IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

AVIN P S
27/1/26

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Insights from EDA
- Interactive Visual Analytics
- Conclusion

Executive Summary

- Collected historical SpaceX Falcon 9 launch data from SpaceX API and Wikipedia.
- Performed data wrangling, EDA, SQL analysis, and interactive visualization.
- Built and tuned multiple classification models.
- Identified key factors influencing landing success.
- Selected the best-performing model based on test accuracy

Introduction

- SpaceX aims to reduce launch costs through reusable rocket technology.
- Falcon 9 first stage landing success is critical for reusability.
- This project predicts whether the first stage will successfully land

Key Questions

- What factors influence landing success?
- Can landing success be predicted using machine learning?

Section 1

Methodology

Methodology

Executive Summary

- Data wrangling and feature engineering
- Exploratory Data Analysis (EDA) using visualization and SQL
- Interactive analytics using Folium and Plotly Dash
- Predictive analysis using classification modelsData collection using SpaceX REST API and web scraping
- Data wrangling and feature engineering
- Exploratory Data Analysis (EDA) using visualization and SQL
- Interactive analytics using Folium and Plotly Dash
- Predictive analysis using classification models

Data Collection

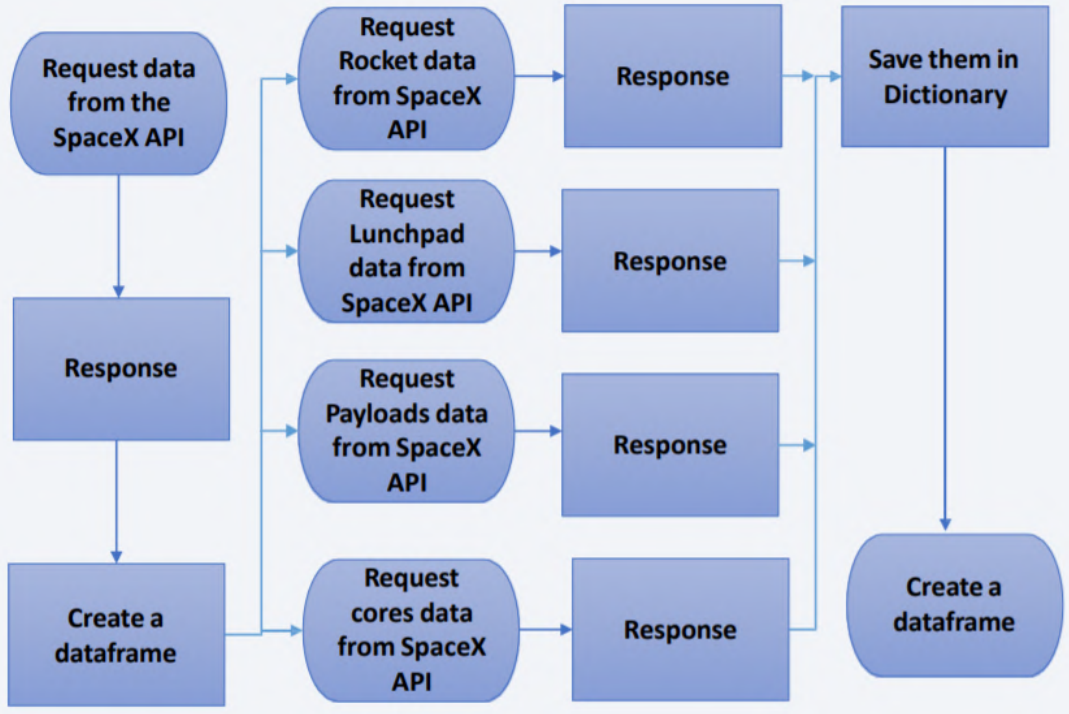
The data set used in the project is the SpaceX Launch data.
The data set contains information about different flights including date, launch site, booster version and more.
The data is collected by two main approaches:

1. The SpaceX API
2. Web Scrapping

Data Collection – SpaceX API

- Used SpaceX REST API to retrieve launch, rocket, and landing data.
- API calls automated using Python requests

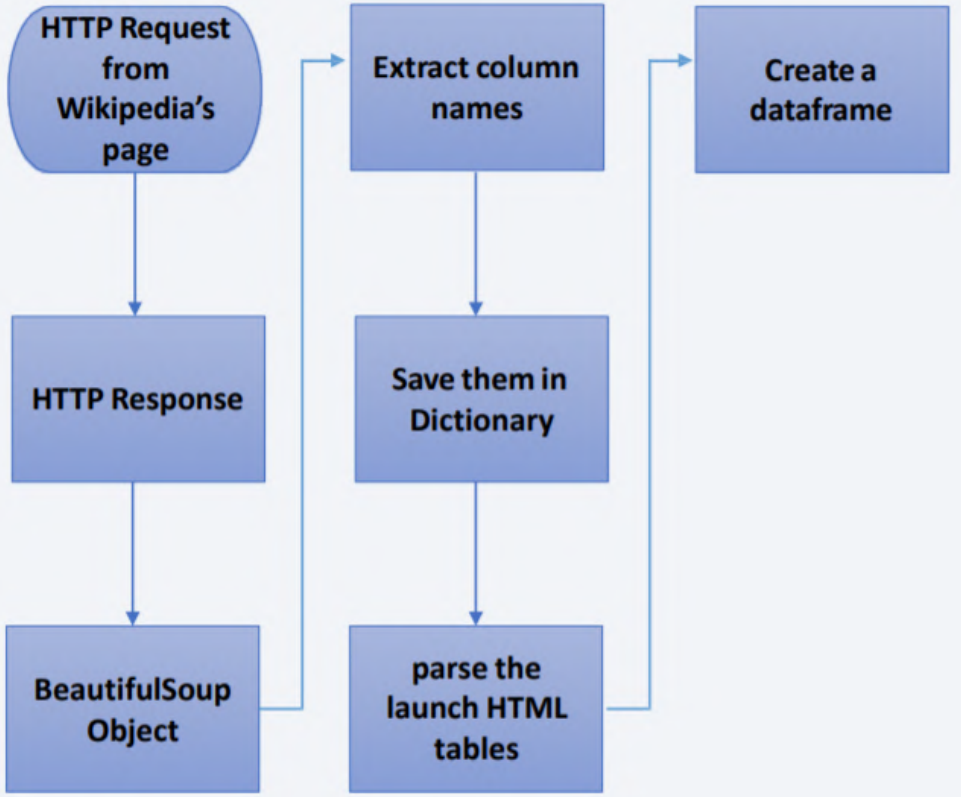
- GitHub URL of the completed SpaceX API calls notebook - https://github.com/avinps/IBMDataScience-Jupyter-Notebook/blob/main/spacex-data-collection-api.ipynb



Data Collection - Scraping

- We will be performing web scraping to collect Falcon 9 historical launch records from a Wikipedia page.
- First we perform an HTTP GET(using requests.get command)method to request the Falcon9 Launch HTMLpage, as an HTTP response.
- Then we create a BeautifulSoup object from the HTML response, We extract the column names from the object and use it as dictionary keys.
- We parse the HTML tables and fill the dictionary keys with launch records from table rows, and finally we transform it into a dataframe.

GitHub URL of the completed web scraping notebook - https://github.com/avinps/IBMDataScience-Jupyter-Notebook/blob/main/webscraping.ipynb



Data Wrangling

Exploratory data analysis is an important step while preprocessing data, it is useful to find some patterns in the data and determine what would be the label for training supervised models.
This process was done in the following order:

1. First thing to do is to identify the data types of the columns.
2. Determine the number of values for each attribute.
3. Calculate the percentage of the missing values.
4. To determine the label, weapply zero/one hot encoding to the “Outcome” column to classify landing to either 1(Success) of 0 (Failure)

- GitHub URL of completed data wrangling related notebook - https://github.com/avinps/IBMDataScience-Jupyter-Notebook/blob/main/spacex-Data%20wrangling.ipynb

EDA with Data Visualization

In order to understand the relations between different features, we visualize the data by plotting scatter plots, bar charts and line charts, it helps finding hidden patterns in data and gain insights about the dataset.

- Pay load mass against the Flight number.
- Lunch site against the Flight number.
- Lunch site against the Pay load mass.
- Orbit type against Class success rate.
- Flight number against Orbit type.
- Orbit type against the Pay load mass.
- launch success yearly trend.
- GitHub URL of completed EDA with data visualization notebook - https://github.com/avinps/IBMDataScience-Jupyter-Notebook/blob/main/edadataviz.ipynb

EDA with SQL

In order to better understand the datasets, we ran the following SQL queries:

- Display the names of the unique launch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'.
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display average payload mass carried by booster version F9 v1.1 .
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- List the total number of successful and failure mission outcomes.
- List the names of the booster versions which have carried the maximum payload mass. Use a subquery.
- List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.
- GitHub URL of completed EDA with SQL notebook - https://github.com/avinps/IBMDataScience-Jupyter-Notebook/blob/main/eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Here, we complete the interactive visual analytics using Folium.

- First we create Folium map object, with an initial center location around Nasa Johnson space center, Houston-Texas.
- We add a circle on the map for each launch site from the dataset by creating a folium circle and folium marker, now the launch sites are marked on the map which means we can see which one is proximate to the equator line or close to a coastline.
- In order to mark the success/failure launches, we create a marker on the map for each launch record from the dataset, a green marker indicates a successful lunching and a red one indicates failure,
- we need to explore and analyze the proximities of launch sites, we calculate the distance between the launch site and its proximities and then we draw a polyline between them.
- GitHub URL of completed interactive map with Folium map - https://github.com/avinps/IBMDataScience-Jupyter-Notebook/blob/main/analytics_folium_map.ipynb

Build a Dashboard with Plotly Dash

- Dropdown to select launch site
- Pie charts for success ratio
- Scatter plots for payload vs outcome.
- Enabled interactive data exploration
- GitHub URL of completed Plotly Dash lab - https://github.com/avinps/IBMDataScience-Jupyter-Notebook/blob/main/SpaceX_plotly.py

Predictive Analysis (Classification)

Now that we finished the exploratory analysis, the next step is to determine the training labels and build a predictor using machine learning algorithms. After using the ‘Class’ column as the label, first thing to do is normalizing the data. We split the normalized data into test/train sets, The training data is divided into validation data, a second set used for training data.
For the model development phase, we use the following algorithms:

1. Logistic regression
2. Support vector machine
3. Decision trees
4. K nearest neighbor

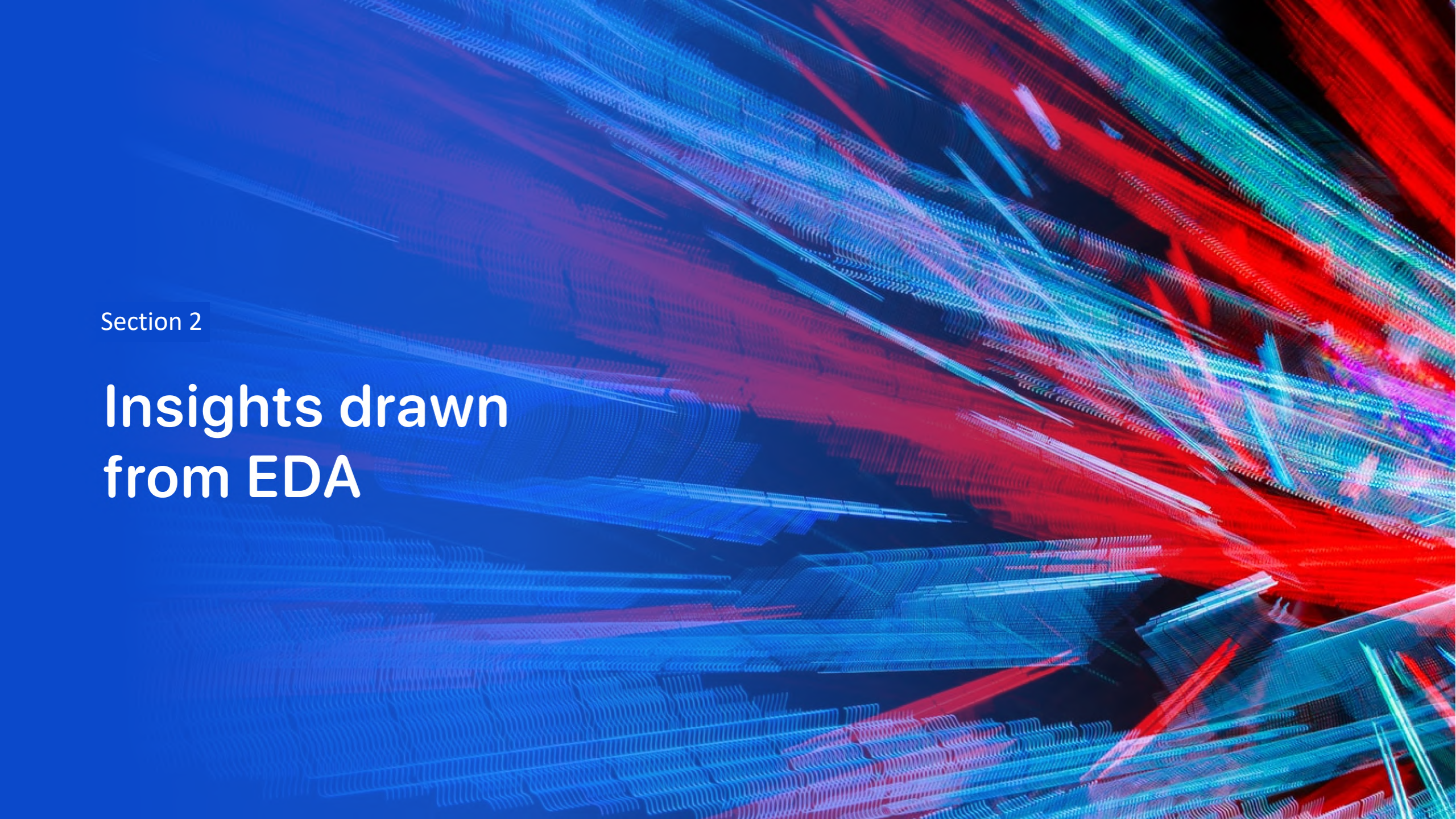
We build a grid search object for each of the algorithms and f i t it to find the best parameters of the model(hyper parameters tuning), then we choose the most accurate model.
GitHub URL of completed predictive analysis lab - https://github.com/avinps/IBMDataScience-Jupyter-Notebook/blob/main/SpaceX_Machine%20Learning%20Prediction.ipynb

Results

- Success rate increased noticeably from 2013 and on.
- Launch site and the orbit type are the features with the largest effect on the outcome.
- KNN and SVM models have a validation set accuracy of 83% and an out of sample accuracy of 77%.

Section 2

Insights drawn from EDA

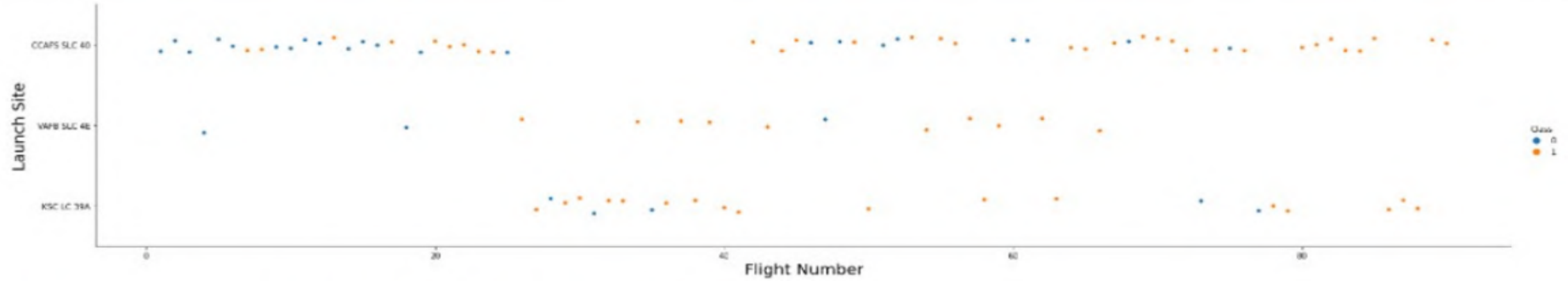


Flight Number vs. Launch Site

TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

In [4]:
```
# 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("Launch Site",fontsize=20)
plt.show()
```



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

In [5]:
```
#CCAFS SLC-40 flights have higher success rate as the flight number increases
```

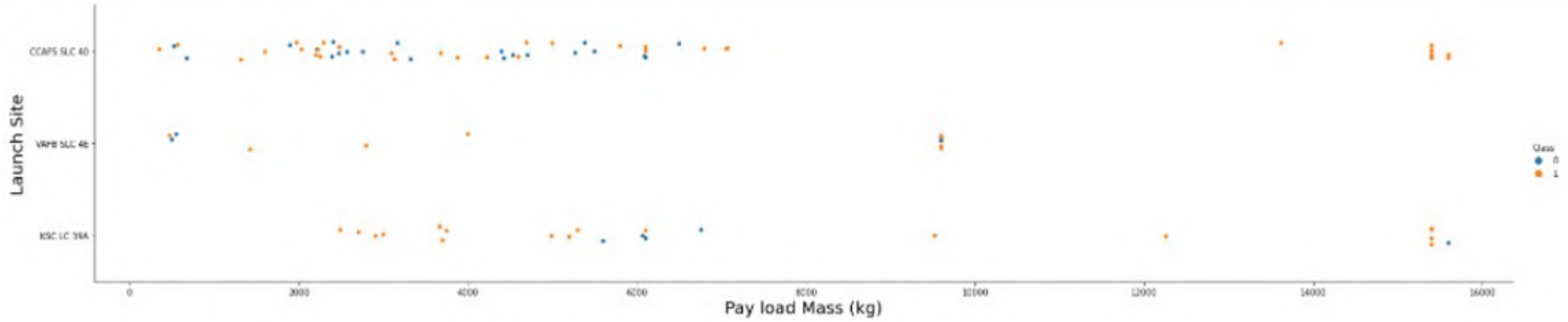
- Flight with number range 0 to 20 and range 40 to 90 are more on site CCAPS SLC 40.
- Flight with number range 21 to 39 is more on site KSC LC 39A.

Payload vs. Launch Site

TASK 2: Visualize the relationship between Payload and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

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("Launch Site",fontsize=20)
plt.show()
```



Now try to explain any patterns you found in the Payload Vs. Launch Site scatter point chart.

In [7]:
```
#small pay load mass values have bad influence on CCAFS SLC-40 flights
```

- More data spread on payloas mass range 0 to 8000 kg.
- When the payload is in the range of 15000, it looks more likely to land successfully.

Success Rate vs. Orbit Type

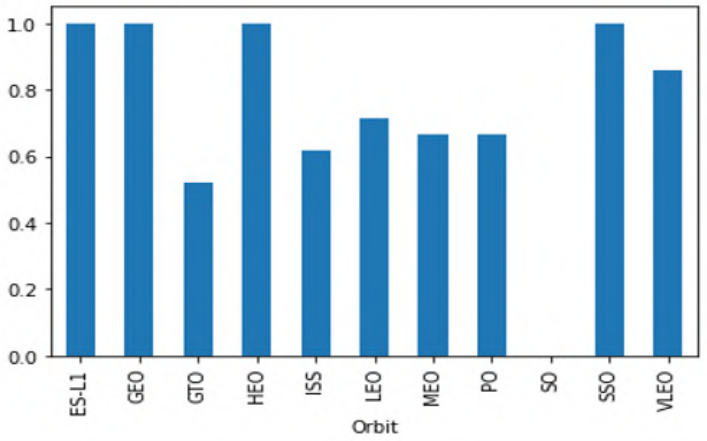
TASK 3: Visualize the relationship between success rate of each orbit type

Next, we want to visually check if there are any relationship between success rate and orbit type.

Let's create a `bar chart` for the sucess rate of each orbit

In [8]:
```
# HINT use groupby method on Orbit column and get the mean of Class column
mean = df.groupby(['Orbit'])['Class'].mean()
mean.plot(kind = 'bar')
```

Out[8]: <AxesSubplot:xlabel='Orbit'>



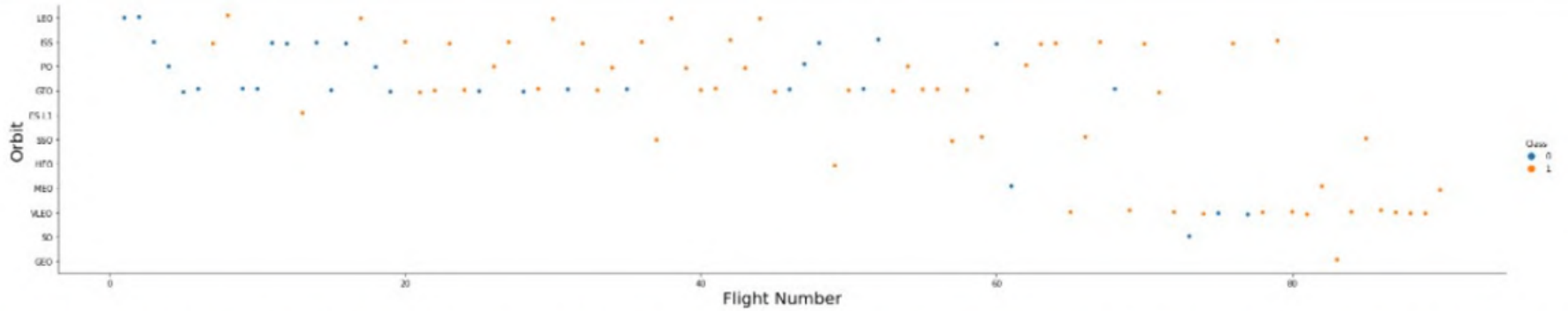
- Orbit type ES L1, GEO, HEO, and SSO have the highest success rate, which is 1, that means it always succeeds.
- Orbit type GTO have the lowest success rate, which is 0.5.

Flight Number vs. Orbit Type

TASK 4: Visualize the relationship between FlightNumber and Orbit type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

In [10]:
```
# 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()
```



You should 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.

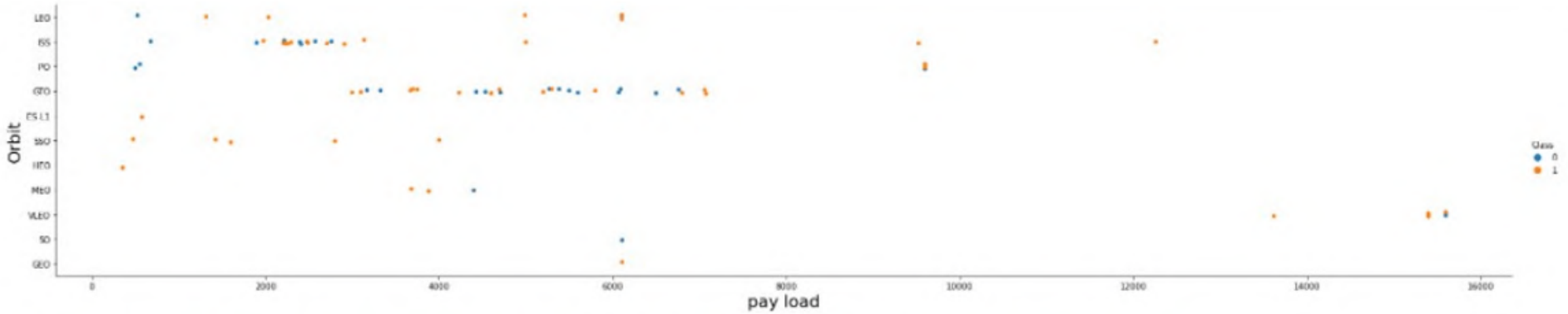
Orbit type LEO, ISS, PO, and GTO have more data spread on flight number range 0 to 60.

Payload vs. Orbit Type

TASK 5: Visualize the relationship between Payload and Orbit type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

In [21]:
```
# 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.ylabel("Orbit",fontsize=20)
plt.xlabel("pay load",fontsize=20)
plt.show()
```



You should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

- Orbit type VLEO that has high success rate also has heavy payload.
- There is a possibility that the heavier the payload, the higher the probability of success.

Launch Success Yearly Trend

You can plot a line chart with x axis to be Year and y axis to be average success rate, to get the average launch success trend.

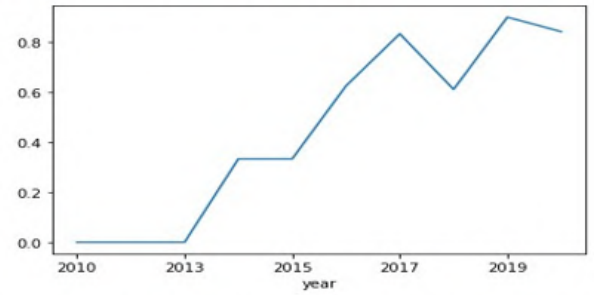
The function will help you get the year from the date:

In [12]:
```
# A function to Extract years from the date
year=[]
def Extract_year(date):
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
```

In [13]:
```
Extract_year(df)
```

In [14]:
```
Class = df['Class'].tolist()
new_df = pd.DataFrame(list(zip(year,Class)),columns = ['year','class'])
mean = new_df.groupby(['year'])['class'].mean()
mean
```

In [15]:
```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
line_plt = mean.plot(kind='line')
```



you can observe that the sucess rate since 2013 kept increasing till 2020

- 2019 is the year that has the highest success rate.
- 2010, 2012, and 2014 are the year that have lowest success rate.

All Launch Site Names

Task 1

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

```
In [5]: %sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACX;
 * ibm_db_sa://sdw99696:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[5]:

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

- There is 4 unique launch site.
- That’s mean there is 4 kind of launch site too.

Launch Site Names Begin with 'CCA'

Task 2

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

In [20]:
```
%sql SELECT * from SPACX where LAUNCH_SITE like 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://sdw99696:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[20]:

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

- There is 5 launch site begin with CCA which mission outcome all success.
- There is 4 launch site that sponsored by NASA.

Total Payload Mass

Task 3

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

In [28]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as total_payload_mass  from SPACX where CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://sdw99696:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[28]:

total_payload_mass
45596

The total payload carried by booster from NASA is 45.596 kg.

Average Payload Mass by F9 v1.1

Task 4

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

```
In [27]: %sql SELECT avg(PAYLOAD_MASS__KG_) as average_payload_mass from SPACX where BOOSTER_VERSION ='F9 v1.1';
 * ibm_db_sa://sdw99696:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[27]:

average_payload_mass
2928

The average payload mass shown is 2.928 kg for 90 payload mass withtotal payload is 45.596 kg.

First Successful Ground Landing Date

Task 5

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

In [30]:
```
%sql select min(DATE) from SPACX where LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://sdw99696:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[30]:

1
2015-12-22

The first successful landing outcome on ground ad at 2015 12 22.

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

31]:
```
%%sql
SELECT BOOSTER_VERSION FROM SPACX
WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
 * ibm_db_sa://sdw99696:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

31]:

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

There is 4 booster version that successfully landed on drone ship and had payload range 4000 to 6000.

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [36]:
```
%sql SELECT MISSION_OUTCOME,COUNT(MISSION_OUTCOME) AS Count FROM SPACX GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://sdw99696:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[36]:

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

The total of successful mission outcome is 100 and failure in flight is 1

Boosters Carried Maximum Payload

Task 8

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

In [37]:
```
%%sql
SELECT BOOSTER_VERSION FROM SPACX WHERE PAYLOAD_MASS__KG_ =(select max(PAYLOAD_MASS__KG_) from SPACX);
```

```
 * ibm_db_sa://sdw99696:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[37]:

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

There is 12 booster version type that carried maximum payload mass.

2015 Launch Records

Task 9

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [44]:
```
%%sql
SELECT LANDING__OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACX
WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE LIKE '%2015%';
```

```
 * ibm_db_sa://sdw99696:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[44]:

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

The failed landing outcomes in drone ship is always happen in CCAFS LC 40 launch site.
There are 2 type of booster that used when failed landing outcomes in drone ship.

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

Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

In [51]:
```
%%sql
SELECT LANDING__OUTCOME,COUNT(LANDING__OUTCOME) AS COUNT FROM SPACX
WHERE DATE > '2010-06-04' and DATE <'2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY COUNT DESC;
```

```
 * ibm_db_sa://sdw99696:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb
Done.
```

Out[51]:

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

There are 9 landing outcomes that success ground pad and 5 landing outcomes that failure drone ship.

Section 3

Launch Sites Proximities Analysis

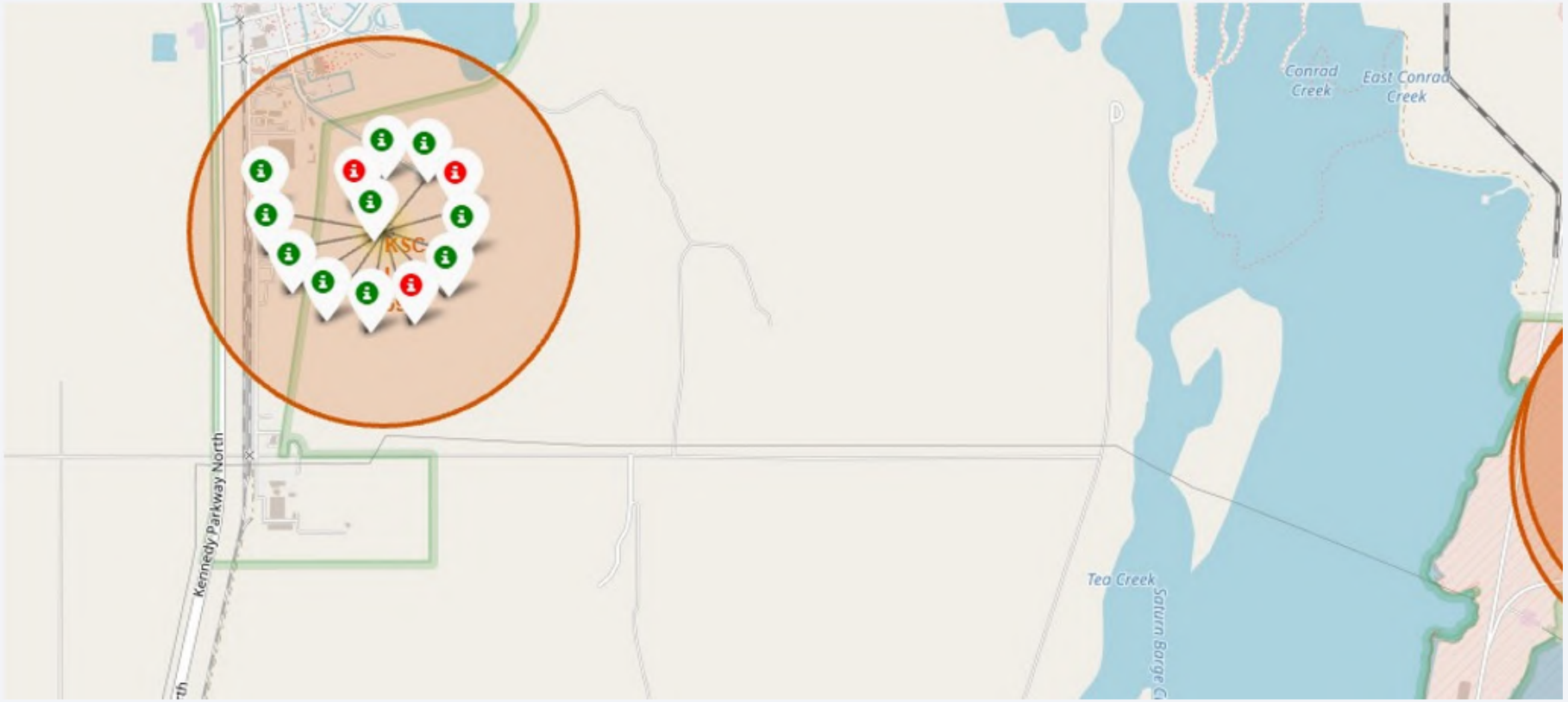
Launch sites Marking

- 

- The markers on this maps show the launch site locations on the map.
-

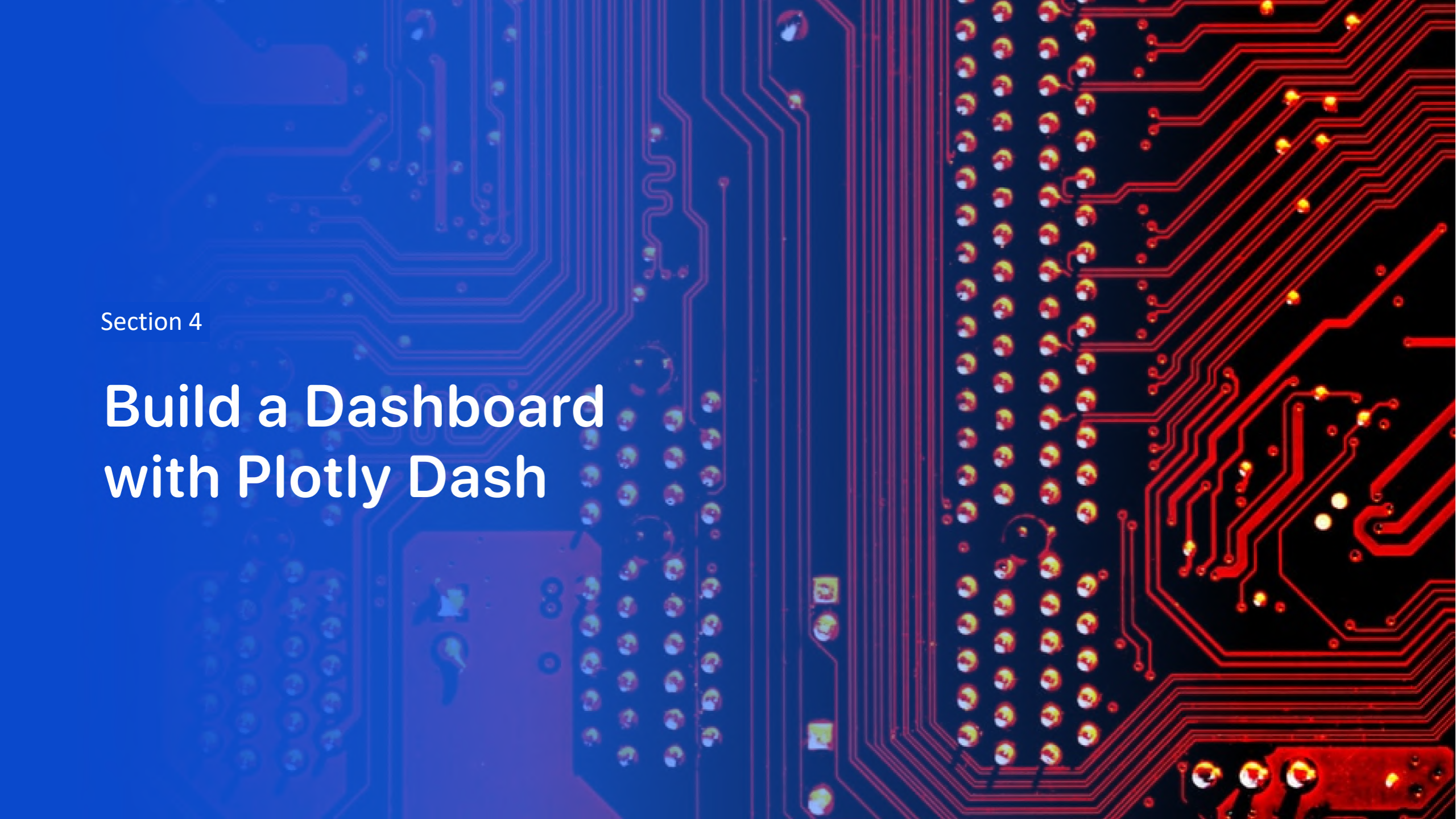
Mark the success/failed launches for each site on the map

A green marker represents a successful landing outcome, while a red one represents failure.



Distances between a launch site to its proximities

The blue line represents the distance between the lunch site and the closest coastline.

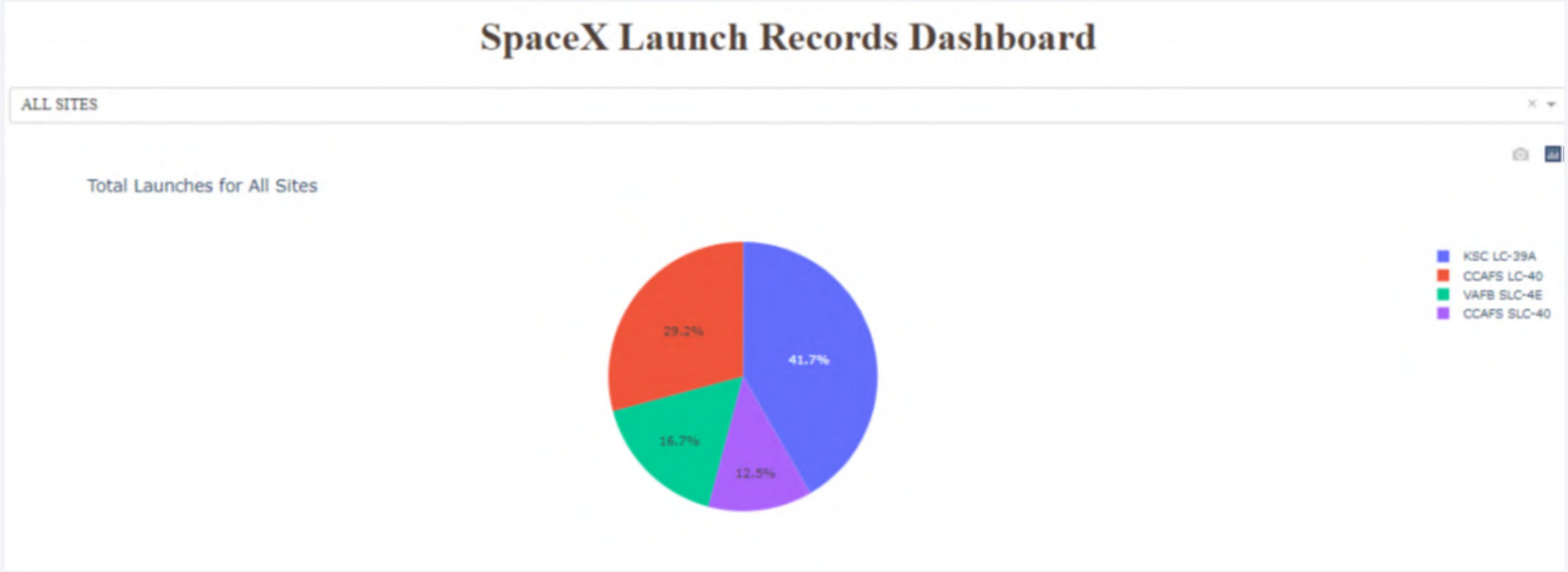


Section 4

Build a Dashboard with Plotly Dash

Launch Success Dashboard for All Sites

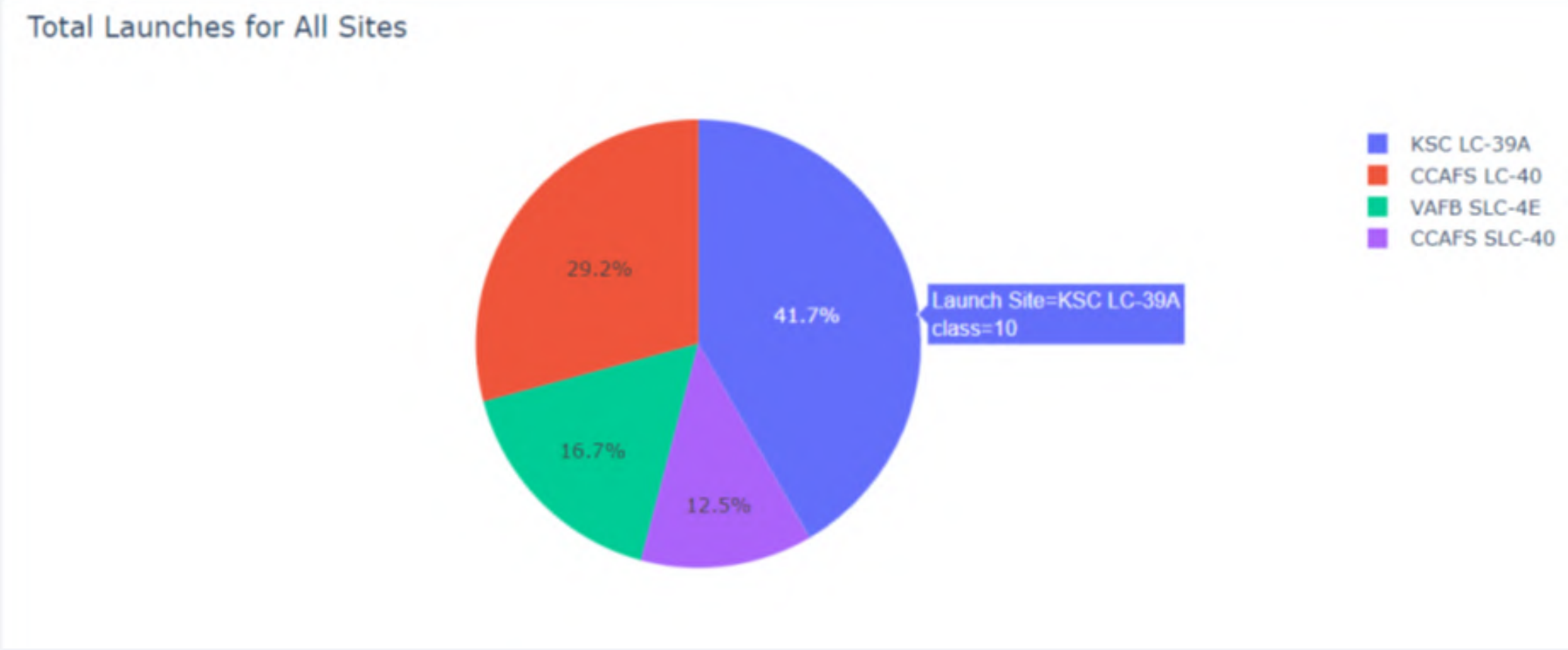
Showing the screenshot of launch success count for all sites, in a Piechart.



Piechart for The Highest Launch Success Site

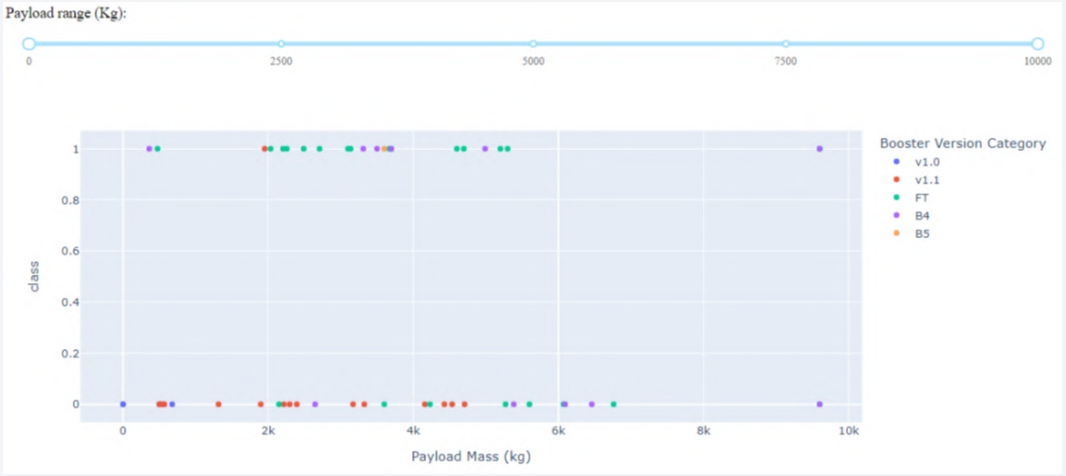
Showing the screenshot of the piechart for the launch site with highest launch success ratio.

The highest total launches is KSC LC-39A site with 41.7%.

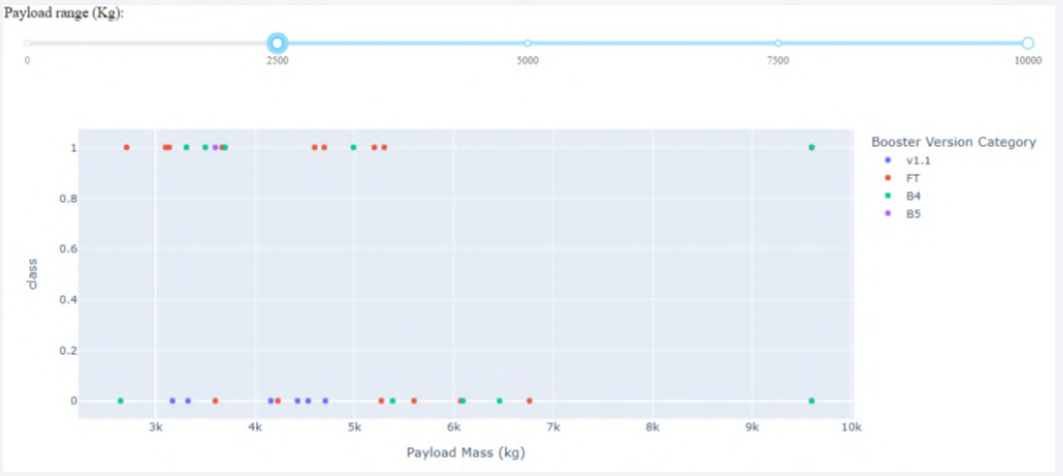


Payload Vs Launch Outcome

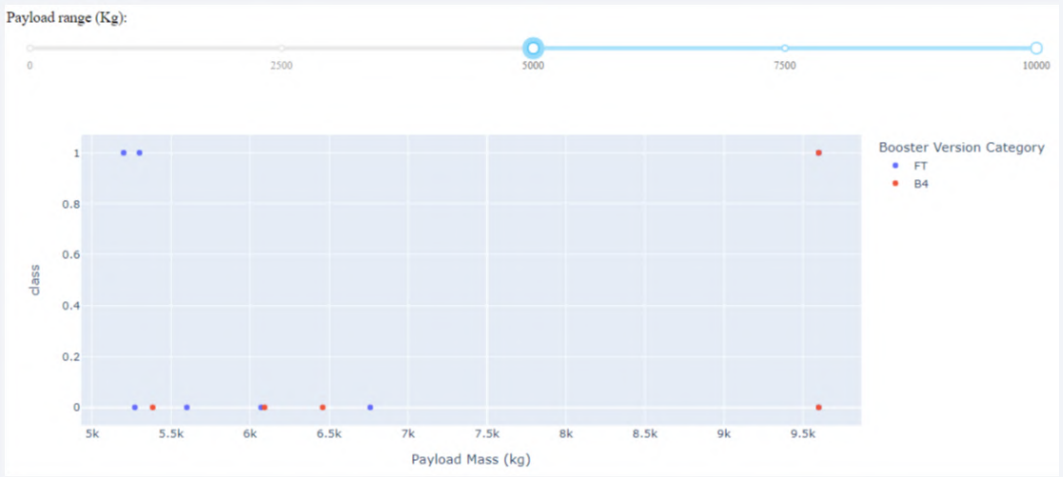
For 0 kg payload



For 2500 kg payload



For 5000 kg payload



For 7500 kg payload

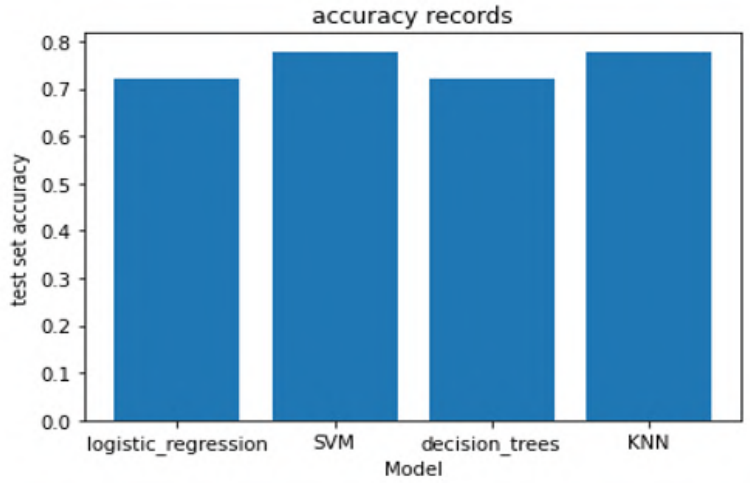
Section 5

Predictive Analysis (Classification)

Classification Accuracy

```
In [44]: x = [lr.score(X_test,Y_test),svm.score(X_test,Y_test),tree.score(X_test,Y_test),KNN.score(X_test,Y_test)]
         y = ['logistic_regression','SVM','decision_trees','KNN']
         plt.bar(y, x)
         plt.title('accuracy records')
         plt.xlabel('Model')
         plt.ylabel('test set accuracy')
         plt.show
```

```
Out[44]: <function matplotlib.pyplot.show(*args, **kw)>
```

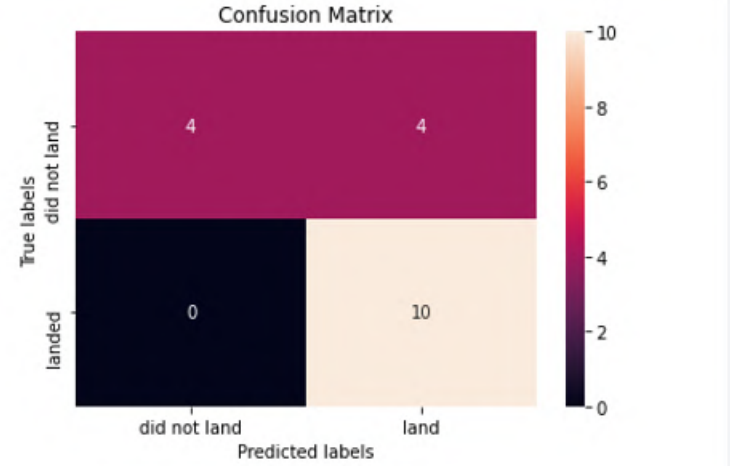



```
In [ ]: #SVM and KNN have the highest out of sample accuracy, thus they are the most accurate models.
```

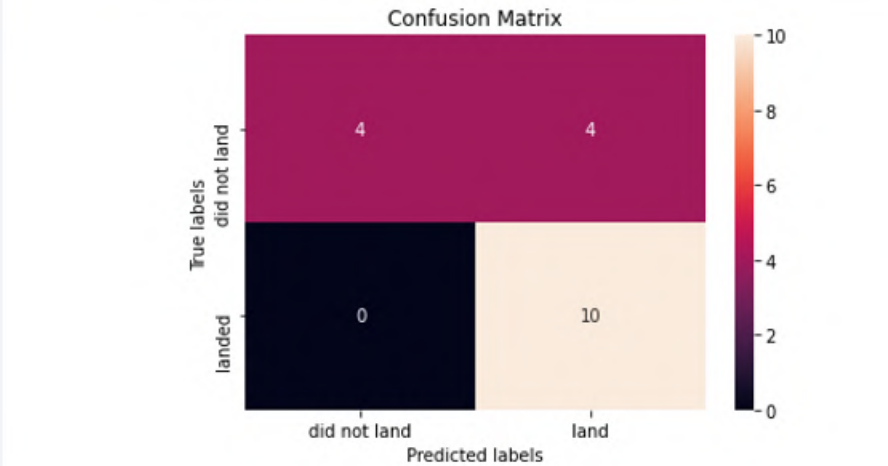
Confusion Matrix

These two graphs represent the confusion matrix for both the SVM and KNNmodels.

```
In [36]: yhat = knn_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```




```
In [22]: yhat=svm_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



These confusion matrices show the largest true positive and true negative values, as well as the least false positive and false negative values.

Conclusions

- Not all the data is important, the collected data may contain irrelevant columns and it is normal to drop them.
- Visualizing data is a good way of determining what features have the strongest effect.
- SQL queries provide wider scope to explore datasets in comparison with traditional EDA.
- SVM and KNN models are the most reliable since they have the highest out of sample accuracy and f1-score.

Thank you!