IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nicole Hermiz
January 16th, 2023

Outline

Executive Summary

- Summary of methodologies
  - Data Collection through SpaceX API & Wikipedia Web Scraping
  - Data Wrangling
  - Data Analysis via SQL & Data Visualization (Plotly)
  - Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Results of Exploratory Data Analysis
  - Screenshots of Interactive Analytics
  - Results of Predictive Analytics

Link to full GitHub repository with all artifacts mentioned in this report.

Introduction

- Background
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars. Other providers cost upwards 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, which can inform SpaceX competitors**.
- We want to determine
  - How does the relationship between different variables impact landing success (payload, orbit, customer, launch site, etc.)?
  - What variables/conditions need to be met to have the highest chance of landing success?

Section 1

Methodology

Methodology

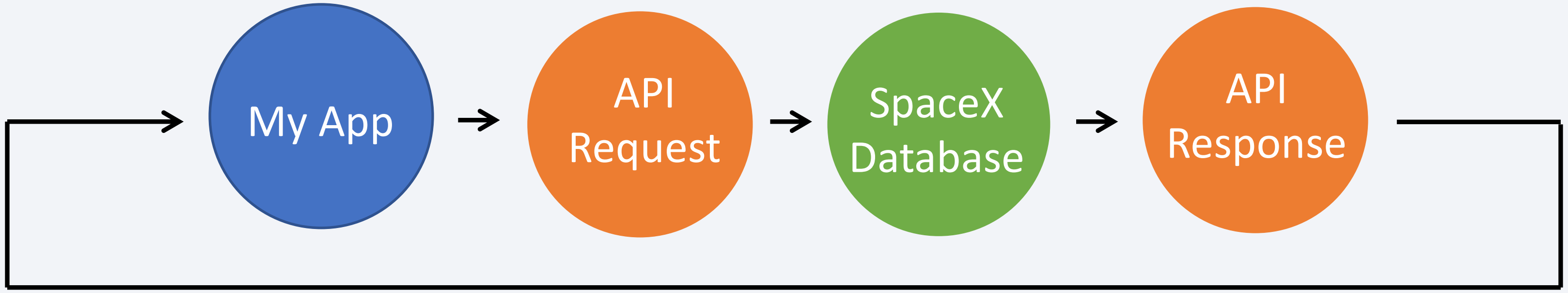
Executive Summary

- Data collection methodology:
    - Data was collected by calling the SpaceX API and via Web Scraping to obtain Rocket, Launchpad, Payload, and Cores data.
- Perform data wrangling
    - The data was narrowed to include only a subset of the data, filtered to remove falcon rockets with extra rocket boosters, and filtered to remove datapoints with multiple payloads.
    - Additional processing was done to standardize the datetime datatype, extracting the date, and narrowing the dates of the launches in the dataset.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Predictive analysis using the flowing classification models were used: Logistic Regression, Support Vector Machine, Decision Tree, and K Nearest Neighbors

Data Collection

Datasets were collected via requests to the SpaceX API.

SpaceX API Data Collection Process



Requests for **Rocket**, **Launchpad**, **Payload**, and **Cores** data were sent to the SpaceX API.

More details is included on the next slide.

For preliminary data analysis, I scraped data from the Falcon 9 Heavy Launches Wikipedia page. This is discussed in more detail on slide 9.

Data Collection – SpaceX API

The process on the right was followed to get the following data from the SpaceX API:

- **Rocket**: Booster Name
- **Launchpad**: Launch Site Name, Longitude, Latitude
- **Payload**: Mass, Orbit
- **Cores**: Landing Outcome, Type, Number of Flights with Core, Gridfins Use Status, Core Reuse Status, Legs Reuse Status, Landing Pad Reuse Status, Core Version, Core Reuse Amount, Core Serial

1. Get Response from SpaceX API

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

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

2. Convert API Response to .json

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

3. Apply custom functions to clean up data

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

4. Assign list to dictionary then data frame

```
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}
```

5. Filter data frame

```
data_falcon9 = launch_df.loc[launch_df['BoosterVersion'] == 'Falcon 9']
```

GitHub URL to the completed notebook

Data Collection – Web Scraping

The process on the right was followed to get launch data from an HTML table on the "List of Falcon 9 and Falcon Heavy launches" Wikipedia page

1. Get response from Wikipedia URL

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
print(response.status_code)
```

2. Use BeautifulSoup to parse the response

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

3. Loop through all the rows in an HTML table with relevant data, and then parsing the data into a Pandas dataframe.

Screenshot is too large to include but can be found in the GitHub repository linked.

4. Create a python dictionary

```
launch_dict= dict.fromkeys(column_names)
```

5. Create a Pandas data frame

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


GitHub URL to the completed notebook

Data Wrangling

- The data wrangling state included data profiling and manipulation, described on the right.

1. Identify and calculate the percentage of the missing values in each attribute

```
df.isnull().sum()/df.count()*100
```

2. Identify which columns are numerical and categorical:

```
df.dtypes
```

3. Create a landing class label from Outcome column

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

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

4. Assign the landing class to the Class column in the data frame

```
df['Class']=landing_class
df[['Class']].head(8)
```

GitHub URL to the completed notebook

EDA with Data Visualization

Scatter plots are used to show the relationship between variables. Line Graphs are used to show trends over time. Bar charts are used to compare categorical data.

Scatter Plots

- Flight Number vs. Payload Mass (kg)
- Flight Number vs. Launch Site
- Flight Number vs. Orbit
- Launch Site vs. Payload Mass (kg)
- Payload Mass vs. Orbit

Line Graphs

- Average Launch Success Trend over Time

Bar Charts

- Success Rate per Orbit Type

GitHub URL to the completed notebook

EDA with SQL

Summary of SQL Queries Performed

- Get the name of each distinct launch site
- Get 5 records where launch site begins with "CCA"
- Get the total payload mass carried by boosters launched by NASA (CRS)
- Get the average payload mass carried by booster version F9 v1.1
- Get the date when the first successful landing outcome in ground pad was achieved
- Get the boosters that have success in drone ship and have payload mass between 4000kg and 6000kg
- Get the total number of successful and failure mission outcomes
- Get the names of the booster versions that have carried the max payload mass
- Get the failed landing outcomes in drone ship with booster versions and launch site names in 2015
- Get the count of landing outcomes between 2010-06-04 and 2017-03-20

GitHub URL to the completed notebook

Build an Interactive Map with Folium

Summary of Map Markers

- Circle Markers were added around each Launch Site.
- Launch Outcomes were represented with green (success) and red (failure) icon markers in a Marker Cluster
- Distance lines were mapped between a Launch Site and the closest city, highway, and coastline.

The above objects were added to the map to assist in answering questions like:

- Are launch sites close to railways?
- Are launch sites close to highways?
- Are launch sites close to coastline?

[GitHub URL](#) to the completed notebook

Build a Dashboard with Plotly Dash

Summary of plots/graphs and interactions on the dashboard

- Pie chart showing number of successes for all launch sites, as well as successes per launch site
- Scatter plot showing the relationship between Payload and Success for all Launch Sites by Booster Version
- Scatter plot showing the relationship between Payload and Success for all Individual Launch Sites by Booster Versions

Pie charts show the proportions of many categories of data. Scatter Plots show the relationship between variables.

[GitHub URL](#) to source code

Predictive Analysis (Classification)

1. Prepare the datasets
   - Create a NumPy array from the Class column and assign it to variable Y as a Pandas series
   - Standardize the data by using StandarScaler().fit()
   - Use the train_test_split() function to break the X and Y data into training and testing datasets
2. For each model, create an evaluation method object and a GridSearchCV object. Fit the object to find the best parameters. Then, calculate the accuracy on the test data using the .score() method and evaluate the Confusion Matrix.
   - This was done for Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbor.
3. Determined the best performing classification model.

[GitHub URL](#) to the completed notebook

1. Prepare the datasets

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

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

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

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

2. Model execution and evaluation (ex. Logistic Regression)

```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

```
lr=LogisticRegression()

logreg_cv = GridSearchCV(lr, parameters, cv = 10)
logreg_cv.fit(X_train, Y_train)
```

```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

```
logreg_cv.score(X_test, Y_test)
```

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

Results

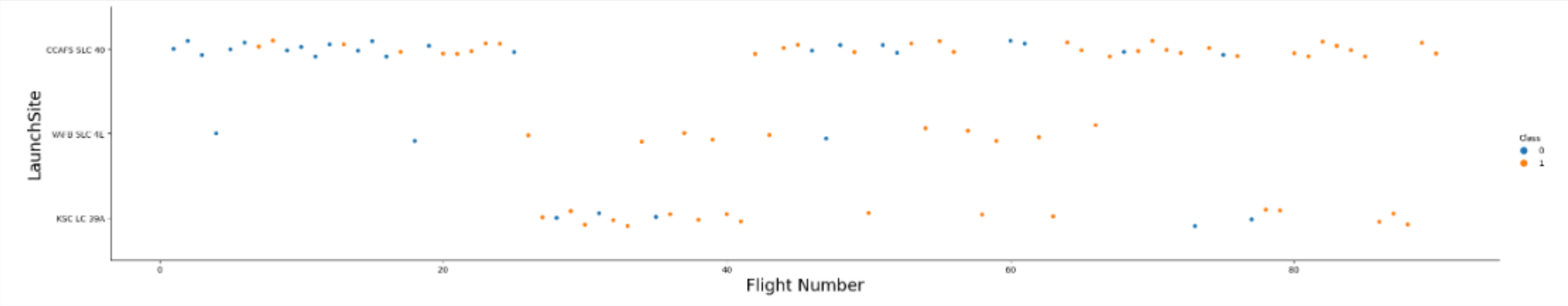
In the following slides, you'll find

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

Section 2

Insights drawn from EDA

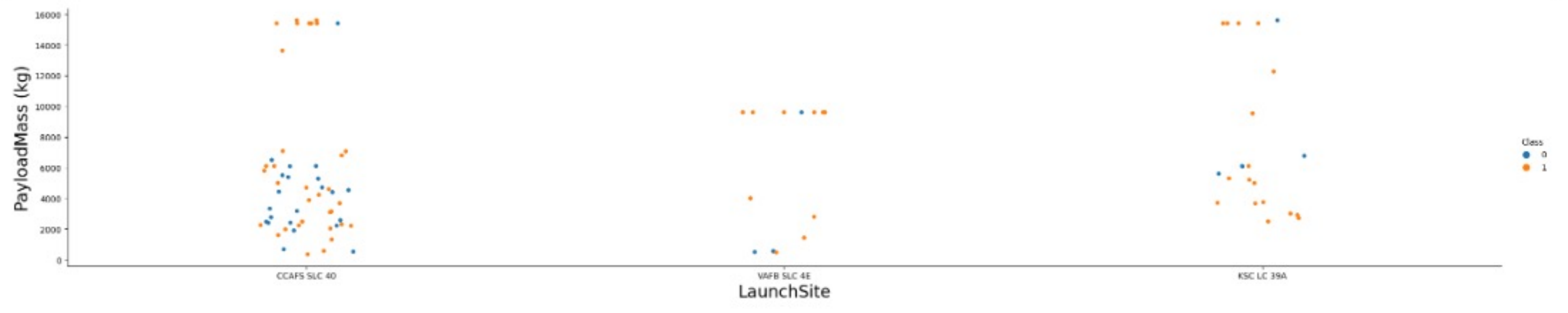
Flight Number vs. Launch Site



- The above scatter plot shows the correlation between Launch Site and Flight Number
- You can see that for Launch Site CCAFS SLC-40, the flights were more successful as the Flight Number increased.
- Although there were fewer flights at site VAFB SLC-4E, after 20 flights, it does not appear that Flight Number has an impact on success

**The Python code that generated this figure can be found in the Appendix section of the report*

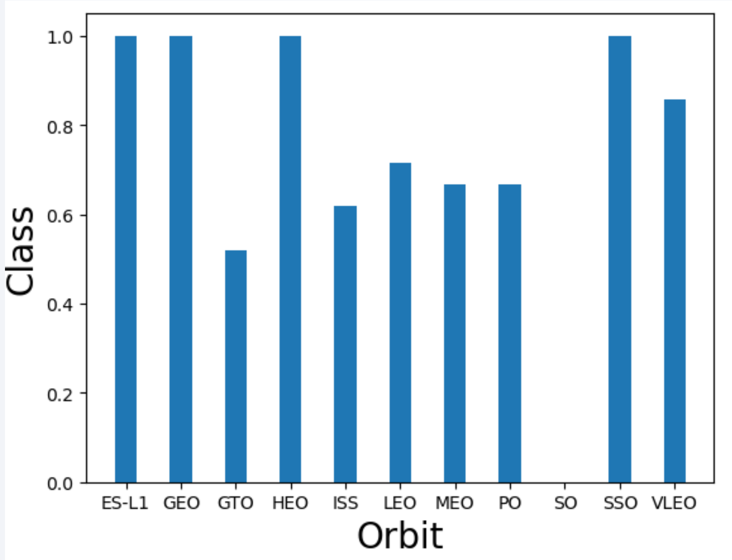
Payload vs. Launch Site



- The above scatter plot shows the correlation between Launch Site and Payload Mass (kg)
- You can see that for Launch Site CCAFS SLC-40, success likelihood is quite even until the payload mass is quite heavy, over 14,000 kg.
- At the VAFB-SLC Launch Site, there are no rockets launched for a heavy payload mass (greater than 10,000 kg).
- Finally, site KSC LC 39A has a high success rate at all payload masses.

**The Python code that generated this figure can be found in the Appendix section of the report*

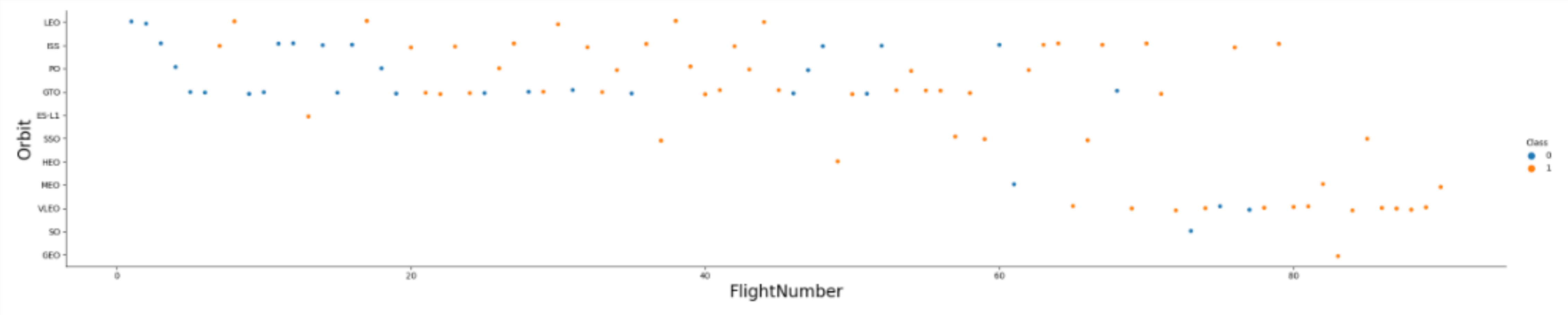
Success Rate vs. Orbit Type



- The above bar chart shows the success rate of each orbit
- The ES-L1, GEO, HEO, and SSO orbits have the highest success rates
- The GRO and SO orbits have the lowest success rates

**The Python code that generated this figure can be found in the Appendix section of the report*

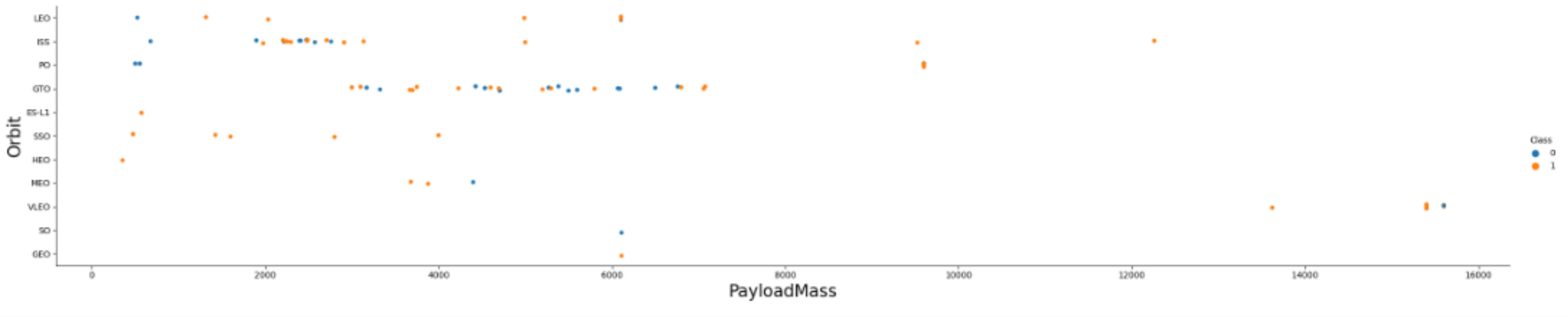
Flight Number vs. Orbit Type



- The above scatter plot shows the relationship between Orbit and Flight Number
- In the LEO orbit, success appears to be related to the number of flights.
- In the GTO orbit, there does not appear to be a relationship between flight number and success.

**The Python code that generated this figure can be found in the Appendix section of the report*

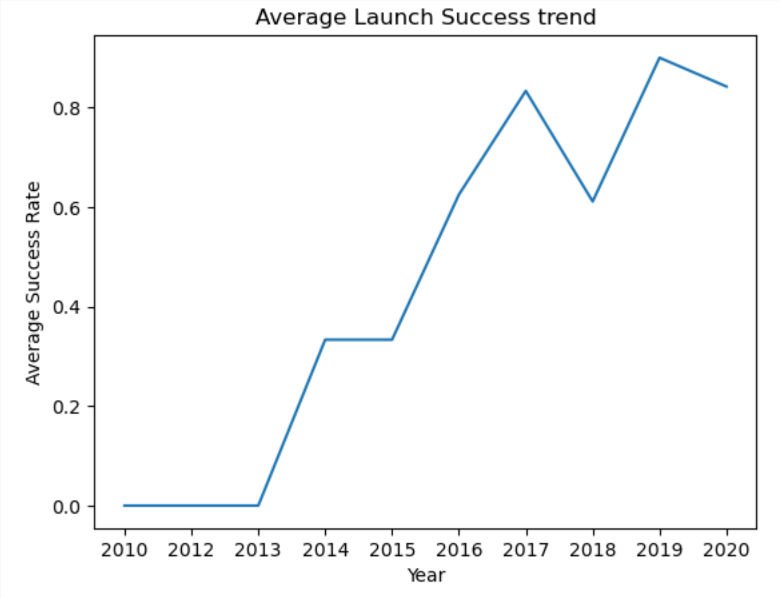
Payload vs. Orbit Type



- The above scatter plot shows the relationship between Orbit and Payload Mass (kg)
- In the LEO orbit, success appears to be related to the Payload Mass (kg). Success value is more prevalent as Payload Mass increases.
- In the GTO orbit, there does not appear to be a relationship between Payload Mass (kg) and success.

**The Python code that generated this figure can be found in the Appendix section of the report*

Launch Success Yearly Trend



- The above line chart shows the trend of Average Success Rate and year.
- You can see that the success rate has increased on average since 2013.

**The Python code that generated this figure can be found in the Appendix section of the report*

All Launch Site Names

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

- The above table shows the results of a SQL query showing the distinct Launch Sites within the SpaceX dataset, which are:
  - **CCAFS LC-40**
  - **CCAFS SLC-40**
  - **KSC LC-39A**
  - **VAFB SLC-4E**

**The SQL query that generated this table can be found in the Appendix section of the report*

Launch Site Names Begin with 'CCA'

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The above table shows the results of a SQL query showing 5 records where the launch site name begins with “CCA” (column 4). The above table shows the launch site **“CCAFS LC-40”**, and there is another launch site called **“CCAFS SLC-40”**.

**The SQL query that generated this table can be found in the Appendix section of the report*

Total Payload Mass

customer	Total Payload Mass
NASA (CRS)	45596

- The above table shows the results of a SQL query displaying the sum of payload mass for all flights where NASA (CRS) was the customer, which was **45,596 kg**.

**The SQL query that generated this table can be found in the Appendix section of the report*

Average Payload Mass by F9 v1.1

booster_version	Avg Payload Mass
F9 v1.1	2928

- The above table shows the results of a SQL query displaying the average payload mass for all flights where the booster version was F9 v1.1, which was **2,928 kg**.

**The SQL query that generated this table can be found in the Appendix section of the report*

First Successful Ground Landing Date

DATE	time__utc_	booster_version	launch_site
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40

- The above table shows the results of a SQL query displaying the first successful landing outcome on ground pad, which was **on December 22nd, 2015**.

**The SQL query that generated this table can be found in the Appendix section of the report*

Successful Drone Ship Landing with Payload between 4000 and 6000

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

- The above table shows the results of a SQL query displaying the names of boosters which have successfully landed on drone ship and had payload mass between 4,000 kg and 6,000 kg. The boosters are:
  - **F9 FT B1022**
  - **F9 FT B1026**
  - **F9 FT B1021.2**
  - **F9 FT B1031.2**

**The SQL query that generated this table can be found in the Appendix section of the report*

Total Number of Successful and Failure Mission Outcomes

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

- The above table shows the results of a SQL query displaying the total number of successful and failure mission outcomes
- There was **1 in-flight Failure and 100 Successes**

**The SQL query that generated this table can be found in the Appendix section of the report*

Boosters Carried Maximum Payload

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

- The above table shows the results of a SQL query displaying the names of the booster which have carried the maximum payload mass, which is **15,600 kg**

**The SQL query that generated this table can be found in the Appendix section of the report*

2015 Launch Records

landing__outcome	booster_version	launch_site	Year
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015

- The above table shows the results of a SQL query displaying the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015. **There are 2 instances of drone ship failures in 2015.**

**The SQL query that generated this table can be found in the Appendix section of the report*

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

landing__outcome	Num Landing Outcomes
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

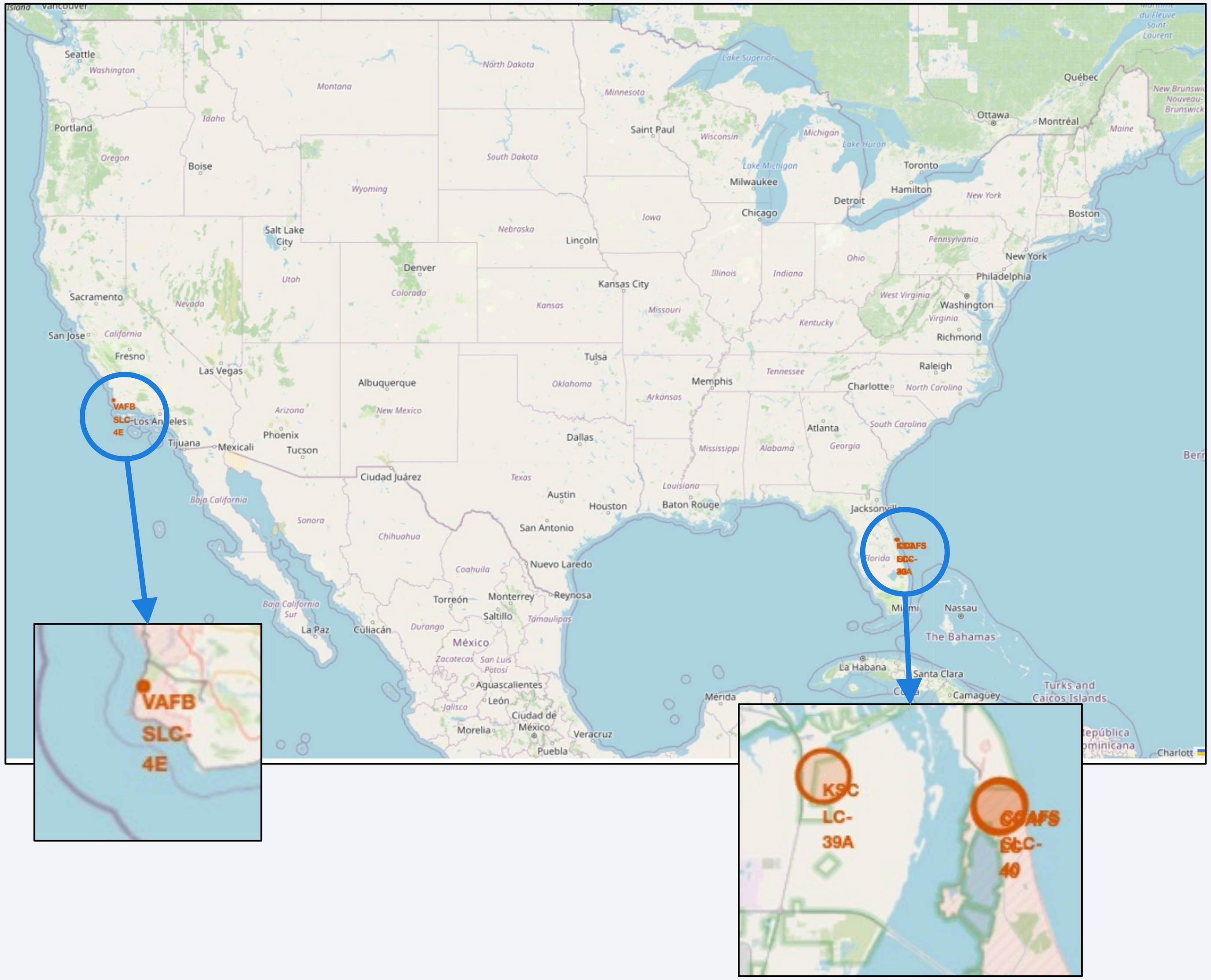
- The above table shows the results of a SQL query displaying the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

**The SQL query that generated this table can be found in the Appendix section of the report*

Section 3

Launch Sites Proximities Analysis

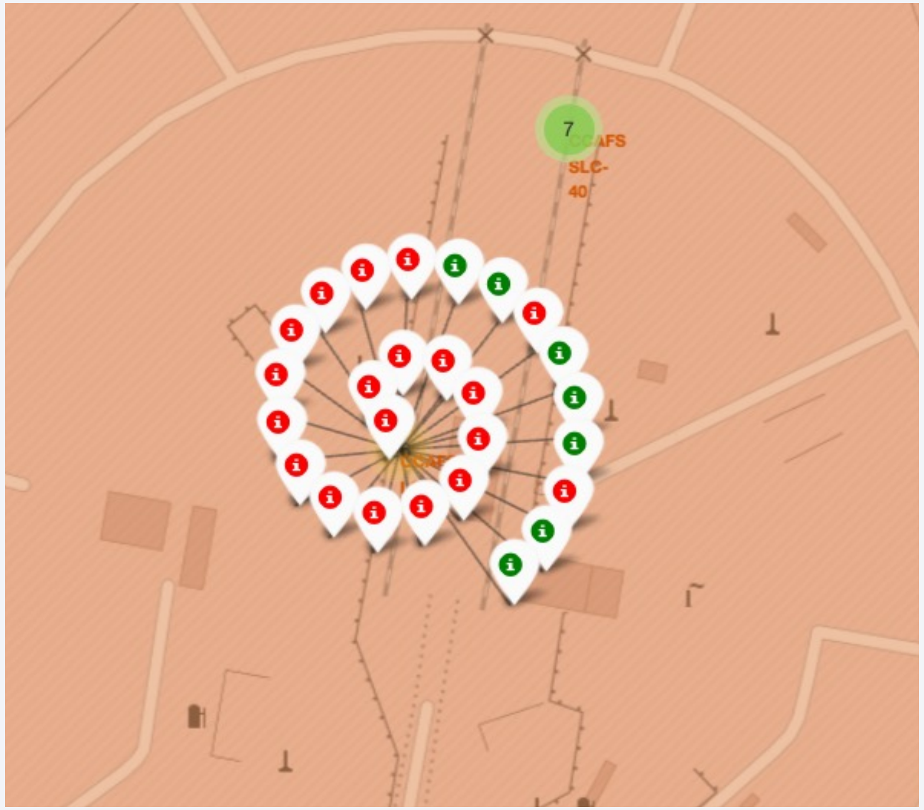
Map of Launch Sites



- There are three launch sites represented on the map
- One in California (VAFB SLC-4E) and three on the easter coast of Florida (KSC LC-39A, CCAFS SLC-40, CCAFS LC-40)

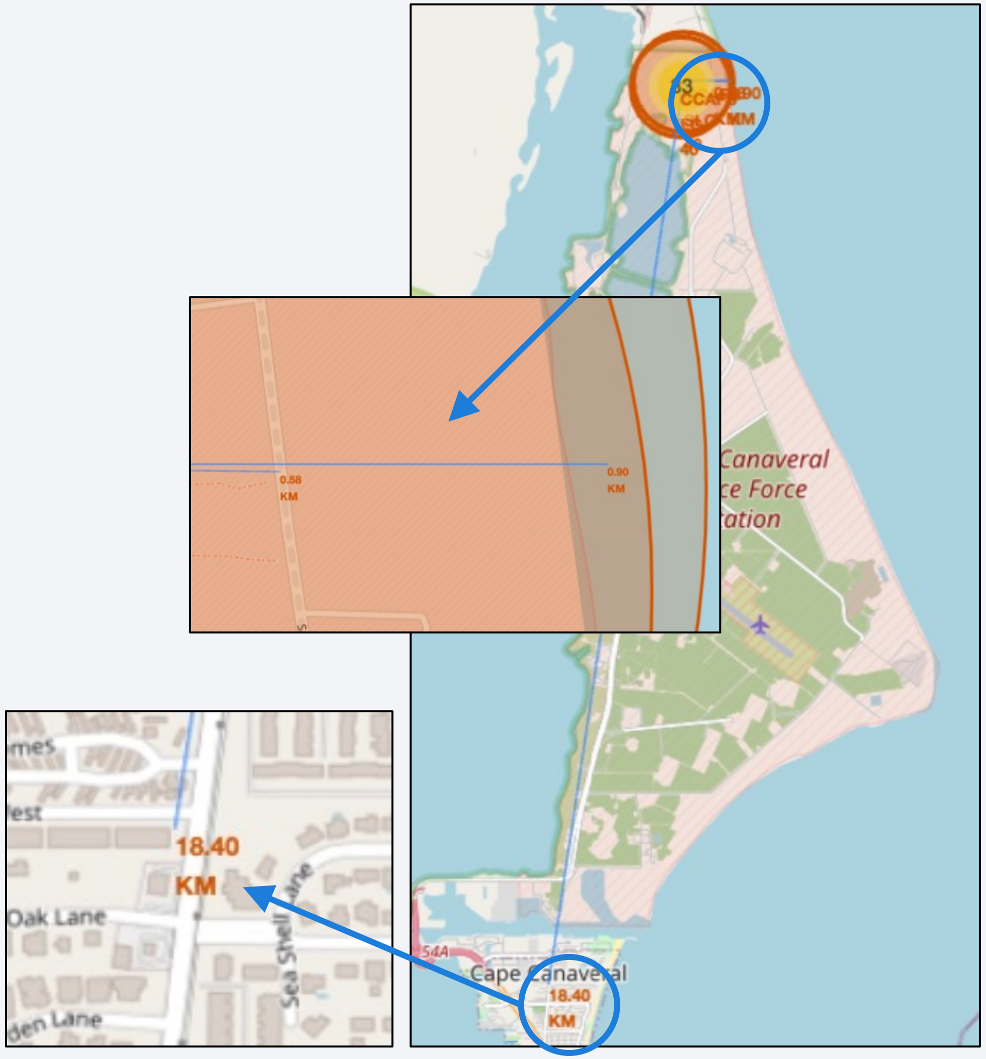
Color Coded Map of Successful and Failed Launches at CCAFS LC-40

- Here is a screenshot of site CCAFS LC-40 with 26 launches. Green icons represent successful launches and red icons represent failed launches. Within the dataset, about 73% of the launches were failed at this site.



Launch Site CCAFS SLC-40 and Location Distances

- Here is a map of approximate distances from Launch Site CCAFS SLC-40 and the nearest highway, coastline, & city
  - Highway: 0.58km
  - Coastline: 0.90 km
  - City: 18.40 km
- We can see that Launch Sites are kept relatively far away from cities compared to highways and coastlines.

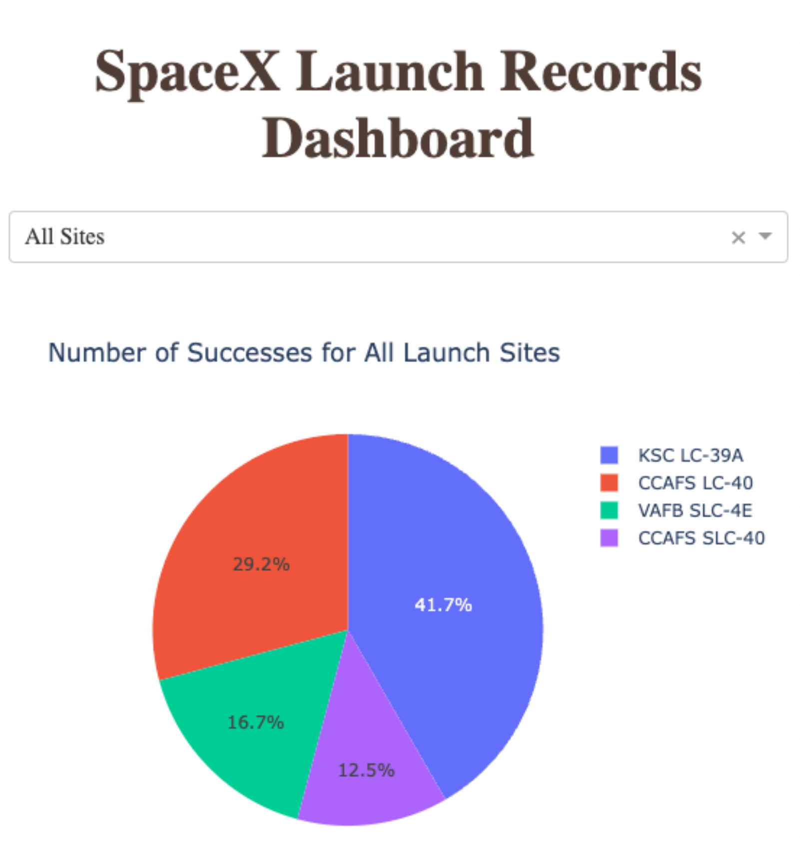


Section 4

Build a Dashboard with Plotly Dash

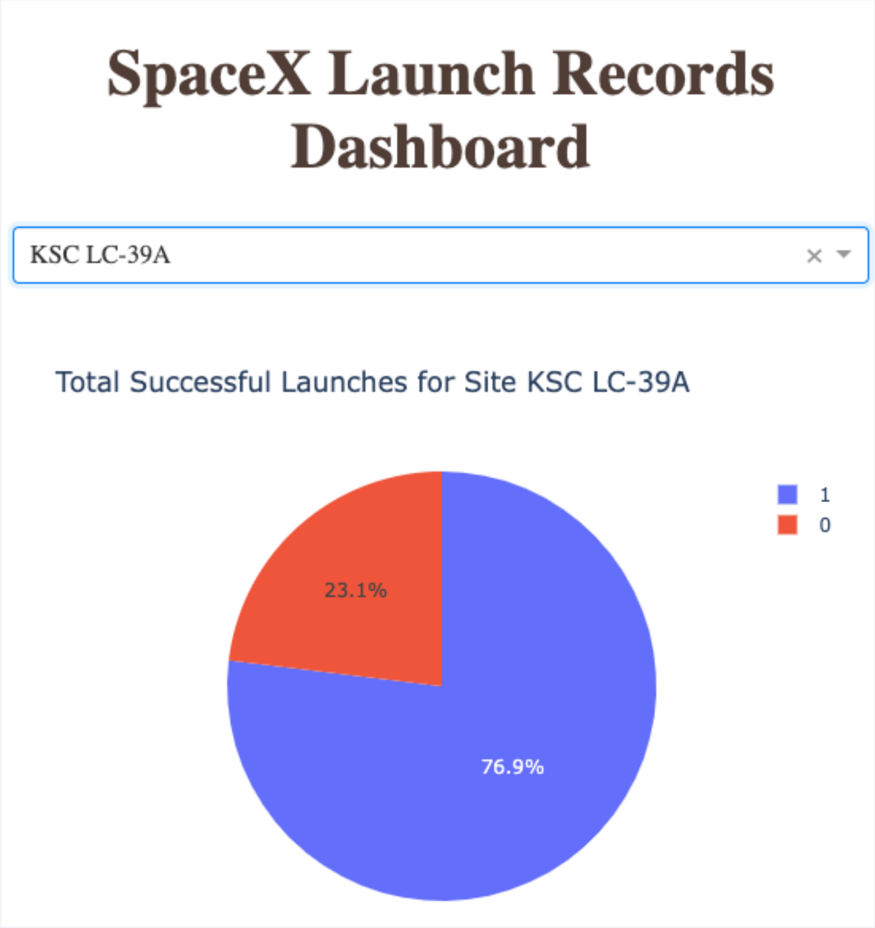
SpaceX Launch Successes per Site

- Launches at site KSC LC-491 have the highest success rate, at 41.7%. CCAFS LC-40 has the second highest success rate, at 29.2%. The other two locations have relatively lower success rates.



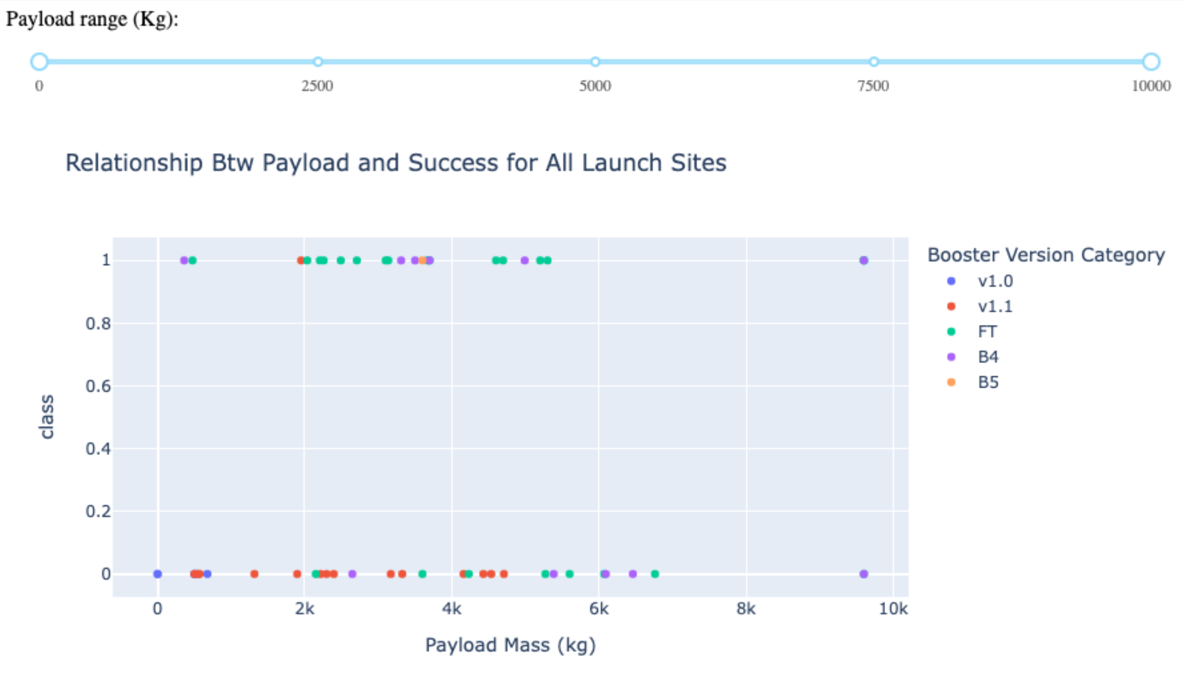
KSC LC-39A Successful and Failed Launch Ratio

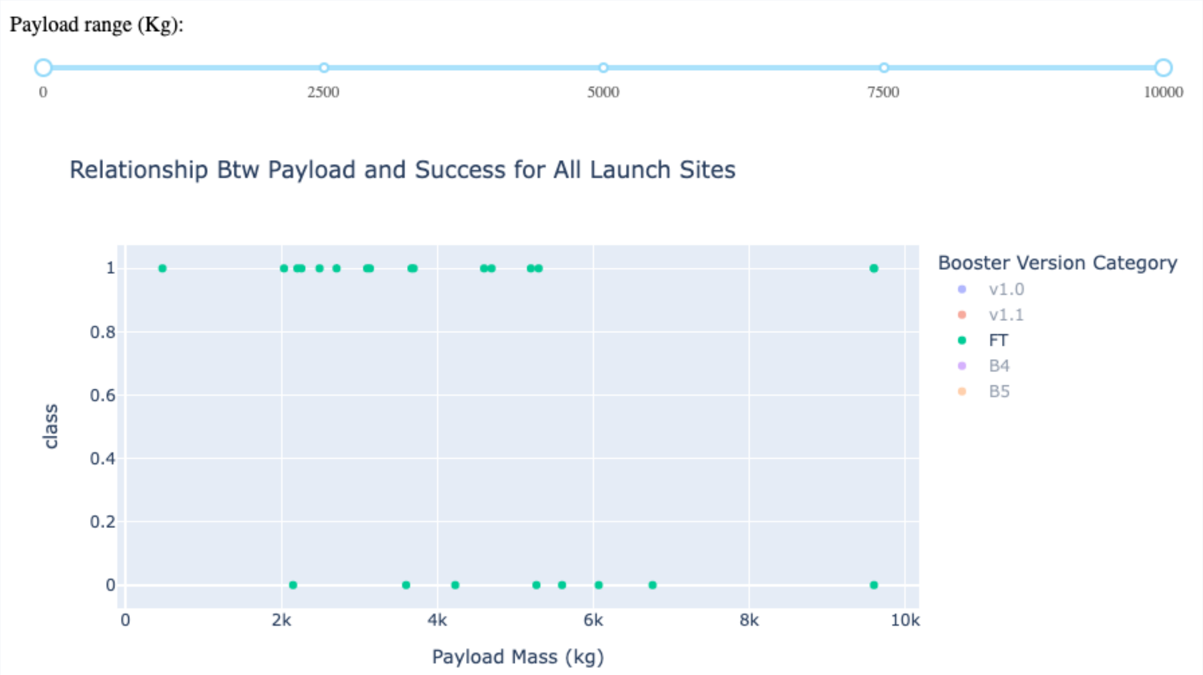
- Out of the 13 launches at KSC LC-39A, 10 (76.9%) were successful and 3 failed (23.1%)



Relationship btw Payload and Success for All Launch Sites by Booster Version

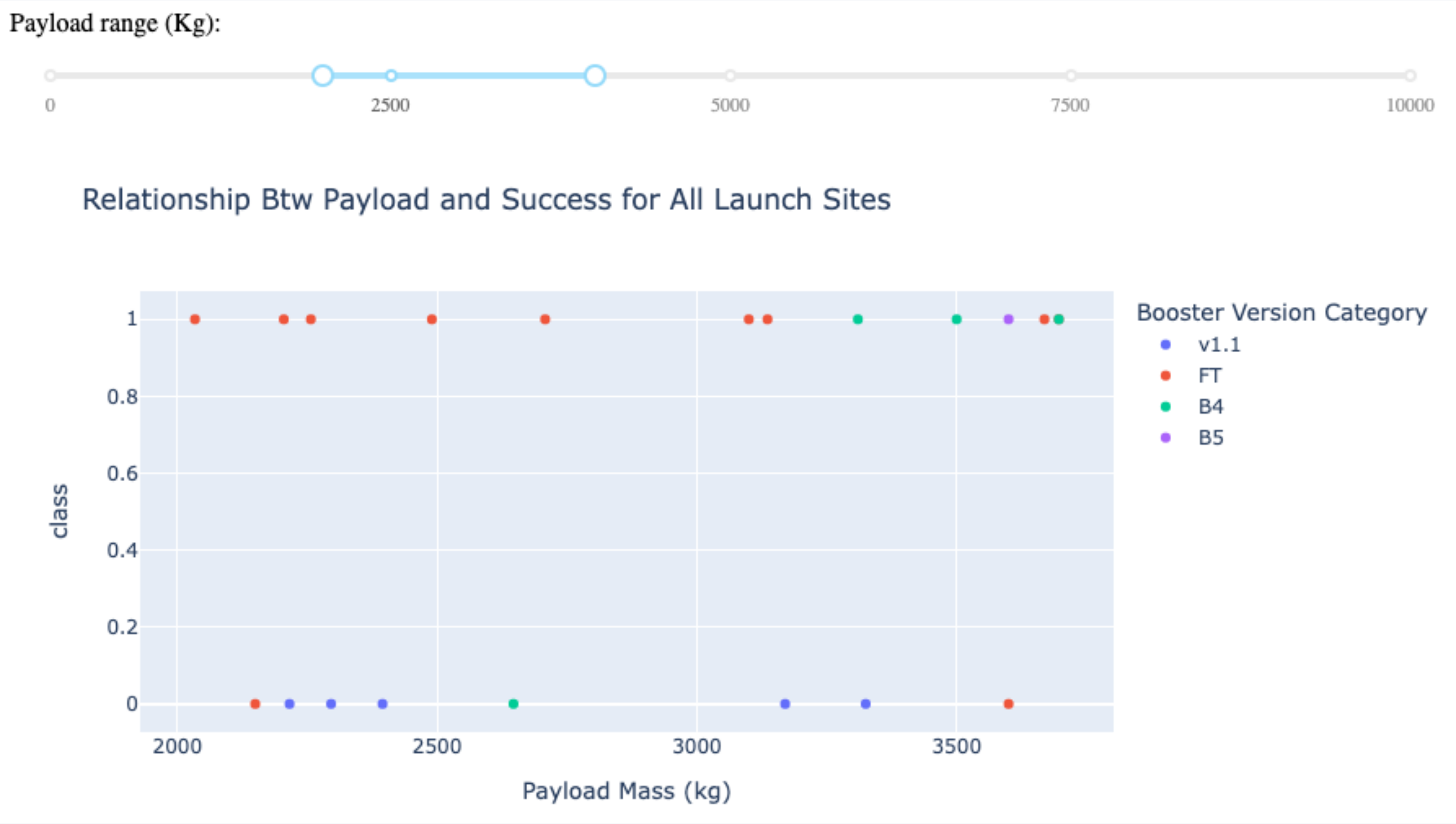
- For all payload ranges, Booster Version FT has the highest success rate.





Relationship btw Payload and Success for All Launch Sites by Booster Version

- The must successful payload range for all Booster Versions seems to be about 2000 kg to 4000 kg.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The Decision Tree Classifier model has the highest classification accuracy, with a score of **83.33%.**
- After finding the best parameters, the score improves to **88.93%**

Accuracy of Decision Tree Classifier Model

Calculate the accuracy of tree_cv on the test data using the method `score`:

```
tree_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

Best Parameters

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 2, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_sam
ples_split': 2, 'splitter': 'best'}
accuracy : 0.8892857142857145
```

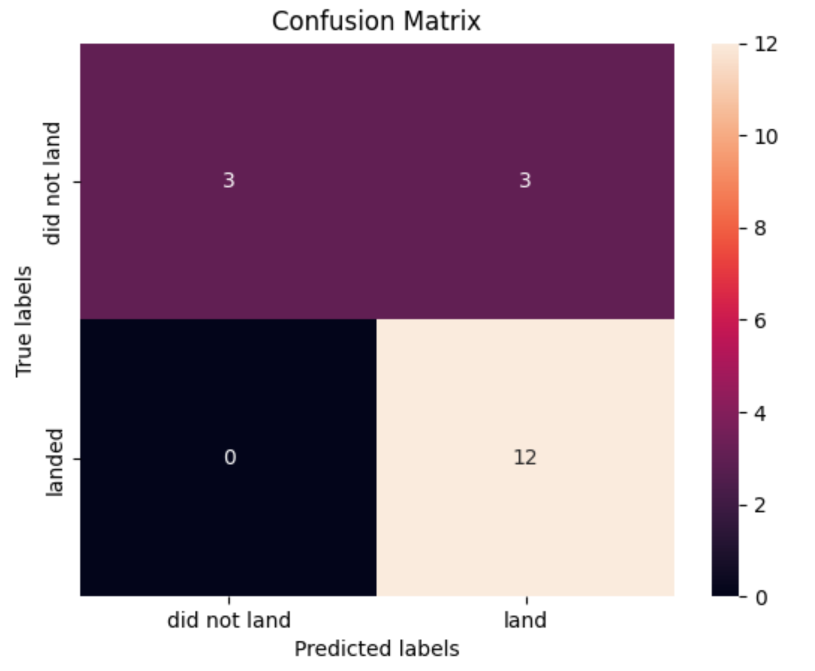
Confusion Matrix

A Confusion Matrix is used to show the following results of a machine learning problem:

- True Positives
- True Negatives
- False Positives
- False Negatives

Examining the Confusion Matrix, we see that Decision Tree can distinguish between the different classes, but that the major problem is false positives.

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



Conclusions

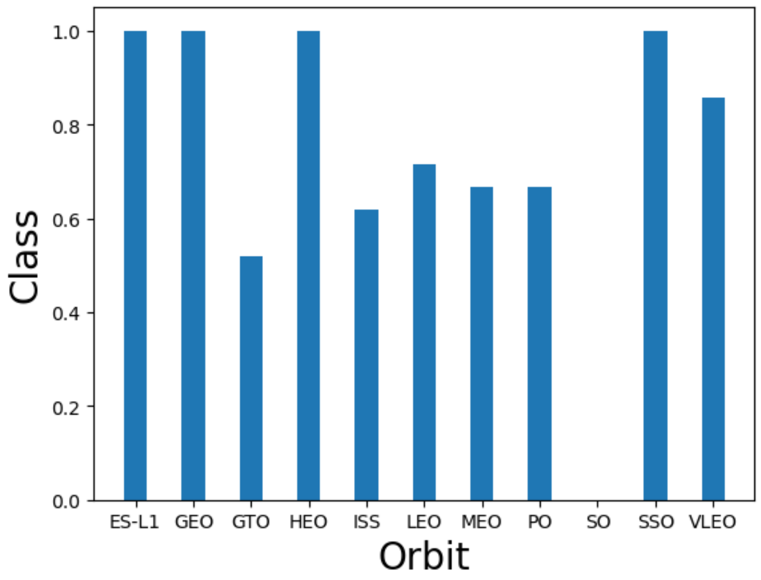
- The **Decision Tree Classifier** is the best algorithm to use to analyze this dataset.
- Higher **Flight Numbers** tend to have a greater chance of success.
- Smaller **Payloads** tend to have a greater chance of success.
- **Orbits** ES-L1, GEO, HEO, and SSO have the highest success rates.
- **Launch Site** KSC LC 39A has a high success rate at all payload masses. At Launch Site CCAFS SLC-40, success likelihood is quite even until the payload mass is quite heavy, over 14,000 kg, which is when the Success Rate drops.
- **Success Rates** have been increasing since 2013.

Thank you!

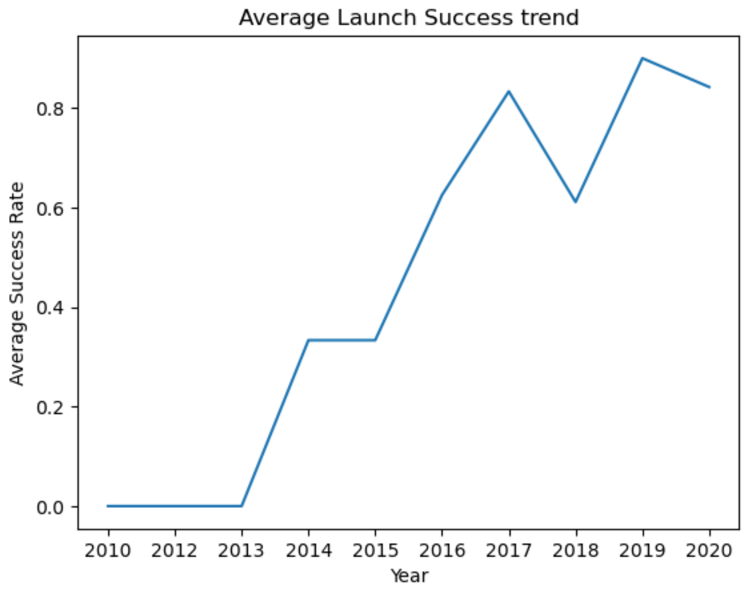
Appendix

Relevant Python Code

```
# HINT use groupby method on Orbit column and get the mean of Class column

df[['Orbit','Class']].groupby('Orbit').mean().sort_values('Class')
plt.bar(np.unique(df["Orbit"]),df.groupby(['Orbit']).mean()['Class'],width = 0.4)
plt.xlabel("Orbit", fontsize = 20)
plt.ylabel("Class", fontsize = 20)
plt.show()
```



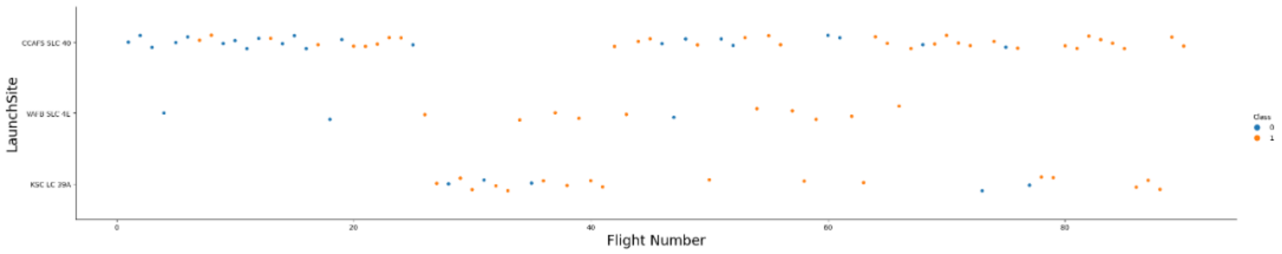

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate

#########
df_year = pd.DataFrame(Extract_year(df['Date']), columns=['year'])
df_year['Class']=df['Class']
sns.lineplot(x = np.unique(df_year['year']), y = df_year.groupby('year')['Class'].mean())

plt.title('Average Launch Success trend')
plt.xlabel('Year')
plt.ylabel('Average Success Rate')
plt.show()
```



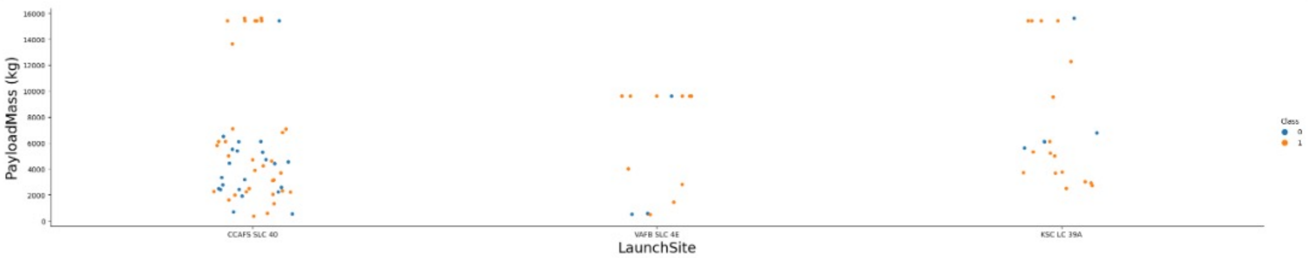
Appendix

Relevant Python Code and SQL Queries

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```




```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="PayloadMass", x="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("LaunchSite",fontsize=20)
plt.ylabel("PayloadMass (kg)",fontsize=20)
plt.show()
```




```
%%sql

SELECT * FROM SPACEX
WHERE landing__outcome LIKE '%Success%'
ORDER BY Date ASC
LIMIT 1
```

```
 * ibm_db_sa://wjx97778:***@125f9f61-9715-46f9
Done.
```

DATE	time__utc_	booster_version	launch_site
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40

```
%%sql

SELECT Booster_Version, landing__outcome, PAYLOAD_MASS__KG_ FROM SPACEX
WHERE landing__outcome = 'Success (drone ship)' AND
PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

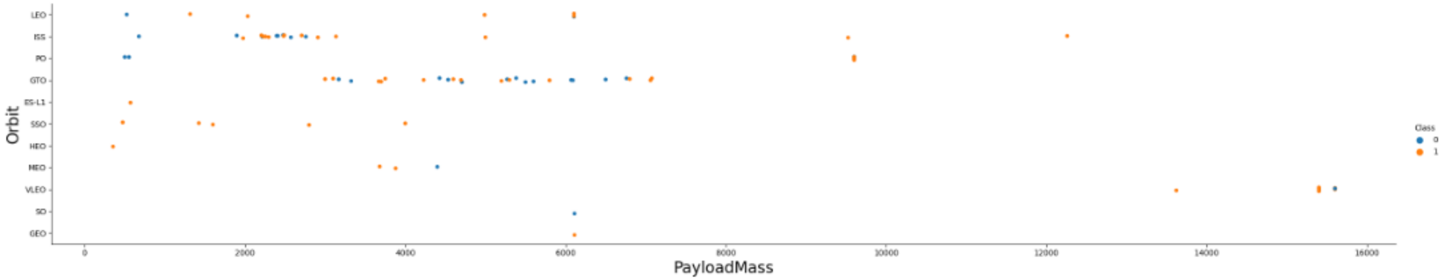
```
 * ibm_db_sa://wjx97778:***@125f9f61-9715-46f9-9399-c8177b21803b.clogj3sd0
Done.
```

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

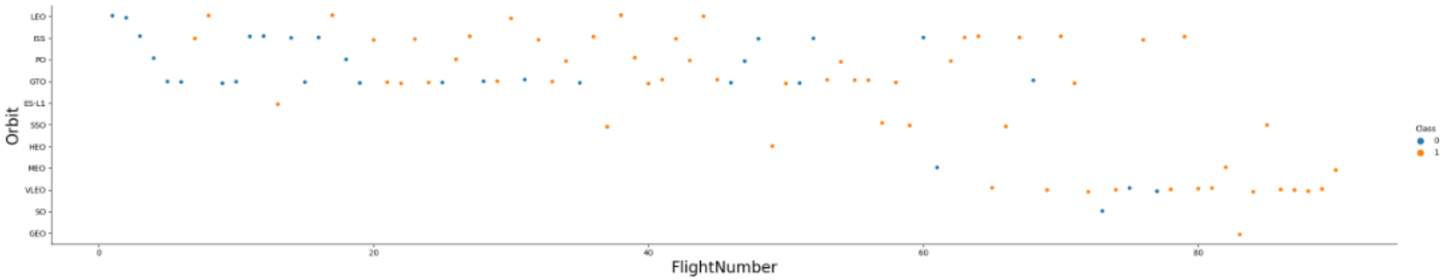
Appendix

Relevant Python Code and SQL Queries

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```




```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```




```
%%sql

SELECT DISTINCT launch_site FROM SPACEX
```

```
 * ibm_db_sa://wjx97778:***@125f9f61-9715-
Done.
```

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

```
%%sql

SELECT Mission_Outcome, COUNT(*) FROM SPACEX
GROUP BY Mission_Outcome
```

```
 * ibm_db_sa://wjx97778:***@125f9f61-9715-46f9-
Done.
```

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

Appendix

Relevant SQL Queries

```
%%sql

SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEX
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX)
```

```
 * ibm_db_sa://wjx97778:***@125f9f61-9715-46f9-9399-c8177b21803b.clogj3
Done.
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

```
%%sql

SELECT landing__outcome, COUNT(*) as "Num Landing Outcomes" FROM SPACEX
WHERE date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY COUNT(*) DESC
```

```
 * ibm_db_sa://wjx97778:***@125f9f61-9715-46f9-9399-c8177b21803b.clogj3sd
Done.
```

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

```
%%sql

SELECT landing__outcome, Booster_Version, Launch_Site, year(date) as "Year" FROM SPACEX
WHERE landing__outcome = 'Failure (drone ship)' AND
year(date) = '2015'
```

```
 * ibm_db_sa://wjx97778:***@125f9f61-9715-46f9-9399-c8177b21803b.clogj3sd0tgtu0lqde00.dat
Done.
```

landing__outcome	booster_version	launch_site	Year
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015

```
%%sql

SELECT Booster_Version, AVG(PAYLOAD_MASS__KG_) as "Avg Payload Mass" FROM SPACEX
WHERE Booster_Version = 'F9 v1.1'
GROUP BY Booster_Version
```

```
 * ibm_db_sa://wjx97778:***@125f9f61-9715-46f9-9399-c8177b21803b.clogj3sd0tgtu0lqd
Done.
```

booster_version	Avg Payload Mass
F9 v1.1	2928

Appendix

Relevant SQL Queries

```
%%sql

SELECT * FROM SPACEX
WHERE launch_site LIKE 'CCA%'
LIMIT 5
```

```
 * ibm_db_sa://wjx97778:***@125f9f61-9715-46f9-9399-c8177b21803b.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/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

```
%%sql

SELECT Customer, SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass" FROM SPACEX
WHERE Customer = 'NASA (CRS)'
GROUP BY Customer
```

```
 * ibm_db_sa://wjx97778:***@125f9f61-9715-46f9-9399-c8177b21803b.clogj3sd0tgtu
Done.
```

customer	Total Payload Mass
NASA (CRS)	45596