IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Bhanu Pratap Singh
August 19 2022

Outline

Executive Summary

- Summary of methodologies

1. Data Collection through SpaceX API
2. Data Collection using Web Scrapping
3. Exploratory Data Analysis
4. Exploratory Data Analysis with SQL
5. Exploratory Data Analysis by Data Visualization
6. Interactive Dashboard Visualization with Folium
7. Classification with Machine Learning

- Summary of all results

1. Results of Exploratory Data Analysis
2. Results of Data Visualizations
3. Results of Machine Learning Analysis

Introduction

- Project background and context

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems we want to find answers to.
  - What all factors can determine the successful landing ?
  - Combination of features that can determine the successful landing.
  - Which launching site is most successful from visualization analysis?
  - Which classification model is best for this problem?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and from Wikipedia page by using Web scrapping.
- Perform data wrangling
  - Data was filtered for Falcon-9 launches. Null values were handled and one-hot coding was applied to categorical values.
- 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.

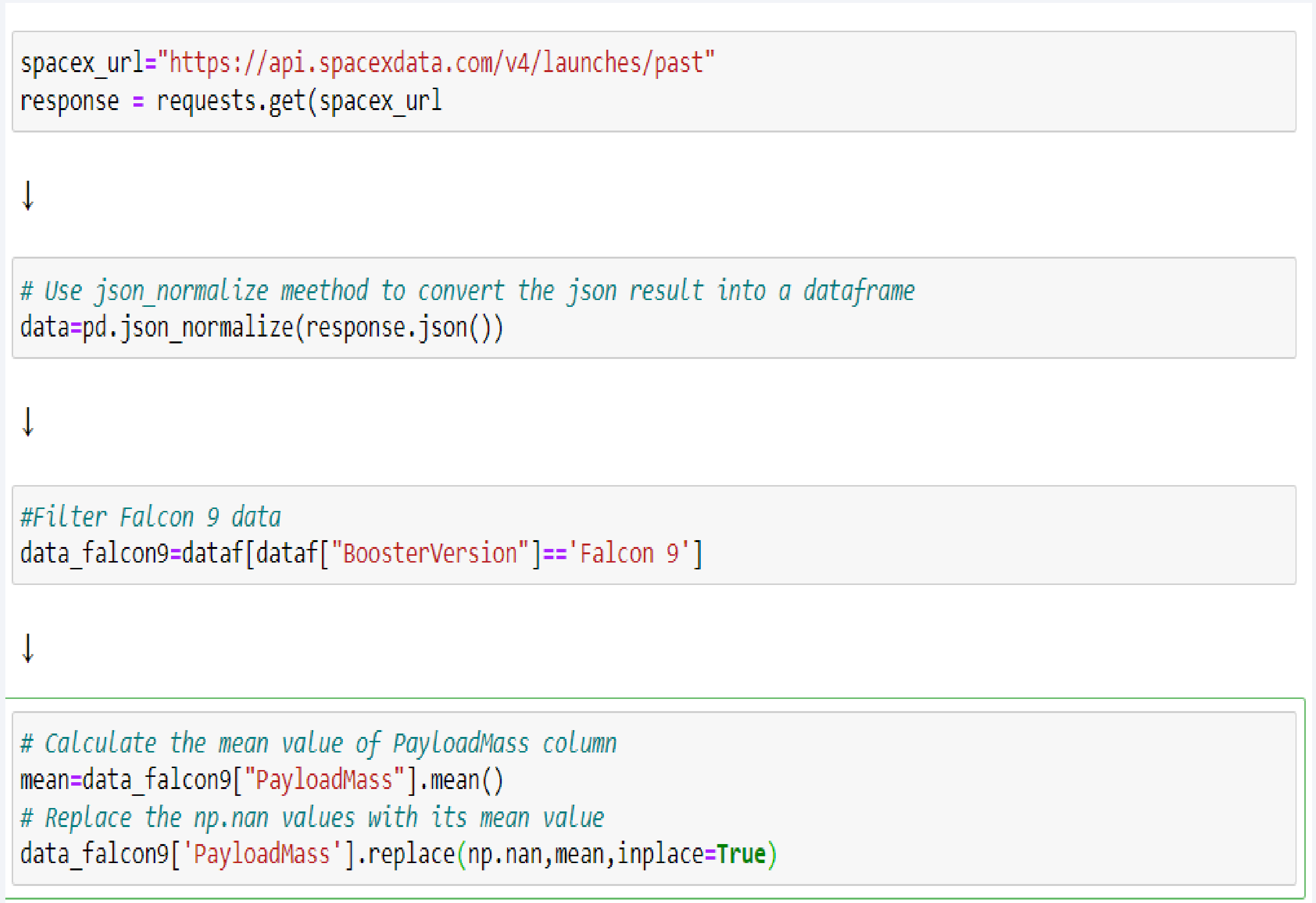
Data Collection

We employed 2 main methods to collect data :

- We used SpaceX API.
  - We request and parsed the launch data using GET request.
  - We transfer the data into Pandas Dataframe using .json_normalize().
  - We filtered the data to include only Falcon 9 launches.
  - We replaced the missing values of Payload Mass with mean.
- We used Web Scrapping on Wikipedia page of SpaceX's launch data.
  - We used BeautifulSoup to web scrap.
  - Extracted the column names from HTML header.
  - Created a Pandas Dataframe by parsing the HTML table for launch details,

Data Collection – SpaceX API

- We sent a get request to the SpaceX API, converted it into a Pandas dataframe. Filtered the data for Falcon9 launches and handled the missing values.
- GitHub URL of the SpaceX API calls notebook: [Data Collection API Notebook Link]()



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

↓

```
# Use json_normalize meethod to convert the json result into a dataframe
data=pd.json_normalize(response.json())
```

↓

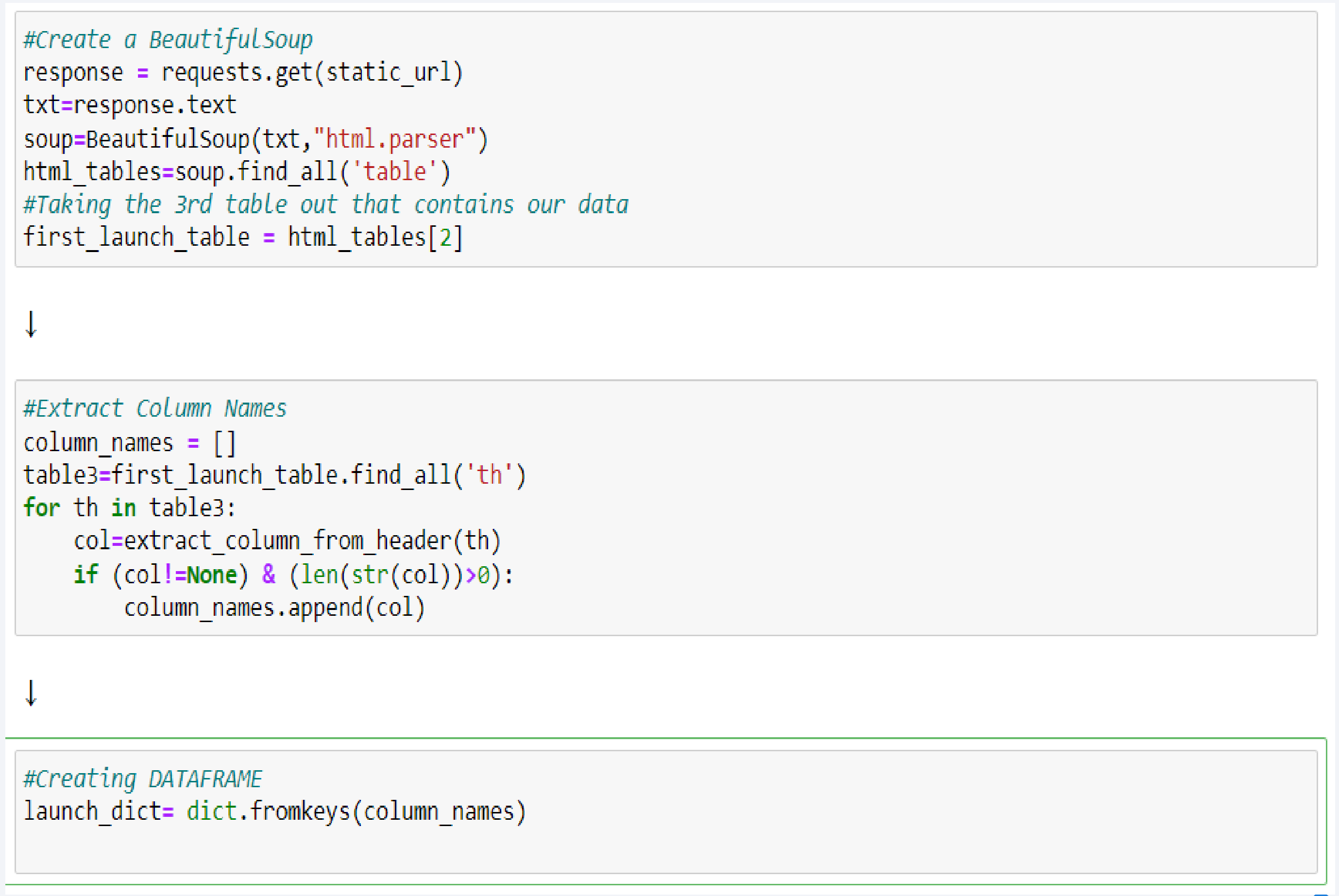
```
#Filter Falcon 9 data
data_falcon9=dataf[dataf["BoosterVersion"]=='Falcon 9']
```

↓

```
# Calculate the mean value of PayloadMass column
mean=data_falcon9["PayloadMass"].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan,mean,inplace=True)
```

Data Collection – Web Scraping

- We used BeautifulSoup to web scrap, extracted the column names and converted the data into a Pandas Dataframe.
- GitHub URL of the web scraping notebook: Data Collection Web Scraping Notebook Link



```
#Create a BeautifulSoup
response = requests.get(static_url)
txt=response.text
soup=BeautifulSoup(txt,"html.parser")
html_tables=soup.find_all('table')
#Taking the 3rd table out that contains our data
first_launch_table = html_tables[2]
```

↓

```
#Extract Column Names
column_names = []
table3=first_launch_table.find_all('th')
for th in table3:
    col=extract_column_from_header(th)
    if (col!=None) & (len(str(col))>0):
        column_names.append(col)
```

↓

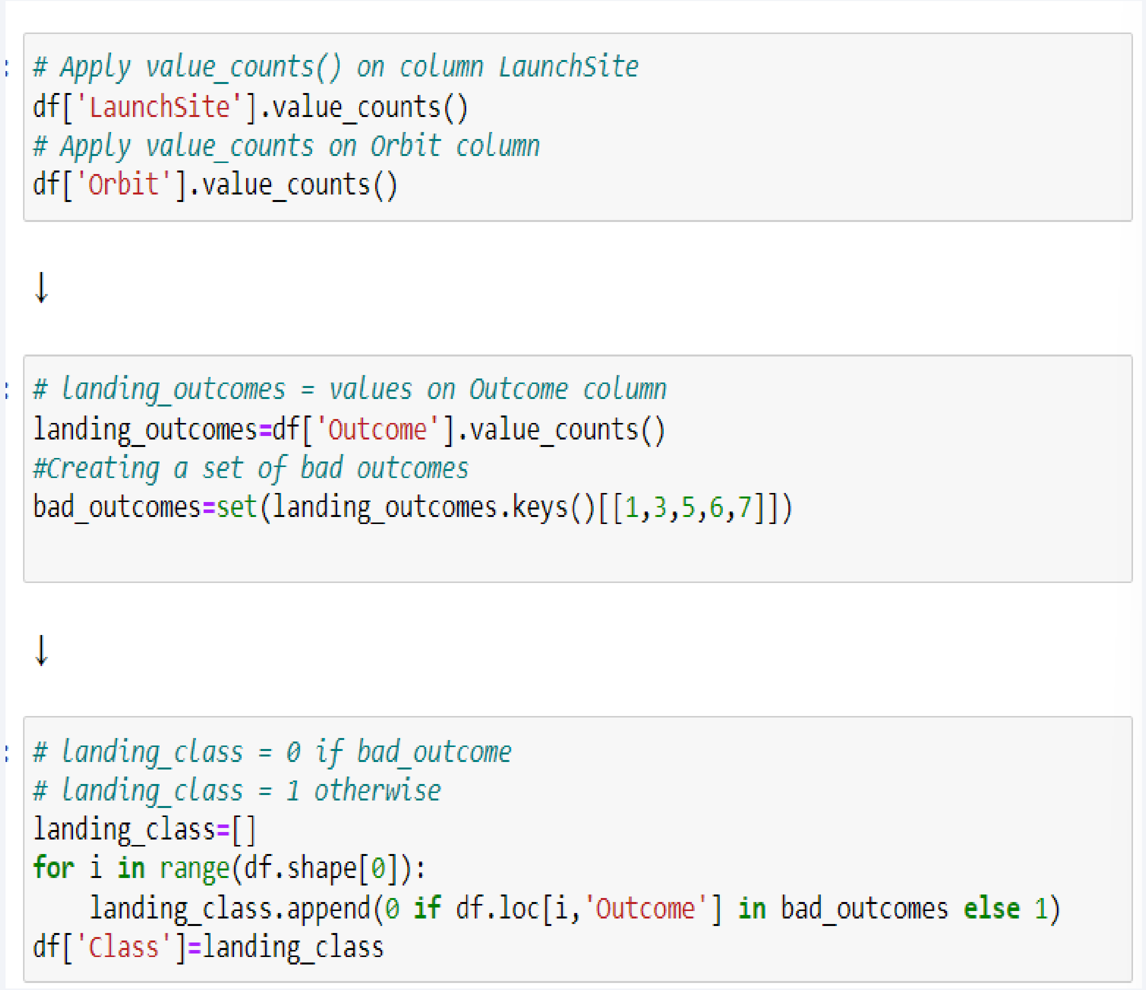
```
#Creating DATAFRAME
launch_dict= dict.fromkeys(column_names)
```

Data Wrangling

- We first used value_counts() on LaunchSite, Orbit and Outcomes to see the occurrences of each.
- Created a bad outcome set to separate bad outcomes from the successful ones.
- We use it to give class label to Class to show the outcome.

  We provide Class value =0 for bad outcomes.

  We provide Class value =1 for successful outcomes.
- GitHub URL : [Data Wrangling Notebook Link]()



```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

↓

```
# landing_outcomes = values on Outcome column
landing_outcomes=df['Outcome'].value_counts()
#Creating a set of bad outcomes
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
```

↓

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class=[]
for i in range(df.shape[0]):
    landing_class.append(0 if df.loc[i,'Outcome'] in bad_outcomes else 1)
df['Class']=landing_class
```

EDA with Data Visualization

- We plotted Scatter Plots for the following :

    Flight Number v/s Launch Sites

    Payload v/s Launch Sites

    Flight Number v/s Orbits

    Payload v/s Orbits

  We used scatter plots to see the relation between these two features for each.

- We plotted Bar Chart for Orbits to show the success rate of each.
- We plotted Line Chart to see how average success ratio changed with years.
- GitHub URL : EDA Visualization Notebook Link

EDA with SQL

- We performed EDA with SQL queries to get insights from data. The queries were:
    - Display names of each unique launch site
    - Display the total payload mass carried by boosters launched by NASA (CRS)
    - Display average payload mass carried by booster version F9 v1.1
    - Display the earliest date for successful landing outcome in ground pad
    - Display total number of successful and failed outcomes
    - Display failed outcome in drone ship with booster version, launch site in 2015.
- GitHub URL : EDA SQL Notebook Link

Build an Interactive Map with Folium

- We marked all launch sites and added map markers to mark the success and failure of each launch from the different sites.
- We assigned class label 0 and 1 for failure and success outcome respectively.
- Calculated distances between the launch sites and its proximities like coastline, highway and city.

  To see if launch sites keep away from cities?

  To see if they are close to coastlines?

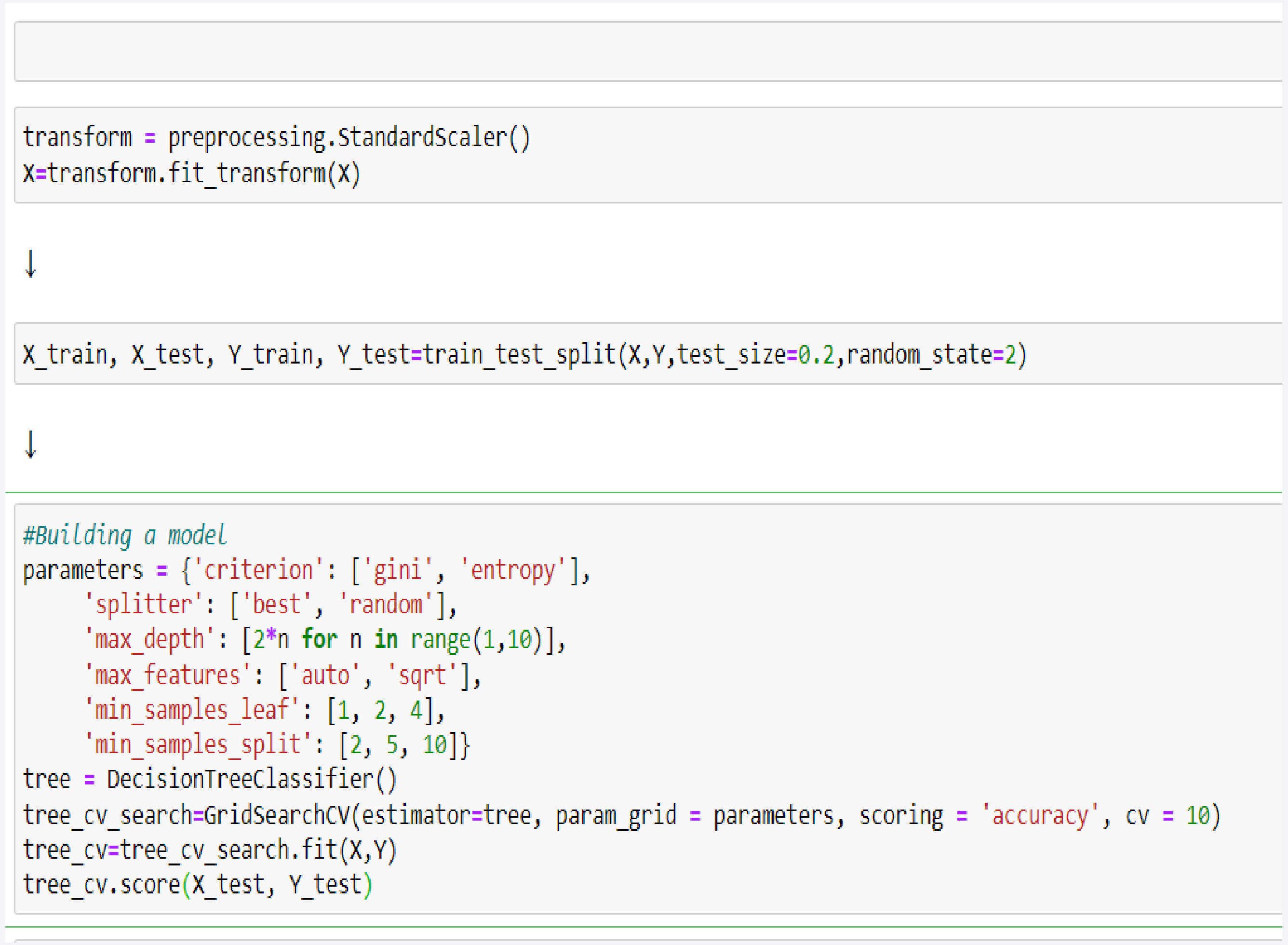
  How are they connected with highways and railways?
- GitHub URL : Interactive Folium Notebook Link

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly Dash.
- A dropdown menu with options to either select all or specific launch sites
- A pie chart showing total successful launches for all sites or to show a pie chart between successful and failure ratio from specific sites.
- We added a payload range slider to select a range of payload mass.
- We plotted a scatter plot between outcome and payload mass picked from slider.
- GitHub URL: Plotly Dash App Python Code Link

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas. Transformed the data using StandardScaler().
- We split the data into train and test data with test size 0.2
- We built different ML models and found best hyperparameters using GrisdSearchCV.
- We found the best performing model.
- GitHub URL : [Machine Learning Analysis Notebook Link]()



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

↓

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

↓

```
#Building a model
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}
tree = DecisionTreeClassifier()
tree_cv_search=GridSearchCV(estimator=tree, param_grid = parameters, scoring = 'accuracy', cv = 10)
tree_cv=tree_cv_search.fit(X,Y)
tree_cv.score(X_test, Y_test)
```

Results

- <u>Exploratory data analysis results</u>

  •Space X uses 4 different launch sites;

  •The first launches were done to Space X itself and NASA;

  •The average payload of F9 v1.1 booster is 2,928 kg;

  •The first success landing outcome happened in 2015 fiver year after the first launch;

  •Many Falcon 9 booster versions were successful at landing in drone ships having payload above the average;

  •Almost 100% of mission outcomes were successful;

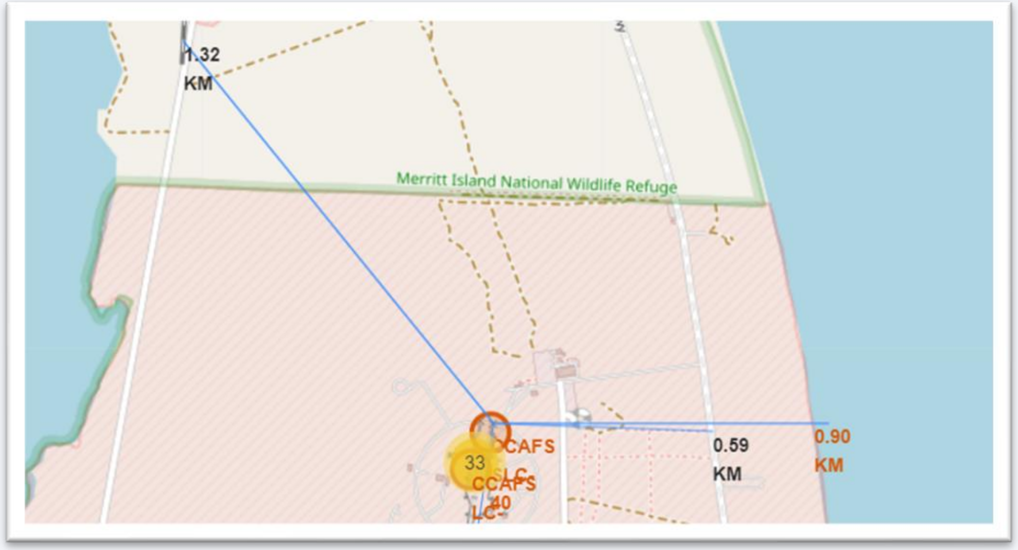
  •Two booster versions failed at landing in drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015;

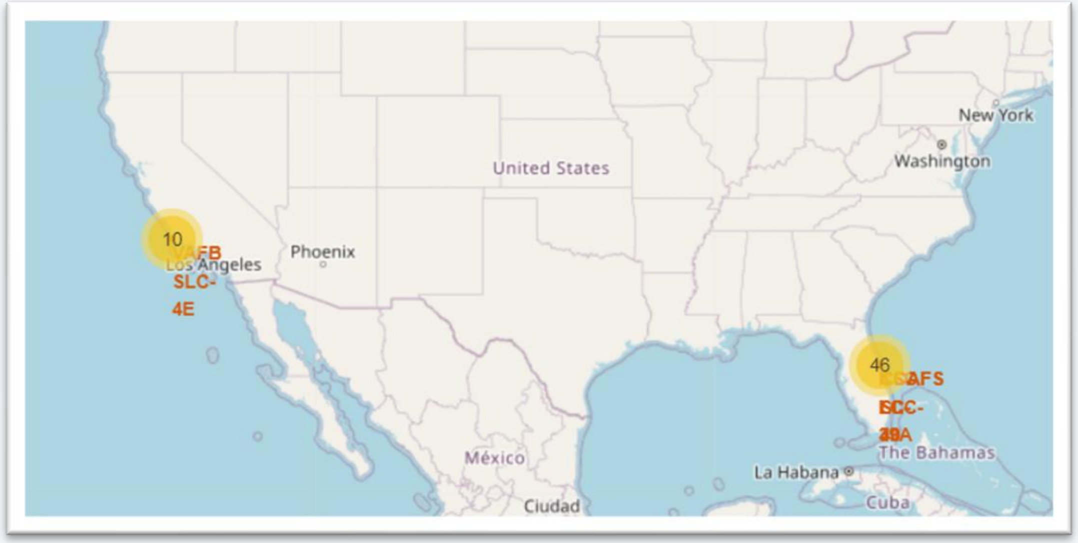
  •The number of landing outcomes became as better as years passed.

- <u>Predictive analysis results</u>
  - Predictive analysis showed that Decision Tree Classifier is the best classification model for our data with accuracy of 90%.
  - Only 1 case each of False Positive and False Negative.

Results

- Interactive analytics demo in screenshots
  - Using interactive analytics was possible to identify that launch sites use to be in safety places, near sea, for example and have a good logistic infrastructure around.
  - Most launches happen at east coast line.

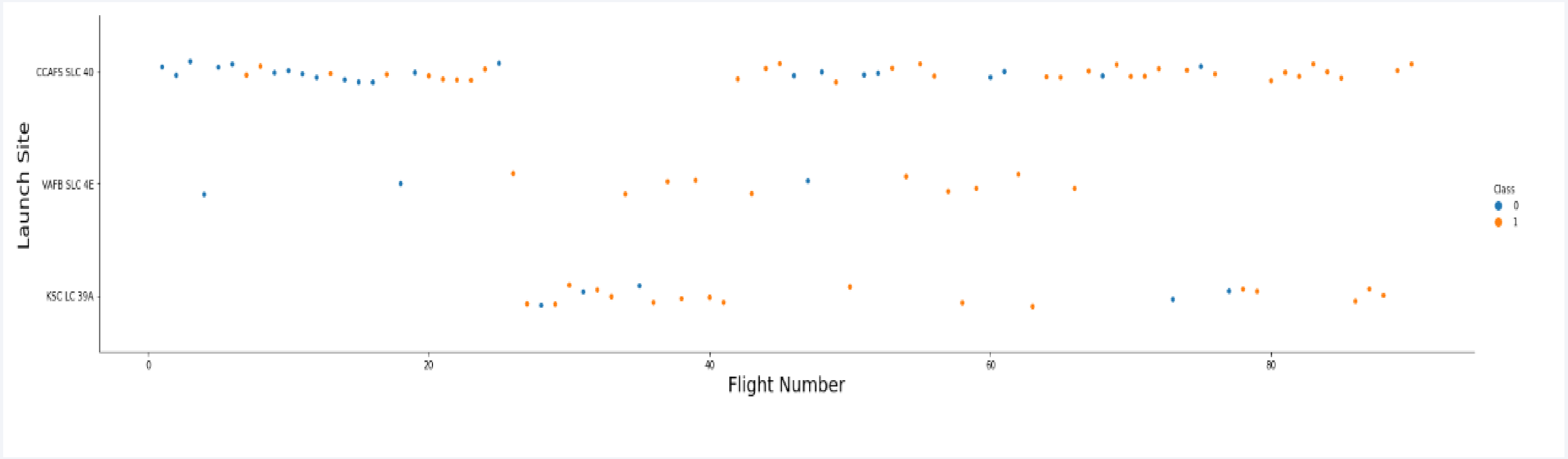




Section 2

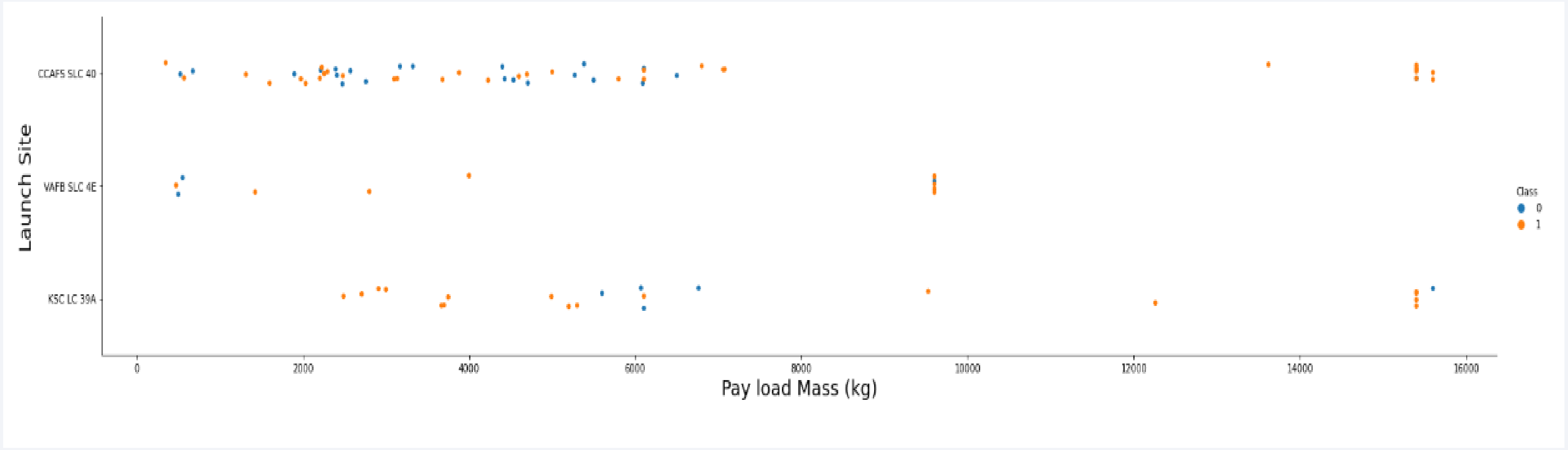
Insights drawn from EDA

Flight Number vs. Launch Site



We see how with increase in Flight numbers, we see more data plots with class label 1. This indicates that with increase in flight number the chances of a successful outcome is high for each launch site.

Payload vs. Launch Site



We see that with high payload mass we have more points with data label 1.

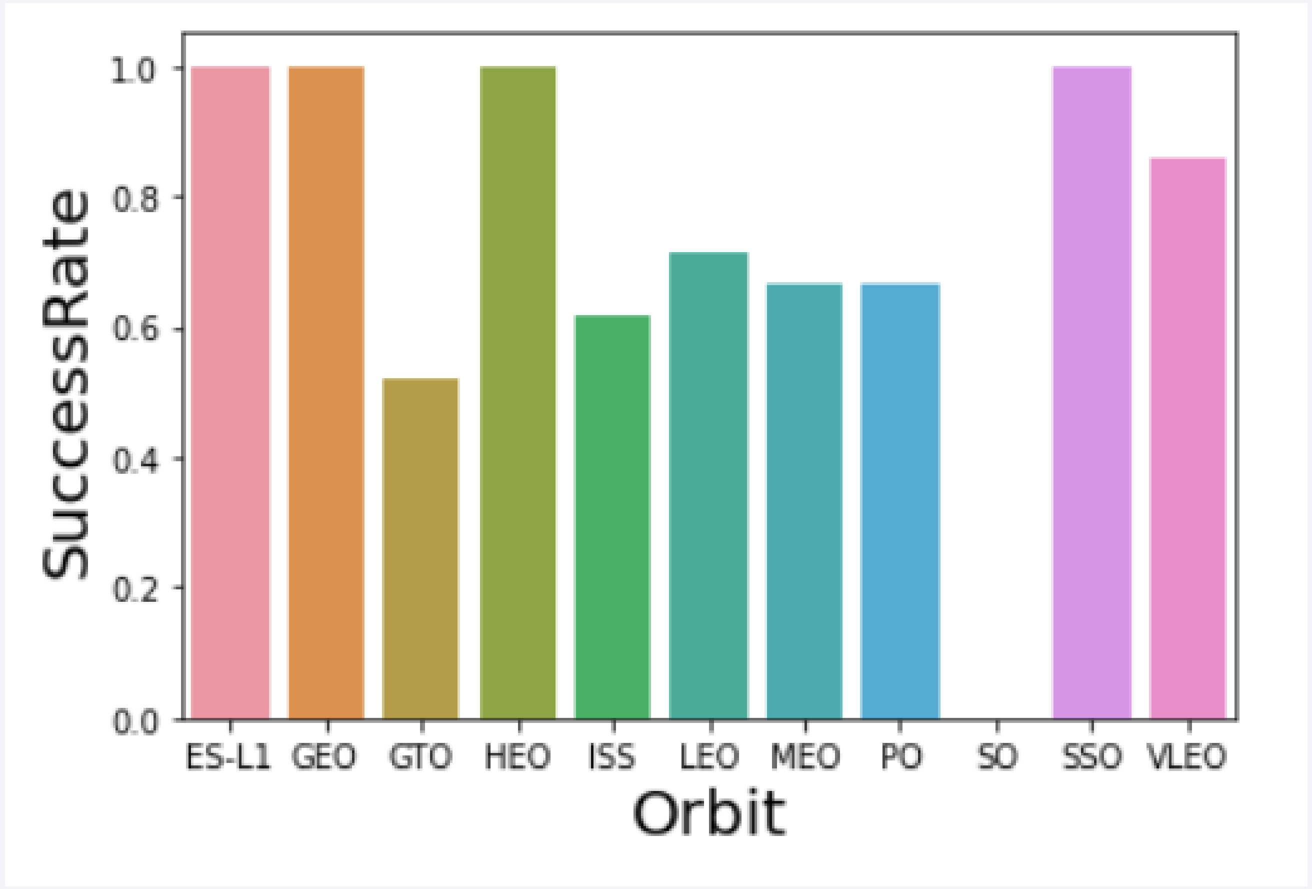
This shows that launches with high payload mass have more chances for successful outcome than a low payload mass launch.

Success Rate vs. Orbit Type

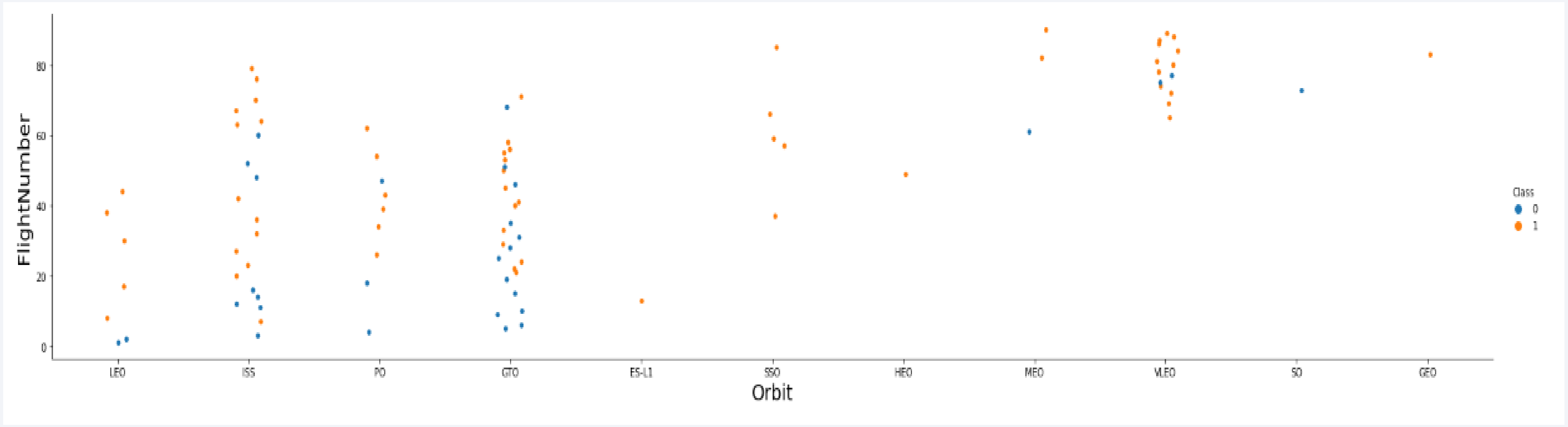
We see that launches for orbit ES-L1, GEO, HEO and SSO have perfect success rate of 1.

The orbit GTO has less success rate with value of about 0.5.

The orbit SO has not had a successful launch and has a 0 score.



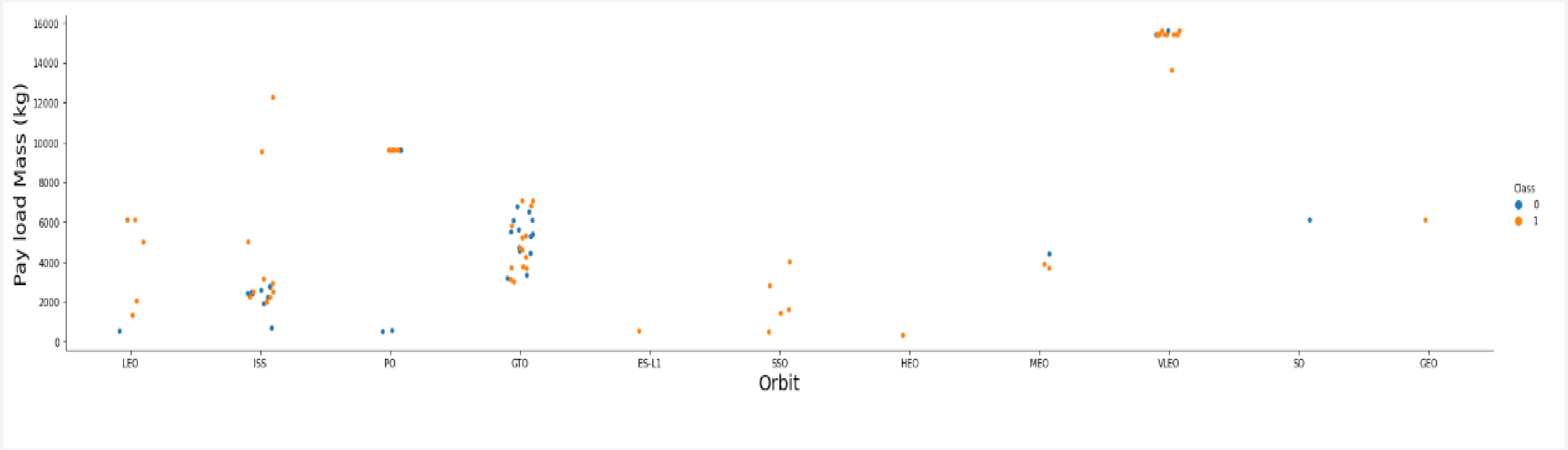
Flight Number vs. Orbit Type



We see that in the LEO orbit the 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



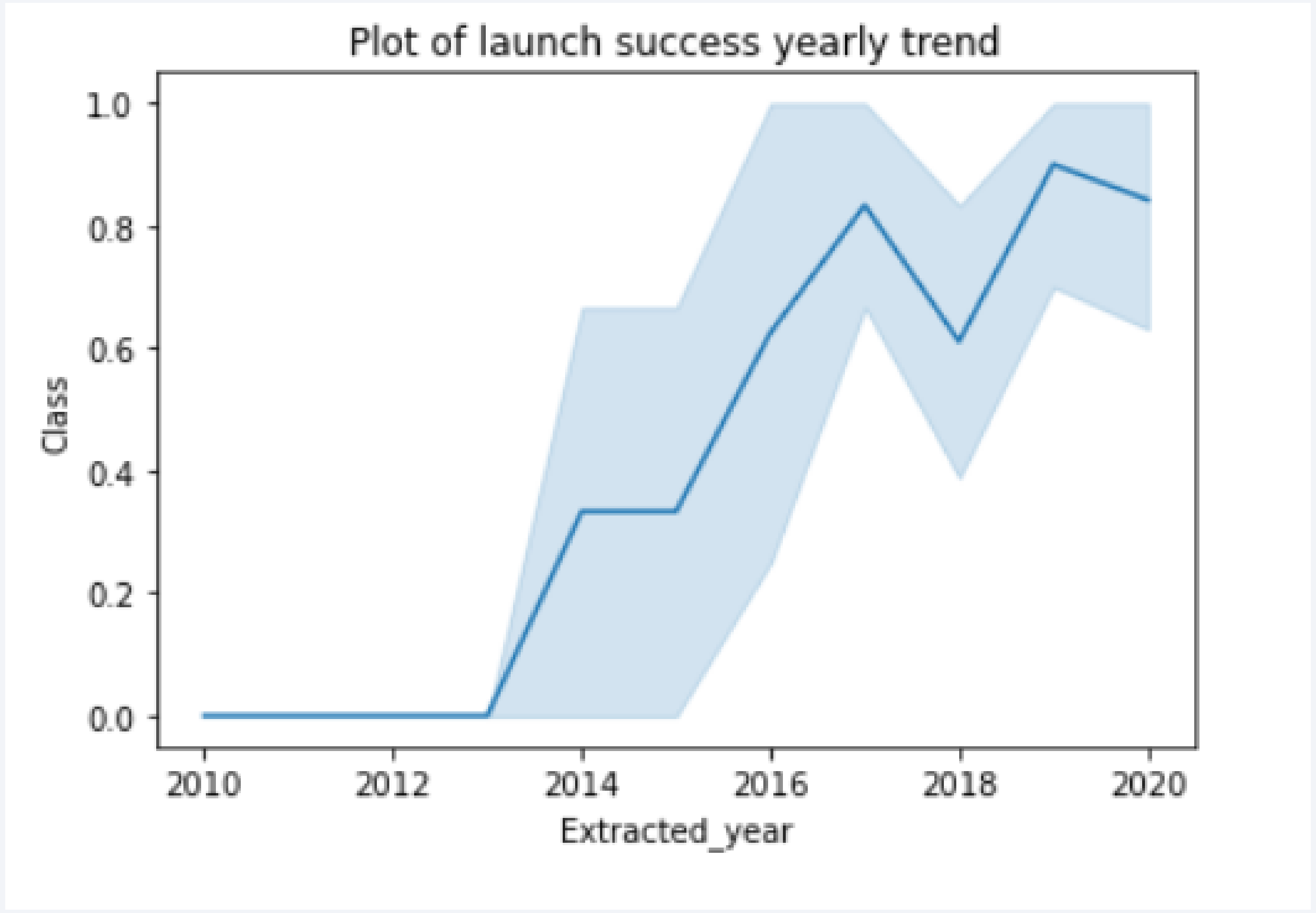
With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

However for GTO we cannot distinguish this very well as both of the positive landing rate and negative landing(unsuccessful mission) are there.

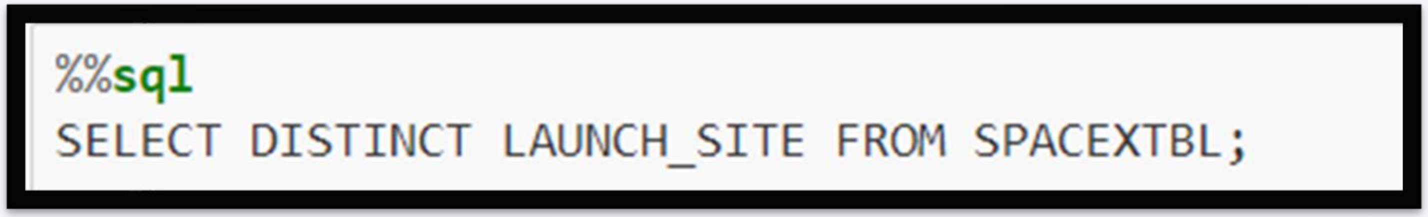
Launch Success Yearly Trend

We see that success rate has been on a rise since year 2013.

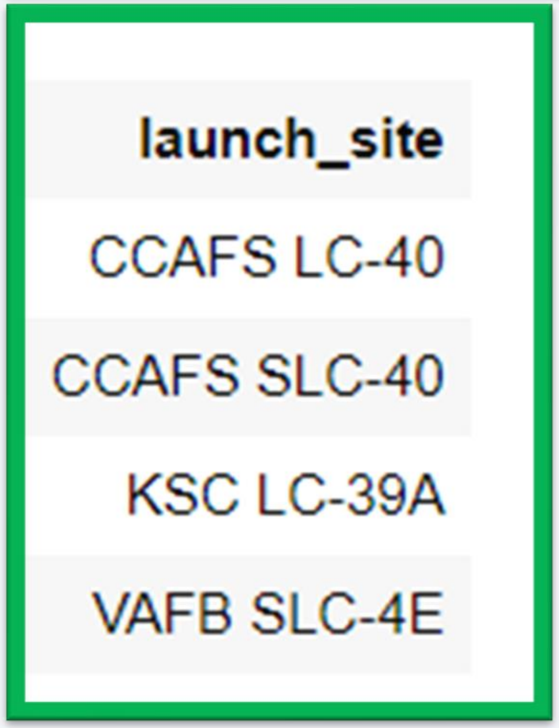
However, there was a downfall for success rate in the year 2018.



All Launch Site Names



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

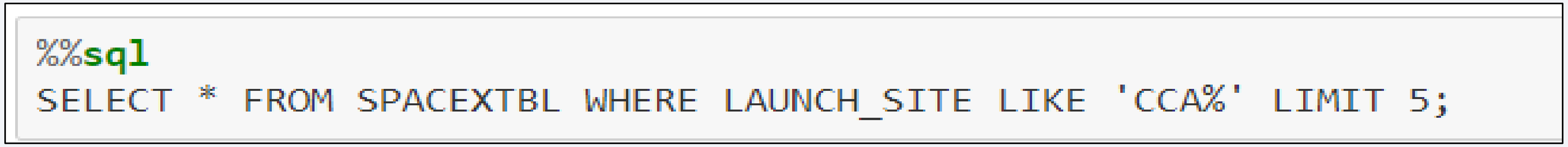


launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

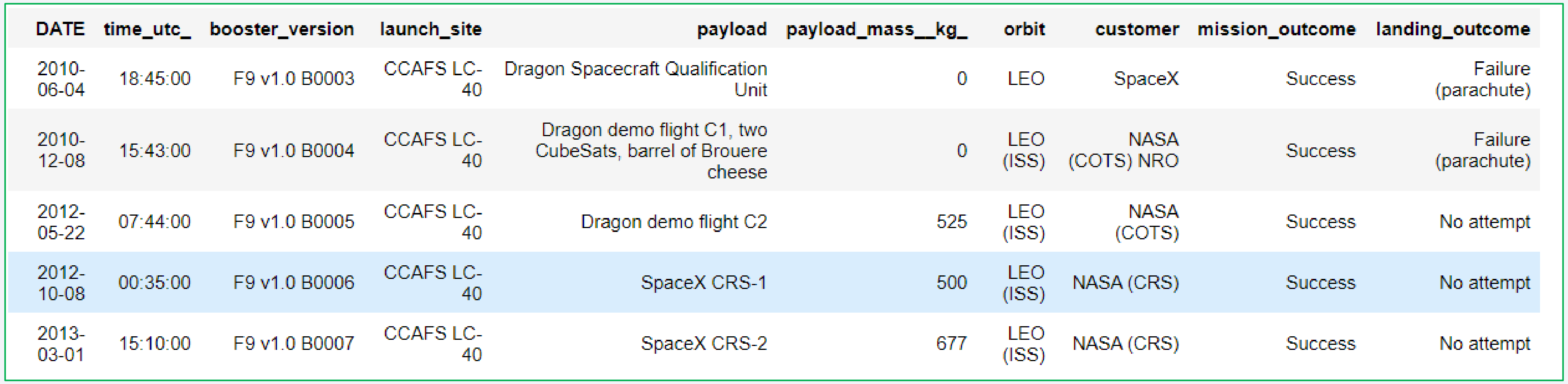
We select DISTINCT launch sites from the table.

We get these 4 outcomes as unique Launch Sites.

Launch Site Names Begin with 'CCA'



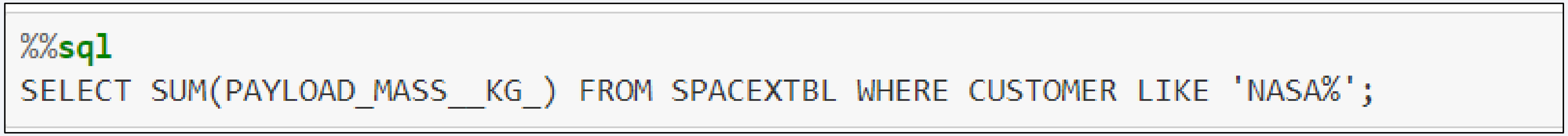
```
%%sql
SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```



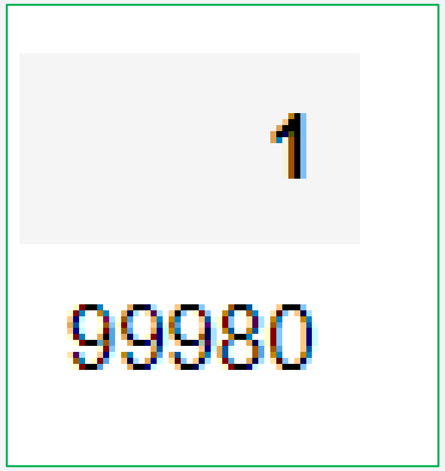
DATE	time_utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight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

We display first 5 record where the Launch Sites begin with CAA using LIKE operation.

Total Payload Mass



```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER LIKE 'NASA%';
```

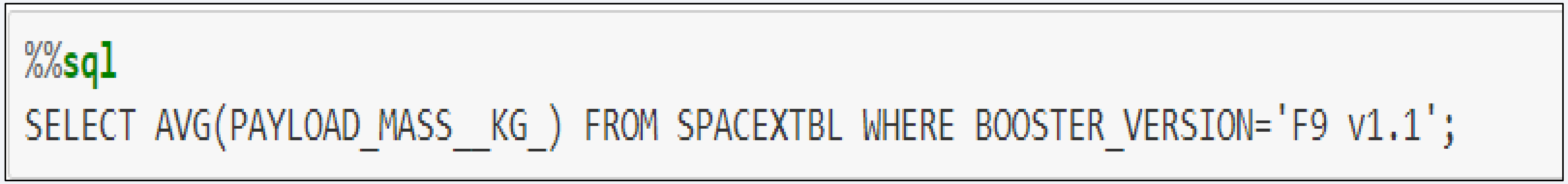


```
    1
99980
```

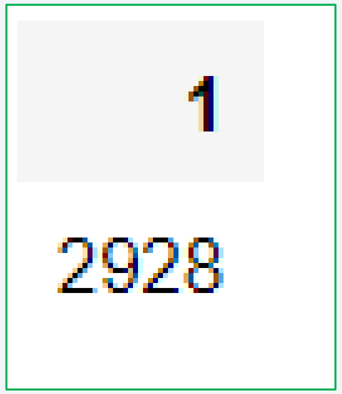
We calculate the total payload mass (kg) carried by boosters launched by NASA and get the value of 99,980 KG.

We do so by using TOTAL operation in our query.

Average Payload Mass by F9 v1.1



```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1';
```

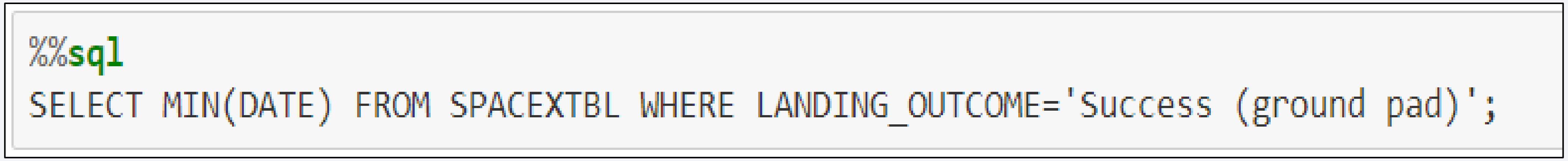


1
2928

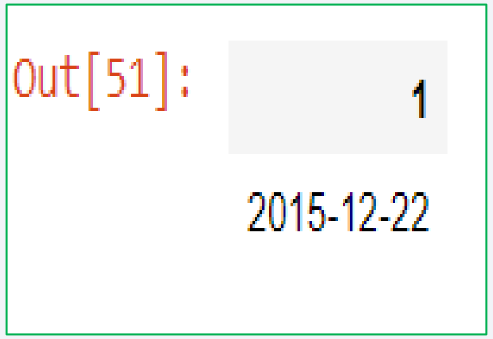
We calculate the average payload mass (kg) carried by booster version F9 v1.1 and get the value of 2928 KG.

We do so by using AVG operation in our query.

First Successful Ground Landing Date



```
%%sql
SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME='Success (ground pad)';
```

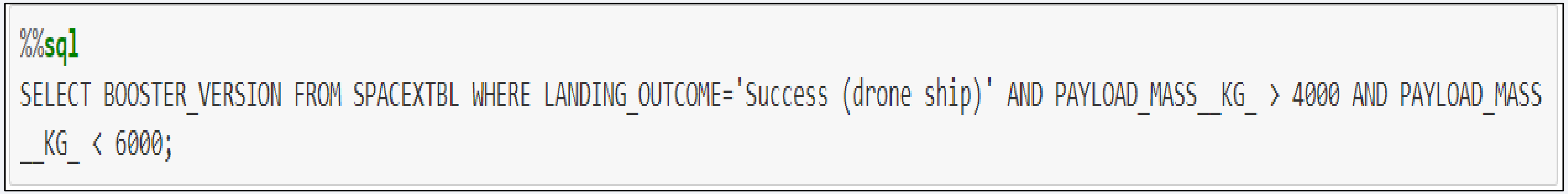


```
Out[51]:          1
         2015-12-22
```

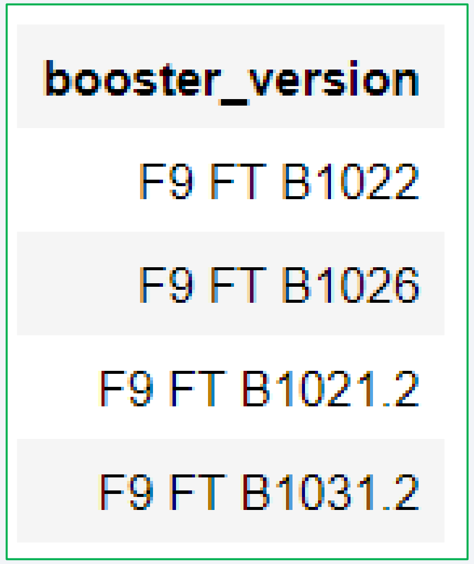
We calculate the earliest date for a successful landing on a ground pad and get the date as 2015-12-22.

We do so by using MIN operation in our query.

Successful Drone Ship Landing with Payload between 4000 and 6000



```
%%sql
SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME='Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS
__KG_ < 6000;
```



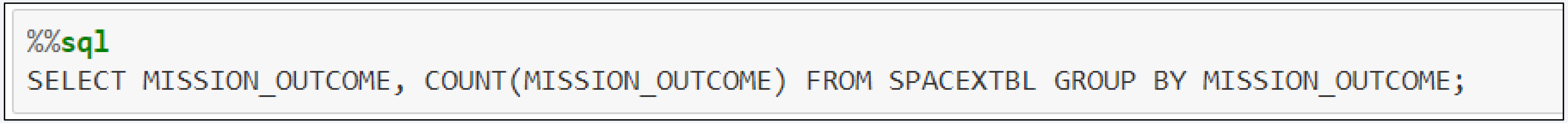
booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

We list the booster versions which have a successful landing on a drone ship and the Payload is between 4000-6000 KG.

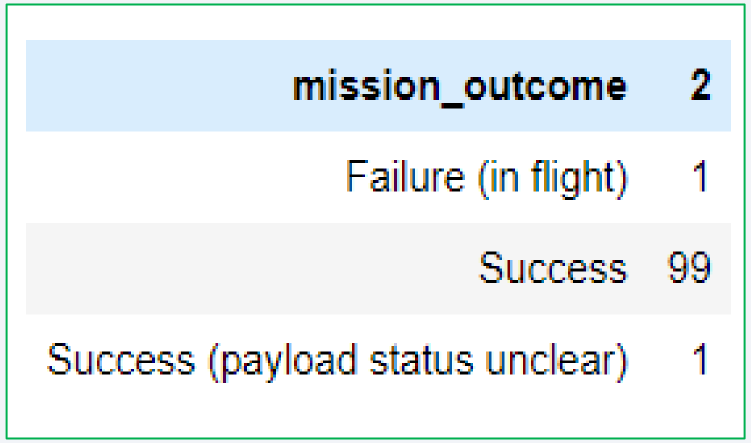
We do so by using AND in our query.

We could have also use BETWEEN in our query.

Total Number of Successful and Failure Mission Outcomes



```
%%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

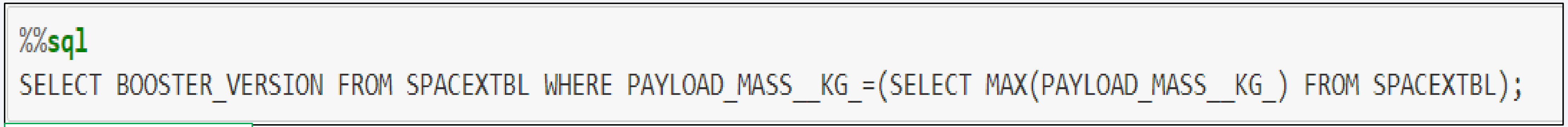


mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

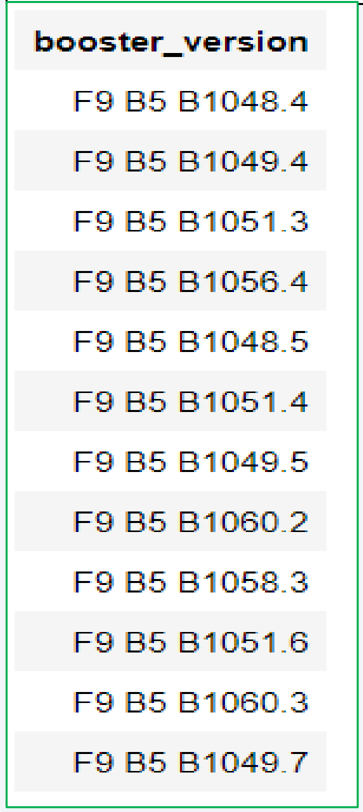
We calculate total successful and failure mission outcomes and get 100 Successful mission and 1 Failure mission.

We do so by using GROUP BY operation in our query.

Boosters Carried Maximum Payload



```
%%sql
SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

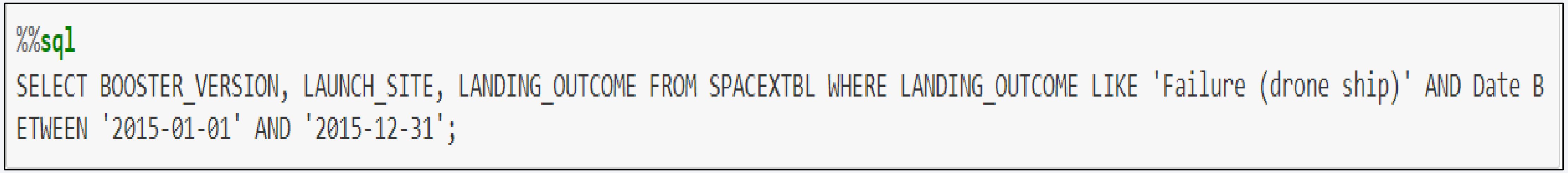


booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

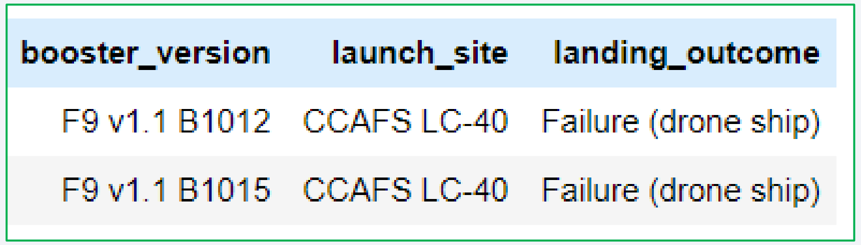
We display the boosters that carried maximum payload weight and get these values.

We do so by using a SUB QUERY in our query and use MAX operation in that sub query.

2015 Launch Records



```
%%sql
SELECT BOOSTER_VERSION, LAUNCH_SITE, LANDING_OUTCOME FROM SPACEXTBL WHERE LANDING_OUTCOME LIKE 'Failure (drone ship)' AND Date B
ETWEEN '2015-01-01' AND '2015-12-31';
```

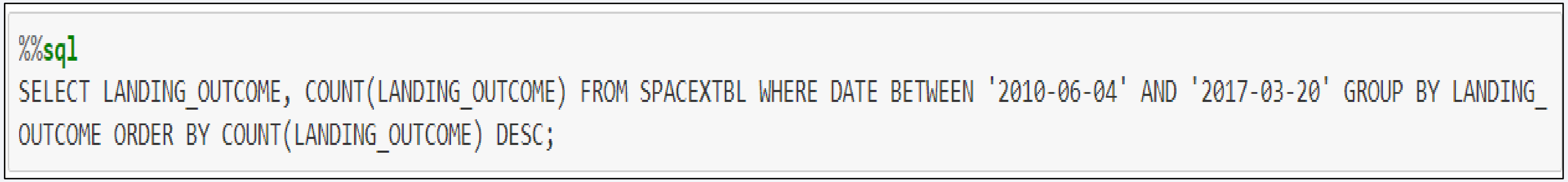


booster_version	launch_site	landing_outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

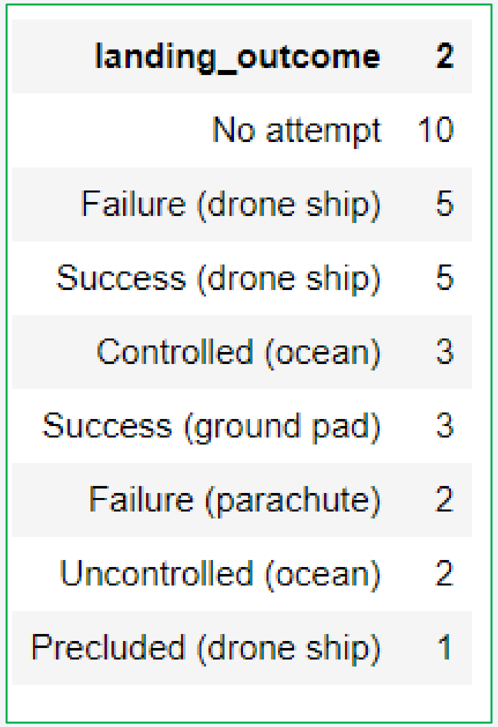
We display booster version, launch site and the landing outcome for failed landing outcome in 2015. We get the result as shown.

We do so by using BETWEEN in our query.

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



```
%%sql
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;
```



landing_outcome	2
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

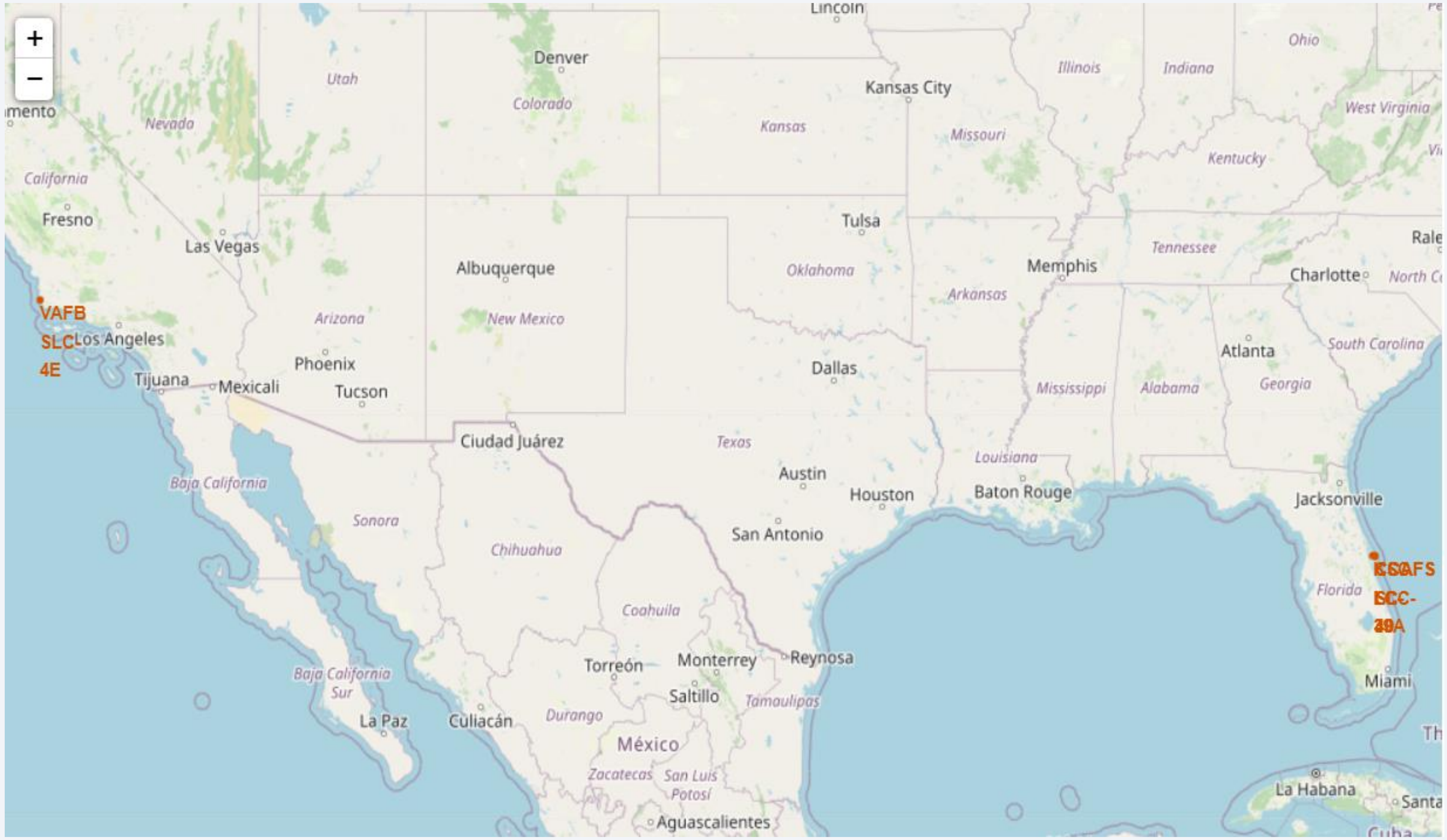
We landing outcome, and their occurrences between specified dates. We get this output as shown.

We do so by using COUNT BETWEEN ORDER BY & GROUP BY in our query.

Section 3

Launch Sites Proximities Analysis

Launch Sites on Map of USA.

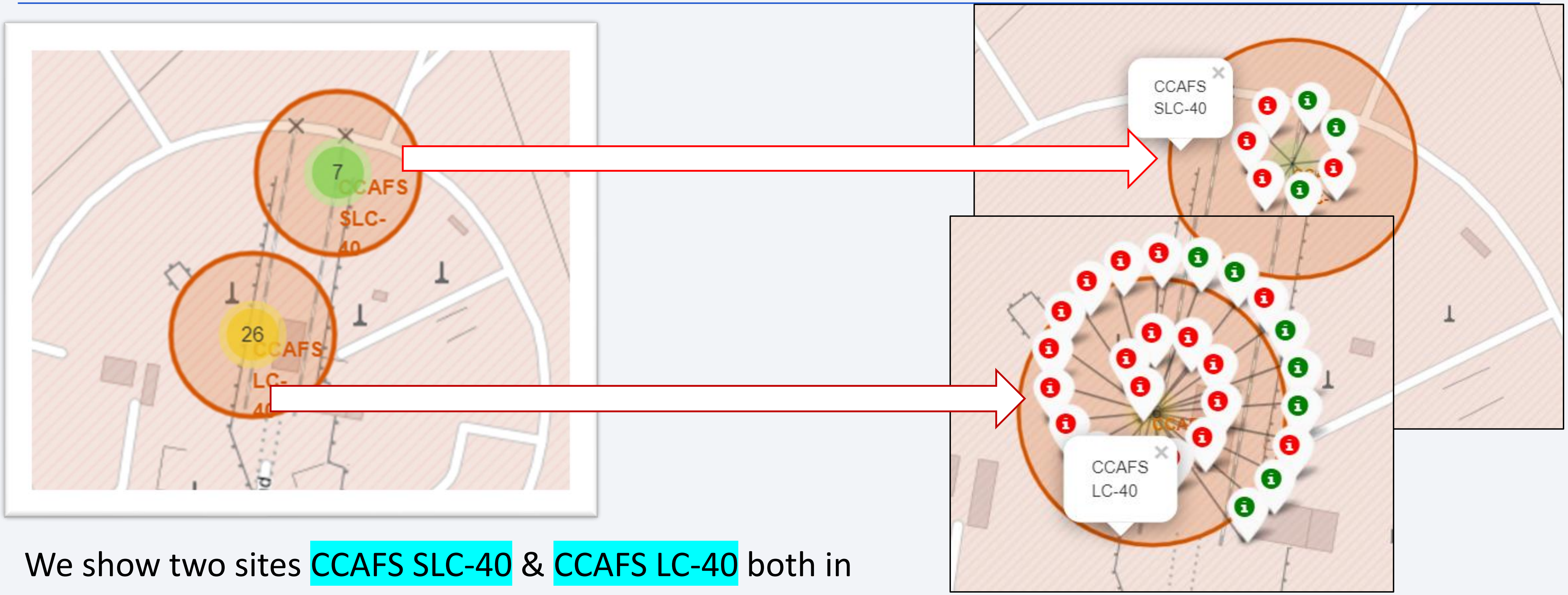


We see that the SpaceX's Launch sites are in United States of America.

All launch sites are in coastal region.

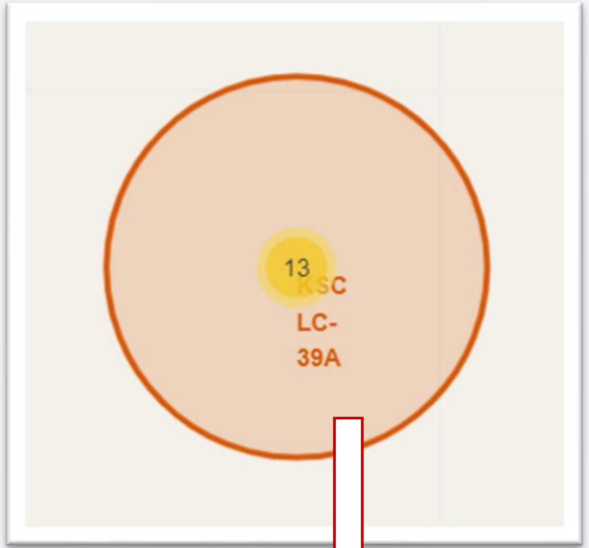
California has 1 Launch Site.
Florida has 3 Launch Sites.

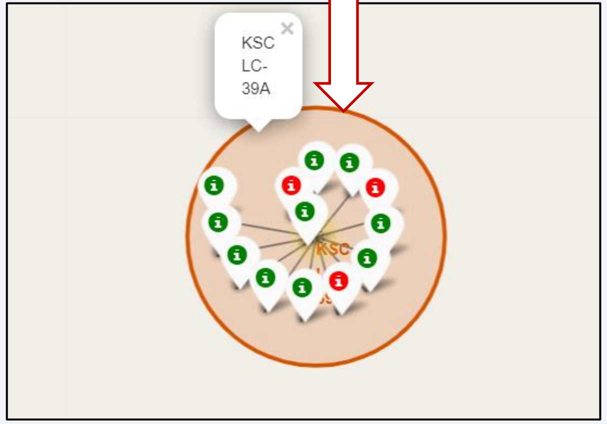
Launch Sites with Success & Fail Landing Markers



We show two sites CCAFS SLC-40 & CCAFS LC-40 both in Florida.
We see that CCAFS LC-40 has a low success ratio. While, CCAFS SLC-40 has a balanced success ratio.

Launch Sites with Success & Fail Landing Markers

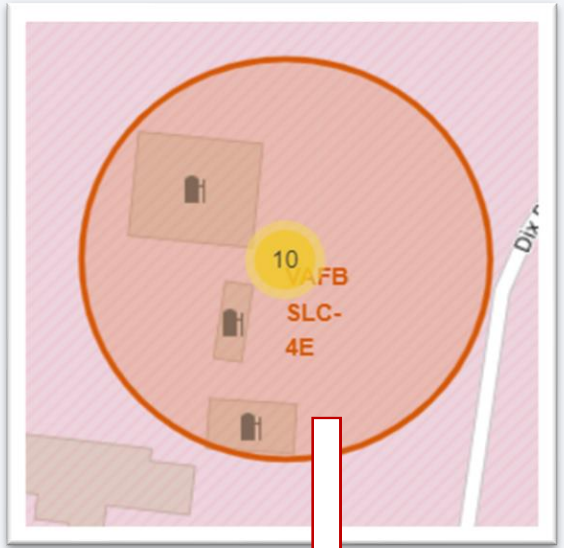


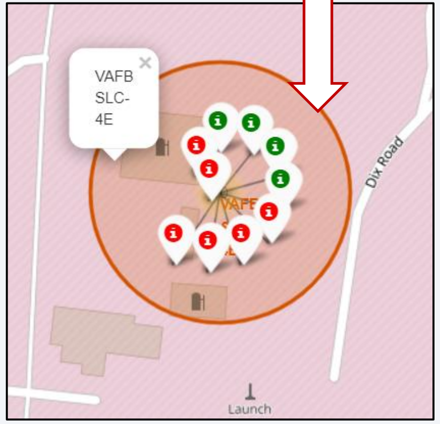


We have two sites:
KSC LC- 39A on left located in Florida.
VAFB SLC- 4E on right located in California.

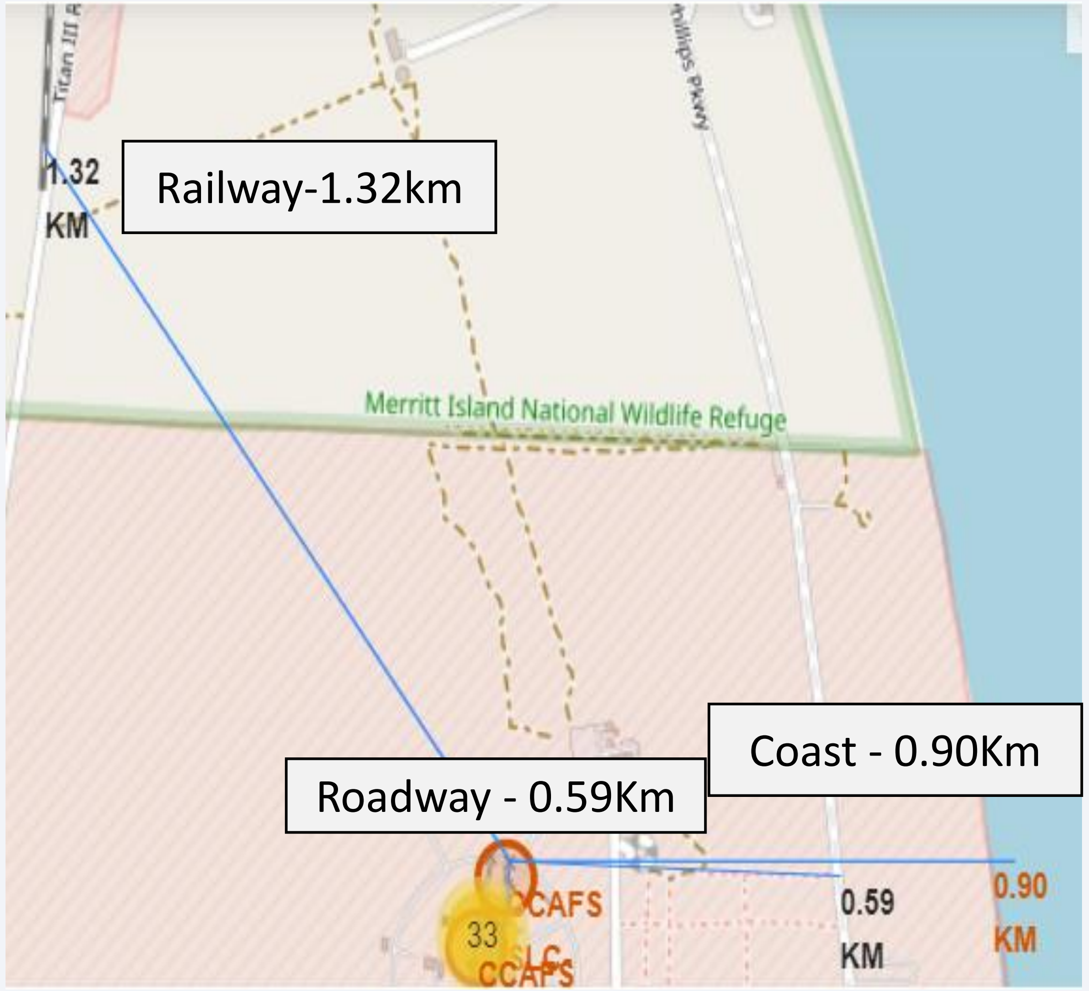
We observe that KSC LC-39A has a very high success ratio.

Relatively, VAFB SLC- 4E has an average success ratio.





Logistics and Safety



We calculate distance of CCAFS SLC-40 from nearest :
Coastline-0.90KM
Railways- 1.32 KM
Roadways-0.59KM
City – 17.84 KM

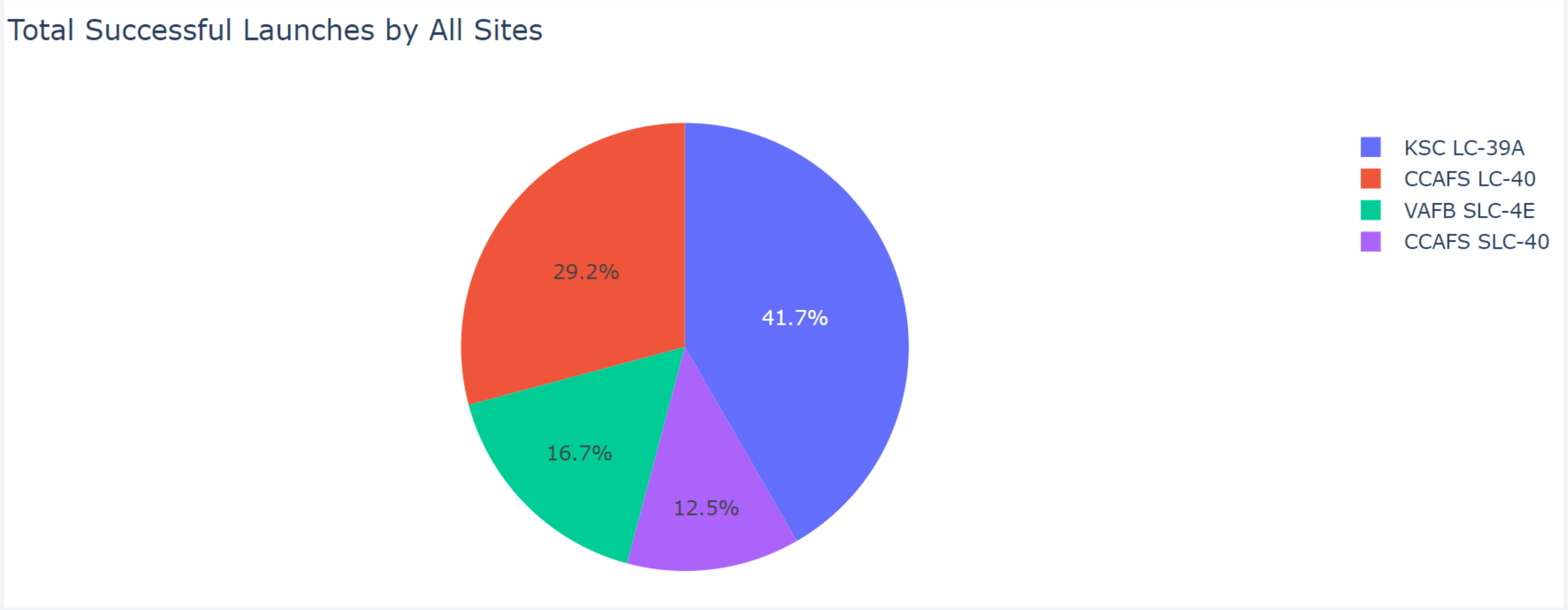
We see that the launch site is conveniently connected to road, rail & coastline for easy transportation. It also keeps a certain distance from City, hence it is also safe.

33
SLC-40
Cape Canaveral Space Force Station
City – 17.84Km
Cape Canaveral
17.84 KM

Section 4

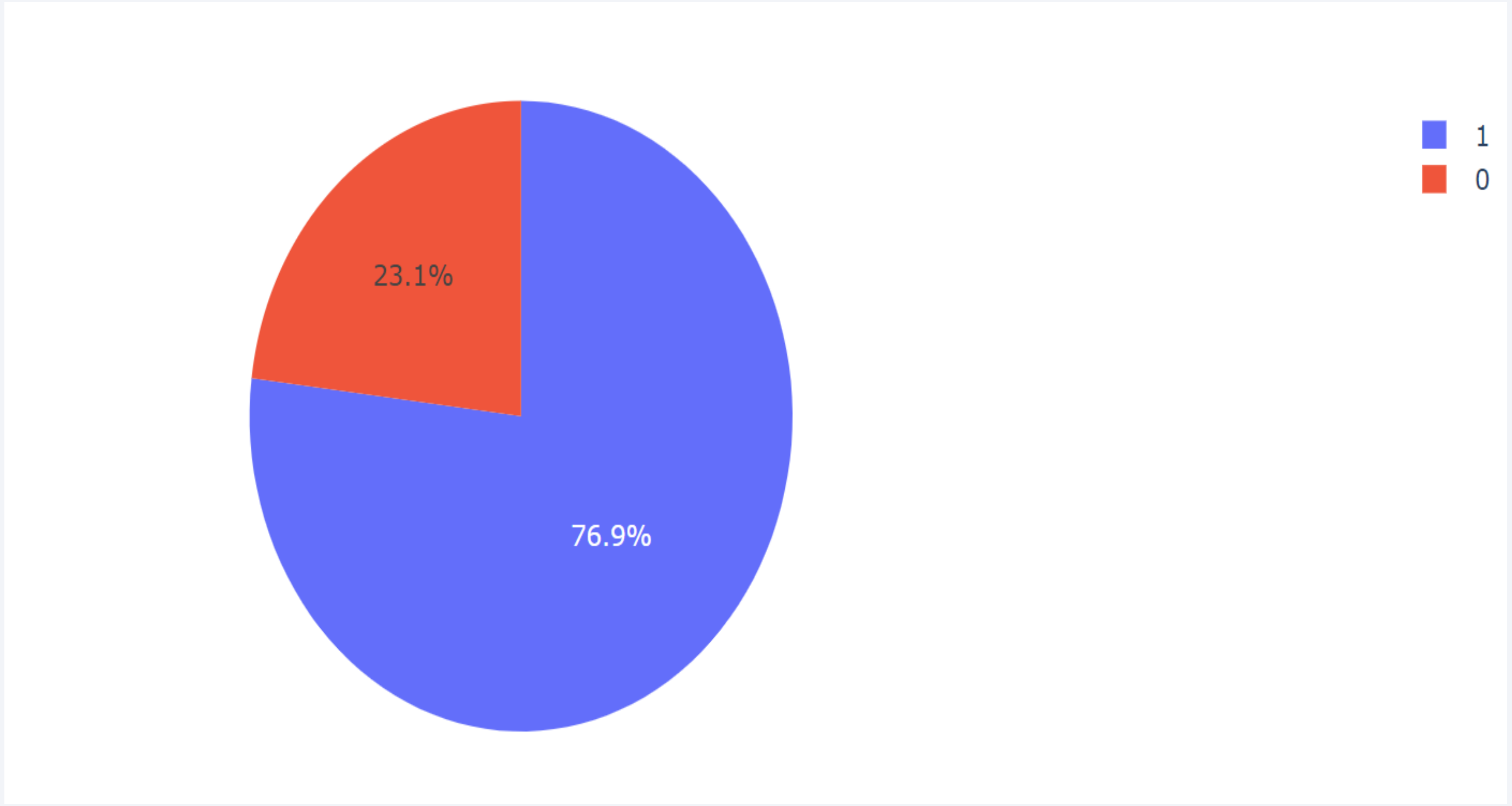
Build a Dashboard with Plotly Dash

Pie Chart for Successful Launches for All Sites



We plot the pie chart and observe that KSC LC-39A has highest number of successful launches. We also observe that CCAFS SLC-40 has lowest number.

Pie Chart for Success ratio of KSC LC-39A

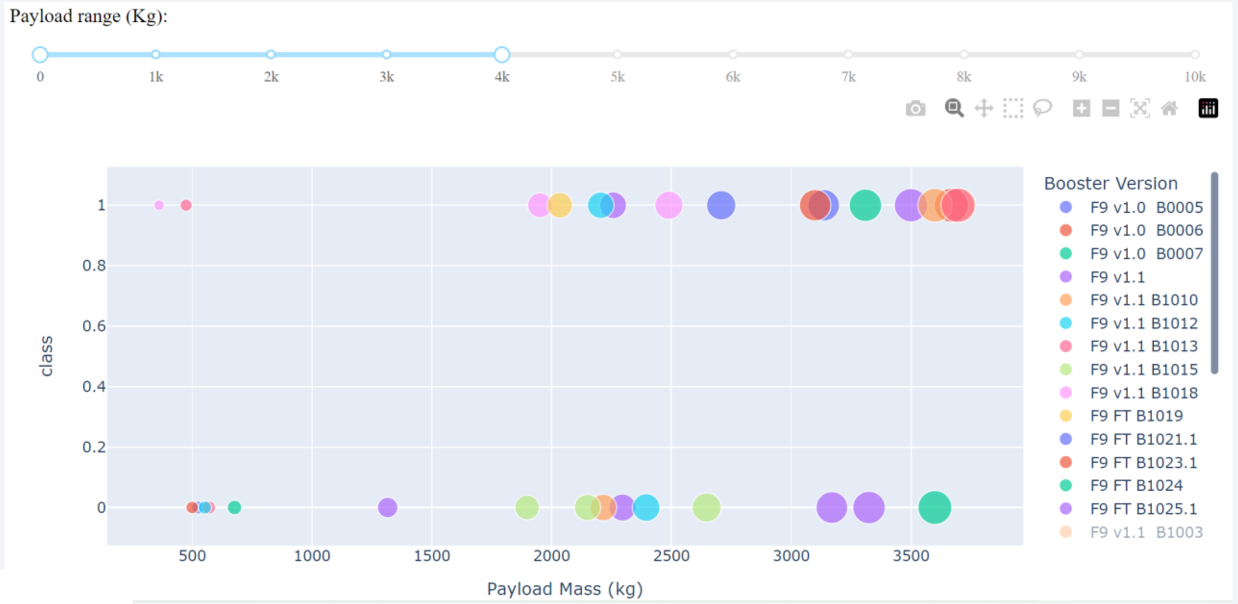


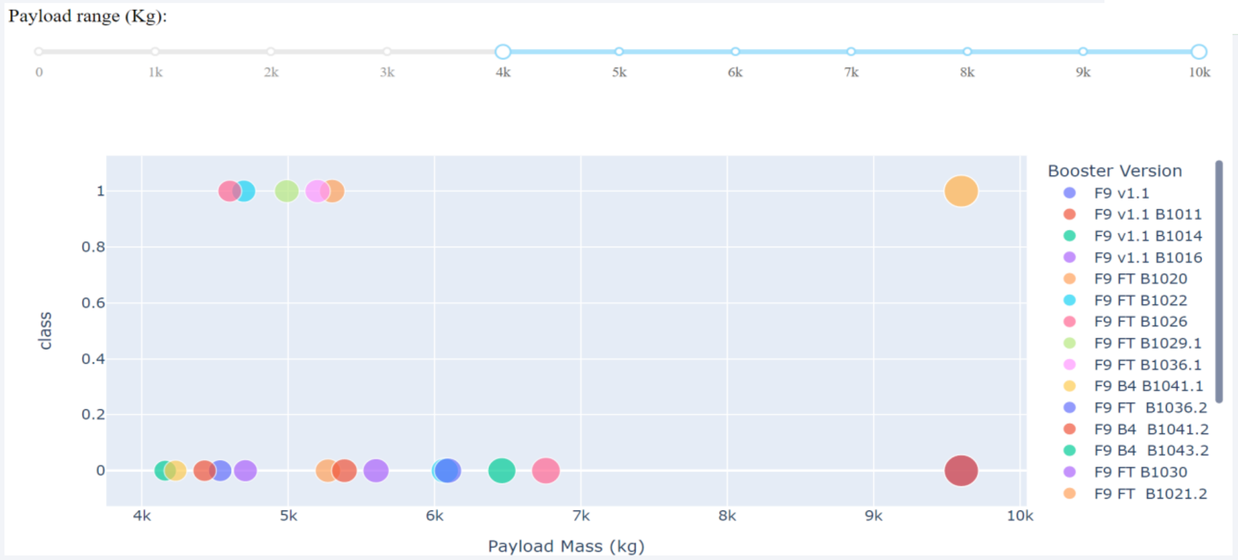
We plot the pie chart for KSC LC-39A and observe that it has a high success rate of 76.9%.

Scatter plot for all sites with different ranges in payload range slider plotting Payload vs. Outcome

We plot two scatter plots for payload ranges of 0-4000Kg & 4000-10,000Kg.

We observe that 0-4000Kg range has more points for class=1. Relatively, 4000-10,000Kg has less.



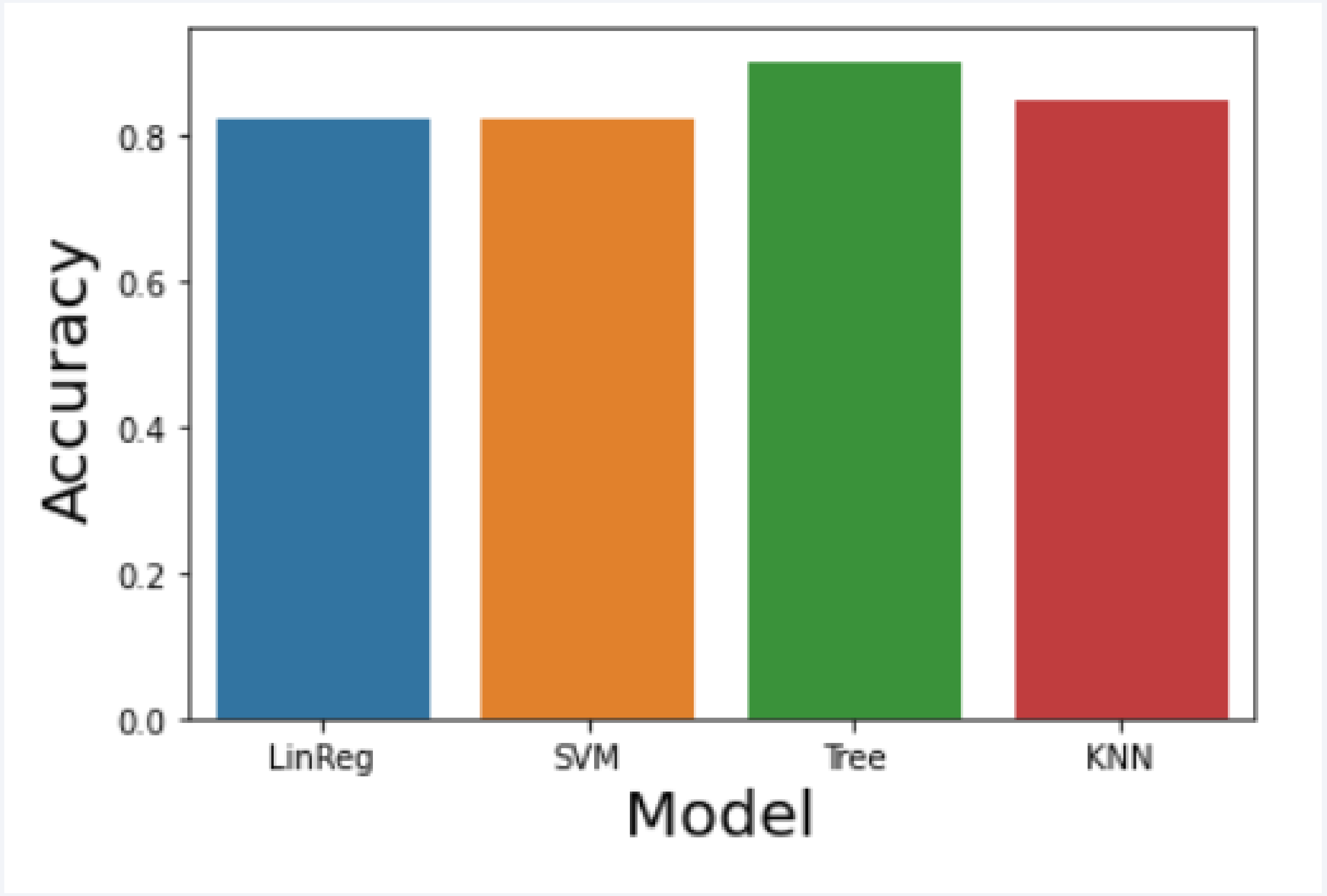


Thus, we can say that if we observe data from all sites, the launches with low payload mass have had more number of successes than those with high payload mass.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



We plot a bar chart for accuracy of our different classification models.

Linear Regression – 0.822

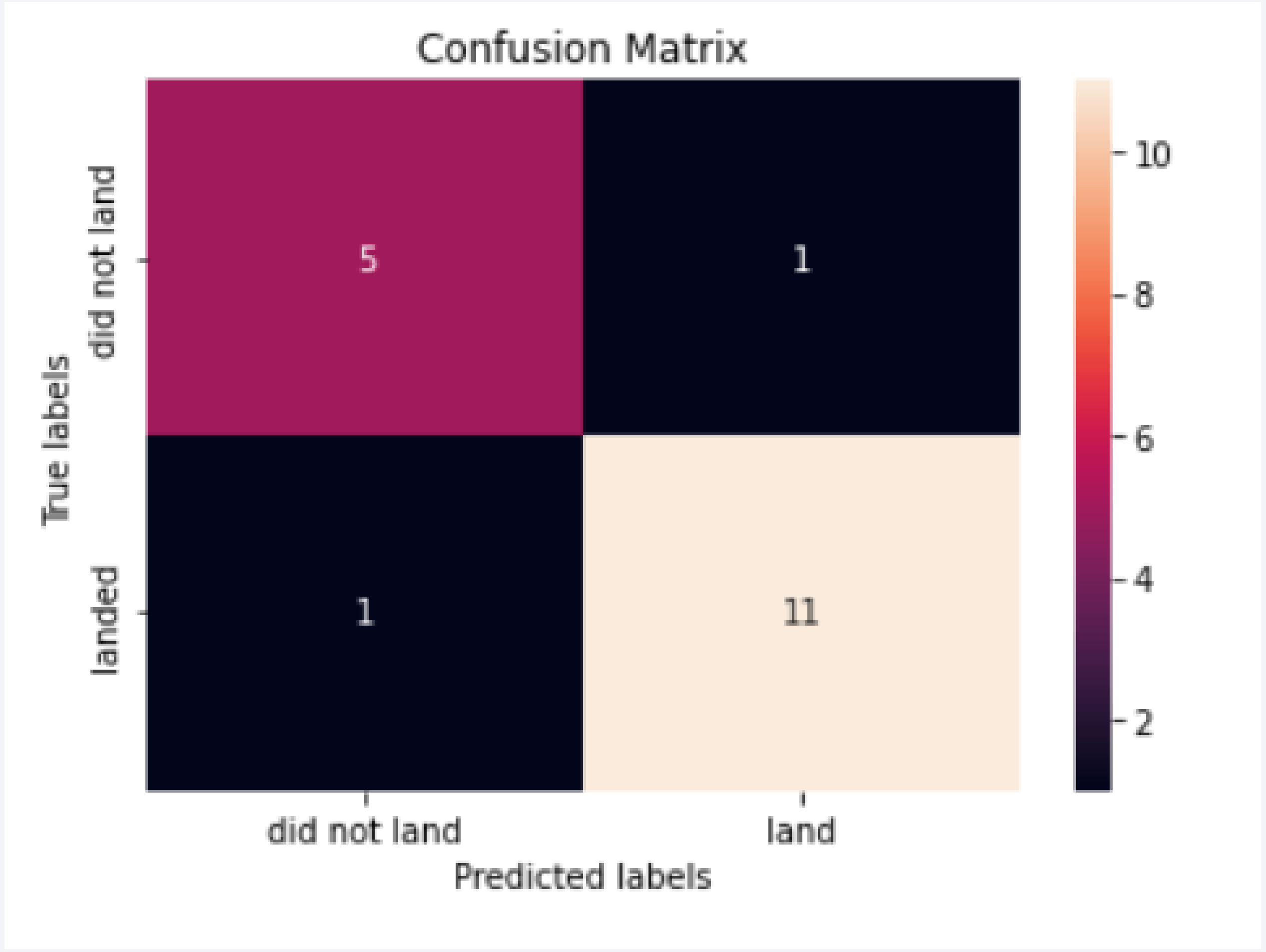
SVM – 0.822

Decision Tree – 0.90

KNN – 0.844

Hence, we see that Decision Tree is the best model for our classification purpose.

Confusion Matrix



We plot the Confusion Matrix for Decision Tree Classifier.

We observe that it has 1 case each of False-Positive and False-Negative.

This is a pretty good model for classification as it has 11 True-Positive and 5 True-Negative.

Conclusions

- We conclude that all chances of success increase with increase in number of flights for all sites.
- We conclude that Success rate is increasing with passing years.
- We conclude that launch site KSC LC-39A has highest success rate and highest number of successful launches.
- We conclude that the launch site has to be conveniently connected with railway, road and coastline. Also, it has to be a certain distance from residential area.
- We conclude that Decision Tree Classifier is the best model for classification whether launch is successful or not.

We have answered all questions we set out to find answers to.

Appendix

- The best tuned hyperparameters of our model are:

  tuned hpyerparameters :(best parameters) {'criterion': 'gini', 'max_depth': 16, 'max_features': 'sqrt’, 'min_samples_leaf': 2, ‘min_samples_split': 10, 'splitter': 'best’}.

- The GitHub notebook will not show the folium maps as it will not trust the notebook.
- Some notebooks may have large content of json file, kindly scroll down for complete code.

Thank you!