IBM Developer SKILLS NETWORK

Winning Space Race with Data Science

Ashlee Harnedy
3/9/2023

Executive Summary

- Methodologies Used:
  - SpaceX Data Collection using SpaceX API
  - SpaceX Data Collection with Web Scraping
  - SpaceX Data Wrangling
  - SpaceX Data Analysis using SQL
  - SpaceX EDA Python Data Visualization using Pandas and Matplotlib
  - SpaceX Launch Site Analysis using Folium
  - SpaceX Machine Learning for Landing Prediction
- Summary of all results
  - EDA
  - Interactive Visuals & Dashboards
  - Predictive Analysis

Introduction

- **Project background:**
  - SpaceX advertises launches of its Falcon9 rocket at a cost of \$65 million due to the reusability of the first stage.Other companies spend upwards of \$165 million.The purpose of this project is to determine if the first stage will in fact land to help determine the cost of the launch. The cost determination could lead to other companies having an opportunity to bid for launches against SpaceX.
- **Problems to solve:**
  - The goal will be to predict if the first stage of Falcon9 will land successfully using data from previous launches provided on the SpaceX website.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - First-party data was obtained from SpaceX
- Perform data wrangling
  - Raw data collected, cleaned, validated, then transformed into tables, visuals, and other formats.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Built, tuned, and evaluate classification models

Data Collection

- Data initially collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API.
- This was done completed by defining a series helper functions that would help in the use of the API to extract information using identification numbers in the launch data and then requesting rocket launch data from the SpaceX API url.
- Finally the SpaceX launch data was requested and parsed using the GET request and then decoded the response content as a Json result which was then converted into a Pandas dataframe.

Data Collection – SpaceX API

- Image illustrates the process used to get to the point of creating Pandas dataframe

GitHub Notebook:
[(DataScienceCourseFinal/Data Collection API Lab.ipynb at main · Thursday122786/DataScienceCourseFinal (github.com))]

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/data
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
data.head()
```

Data Collection - Scraping

- Web scraping used to collect Falcon9 historeical launch records using BeautifulSoup and request to extract the records.
- With data extracted, dataframe was created to be able to parse the lunch data

GitHub Notebook:
[(DataScienceCourseFinal/Web Scraping.ipynb at main · Thursday122786/DataScienceCourseFinal (github.com)]

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

Create a `BeautifulSoup` object from the HTML `response`

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

Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
print(html_tables)
```

Data Wrangling

- Created DF using Pandas from collected data, after this data was filtered using BoosterVersion column to limit results to Falcon9 launches.
- Proceeded to deal with missing data values from LandingPad, PayloadMass columns by replacing missing values using mean column value
- Subsequent EDA done to find patterns in data to be used for training models

GitHub Notebook:(DataScienceCourseFinal/EDA-Data Wrangling.ipynb at main · Thursday122786/DataScienceCourseFinal (github.com))

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is
variable `landing_class`:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
df['Class'].value_counts()
```

```
1    60
0    30
Name: Class, dtype: int64
```

This variable will represent the classification variable that represents the outcome of each launch. If t
first stage landed Successfully

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

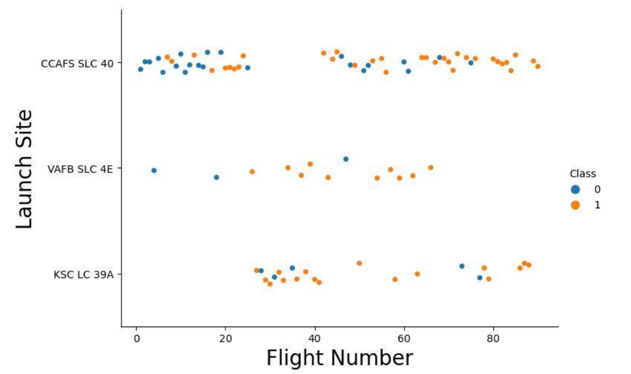
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

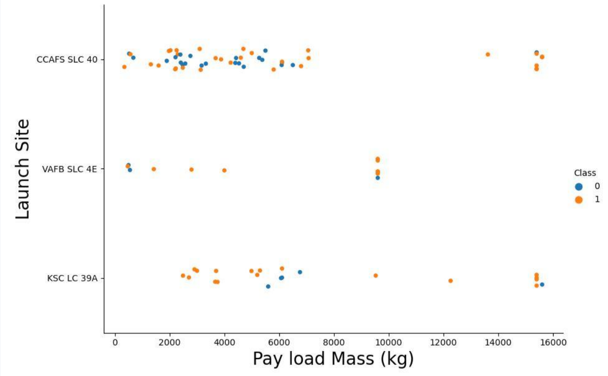
EDA with Data Visualization

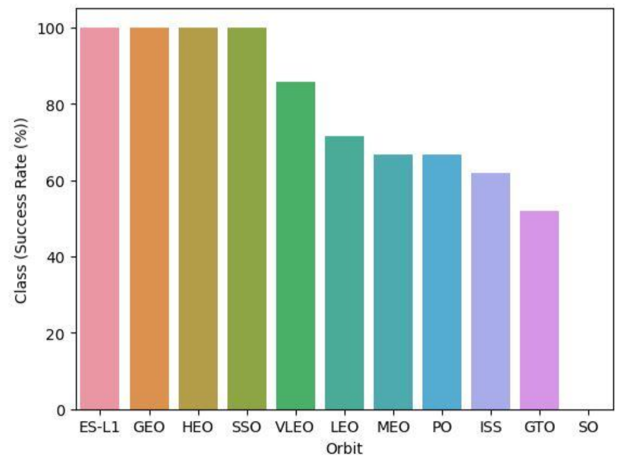
- Data analysis performed using Python pandas, matplotlib
- Scatter plots used to visualize relationships between flight number and launch site, payload and launch site, flight number and orbit type, as well as payload and orbit type
    - Scatter plots allowed for conclusions between the data to be drawn
- Bar chart used to visualize relationship between success rate of each orbit type for easy comparison
- Line plot used to visualize launch success rate over time

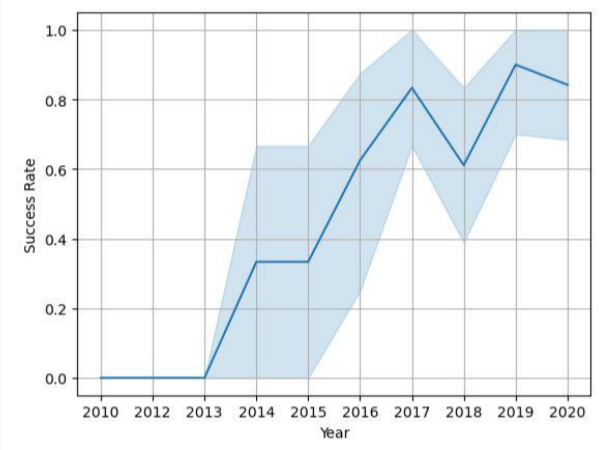
Github Notebook:(DataScienceCourseFinal/EDA with Visualizations.ipynb at main · Thursday122786/DataScienceCourseFinal (github.com))

EDA with Data Visualization(Plots & Charts)









EDA with SQL

The following SQL queries were performed for EDA:

- Display the names of the unique launch sites in the space mission
  - sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
- Display 5 records where launch sites begin with the string 'CCA'
  - sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
- Display the total payload mass carried by boosters launched by NASA (CRS)
  - sql SELECT SUM (PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
- Display average payload mass carried by booster version F9 v1.1e
  - sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%'

EDA with SQL (Cont.)

- List the date when the first successful landing outcome in ground pad was achieved
  - %sql select min(DATE) from SPACEXTBL;
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - %sql select BOOSTER_VERSION from SPACEXTBL where "LANDING _OUTCOME" ='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
- List the total number of successful and failure mission outcomes
  - sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBL GROUP BY MISSION_OUTCOME
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - sql SELECT BOOSTER_VERSION,PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

GitHub Notebook: (DataScienceCourseFinal/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · Thursday122786/DataScienceCourseFinal (github.com))

Build an Interactive Map with Folium

- Folium used to create a map to marked all the launch sites, create objects such as markers, circles, lines to identify the success or failure of launches for each launch site.
- Launch outcomes (failure=0 or success=1).

GitHub URL for SQL Notebook: (Thursday122786/DataScienceCourseFinal: IBM Cert Final Course Project (github.com)

Build a Dashboard with Plotly Dash

Interactive dashboard application create with Plotly dash by:

- Adding a Launch Site Drop-down
- Adding a callback function to render success-pie-chart based on selected site
- Adding a callback function to render the success-payload-scatter-chart scatter plot based on selected site

GitHub URL for SQL Notebook: (DataScienceCourseFinal/spacex_interactive_dash_app.py at main · Thursday122786/DataScienceCourseFinal (github.com))

Predictive Analysis (Classification)

- Data accuracy score was obtained by comparing SVM, Classification Trees, K nearest neighbor, and Logistic regression.
- All Scored the same

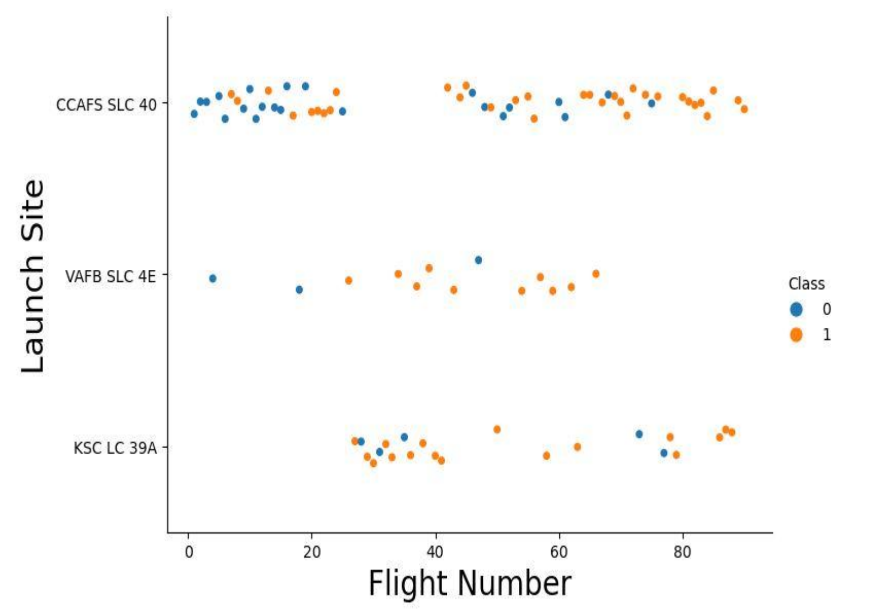
	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

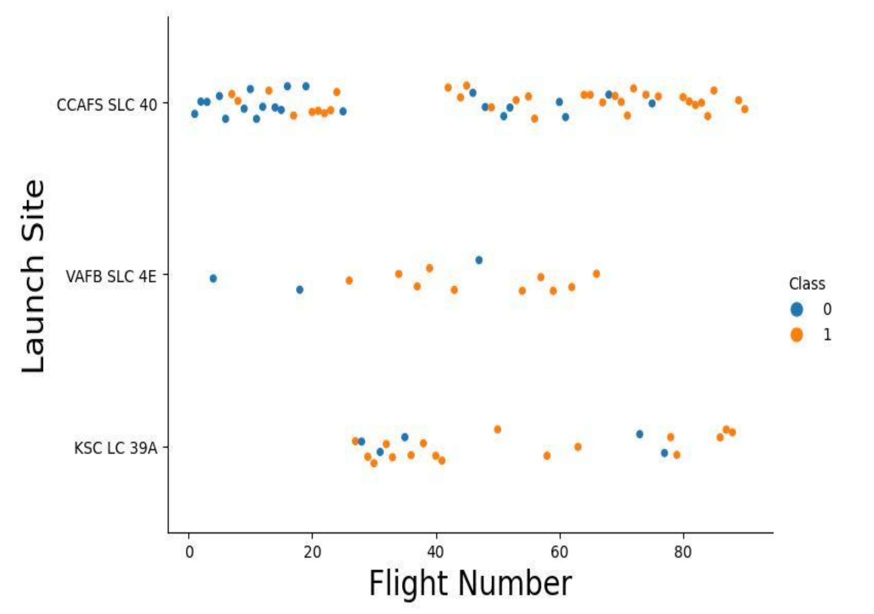
GitHub Notebook:[(DataScienceCourseFinal/Machine Learning.ipynb at main · Thursday122786/DataScienceCourseFinal (github.com)]

Section
2

Insights drawn from EDA

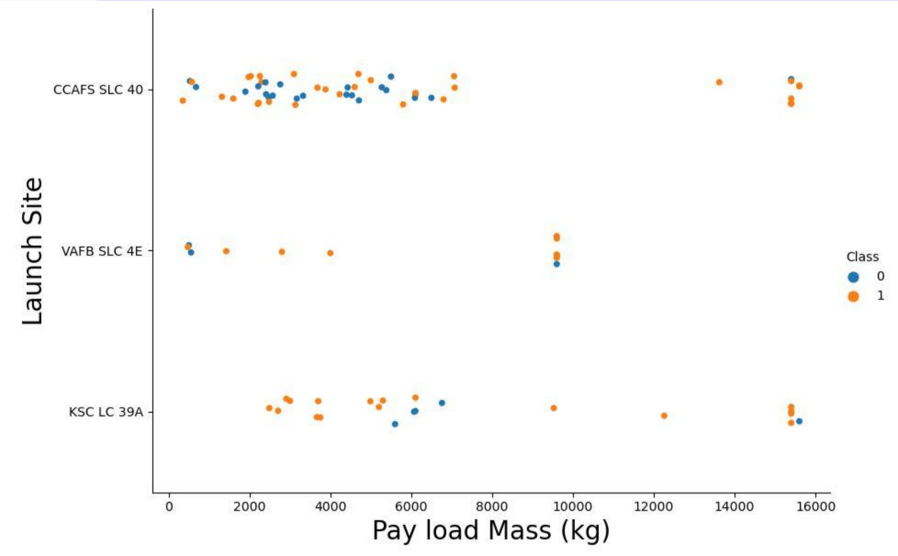
Flight Number vs. Launch Site

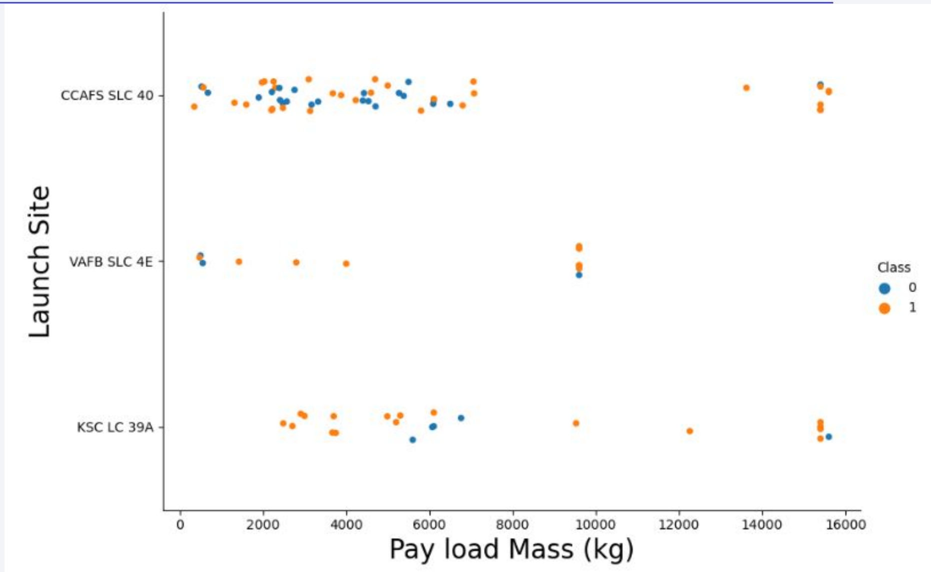




As the number of flights increase at each launch site the success rate increases. You can see for VAFB SLC-4E , 100% success after 50 flights, and for sites CCAFS SLC-40 and KSC LC-39A 100% success rate after 80 flights.

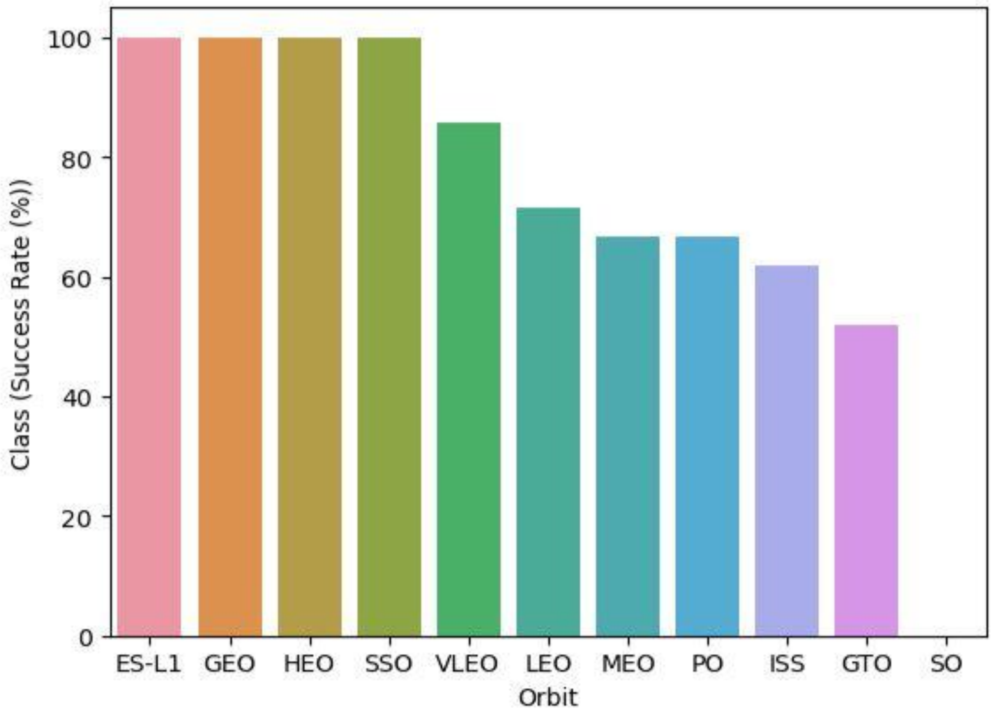
Payload vs. Launch Site





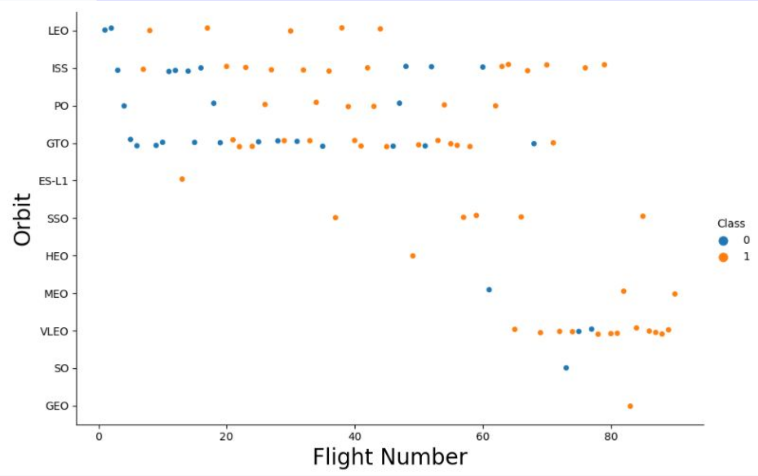
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

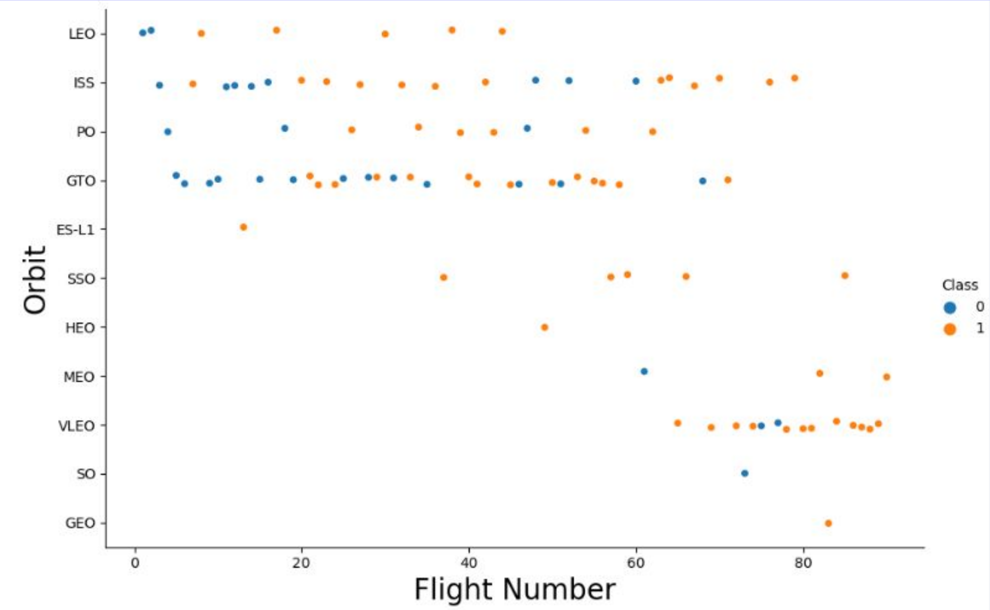
Success Rate vs. Orbit Type



Orbits ES-L1, GEO, HEO, and SSO have the highest success rates at 100%, SO has the lowest success rate at 0%.

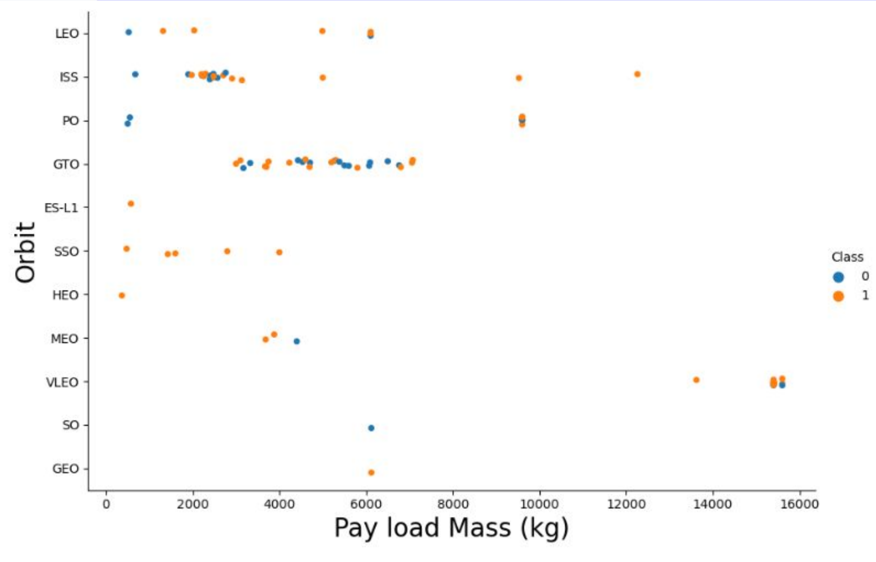
Flight Number vs. Orbit Type

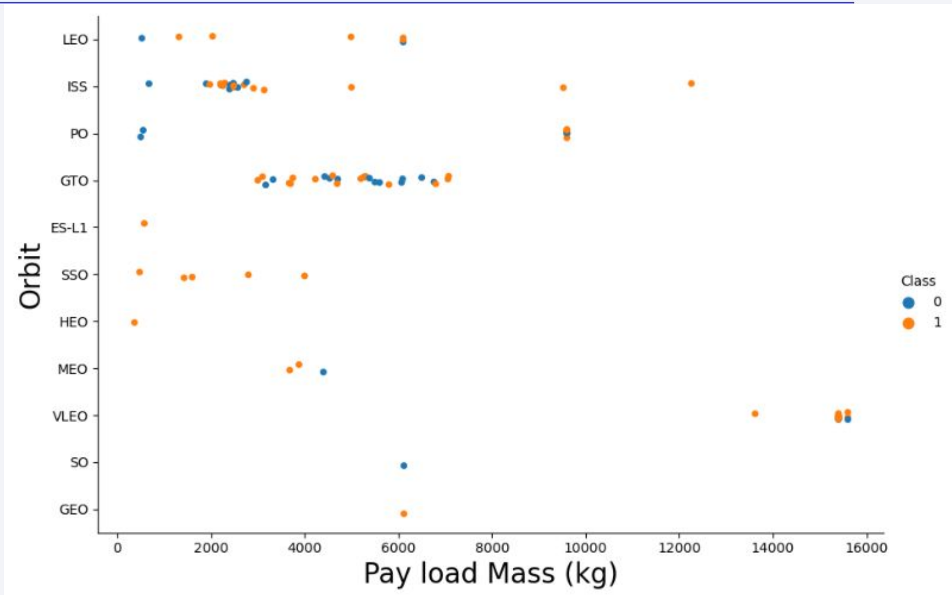




You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

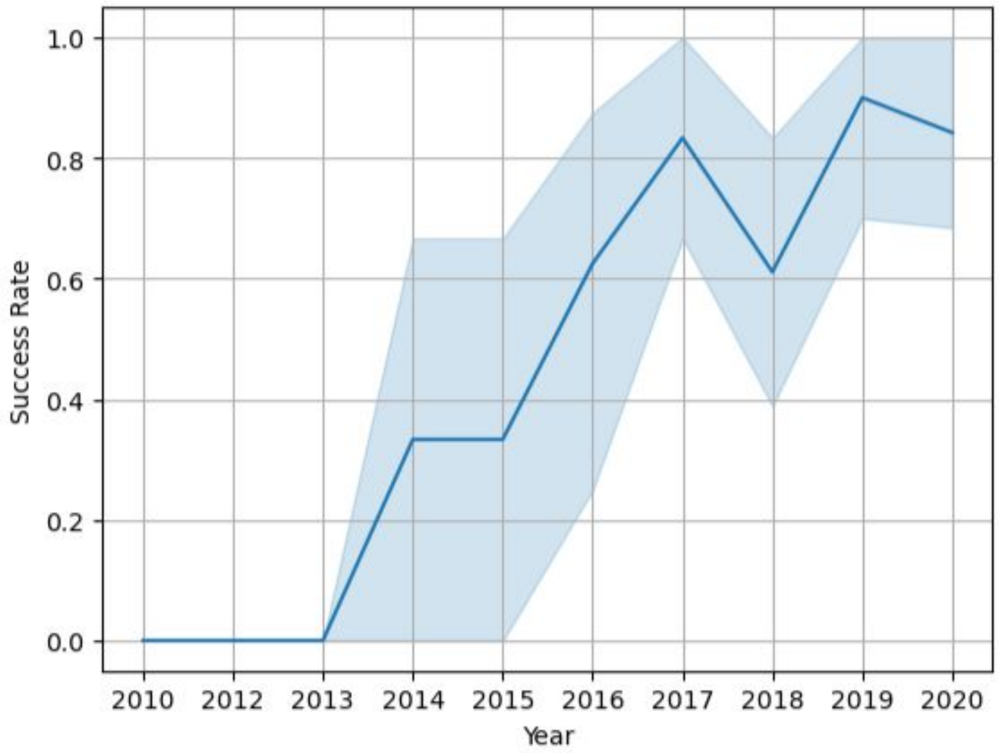




With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

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

Launch Success Yearly Trend



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

All Launch Site Names

- Used SELECT DISTINCT to identify the unique launch sites

```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

```
 * sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

- WHERE LAUNCH_SITE LIKE 'CCA%' to identify the launch sites.

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

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

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Used SUM(PAYLOAD_MASS_KG_) combined with CUSTOMER= 'NASA (CRS)' to obtain mass payload in kg.

```
sql SELECT SUM (PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

SUM (PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
  - Used AVG(PAYLOAD_MASS_KG_) combined with LIKE ‘F9 v1.1%’ to determine average payload in kg.

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

```
sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%'
```

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Select min(DATE) used to find the first successful landing

```
%sql select min(DATE) from SPACEXTBL;
```

```
 * sqlite:///my_data1.db
Done.
```

min(DATE)
01-03-2013

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

- Refined seach using “Landing _Outcome’ = ‘Success (drone ship)’ and PAYLOAD_MASS_KG_ BETWEEN 4000 and 6000 to obtain results

```
%sql select BOOSTER_VERSION from SPACEXTBL where "LANDING _OUTCOME" ='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
```

```
 * sqlite:///my_data1.db
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- SELECT MISSION_OUTCOME, and COUNT used to count the different missing outcomes and display the results

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

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	COUNT(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Used nested SELECT statement to obtain the booster version and mass payload data for the version

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

```
 * sqlite:///my_data1.db
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

2015 Launch Records

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

- Used DATA and LIKE to refine results to 2015 and Landing Outcomes matching Failure (drone ship)

```
%sql SELECT MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE, [LANDING _OUTCOME] FROM SPACEXTBL WHERE DATE('2015') AND [Landing _outcome] LI
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	Booster_Version	Launch_Site	Landing _Outcome
Success	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
Success	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)
Success	F9 v1.1 B1017	VAFB SLC-4E	Failure (drone ship)
Success	F9 FT B1020	CCAFS LC-40	Failure (drone ship)
Success	F9 FT B1024	CCAFS LC-40	Failure (drone ship)

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

- 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
- Isolated Landing Outcomes by data range then ordered by date (see notebook for full dates)

```
%sql SELECT [Landing _Outcome], DATE FROM SPACEXTBL WHERE DATE BETWEEN '06-04-2010' AND '20-03-2017' ORDER BY DATE;
```

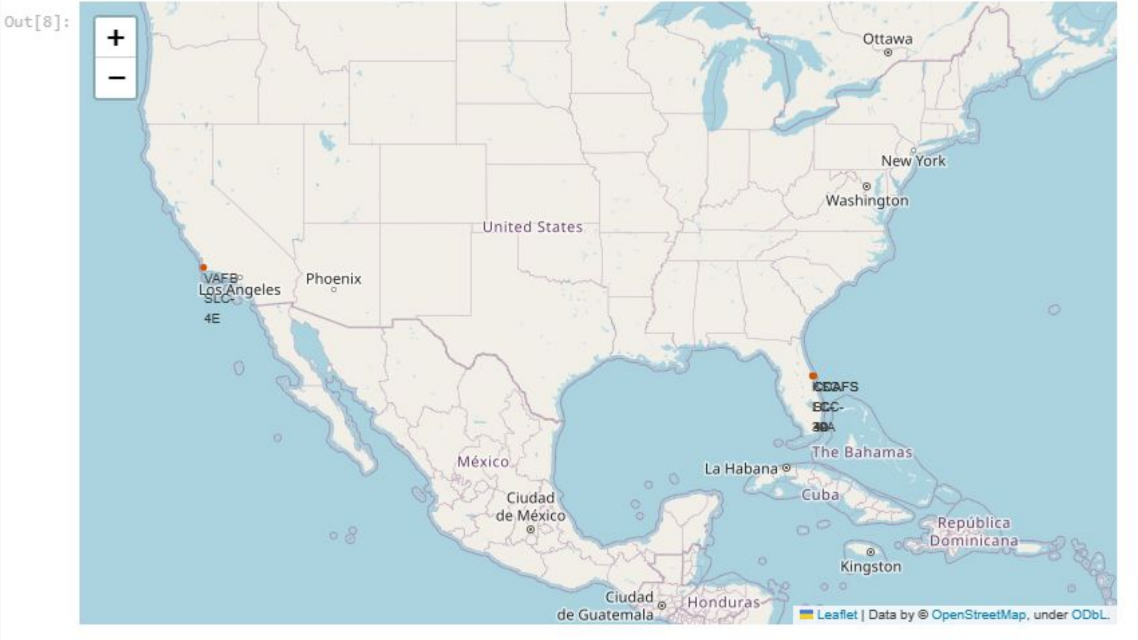
```
 * sqlite:///my_data1.db
Done.
```

Landing _Outcome	Date
Success (drone ship)	06-05-2016

Section
3

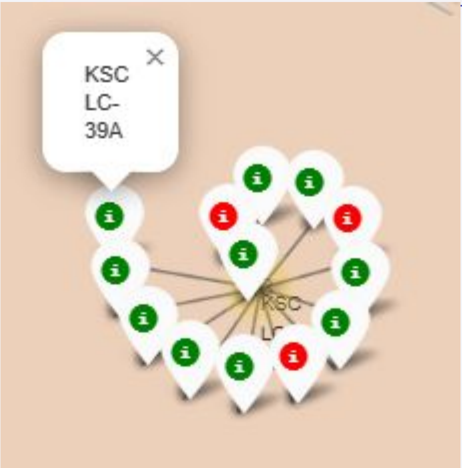
Launch Sites Proximities Analysis

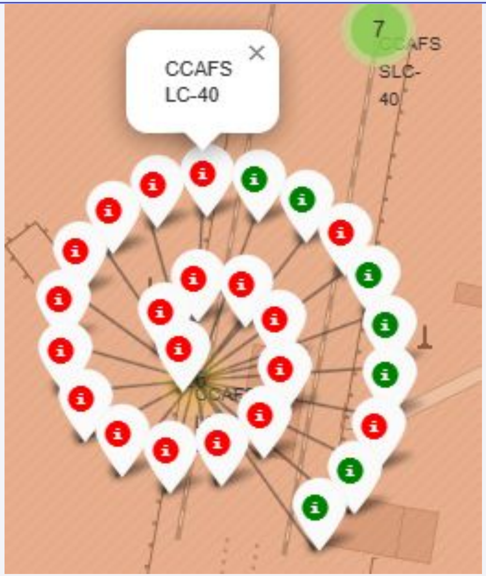
SpaceX Launch Sites

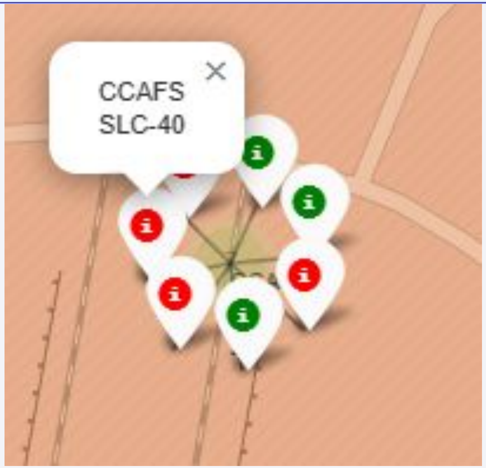


Launch sites are located in coastal areas on the east and west coast.

Florida Launch Site

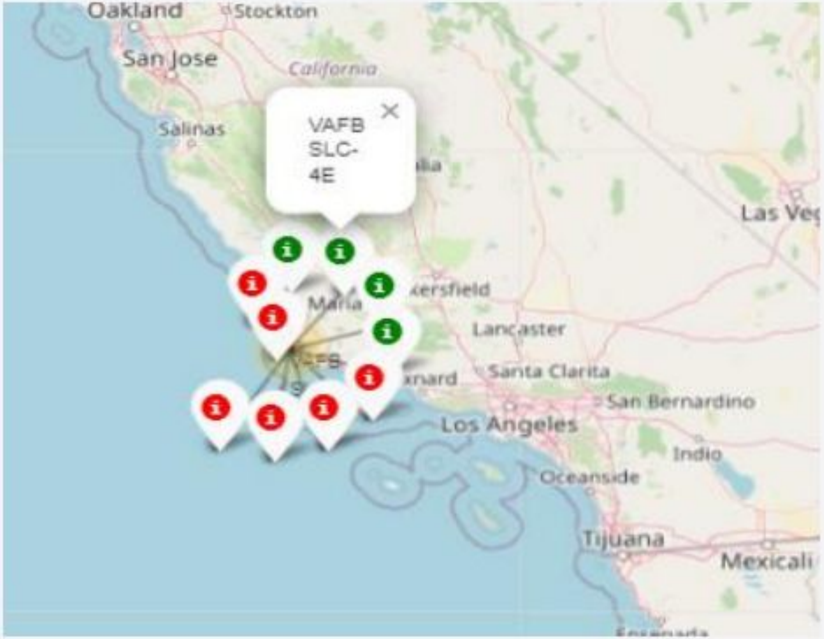






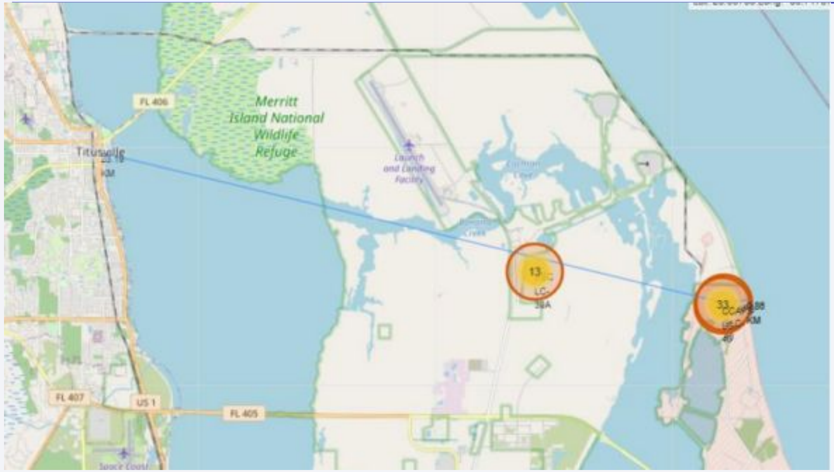
KSC LC-39A has a higher success rate compared to CCAFS LC-40 and CCAFS SLC-40.

Launch outcomes for California



California launch sites have alower success rate than most of the Florida launch sites.

FL launch site proximities to highways and coast



Launch site CCAFS SLC-40 23.19km from the highway.

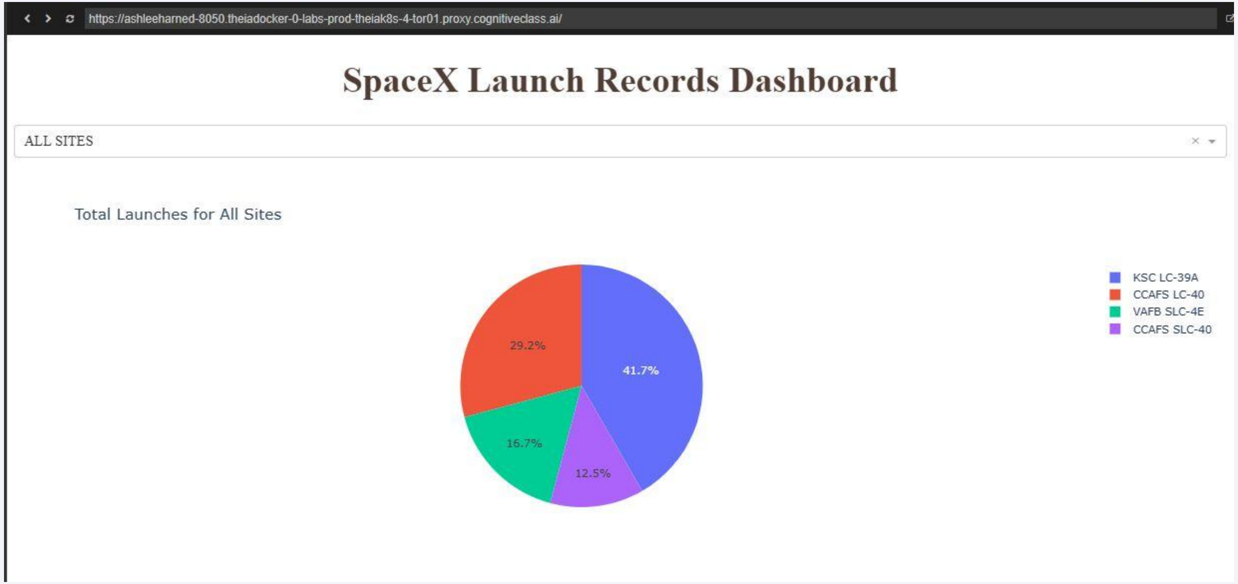
Launch site CCAFS SLC-40 is .86km from the coast.

Section
4

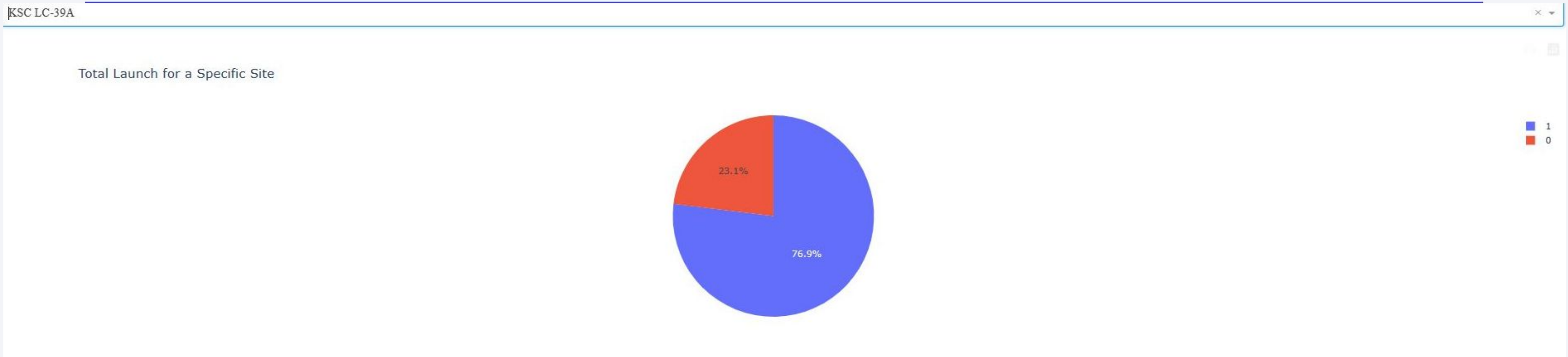
Build a Dashboard with Plotly Dash

Pie-Chart: Launch Success All Sites

- Launch Site KSC LC-39A had the highest success rate at 41.7%, CCAFS SLC-40 had the lowest success rate at 12.5%



Pie Chart: Launch Site Highest Success Rate



- KSC LC-39 Had the highest success rate
- 76.9% percent of launches were successful
- 23.1% of launches were failures

<Dashboard Screenshot 3>

- Launch site CCAFS LC-40 and booster version FT has the largest success rate from a payload mass of >2000kg

Section
5

Predictive Analysis (Classification)

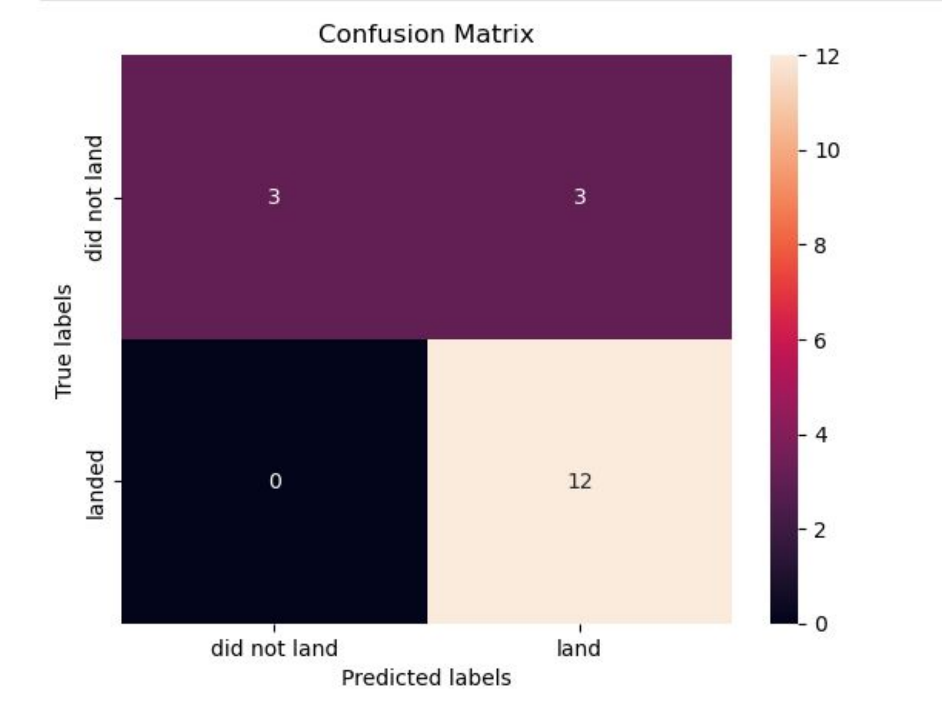
Classification Accuracy

All models tested the same with .0833% accuracy

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Confusion Matrix

- All models tested the same to the right is what all models returned when tested.



Conclusions

1. Success rates differed based on the launch sites
2. As the number of flights increased the success rate increased.
3. Payloads were limited to mass under 10000kg
4. Success rates increased from 2013 until 2020.

Thank you!