - Data visualization with seaborn, folium
  - Predictive analysis with scikit-learn
- Results
  - Interactive visualizations with Dash
  - Identified optimum predictive model

Introduction

- SpaceX
  - Highly successful private manufacturer of rockets and spacecraft
  - Advertises Falcon 9 launches at $62 million—$100 million *less* than some competitors
  - Low cost due to re-usable expensive Stage 1 rockets
  - Landing success of Stage 1 rockets correlates with payload, orbit, etc.
- SpaceY
  - To compete with SpaceX, we wish determine the price of each launch. To do so, we will use machine learning to predict whether Stage 1 rockets will land successfully based on publicly available data.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected via SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - Handled missing values and derived new column
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection – SpaceX API

Initial SpaceX API request

```
response = requests.get(spacex_url)
```

Convert json data into dataframe

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

Use functions for additional API requests

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

Create dictionary from data lists

```
launch_dict = {'list1': list1, 'list2': list2, ...}
```

Convert dictionary into dataframe

```
launch_df = pd.DataFrame(launch_dict)
```

https://github.com/ibmcoursestudent/ibm_data_science_capstone

Data Collection - Scraping

1. Request data from Wikipedia
   `response = requests.get(wiki_url)`
2. Create BeautifulSoup object
   `soup = BeautifulSoup(response.content)`
3. Extract tables from html
   `html_tables = soup.find_all('table')`
4. Extract column names (looped)
   `col_name = extract_column_from_header(th)`
5. Create dictionary from column names
   `launch_dict= dict.fromkeys(column_names)`
6. Loop over rows to populate dictionary values
   `launch_dict['column1'].append(column1)`
   `launch_dict['column2'].append(column2)`
   ...
7. Convert into dataframe
   `df=pd.DataFrame(launch_dict)`

https://github.com/ibmcoursestudent/ibm_data_science_capstone

Data Wrangling

1. PayloadMass: replace null values with mean

```
df.isnull().sum()

FlightNumber      0
Date              0
BoosterVersion    0
PayloadMass       5
Orbit             0
```

```
# Calculate the mean value of PayloadMass column
mean_payloadmass = df['PayloadMass'].mean()

# Replace the np.nan values with its mean value
df['PayloadMass'].replace(np.nan, mean_payloadmass, inplace=True)
```

```
df.isnull().sum()

FlightNumber      0
Date              0
BoosterVersion    0
PayloadMass       0
Orbit             0
```

2. Derive column for success from Outcome

```
# View unique Outcomes
df['Outcome'].value_counts()

True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
```

```
# landing_class = 0 if bad outcome, 1 if good
landing_class = []
for x in df.loc[:,'Outcome']:
    landing_class.append(x.startswith('True'))
```

```
# Add class to dataframe
df['Class']=landing_class
```


https://github.com/ibmcoursestudent/ibm_data_science_capstone

EDA with Data Visualization

- Scatterplots (plotting values against each other to show dependency)
  - Flight Number and Launch Site
  - Payload and Launch Site
  - Flight Number and Orbit Type
  - Payload and Orbit Type
- Bar Chart (comparing proportional values of discrete categories of data)
  - Success Rate by Orbit Type
- Line Chart (visualizing time-dependent variables to show trends)
  - Yearly Success Rate

https://github.com/ibmcoursestudent/ibm_data_science_capstone

EDA with SQL

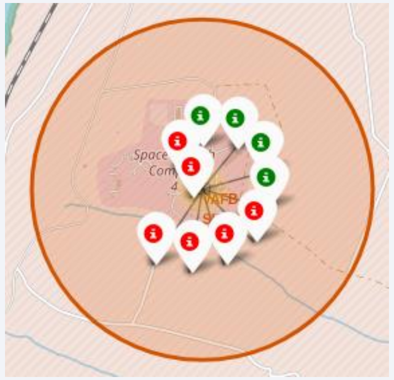
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first succesful landing outcome in ground pad was acheived.
- 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 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 successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

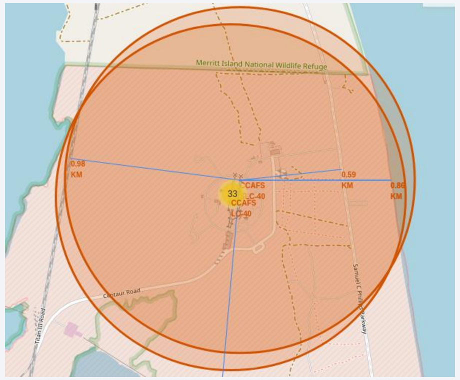
https://github.com/ibmcoursestudent/ibm_data_science_capstone

Build an Interactive Map with Folium

Markers and circles identify launch sites

Marker clusters show successful (green) and unsuccessful (red) launches.



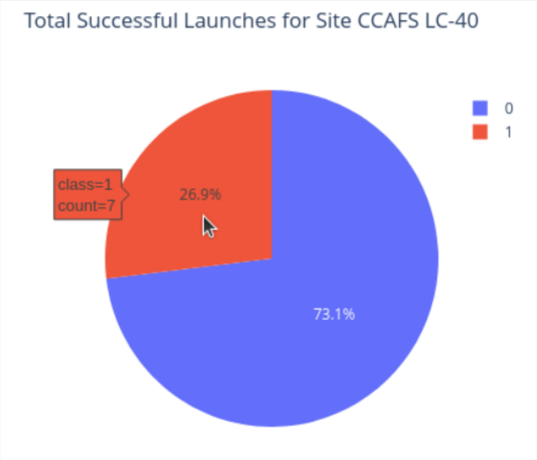


Lines mark distance to nearest railroad, highway, coast, and city.

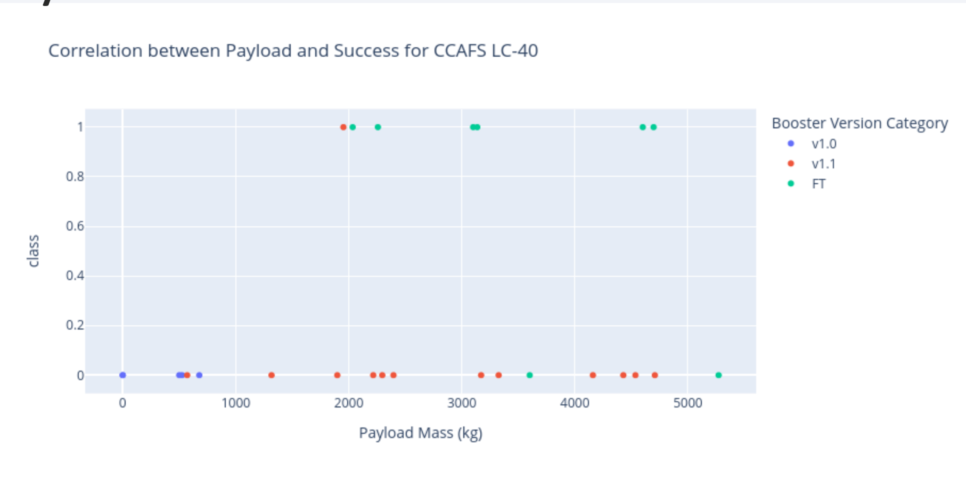
https://github.com/ibmcoursestudent/ibm_data_science_capstone

Build a Dashboard with Plotly Dash

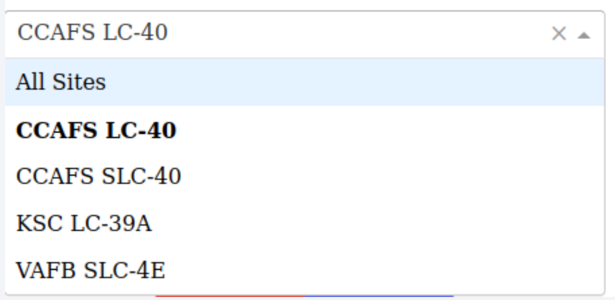
Pie chart shows proportion of successful launches



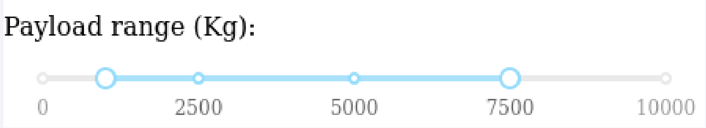
Scatterplot shows correlation between payload and success



Select site with dropdown menu



Select payload range with slider



https://github.com/ibmcoursestudent/ibm_data_science_capstone

Predictive Analysis (Classification)

1. Convert target column into numpy array Y
2. Standardize the data in X
3. Split into training and test sets
4. For each model:
   a. Set search parameters
   b. Create GridSearchCV
   c. Fit model with training data
   d. Identify best parameters
   e. Check accuracy on training data
   f. Calculate accuracy on test data
   g. Compare predictions with actual scores

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

X = preprocessing.StandardScaler().fit_transform(X)

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

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

model_cv = GridSearchCV(Model(), parameters, cv=10)

model_cv.fit(X_train, Y_train)

print(model_cv.best_params_)

print(logreg_cv.best_score_)

model_cv.score(X_test, Y_test)

yhat=model_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat) # custom function
```


https://github.com/ibmcoursestudent/ibm_data_science_capstone

Results - Outline

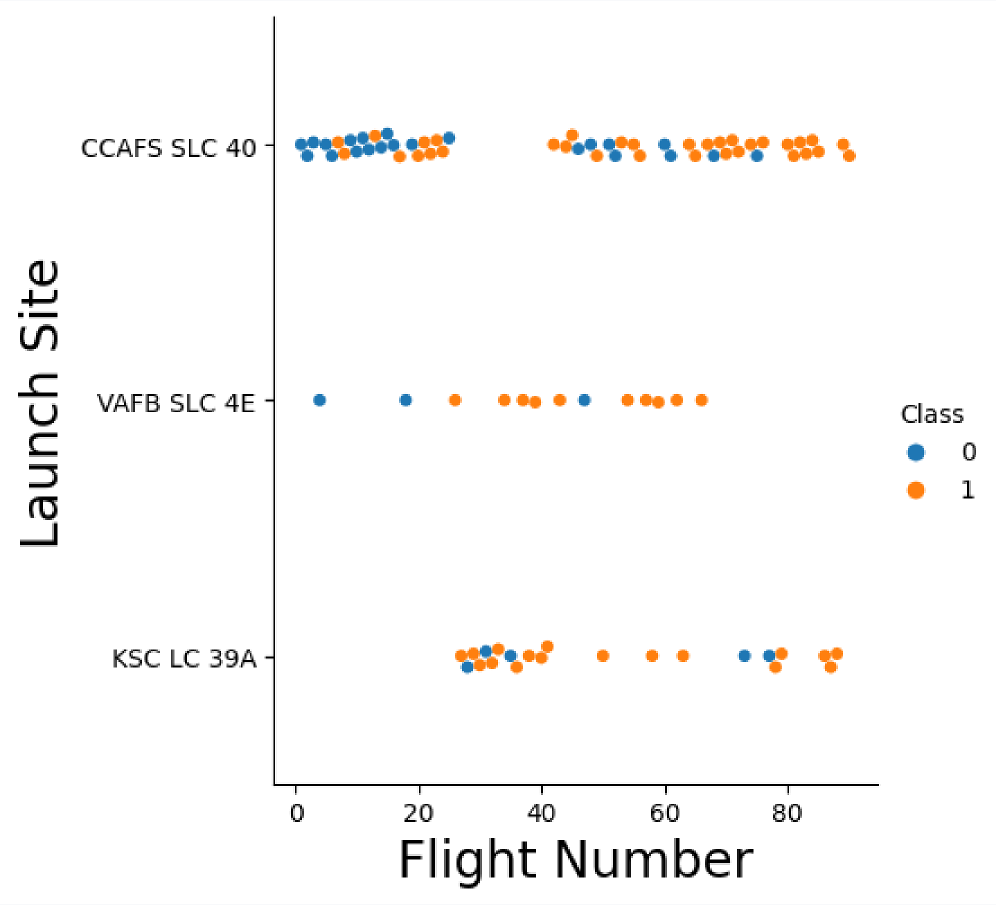
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

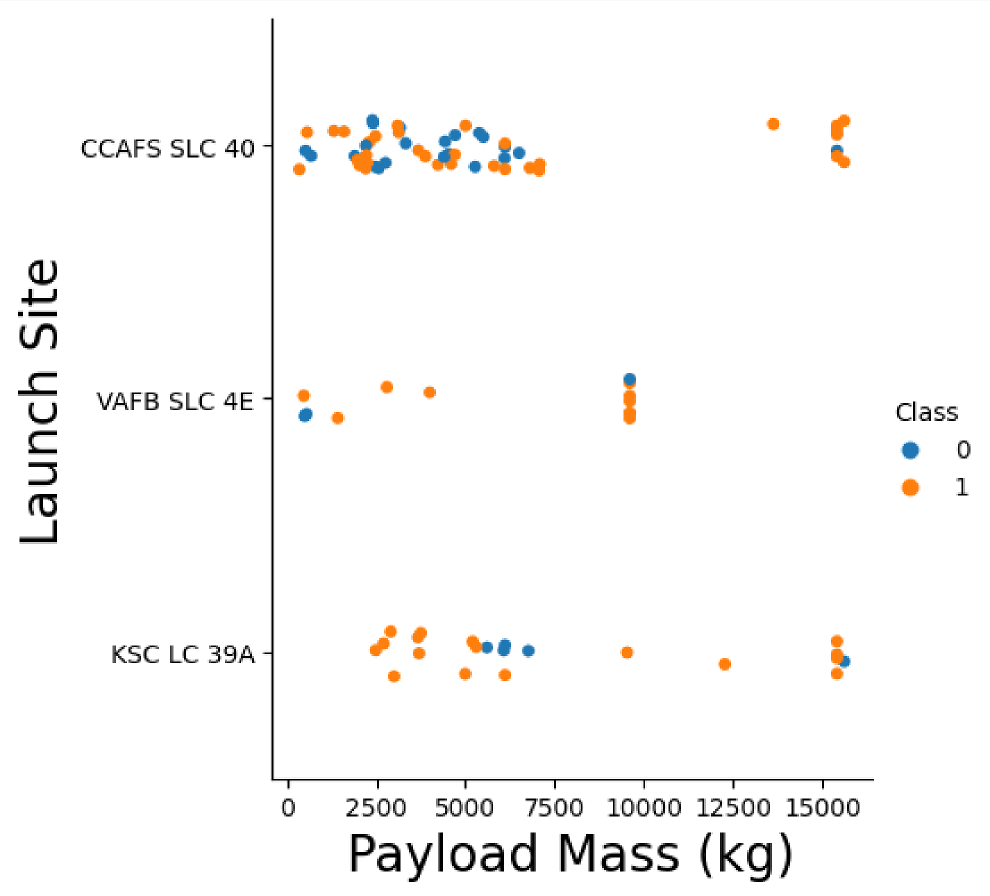
Flight Number vs. Launch Site

- Most used launch site: CCAFS SLC40
- Least used launch site: VAFB SLC 4E
- Overall: higher flight number → greater chance of success



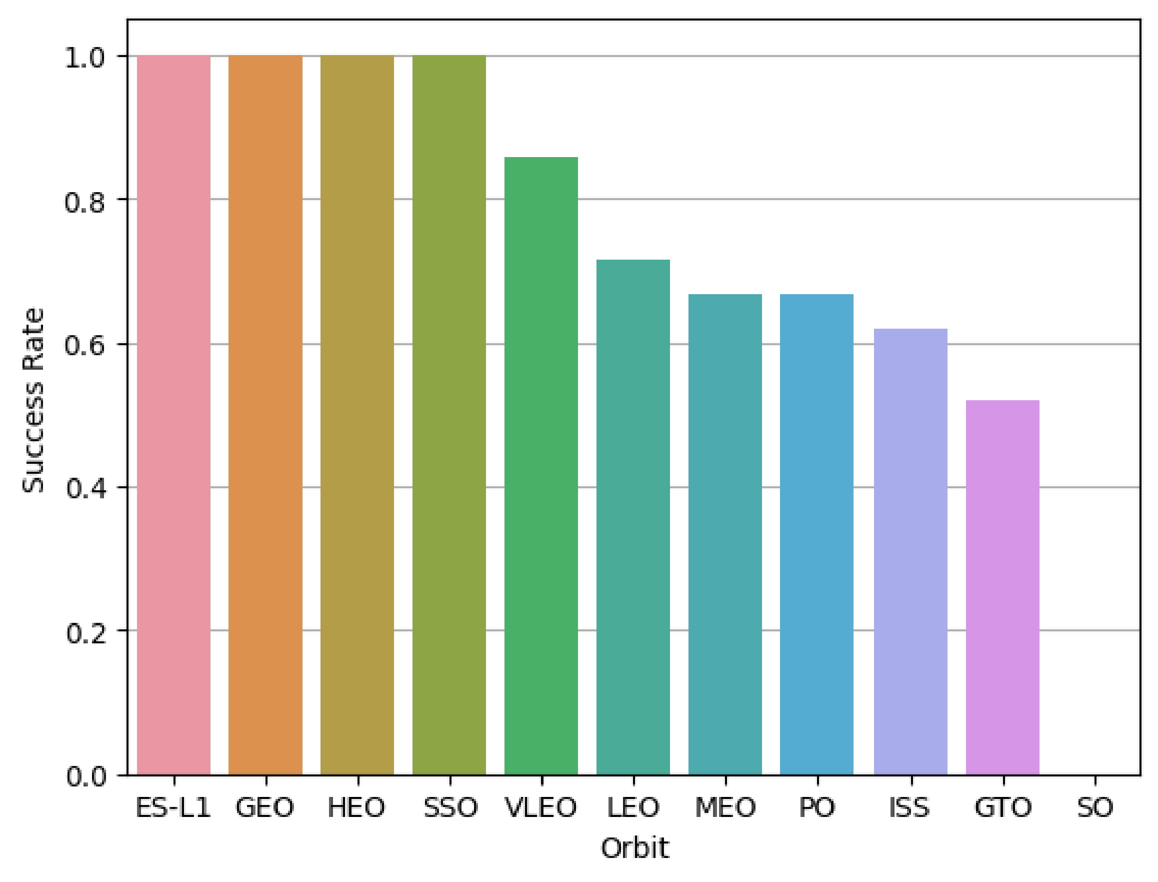
Payload vs. Launch Site

- No payloads > 10,000kg at VAFB SLC 4E



Success Rate vs. Orbit Type

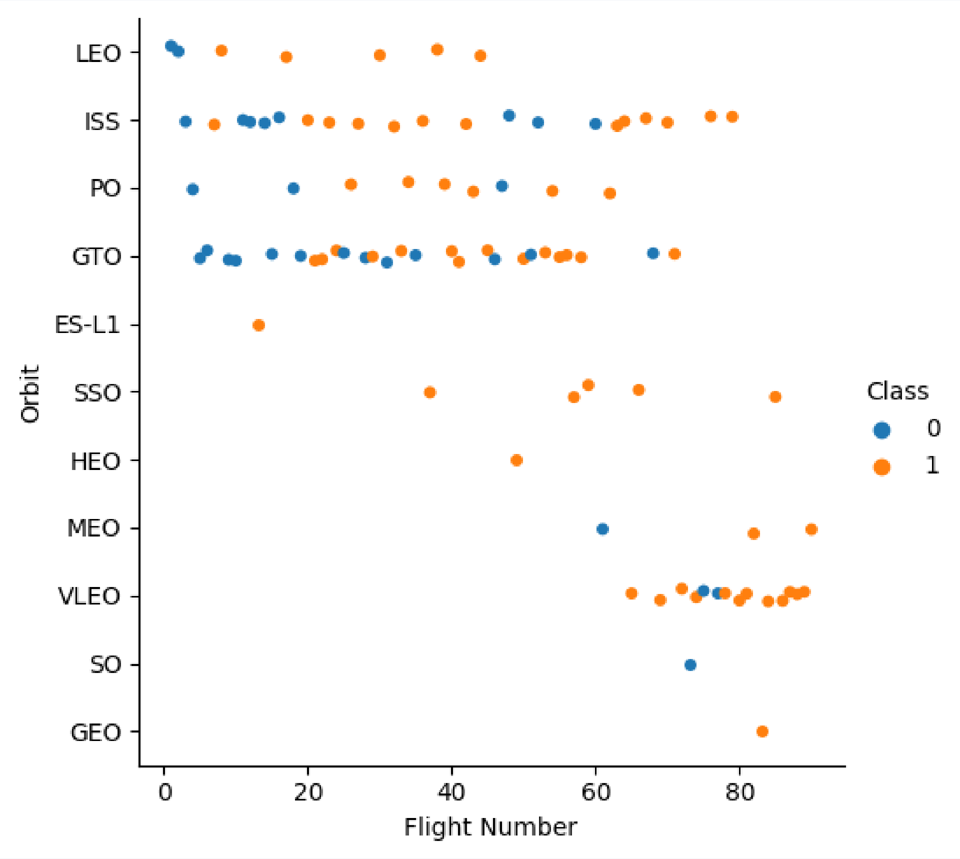
- Highest success rate:
  - ES-L1 (1 launch)
  - GEO (1 launch)
  - HEO (1 launch)
  - SSO (5 launches)
- Lowest success rate:
  - SO (1 launch)



Flight Number vs. Orbit Type

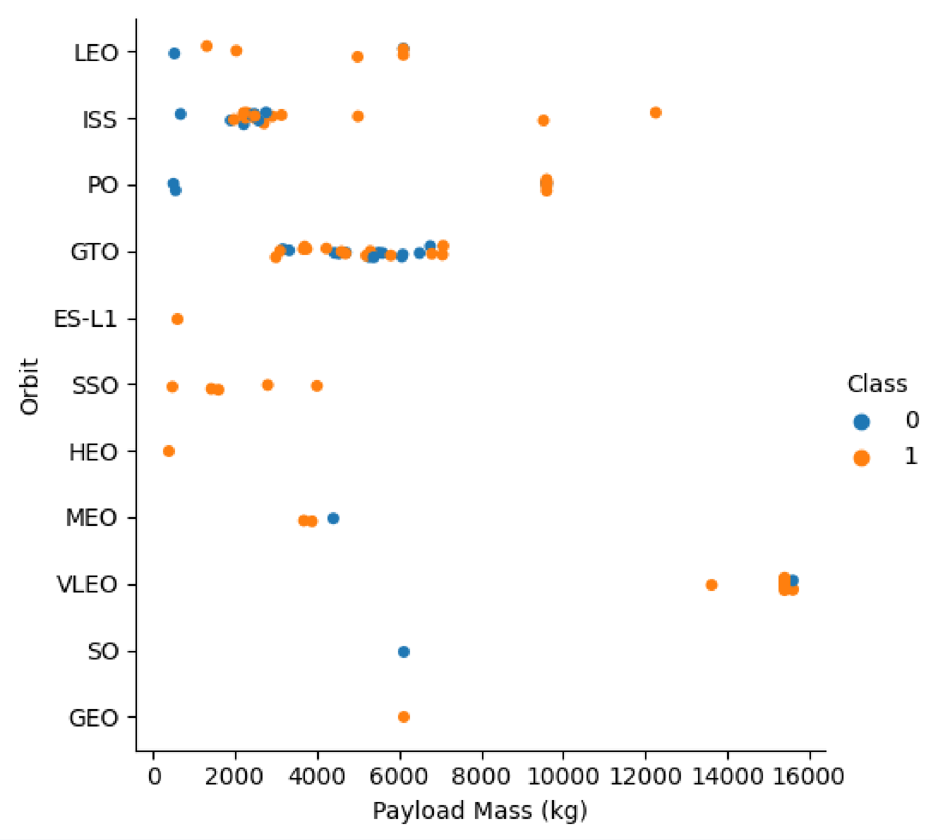
- Overall, higher success rate as flight number increases

- No correlation evident for GTO orbit



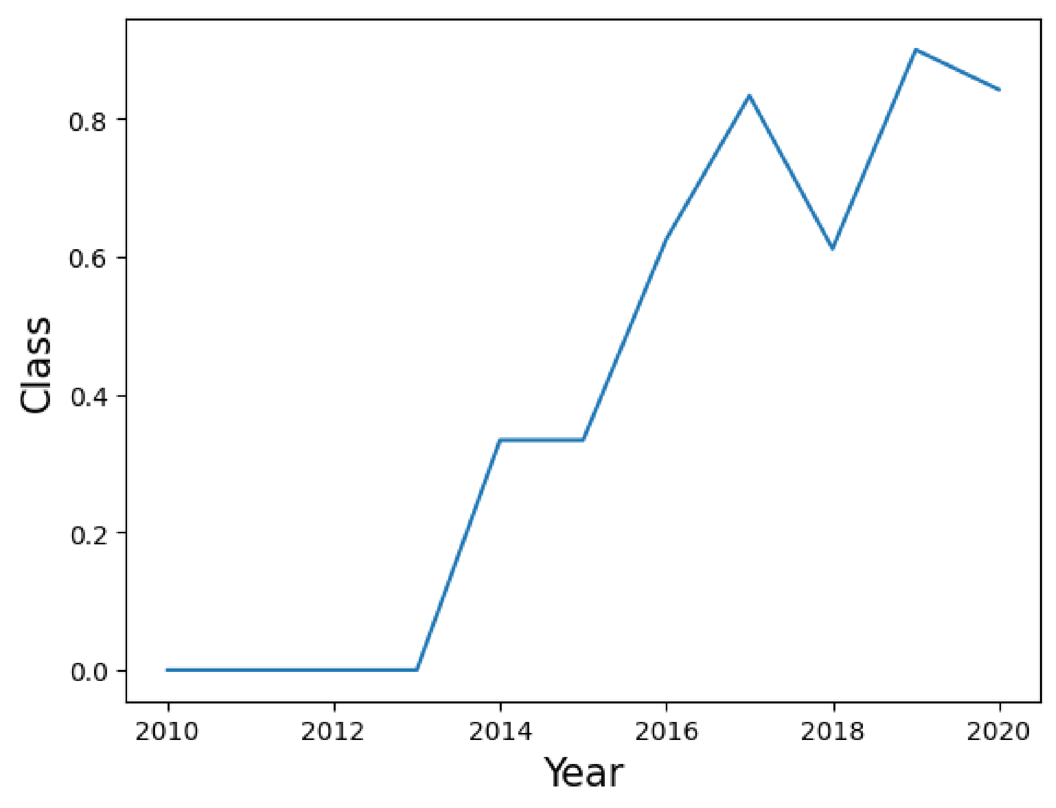
Payload vs. Orbit Type

- Heavier payloads correlate with greater success in LEO, ISS, and PO



Launch Success Yearly Trend

- Sharp increase in success rate from 2013-2017
- Dip in 2018
- Rise to highest point in 2019



Exploratory Data Analysis with SQL Queries

LAUNCH SITES

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

All Launch Site Names

```
select distinct launch_site
from spacextbl;
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Launch Site Names Begin with 'CCA'

```
select *
from spacextbl
where launch_site like 'CCA%'
limit 5;
```

TOTAL PAYLOAD MASS (KG)

45596

Total Payload Mass

```
select sum(payload_mass__kg_)
from spacextbl
where customer = 'NASA (CRS)';
```

AVERAGE PAYLOAD MASS (KG)

2928.4

Average Payload Mass by F9 v1.1

```
select avg(payload_mass__kg_)
from spacextbl
where booster_version = 'F9 v1.1';
```

min(DATE)
2015-12-22

First Successful Ground Landing Date

```
update spacextbl set date = substr(date, 7,4) || '-' || substr(date, 4,2) || '-' || substr(date, 1,2); # convert date format

select min(DATE) from spacextbl where "landing _outcome" = 'Success (ground pad)'; # query
```

BOOSTER VERSION
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Successful Drone Ship Landing with Payload between 4000 and 6000

```
select distinct booster_version

from spacextbl

where "landing _outcome" = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000;
```

Mission Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Total Number of Successful and Failure Mission Outcomes

- select distinct mission_outcome, count(*) as total
- from spacextbl
- group by mission_outcome;

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

Boosters Carried Maximum Payload

```
select distinct booster_version

from spacextbl

where payload_mass__kg_ = (select max(payload_mass__kg_)   from spacextbl);
```

Landing Outcome	Booster Version	Launch Site	Month
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	01
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	04

2015 Launch Records

```
select "landing _outcome", booster_version,
launch_site, substr(date, 6,2) as 'month'

from spacextbl

where substr(date,1,4) = '2015' and "landing _outcome" like
'Failure (drone ship)';
```

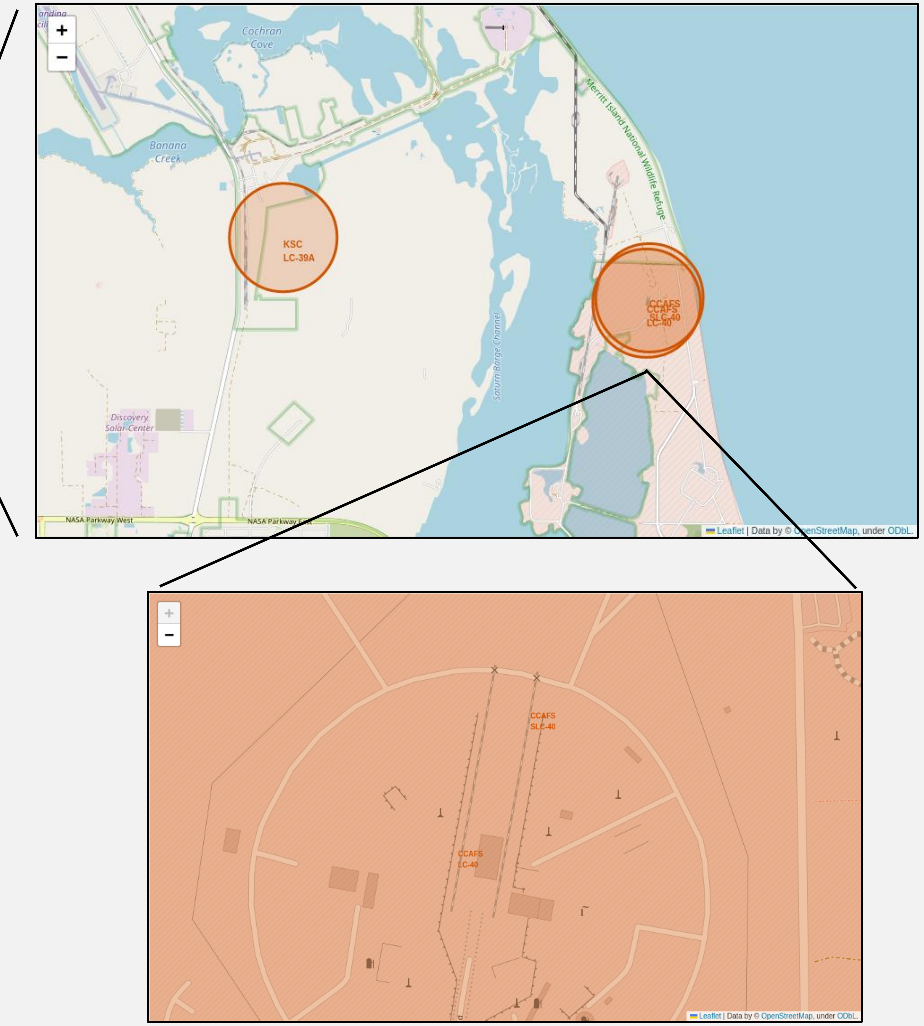
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

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

```
select "landing_outcome", count("landing_outcome") from spacextbl
where date between '2010-06-04' and '2017-03-20'
group by "landing_outcome" order by count("landing_outcome") desc;
```

Section 3

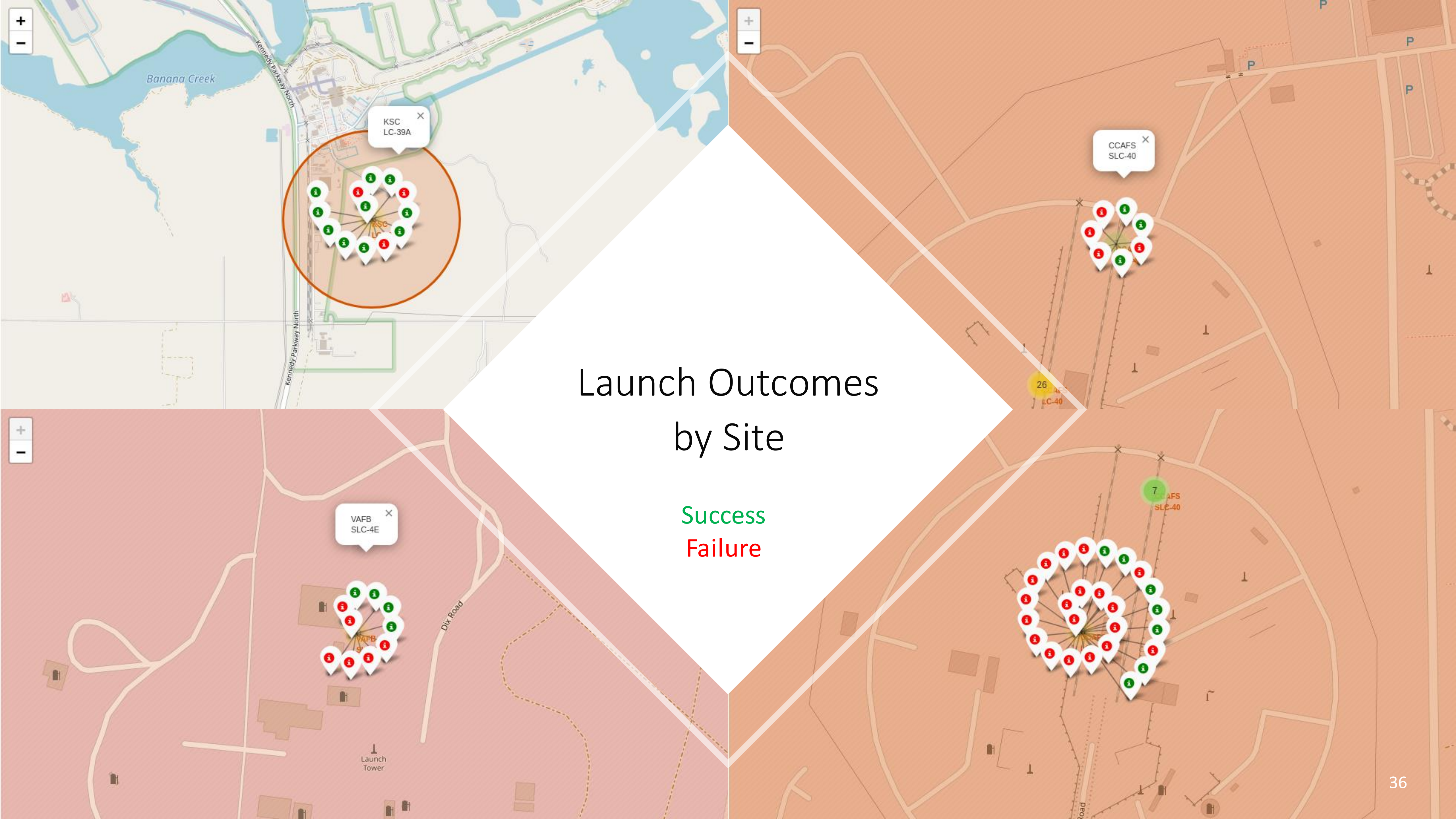
Launch Sites Proximities Analysis



Launch Site Locations

California: 1
Florida: 3

Launch Outcomes by Site



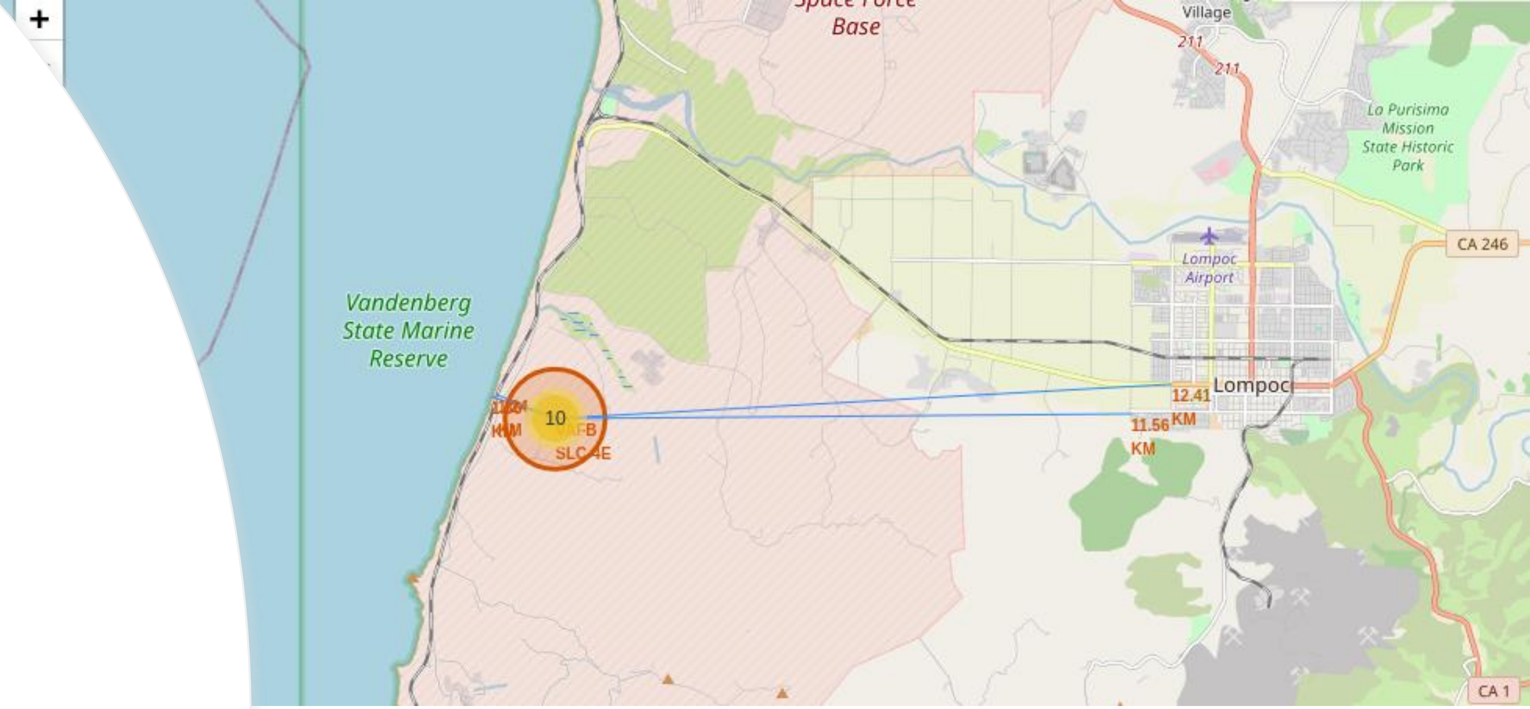
Success

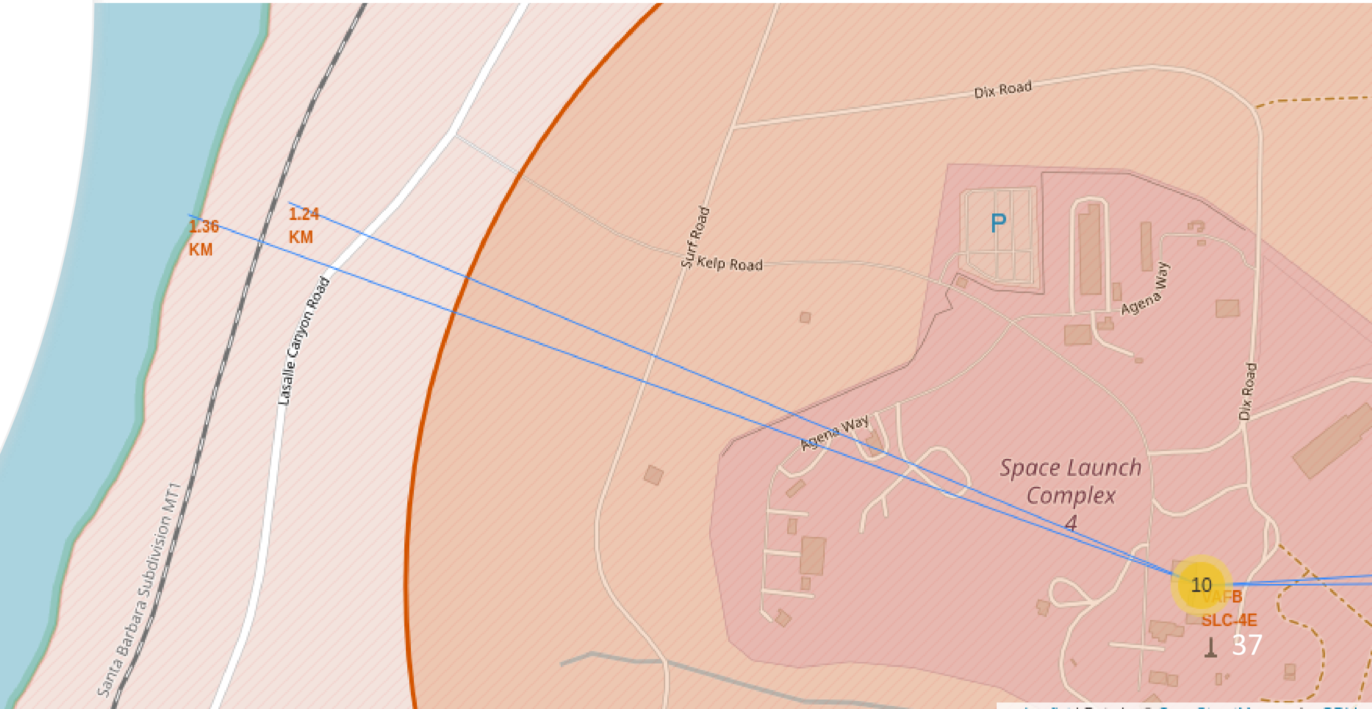
Failure

Proximity to Coastline and Infrastructure

Typically within 1 km to railways, highways, and coastline

Closest city to any site: 11.5 km





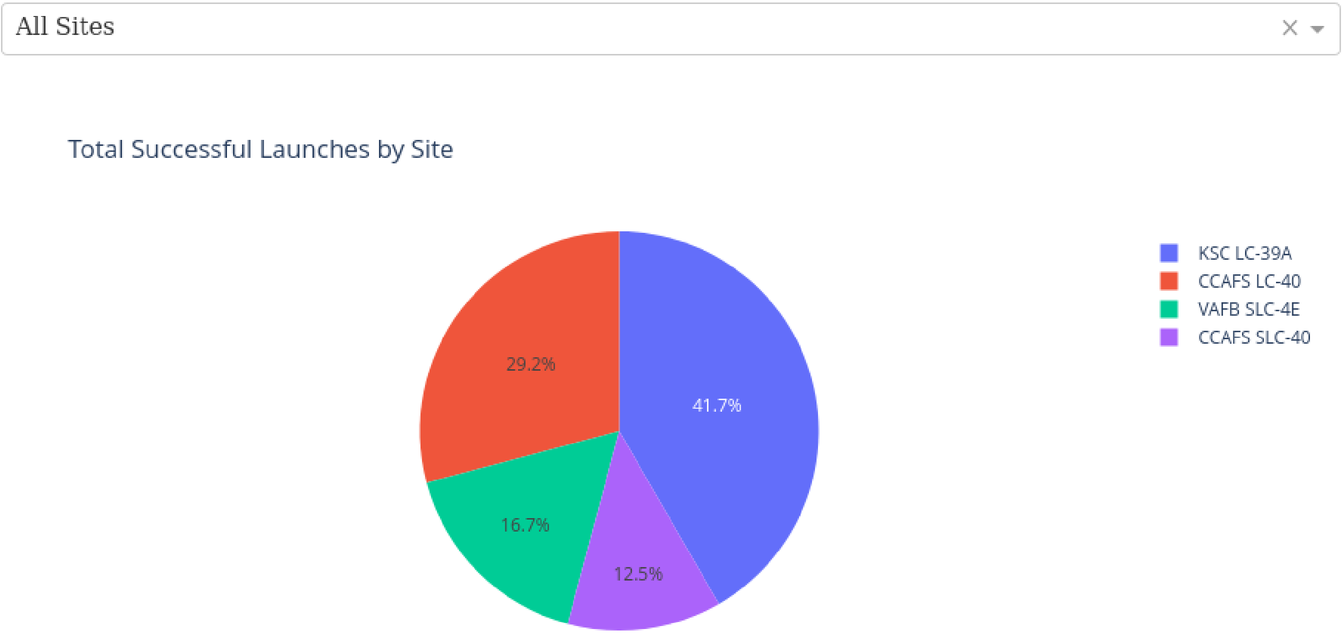
Section 4

Build a Dashboard with Plotly Dash

Successful Launches by Site

- Most successful: KSC LC-39A

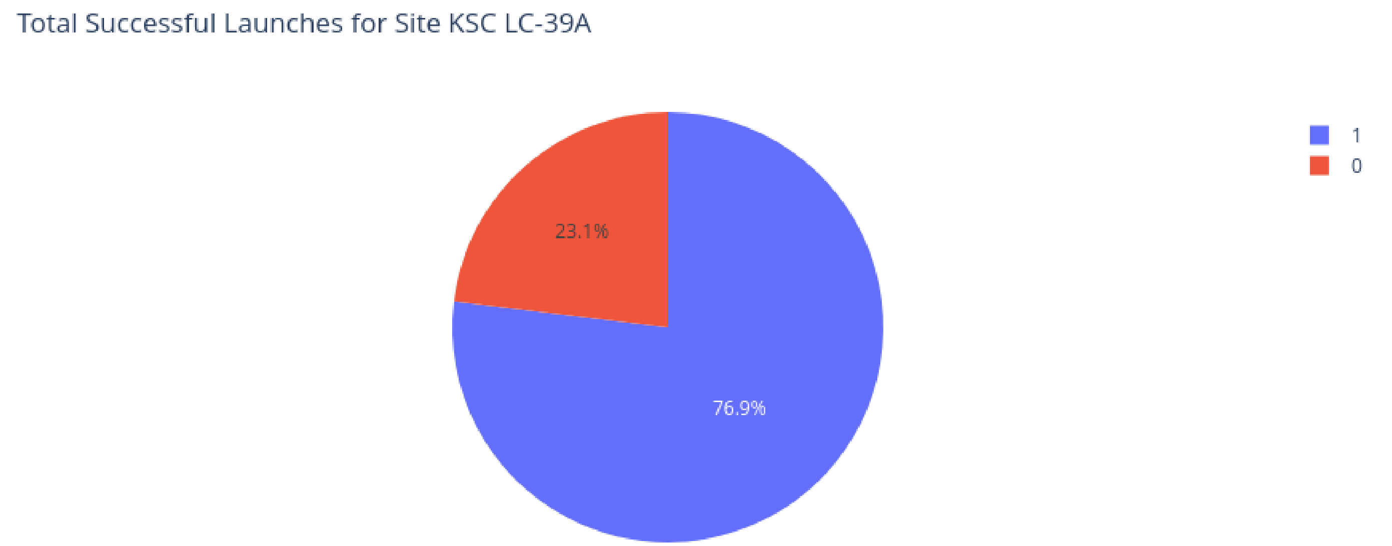
- Least successful: CCAFS SLC-40



Success Rate at KSC LC-39A

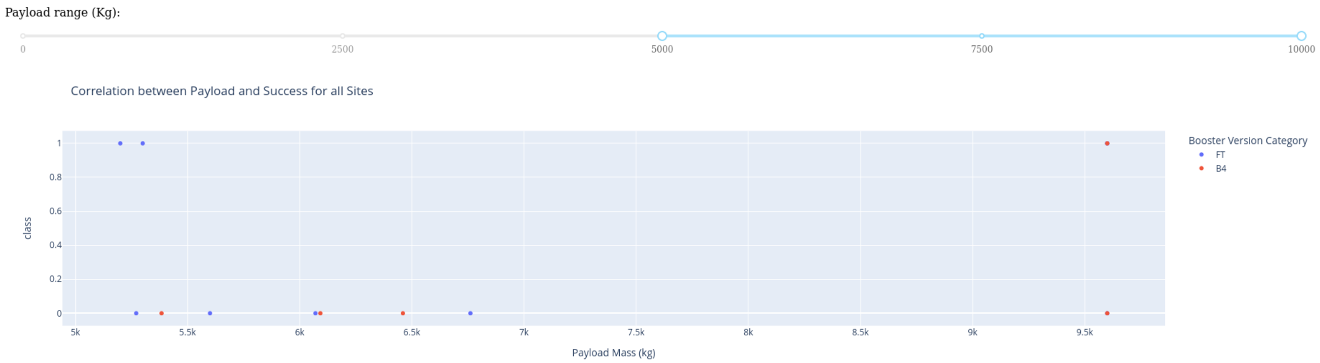
- Highest among all sites

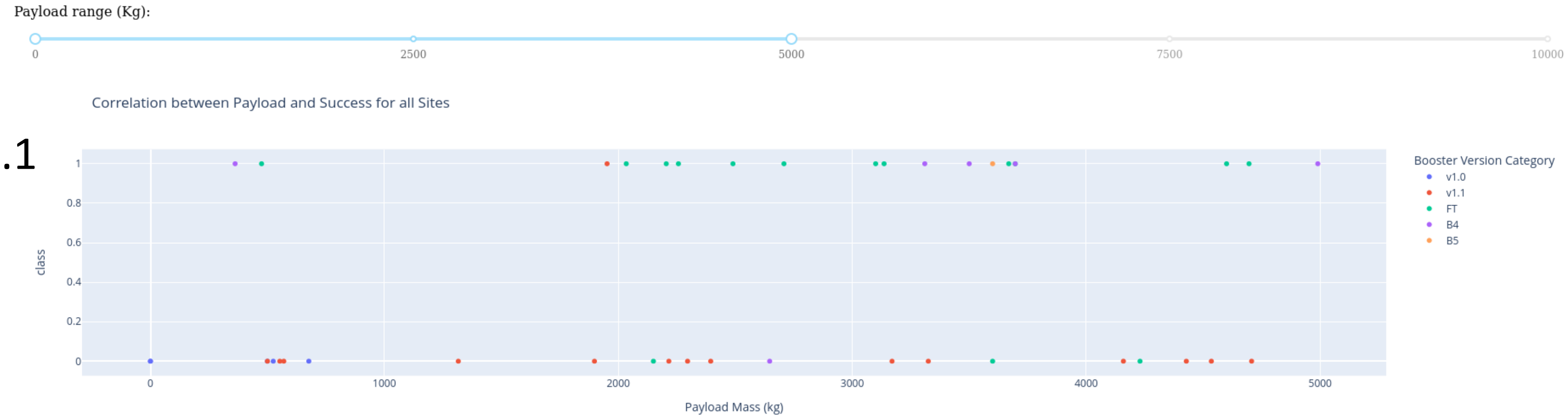
KSC LC-39A



Correlation Between Payload and Success

- Few successes above 5,000Kg
- Optimal range: 1,500-4,000Kg
- Least successful boosters: v1.0, v1.1



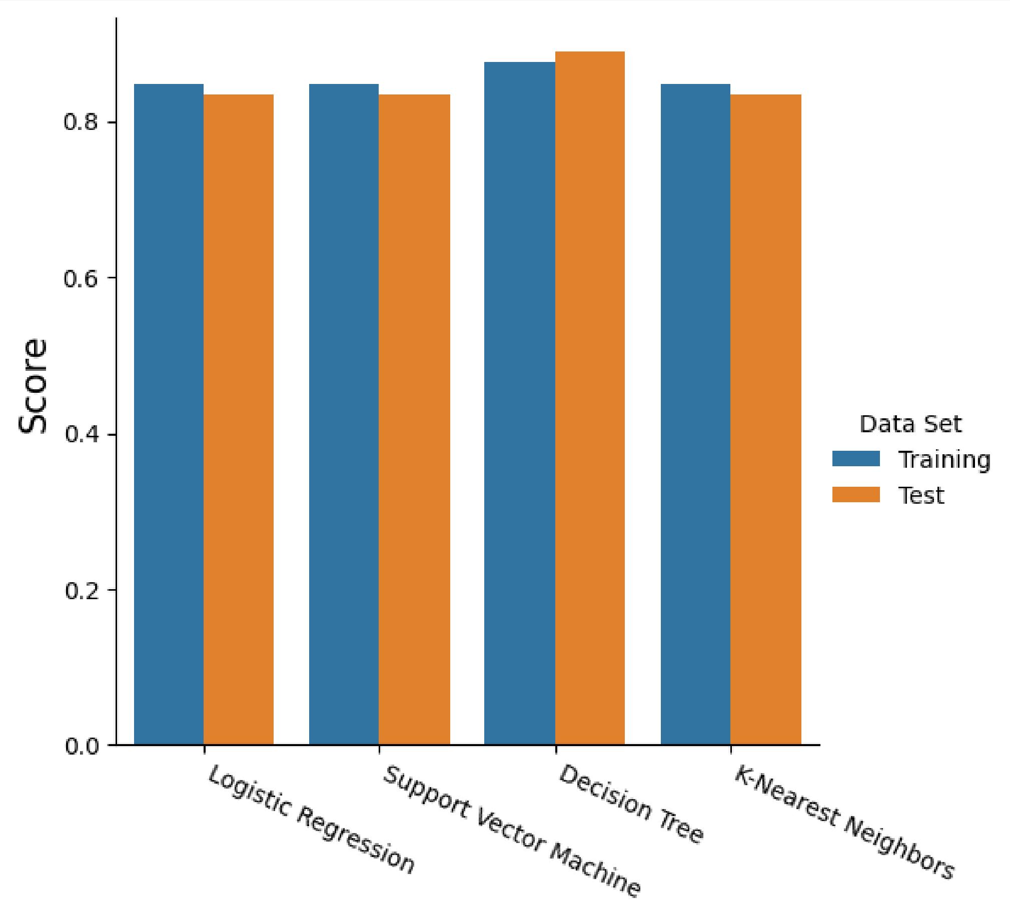


Section 5

Predictive Analysis (Classification)

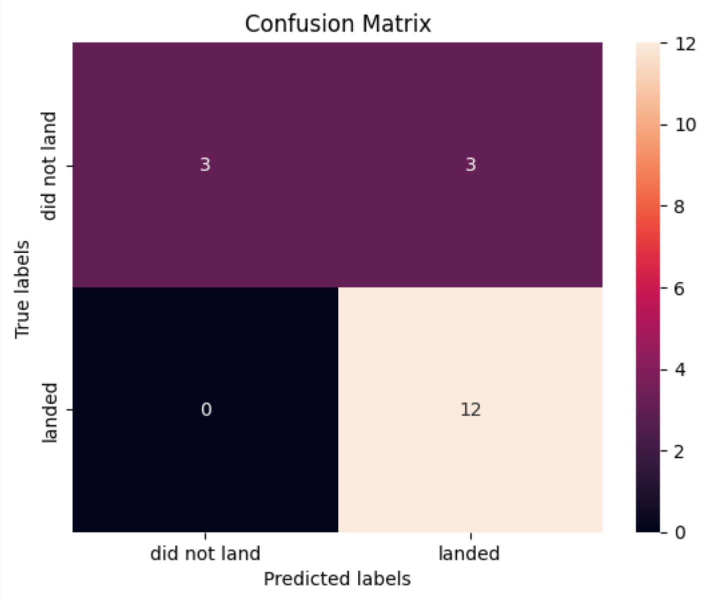
Classification Accuracy

- Highest accuracy: decision tree
- 3 of 4 models showed same accuracy on both training and test data

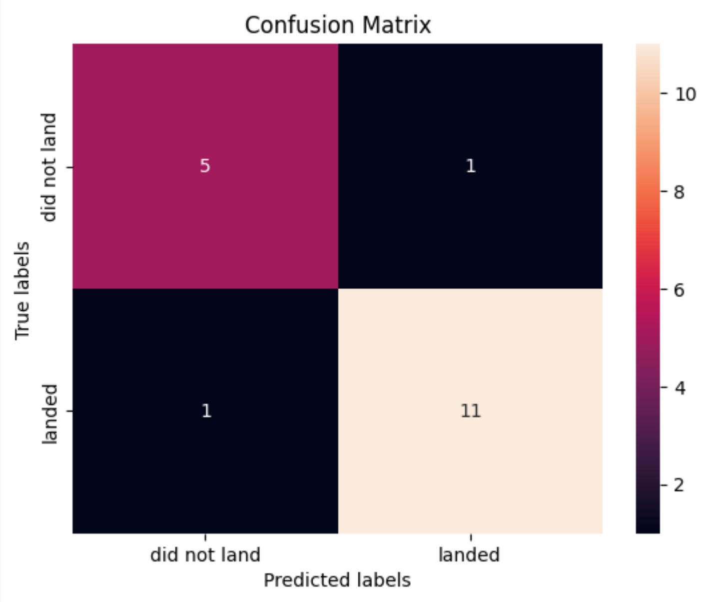


Confusion Matrices

- K-Nearest Neighbors
- Support Vector Machine
- Logistic Regression
- Decision Tree (most runs)



- Decision Tree (best run)



Conclusions

- Steady increase in success rates since 2013
- Site KSC LC-39A has greatest success rate
- Optimal payload range: 1,500-4,000Kg
- Optimal classification model: Decision Tree

Thank you!