IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Cyrus Cai
2022-08-18

Outline

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

Executive Summary

- Data is collected via API and web scaping. EDA and interactive analytics is used to identify key features for prediction. In order to find out the best parameters, grid search method is applied to multiple models (Logistic Regression, SVM, Decision Tree, and KNN).
- Using the LR model with its best parameters, the launch success rate can increase to 80%

Introduction

- 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, and 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.
- The project goal is to predict if the Falcon 9 first stage will land successfully.

Section 1

Methodology

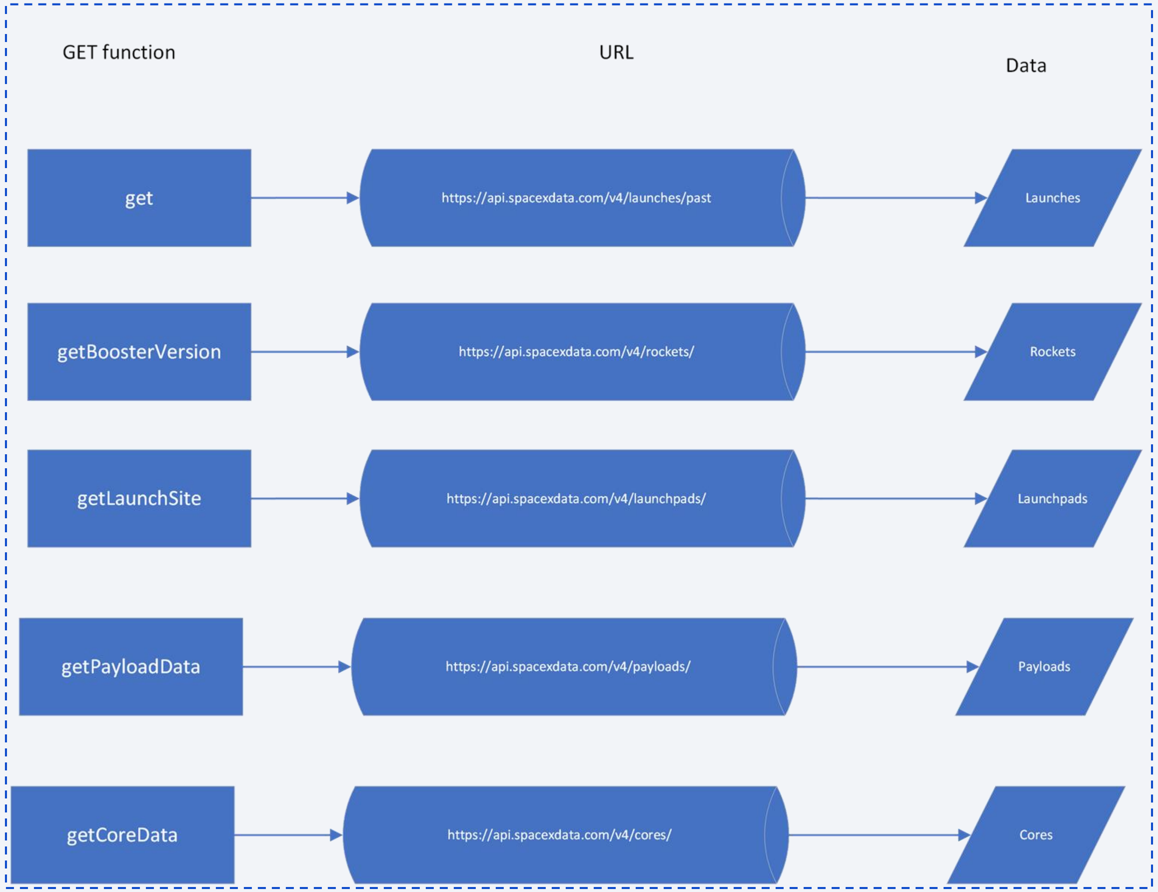
Methodology

Executive Summary

- Data collection methodology:
  - Data is gathered from SpaceX REST API and Web Scrapping
- Perform data wrangling
  - Wrangling Data using an API and Web Scraping; Sampling Data; Dealing with Nulls
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Training and testing dataset split, grid search, and confusion matrix

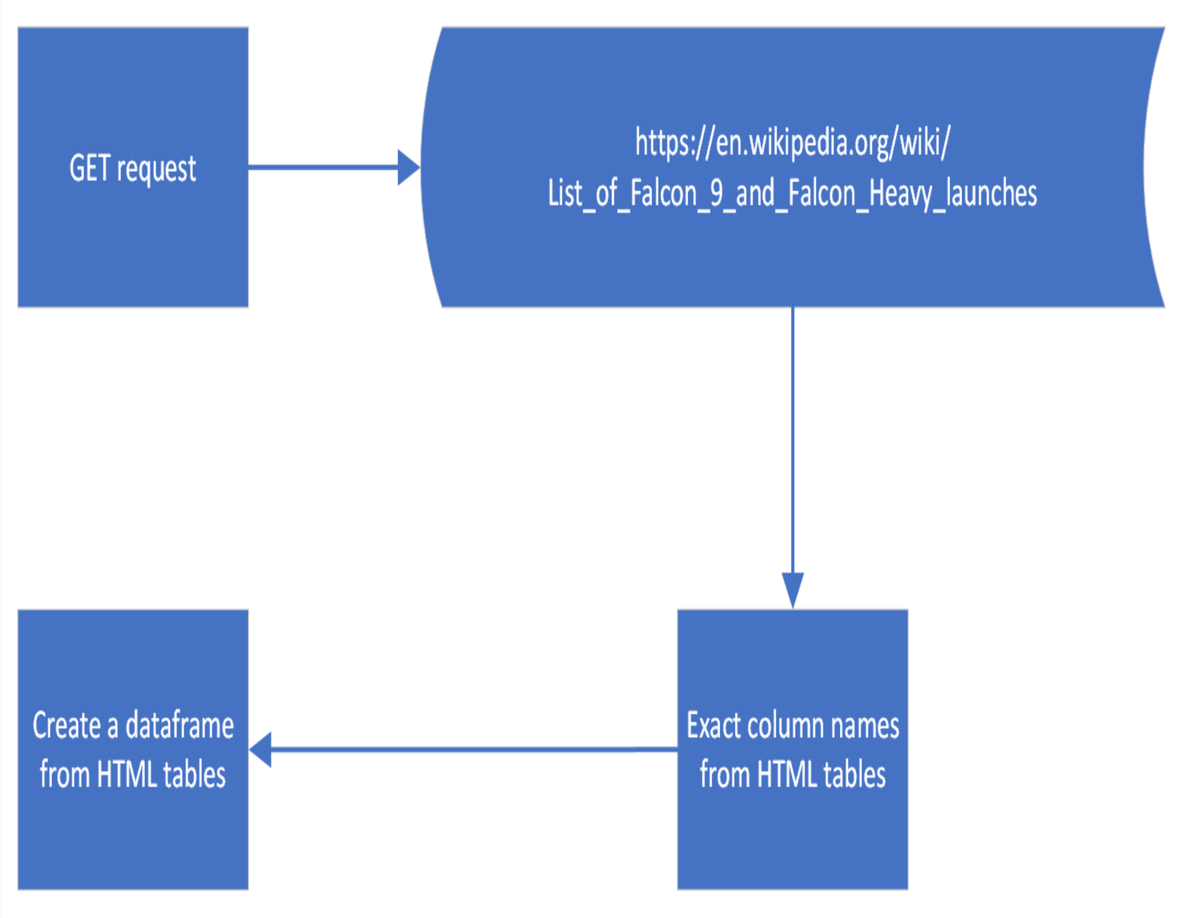
Data Collection – SpaceX API

- Use GET requests to obtain data for the launches, rockets, payloads, launchpads and other core information. (See the flowchart)
- GitHub URL of the completed SpaceX API calls notebook: [Link]()



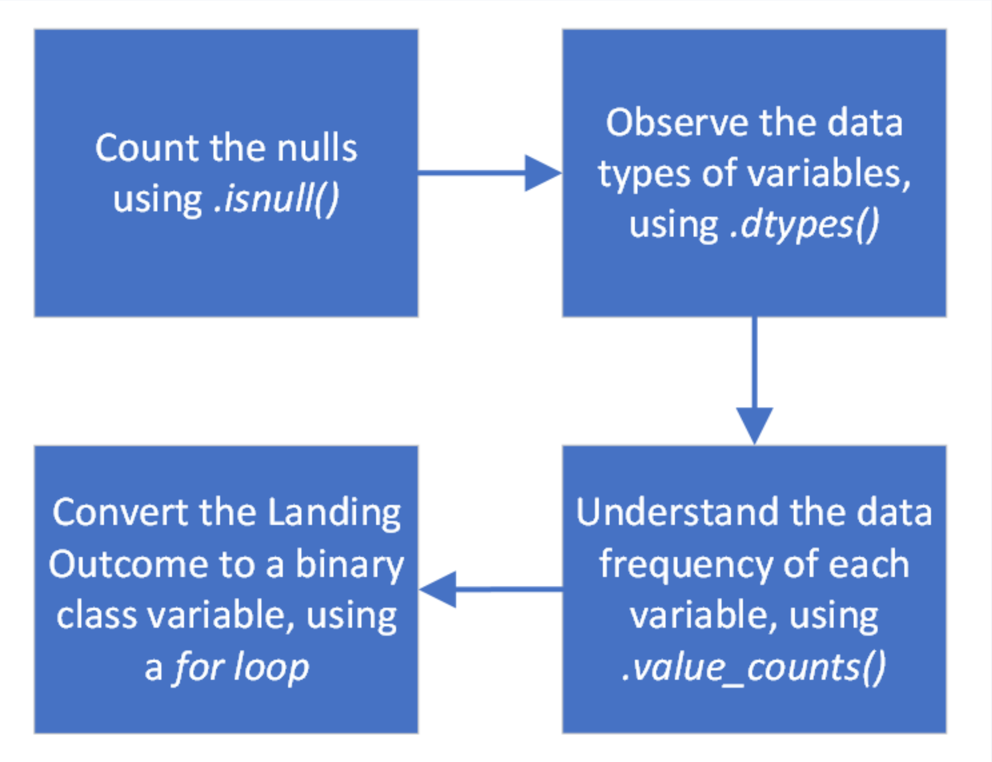
Data Collection - Scraping

- Use Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch record from Wike pages
- GitHub URL of the completed web scraping notebook: [Link]



Data Wrangling

- Observe the data type and the data frequency of each variable and obtain a binary outcome variable
- GitHub URL of the completed data wrangling related notebooks: [Link]



EDA with SQL

- SQL queries that are performed: [Link]
  - Display the names of unique launch sites
  - Sample some records from launch sites starting with string “KSC”
  - Calculate the total payload mass carried by NASA(CRS)
  - Calculate the average payload mass carried by booster version F9 V1.1
  - Find the first successful landing outcome in drone ship was achieved
  - Display the names of the boosters which have success in ground pad with certain payload
  - Display the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass
  - List the records which will display the month names
  - Rank the count of successful landing_outcomes between the certain dates

EDA with Data Visualization

- Charts were plotted to identify features that can be used in predicting successful landing rate: [Link]
  - Visualize the relationship between Flight Number and Launch Site
  - Visualize the relationship between Payload and Launch Site
  - Visualize the relationship between success rate of each orbit type
  - Visualize the relationship between FlightNumber and Orbit type
  - Visualize the relationship between Payload and Orbit type
  - Visualize the launch success yearly trend

Build an Interactive Map with Folium

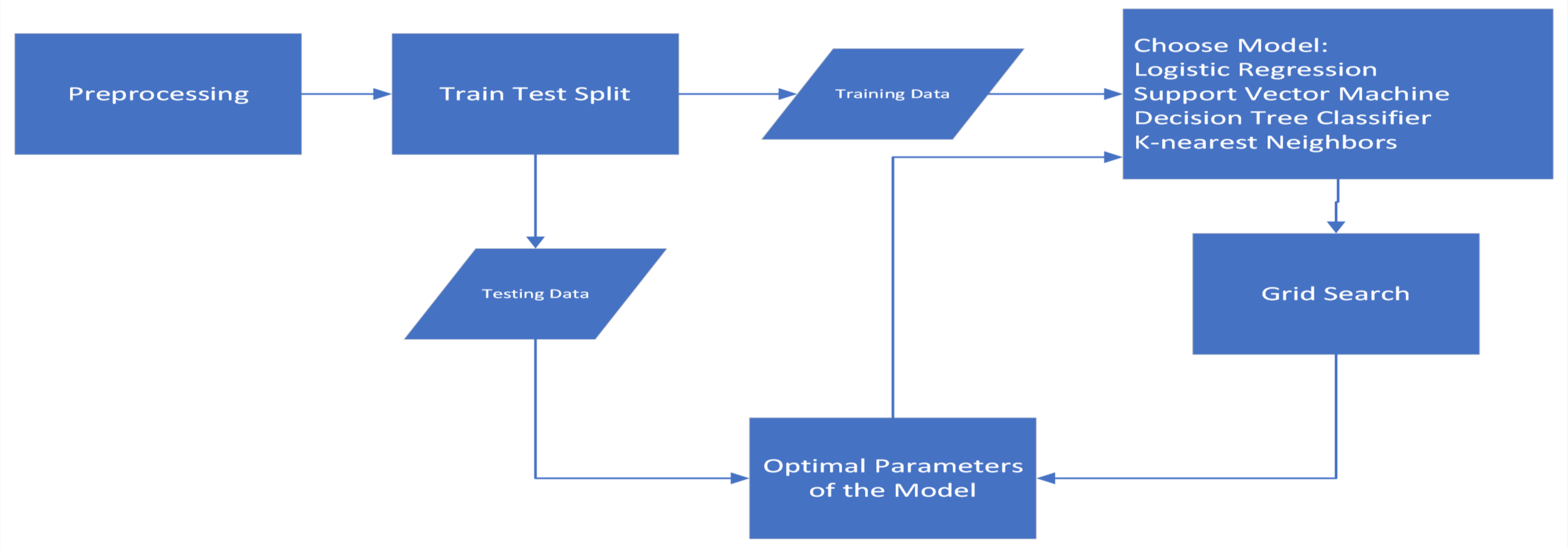
- Objects in the folium Map and theirs reasons: [Link]
  - folium.Circle to add a highlighted circle area with a text label on a specific coordinate
  - MarkerCluster object to simplify a map containing many markers having the same coordinate
  - MousePosition on the map to get coordinate for a mouse over a point on the map
  - PolyLine to show the distance between two selected points

Build a Dashboard with Plotly Dash

- A pie chart to show the success launches and success rate by site
- A scatter plot to visualize the relationship between pay load and outcome
- A dropdown to select sites and a slider to filter by pay loads
- Link

Predictive Analysis (Classification)

- Split the processed dataset into training and testing datasets. Use grid search to find out the best parameters of each model. Compare the best result of each model: [Link]



Results

- Based on the EDA and interactive analytics, Launch Site and Pay Load impacts the Launch Outcome. The KSC site has the most success launches. Some sites are better at launches with heavy Pay Load
- Logistic regression with the best parameter gives a score of 0.833. Its false negative rate is 0% and its false positive rate is 20%

Section 2

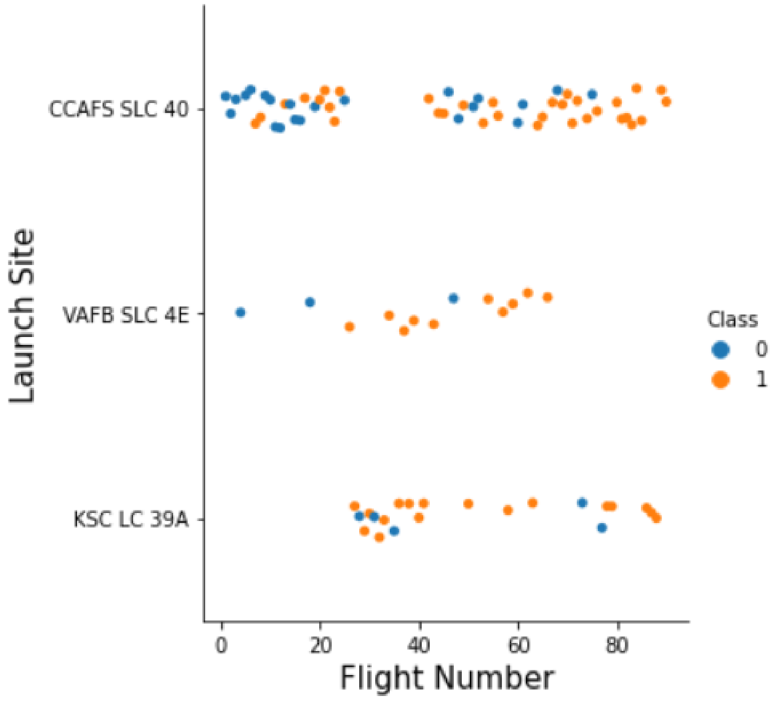
Insights drawn from EDA

Flight Number vs. Launch Site

- Within each launch site, there are more successful launches with higher flight numbers (>40)

In [5]:

```
sns.catplot(y="LaunchSite",x="FlightNumber",hue="Class", data=df, aspect = 1)
plt.ylabel("Launch Site",fontsize=15)
plt.xlabel("Flight Number",fontsize=15)
plt.show()
```

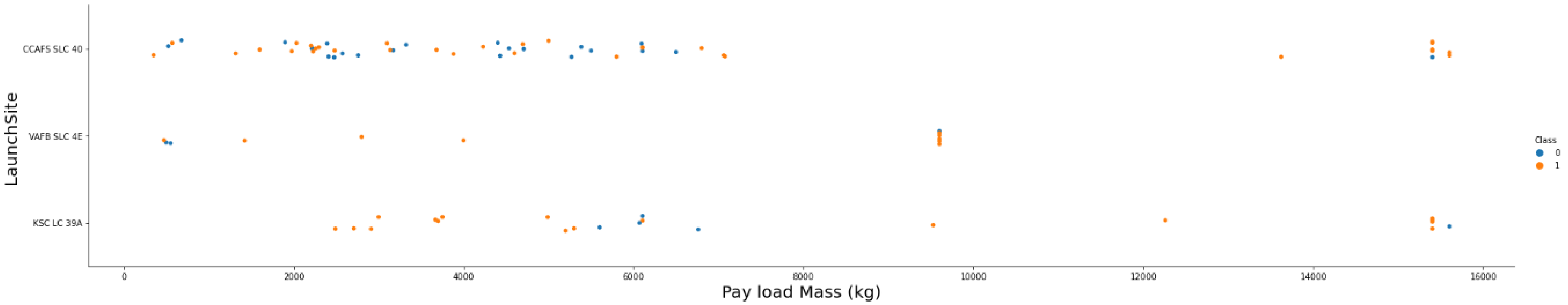


Payload vs. Launch Site

- Within each launch site, there are more successful launches with higher pay load mass (>10K)

In [6]:

```
# 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="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



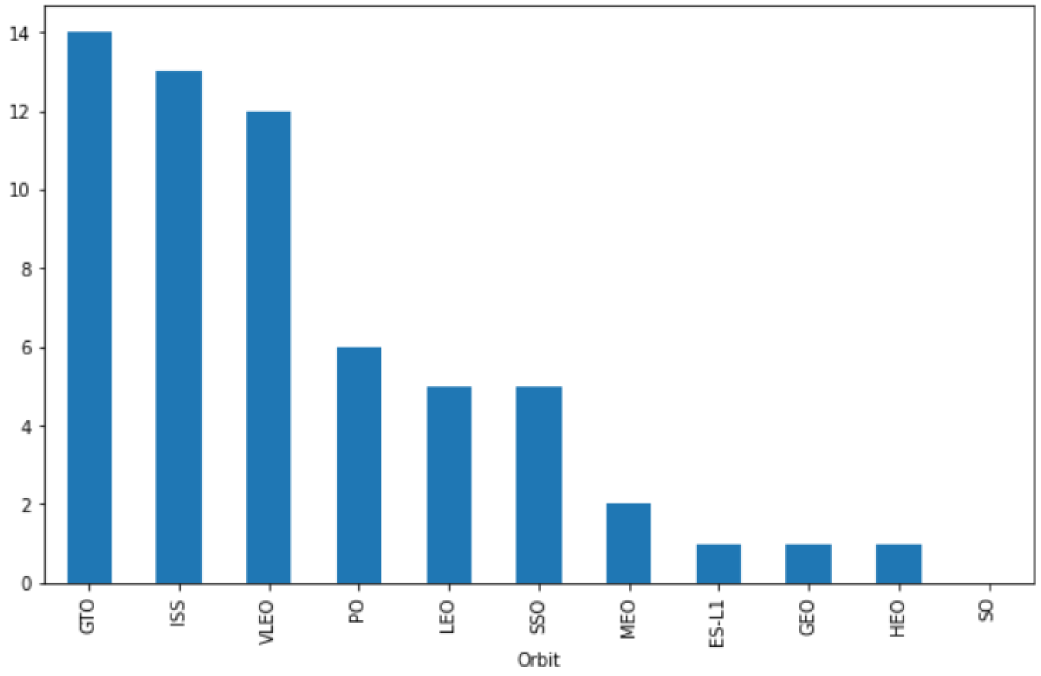
Success Rate vs. Orbit Type

- The successful landing rate is highest when using GTO orbit, and is lowest when using SO
- Link to explain each Orbit Type: [Link]

In [7]:

```python
# HINT use groupby method on Orbit column and get the mean of Class column
df_bar = df.groupby(['Orbit']).sum().sort_values(by=['Class'],ascending = False)['Class']
df_bar.plot(kind='bar', figsize=(10, 6))
```

Out[7]: <AxesSubplot:xlabel='Orbit'>

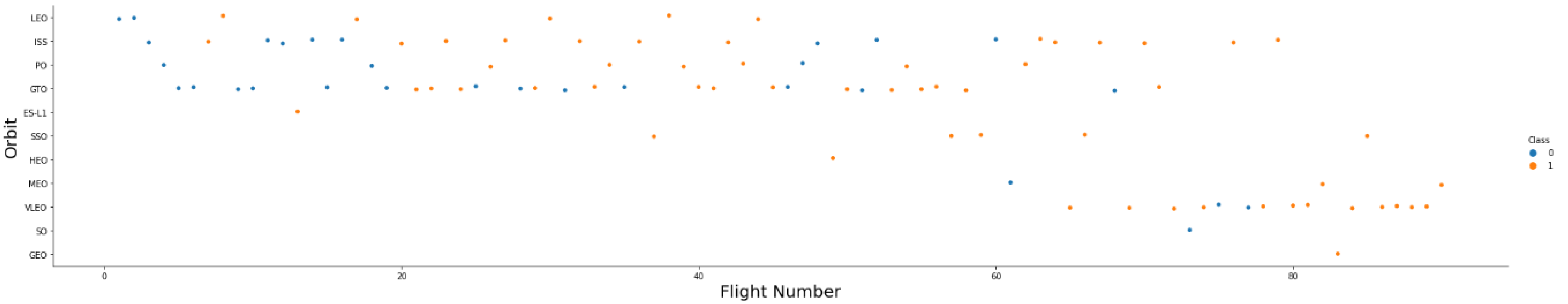


Flight Number vs. Orbit Type

- LEO, ISS, PO and GTO are used for the earlier flights

In [8]:

```
# 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("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

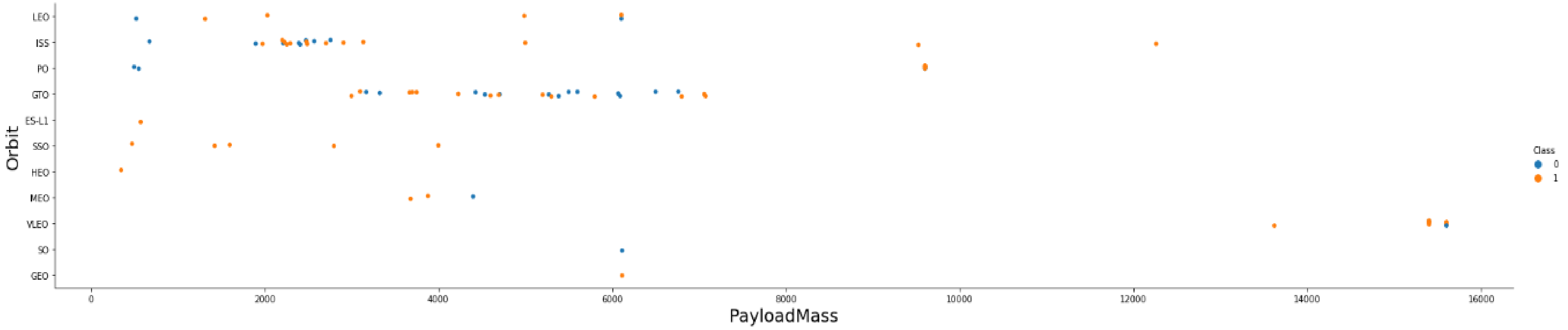


Payload vs. Orbit Type

- With light payloads the successful landing or positive landing rate are more for SSO, HEO and MEO.

In [9]:

```
# 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()
```

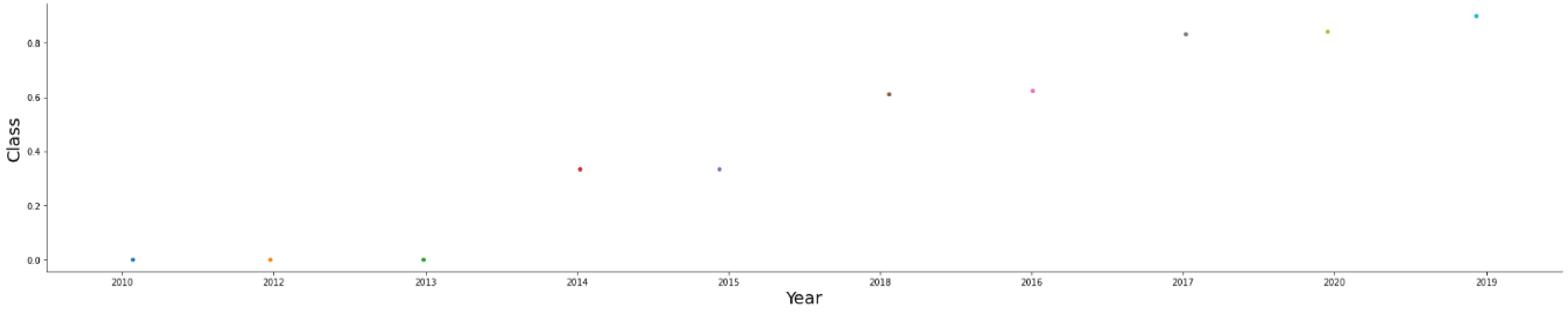


Launch Success Yearly Trend

- The success rate increases year by year!

In [12]:

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
Year = np.asarray(Extract_year(df["Date"]))
df["Year"]=Year
df_Rate_by_Year = df.groupby(['Year']).mean().sort_values(by=['Class'])
df_Rate_by_Year = df_Rate_by_Year.reset_index(level=0)
sns.catplot(y="Class", x="Year", data=df_Rate_by_Year, aspect = 5)
plt.xlabel("Year",fontsize=20)
plt.ylabel("Class",fontsize=20)
plt.show()
```



All Launch Site Names

- Find the names of the unique launch sites

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

In [15]:
```
%sql SELECT UNIQUE launch_site FROM SPACEXTBL
```

```
 * ibm_db_sa://jcn13848:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[15]:

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

Launch Site Names Begin with 'KSC'

- Find 5 records where launch sites' names start with `KSC`

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

In [16]:

```
%sql SELECT  * FROM SPACEXTBL WHERE launch_site like 'KSC%' FETCH FIRST 5 ROWS ONLY;
```

```
 * ibm_db_sa://jcn13848:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[16]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

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

```
In [17]: %sql SELECT  sum(payload_mass__kg_) FROM SPACEXTBL WHERE customer like 'NASA (CRS)'
```

```
 * ibm_db_sa://jcn13848:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[17]:

1
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

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

In [18]:
```
%sql SELECT  avg(payload_mass__kg_) FROM SPACEXTBL WHERE Booster_version like 'F9 v1.1'
```

```
 * ibm_db_sa://jcn13848:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[18]:

1
2928

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on drone ship.

List the date where the first successful landing outcome in drone ship was acheived.

Hint:Use min function

In [19]:
```
%sql SELECT  * FROM SPACEXTBL WHERE LANDING__OUTCOME like 'Success (drone ship)' FETCH FIRST 1 ROWS ONLY;
```

```
 * ibm_db_sa://jcn13848:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[19]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

In [20]:

```
%sql SELECT  BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING__OUTCOME like 'Success (ground pad)' AND PAYLOAD_MASS__KG_ >4000 AND PAYLOAD_MASS__KG_ < 600
```

```
 * ibm_db_sa://jcn13848:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[20]:

booster_version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

In [21]:

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

```
 * ibm_db_sa://jcn13848:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[21]:

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [22]:
```
%sql SELECT * FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT  MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://jcn13848:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[22]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2019-11-11	14:56:00	F9 B5 B1048.4	CCAFS SLC-40	Starlink 1 v1.0, SpaceX CRS-19	15600	LEO	SpaceX	Success	Success
2020-01-07	02:33:00	F9 B5 B1049.4	CCAFS SLC-40	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600	LEO	SpaceX	Success	Success
2020-01-29	14:07:00	F9 B5 B1051.3	CCAFS SLC-40	Starlink 3 v1.0, Starlink 4 v1.0	15600	LEO	SpaceX	Success	Success
2020-02-17	15:05:00	F9 B5 B1056.4	CCAFS SLC-40	Starlink 4 v1.0, SpaceX CRS-20	15600	LEO	SpaceX	Success	Failure
2020-03-18	12:16:00	F9 B5 B1048.5	KSC LC-39A	Starlink 5 v1.0, Starlink 6 v1.0	15600	LEO	SpaceX	Success	Failure
2020-04-22	19:30:00	F9 B5 B1051.4	KSC LC-39A	Starlink 6 v1.0, Crew Dragon Demo-2	15600	LEO	SpaceX	Success	Success
2020-06-04	01:25:00	F9 B5 B1049.5	CCAFS SLC-40	Starlink 7 v1.0, Starlink 8 v1.0	15600	LEO	SpaceX, Planet Labs	Success	Success

2017 Launch Records

- List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

```
In [23]: %sql SELECT *, MONTH(DATE) AS MONTH FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)' AND YEAR(DATE) = 2017

 * ibm_db_sa://jcn13848:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[23]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome	MONTH
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)	2
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)	5
2017-06-03	21:07:00	F9 FT B1035.1	KSC LC-39A	SpaceX CRS-11	2708	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)	6
2017-08-14	16:31:00	F9 B4 B1039.1	KSC LC-39A	SpaceX CRS-12	3310	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)	8
2017-09-07	14:00:00	F9 B4 B1040.1	KSC LC-39A	Boeing X-37B OTV-5	4990	LEO	U.S. Air Force	Success	Success (ground pad)	9
2017-12-15	15:36:00	F9 FT B1035.2	CCAFS SLC-40	SpaceX CRS-13	2205	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)	12

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

- Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order

Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

```
In [24]: %sql SELECT  LANDING__OUTCOME, COUNT(*) AS COUNT FROM SPACEXTBL WHERE DATE > '2010-06-04' GROUP BY LANDING__OUTCOME ORDER BY COUNT(*) DESC
```

```
 * ibm_db_sa://jcn13848:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[24]:

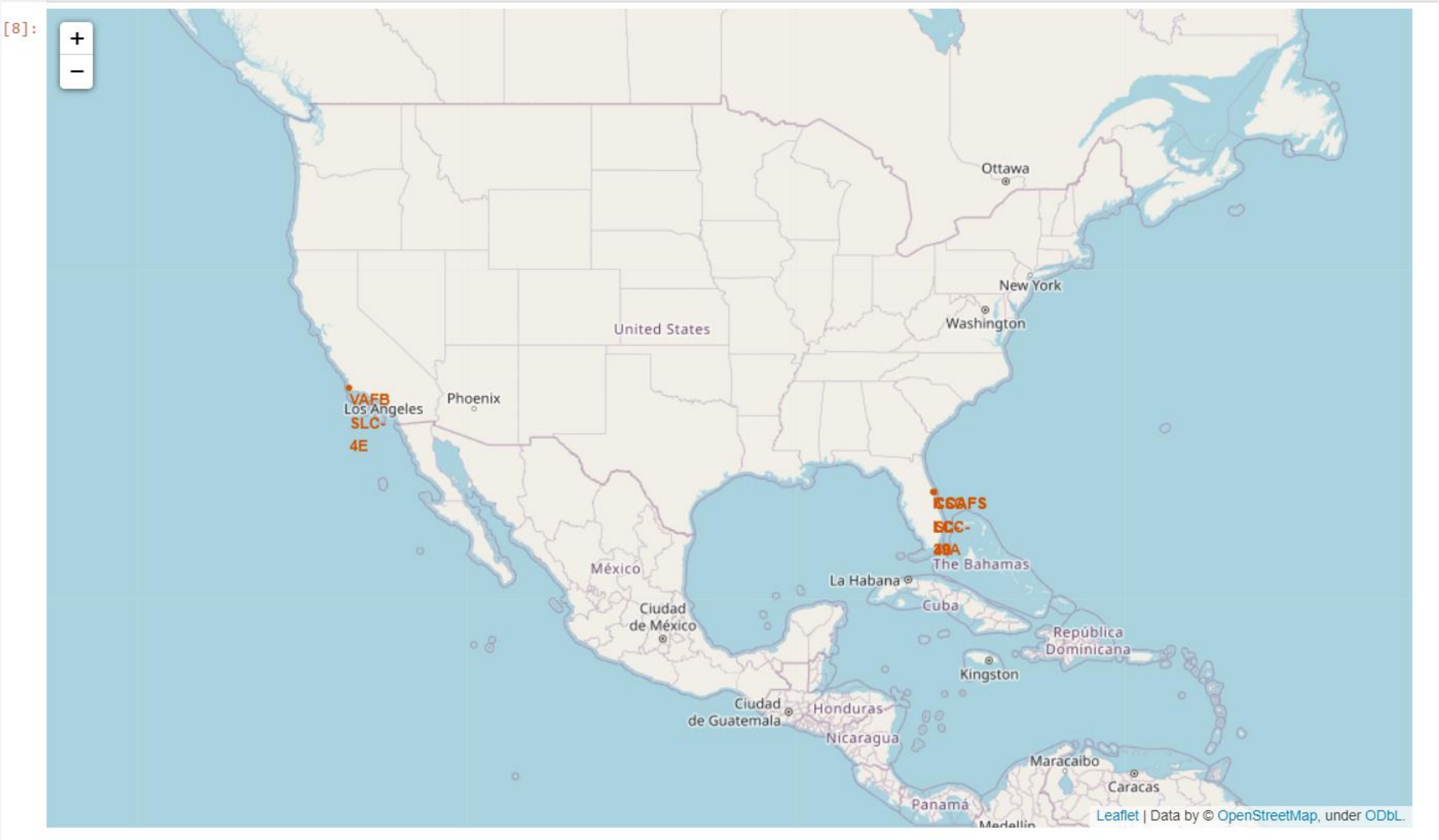
landing__outcome	COUNT
Success	38
No attempt	22
Success (drone ship)	14
Success (ground pad)	9
Controlled (ocean)	5
Failure (drone ship)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	1
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

All Launch Sites on a Map

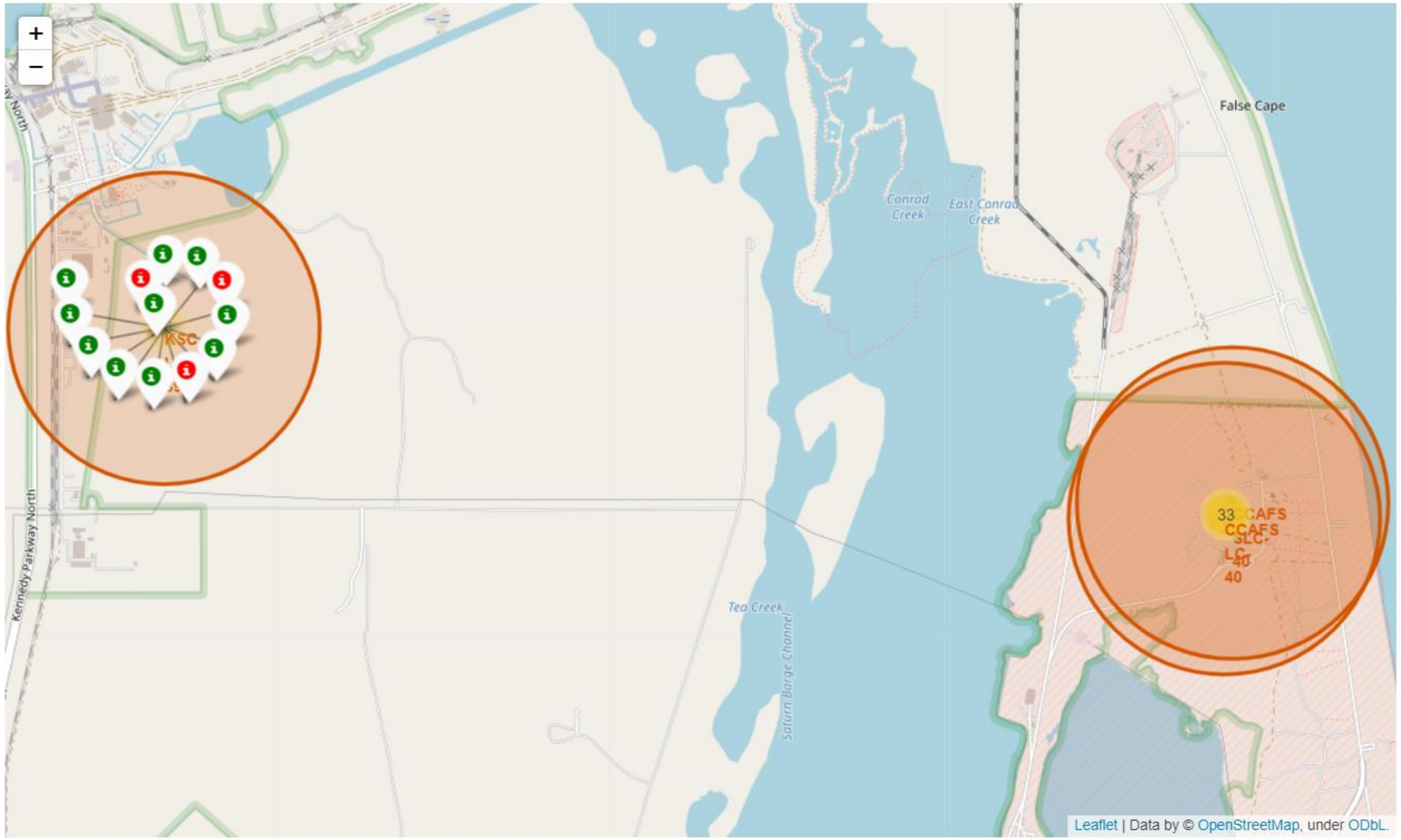
- 1 Launch site is on the west coast while the other 3 on the east coast.
- The 3 sites on the east coast are very close to each other



Success/Failed Launches for each Site

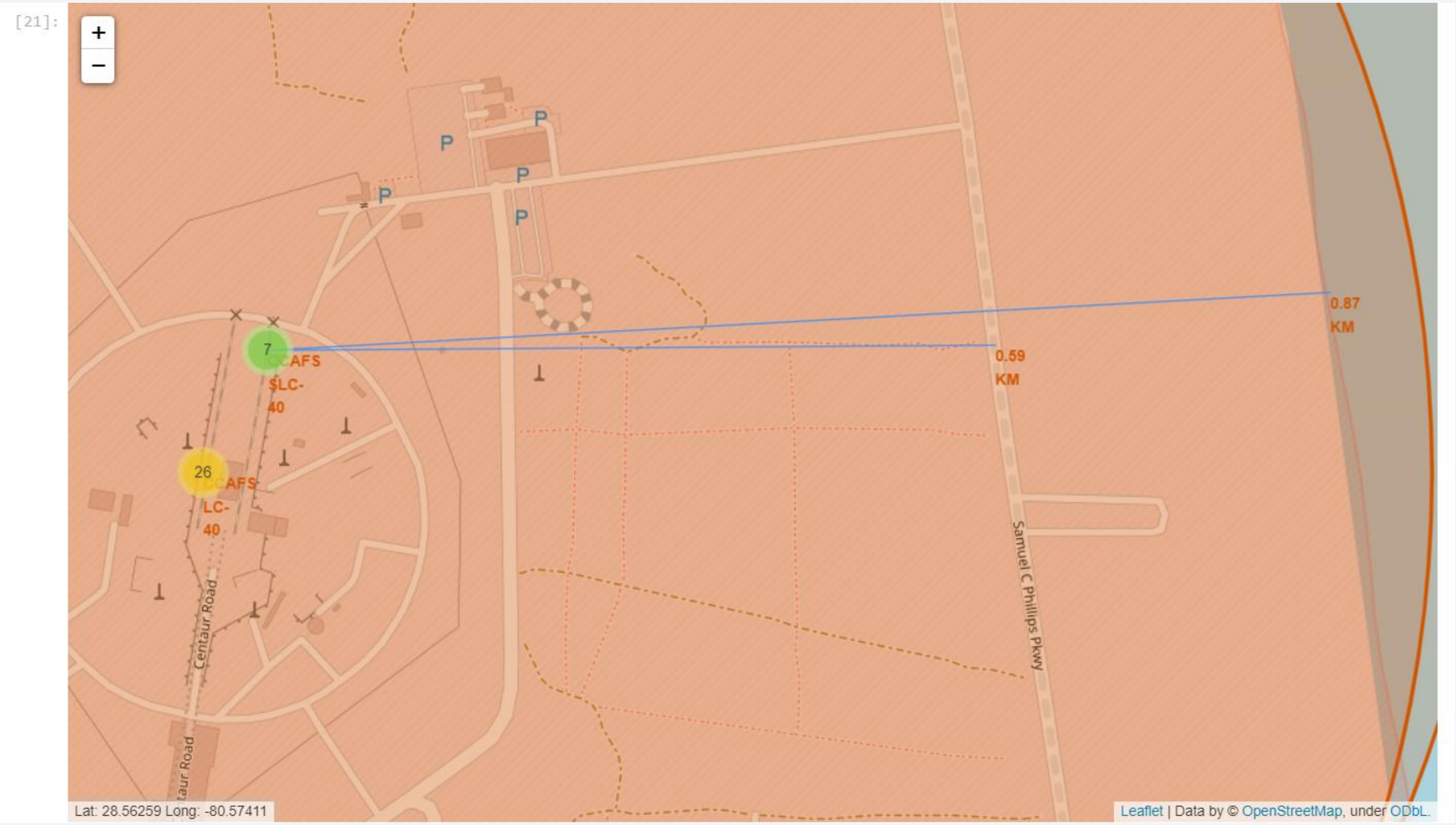
- More launches happens on the sites closer to the coast

[13]:



Distances between a Launch Site to its Proximities

- Launch Sites are close to coast lines

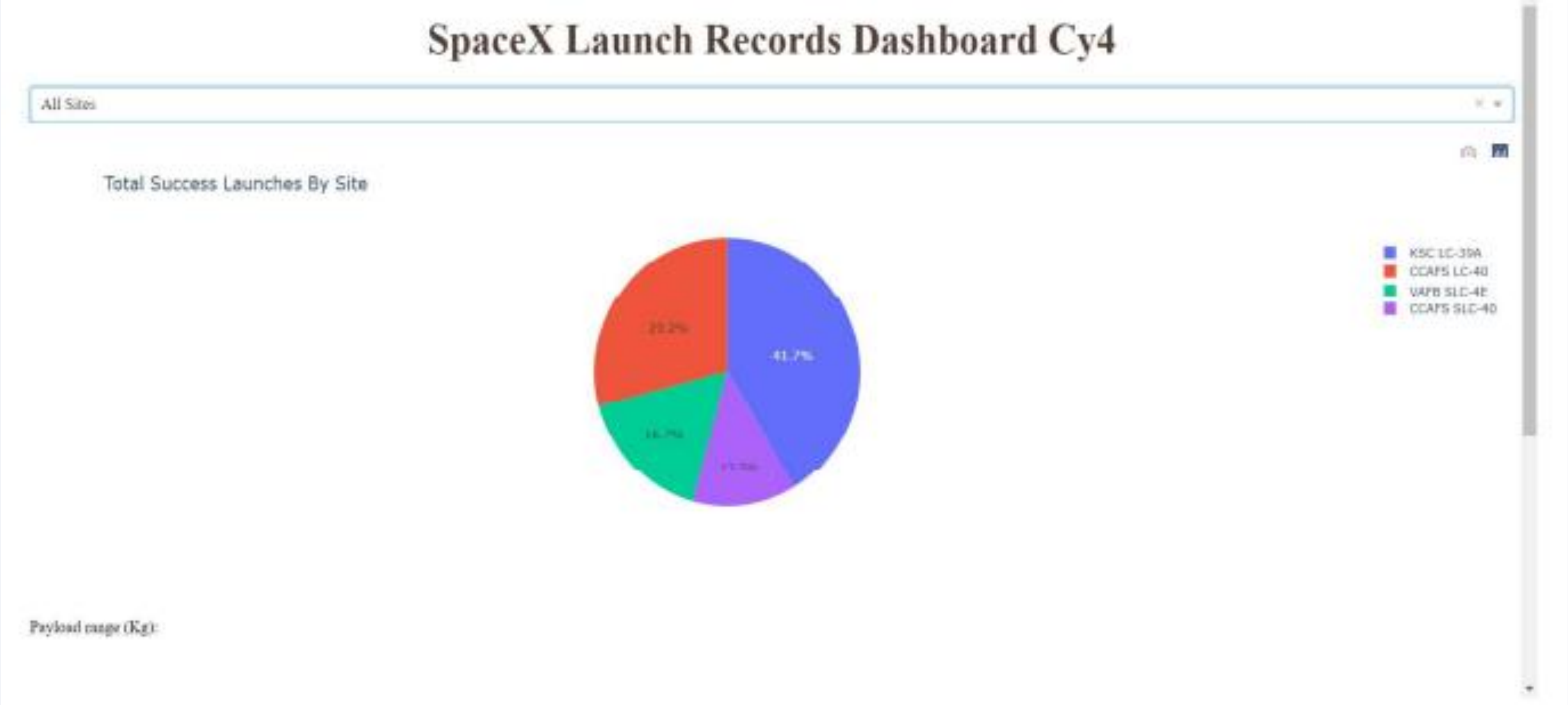


Section 4

Build a Dashboard with Plotly Dash

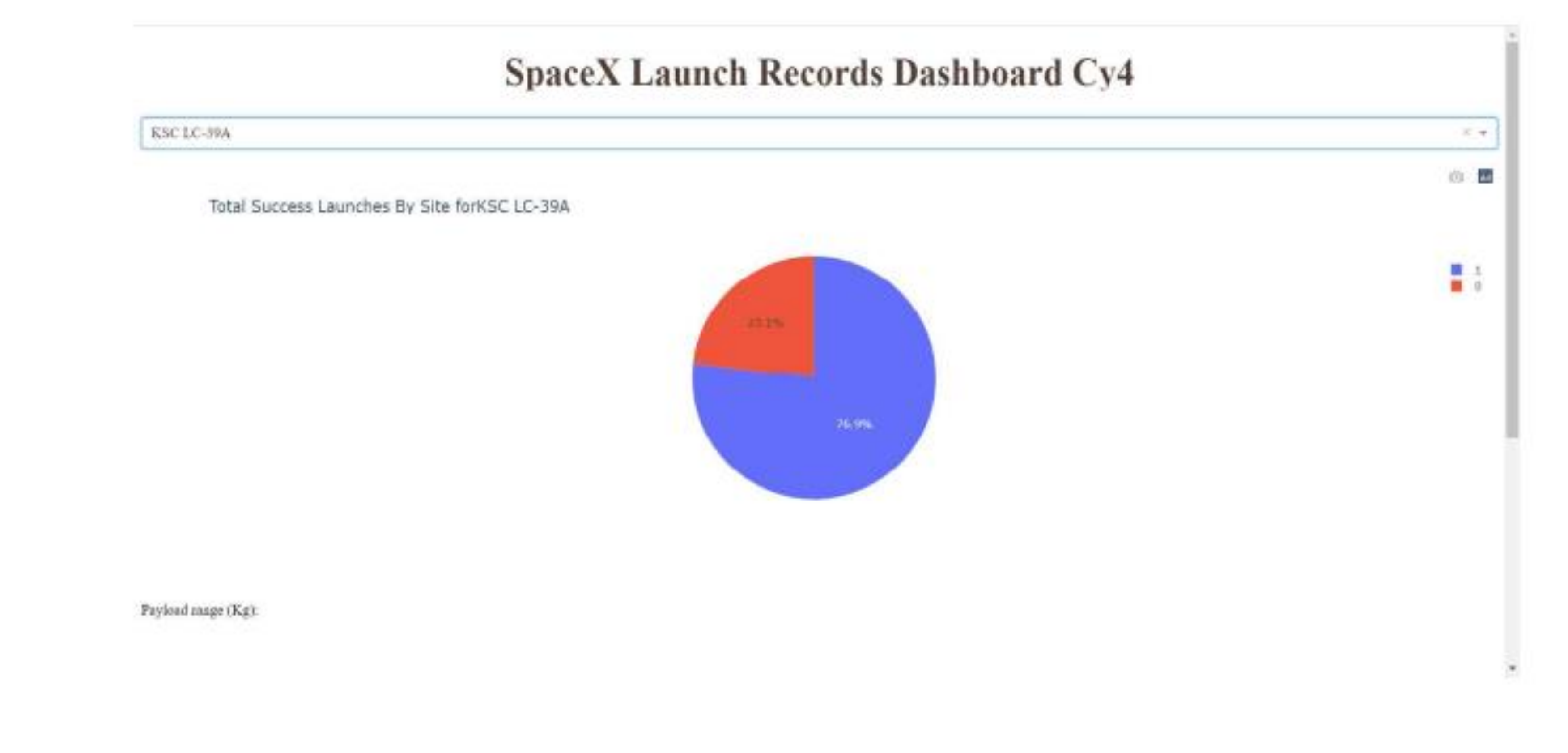
Success Launches by site

- KSC LC site has the most success launches



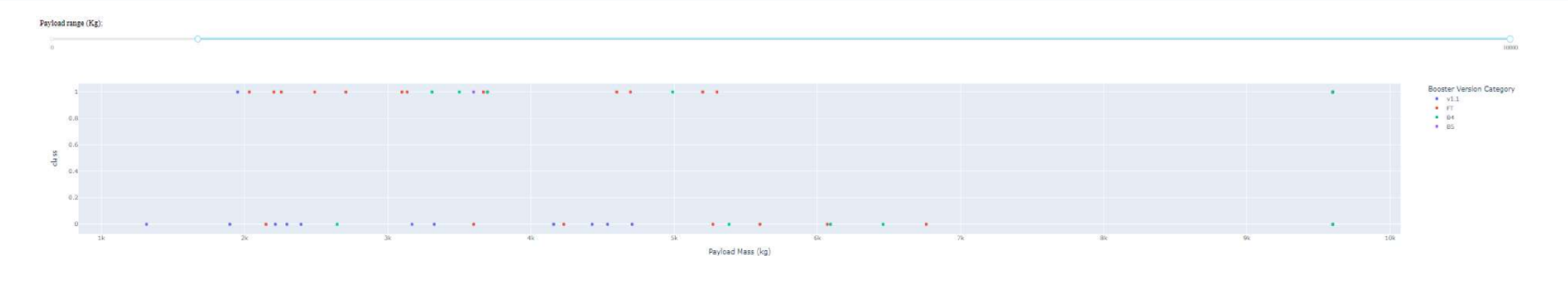
Success Launch Rate at KSC LC Site

- KSC LC site has over 76% success launch rate!



Success Launches vs Pay Loads

- Success Launch Rate is higher between 2K to 6K pay loads than between 6K to 10K.

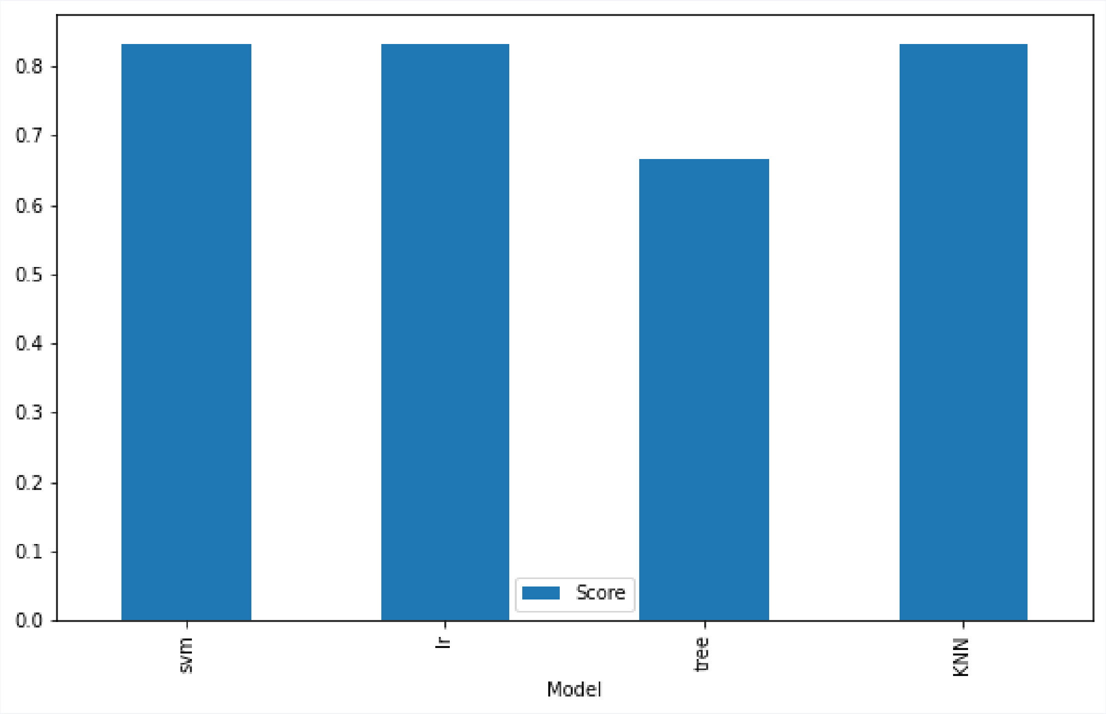


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- SVM, Decision Tree, and KNN have highest scores 0.833



Find the method performs best:

```
[27]: print("svm score",svm2.score(X_test,Y_test))
      print("lr score",lr2.score(X_test,Y_test))
      print("tree score",tree2.score(X_test,Y_test))
      print("KNN score",KNN2.score(X_test,Y_test))
```

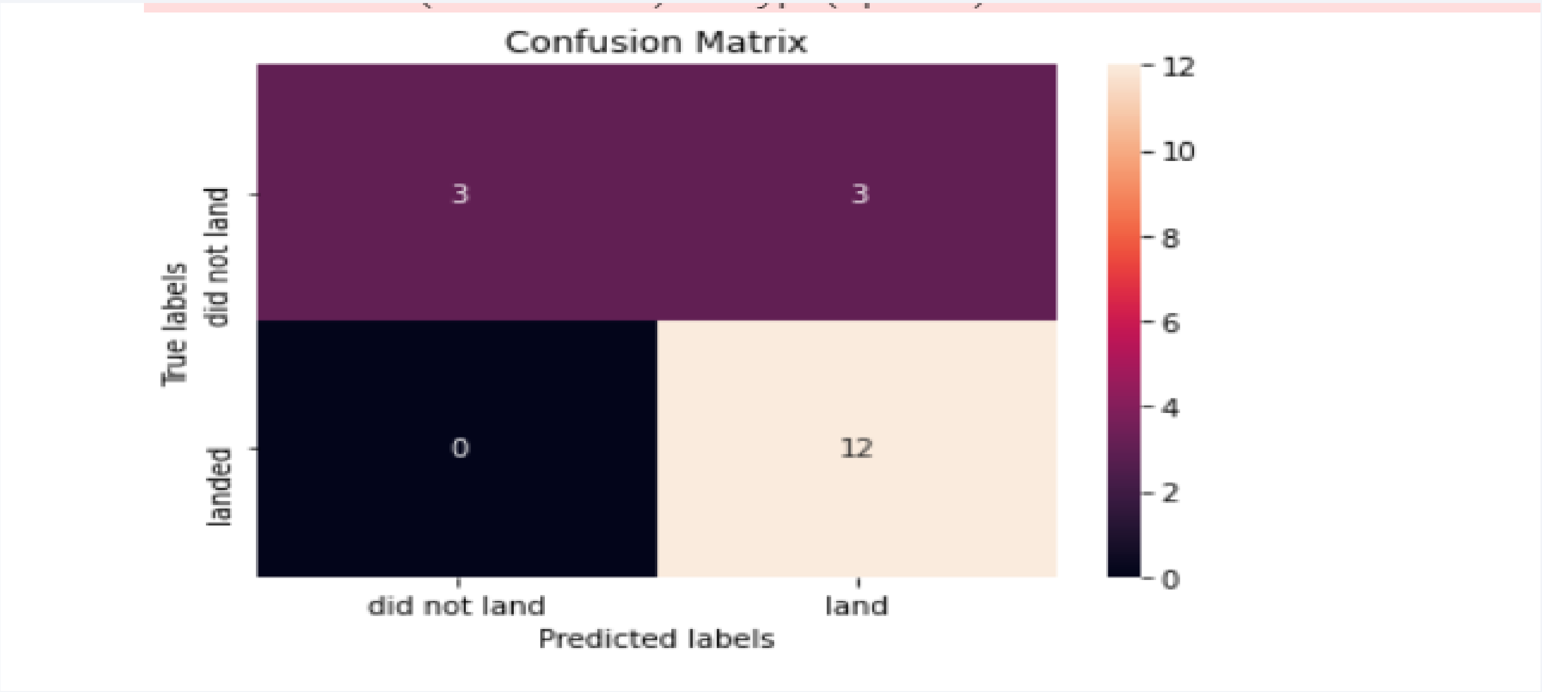
```
svm score 0.8333333333333334
lr score 0.8333333333333334
tree score 0.6666666666666666
KNN score 0.8333333333333334
```

Confusion Matrix

- Logistic regression can differentiate between the classes. The major challenge is false positives

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

tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8472222222222222
```



Conclusions

- Logistic Regression, SVM and KNN are all good at predicting the failed launches (0% False Negative). They can be used to identify risky flights and adjust the flight features. Hence, it can reduce the failed rate and increase the success rate
- Among all flights that are predicted to launch successfully, 80% of them launched successfully. In order words, if SpaceX only flies flights with a positive prediction, the success launch rate can be as high as 80%!

Thank you!