



SPACE Y

Talha Shabqder

Jan 26, 2023



Contents

Introduction
Methodology
Results
Conclusion
Appendix

Summary



Summary of methodologies

Data Collection through SpaceX API

Data Collection with Web Scraping

Data Wrangling

Exploratory Data Analysis with SQL

Exploratory Data Analysis with Data Visualization

Interactive Visual Analytics with Folium

Dashboard using dash by plotly

Machine Learning Prediction



Summary of all results

EDA results

Interactive analytics findings

Predictive Analytics result

01-Introduction



- **Background and context for the project**

Space X offers Falcon 9 rocket launches on their website for 62 million dollars; other suppliers charge up to 165 million dollars apiece; much of the savings is due to Space X's ability to reuse the first stage. As a result, if we can predict whether the first stage will land, we can estimate the cost of a launch. This data can be utilized if another business wishes to compete with Space X for a rocket launch. The project's purpose is to build a machine learning pipeline that can forecast if the first stage will successfully land for the supposition this project uses a hypothetical company named SPACE-Y founded by Alon Musk.

- **Problems and research questions**

1. What variables influence if the rocket lands successfully?
2. What is the interaction of different factors that determines the success rate of a good landing?
3. What operational conditions are required to achieve a successful landing program?

02-Methodologies



Methodologies

1. Data Collection
2. Data Wrangling
3. EDA with SQL and Visualization
4. Interactive Analytics analysis
5. Predictive analysis using classification ML models

Data Collection

- The data collection involved several sources and methods
 - From the Wikipedia page of falcon 9 launches Using Requests and BeautifulSoup4 module
 - The table names were extracted from the HTML table header and each row in the HTML tables was then parsed to create the data frame the data frame was later saved as a CSV for future analysis.
 - From SpaceX API using Requests Module
 - The response content was converted to a pandas data frame using `.json_normalize()` the data frame was later filtered to have only records of Falcon 9 launches

Data Collection—SpaceX API

Collected data using the SpaceX API's get requests, cleaned the obtained data, and also performed somewhat basic data wrangling and formatting.

The link to the Notebook:
<https://github.com/notBruisWayne/Applied-data-science-capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb>

1. get Requests from SpaceX API

```
In [9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_
```

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

```
In [10]: r=requests.get(static_json_url)
r.status_code
```

```
Out[10]: 200
```

2. Replacing missing payload values with mean

```
In [28]: # Calculate the mean value of PayloadMass column
meanvalue=data_falcon9['PayloadMass'].mean()
print(meanvalue)
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan,meanvalue,inplace=True)
data_falcon9.isnull().sum()
data_falcon9.head()
```

```
6123.547647058824
```

3. Saving the DF as a CSV

```
Out[28]:
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004

```
In [29]: #data_falcon9.to_csv('space_y_dataset_part_1.csv', index=False)
```

```
In [30]: data_falcon9.shape
```

```
Out[30]: (90, 17)
```

Data Collection—BeautifulSoup4

Created a BS4 object and parsed through the HTML tables tags to iterate through each td or row and

The link to the Notebook:
<https://github.com/notBruisewayne/Applied-data-science-capstone/blob/main/jupyter-labs-webscraping.ipynb>

1. Getting response object and parsing it into a soup object

```
In [5]: # use requests.get() method with the provided static_url
r=requests.get(static_url)
# assign the response to a object
```

Create a BeautifulSoup object from the HTML response

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup=BeautifulSoup(r.text,'html.parser')
```

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

```
In [7]: # Use soup.title attribute
print(soup.title)
```

2. Looping through each row and appending the dictionary

```
launch_outcome = list(row[''].strings)[0]
print(launch_outcome)
launch_dict['Launch outcome'].append(launch_outcome)
# Booster Landing
# TODO: Append the launch_outcome into launch_dict with key 'Booster Landing'
booster_landing = landing_status(row[8])
print(booster_landing)
launch_dict['Booster landing'].append(booster_landing)
```

```
1
4 June 2010
18:45
F9 v1.0B0003.1
CCAFS
Dragon Spacecraft Qualification Unit
Dragon Spacecraft Qualification Unit
LEO
SpaceX
Success
```

3. Casting the dictionary to a DF and saving it as CSV file

```
In [15]: df=pd.DataFrame.from_dict(launch_dict)
```

We can now export it to a CSV for the next section, but to make the answers consistent and in case you have difficulties finishing this lab.

Following labs will be using a provided dataset to make each lab independent.

```
In [16]: df.to_csv('spacex_web_scraped_first_table.csv', index=False)
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```


EDA with Visualization

Explored success patterns in several relations of the launches i.e., Launch site and total flights, Launch site and payload mass, the success rate for each orbit, orbits and flight numbers, orbits and payload mass, the yearly trend of successful rates, and furthermore used one-hot encoding for the columns Orbits, LaunchSite, LandingPad, and Serial which will be useful in modeling.

The link to the Notebook:

<https://github.com/notBruiseWayne/Applied-data-science-capstone/blob/main/jupyter-labs-eda-dataviz.ipynb>

EDA with SQLLITE3

Since IBM.DB2 was not available we used the sqlite3 database to perform SQL EDA and analyzed:

- Unique names in Launch sites and launch sites with names 'CCA' that refers to Cape Canaveral Space.
- Total payload mass carried by NASA boosters (was found to be 45,596) and the average mass of booster Falcon F9 v1.1 (was 2989.4 units).
- Boosters with success in drone ship with mass in the range 4000-6000 units.
- The Total number of successful and unsuccessful missions.

The link to the Notebook:

https://github.com/notBruiseWayne/Applied-data-science-capstone/blob/main/jupyter-labs-eda-sql-coursera_sqlite.ipynb

Interactive visualizations with folium

- We identified all launch locations and inserted map elements such as markers, circles, and arcs to the folium map to indicate the effectiveness or failure of missions for every site.
- We found which launch sites had a pretty high success rate using color-labeled marker clusters.
- We measured the distances between a launch location and its surroundings. For example, we responded to the following question:
 - Were launch locations located near trains, motorways, or coastlines?
 - Did launch sites maintain a particular distance from metropolitan areas?

The link to the Notebook:

https://github.com/notBruiseWayne/Applied-data-science-capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Interactive Dashboard with dash App

- Using Plotly dash, we created an interactive dashboard.
- We created pie charts depicting the overall number of launches by specific sites.
- We plotted a scatter graph showing the relationship between Outcome and Payload Mass (Kg) for the different booster versions.

The link to the Dash App code:

<https://github.com/notBruiseWayne/Applied-data-science-capstone/blob/main/spacexDashApp.py>

Classification/ Predictive Analysis

- We imported the data and then (using NumPy, Scikitlearn, and pandas), converted it, then divided it into training and testing sets.
- Using GridSearchCV, we created several machine-learning models(Decision tree, KNN, SVM and Logistic Regression) and tuned various hyperparameters. We utilized accuracy as our model's measure and increased it through feature extraction and algorithm tweaking, we also used a heat map/ confusion matrix to view True positives, True Negatives False Positives, and False Negatives.
- Hence the best-performing model was found which was Decision Tree Classifier.

The link to the Notebook:

https://github.com/notBruiseWayne/Applied-data-science-capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

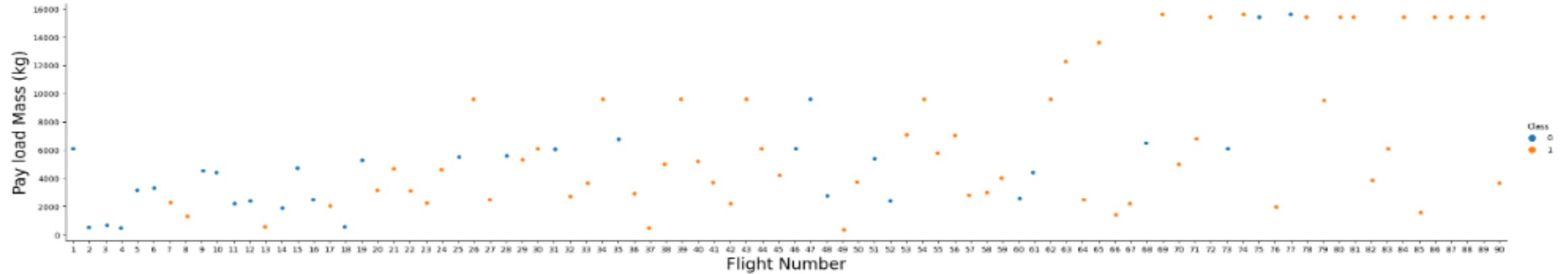
Results

- EDA results
- Interactive Visualizations results
- Predictive analysis results

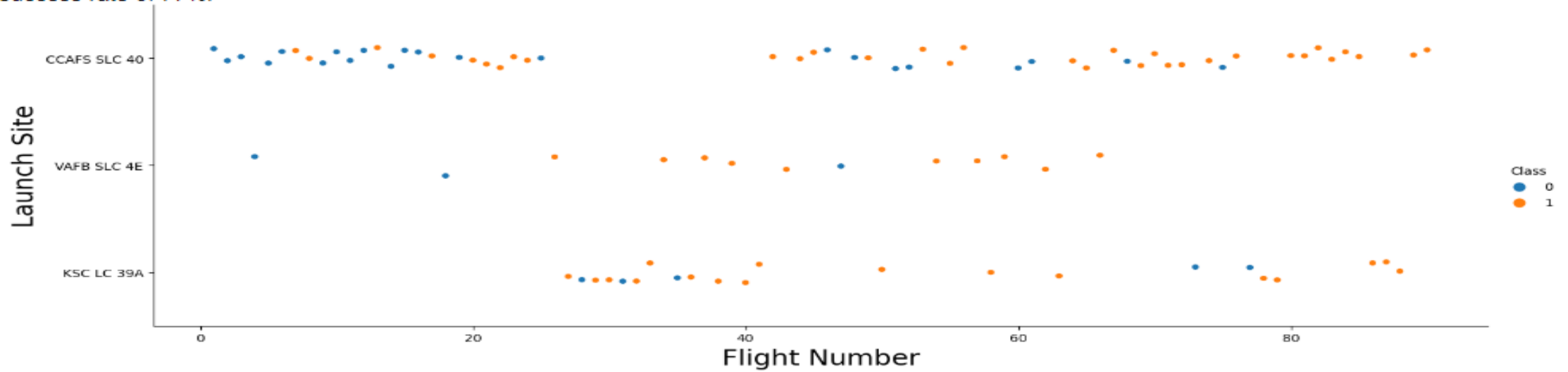
A satellite view of Earth at night, showing the illuminated landmasses of North and South America against the dark background of the oceans. The city lights are visible as bright yellow and orange spots across the continents. A thin white line runs vertically along the left edge of the image.

03 - Findings and insights from EDA

Payload mass vs Flight no and Launch sites vs flight numbers

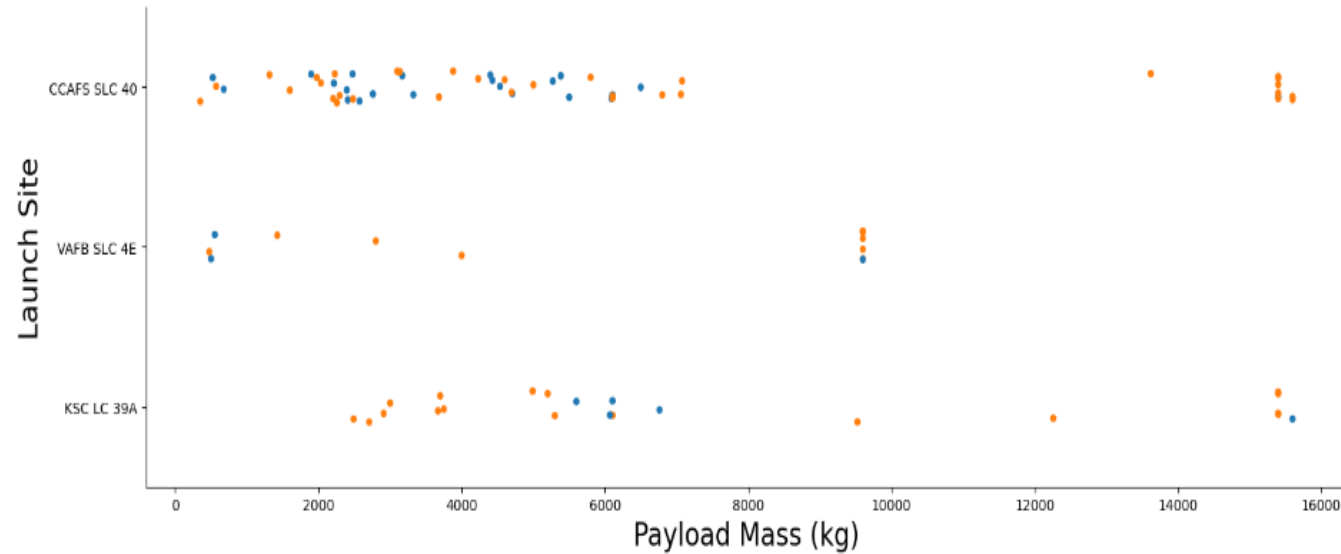


We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

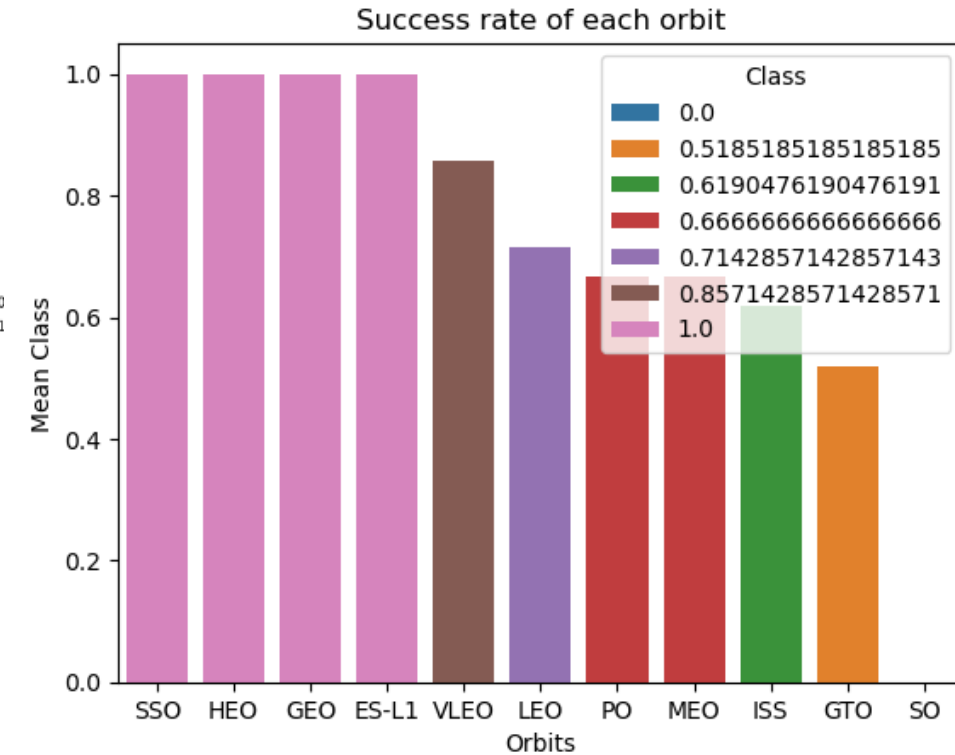


Higher flight No. had more frequent successful missions suggesting newer launches were more successful than previous ones

Launch site vs Payload mass and Orbits vs Success rate

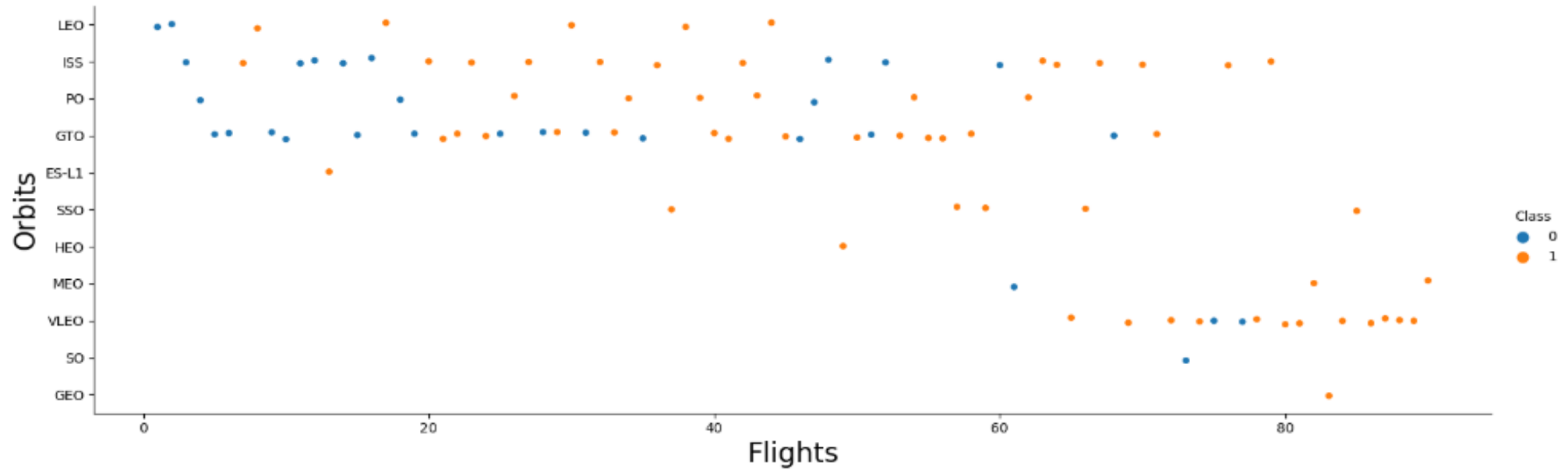


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavy payload mass (greater than 10000).



Above 8000 payload mass resulted in better success rates and Orbits SSO, HEO, GEO and ES-L1 Had most successful missions

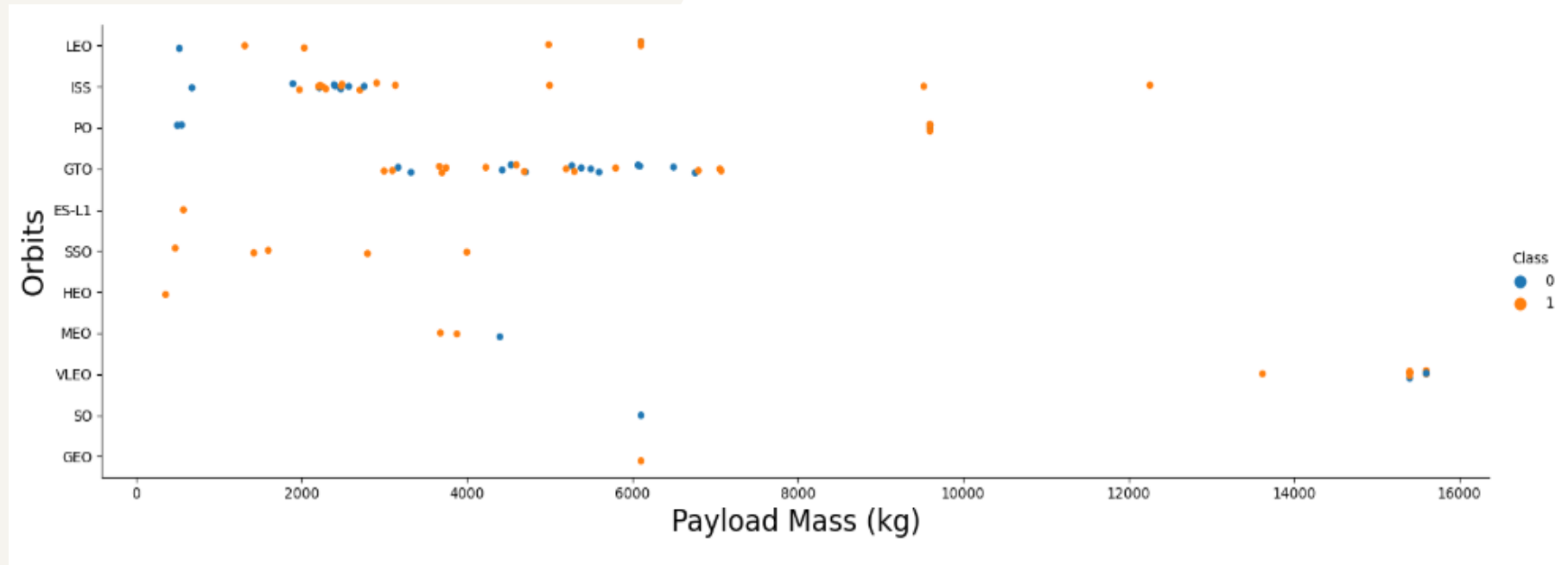
Orbits vs Flights



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.



Orbits vs payload mass

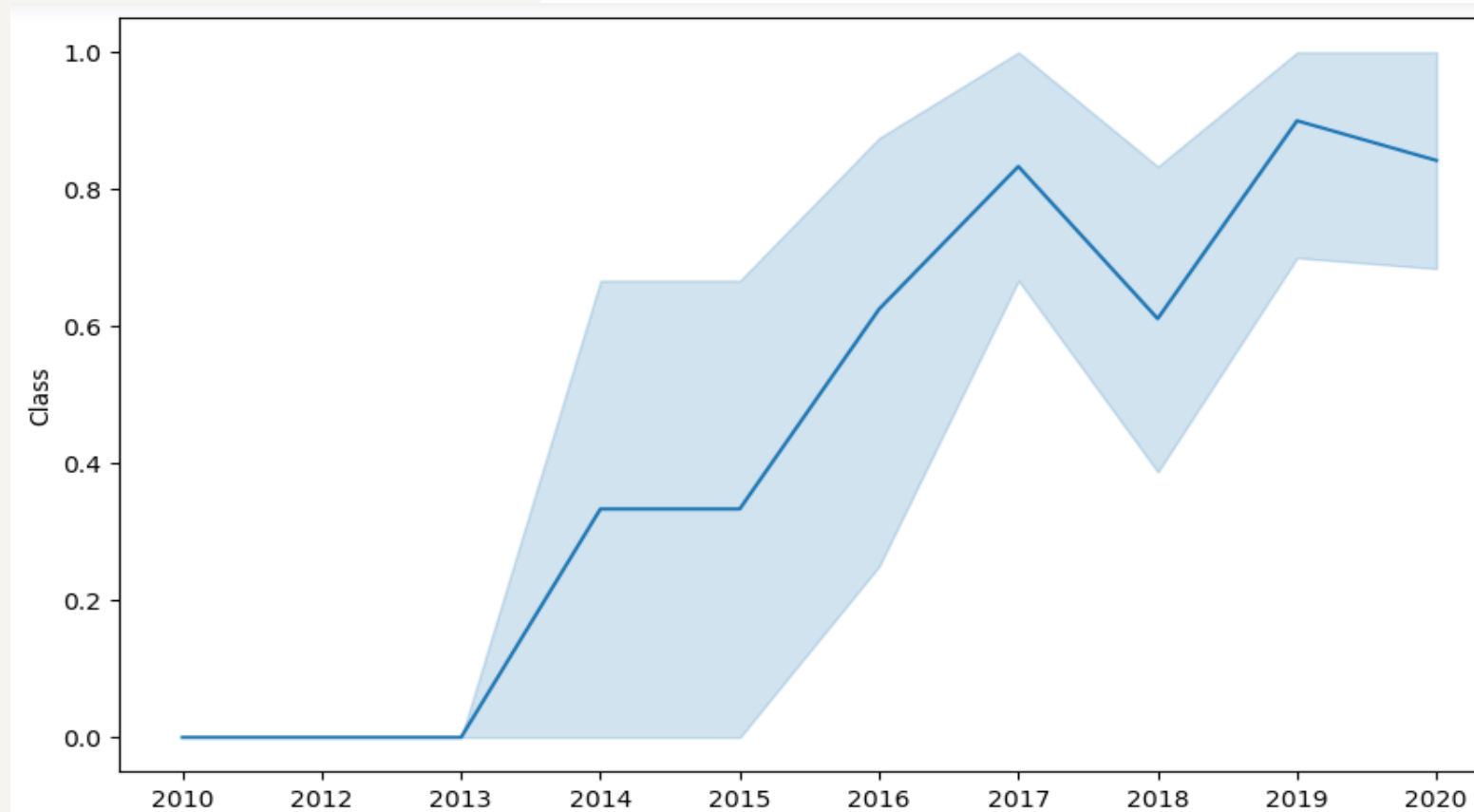


With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

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



Yearly Trend of success rates



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



SQL EDA – Site Names

Unique site names from datasets
And sites with names beginning with 'CCA'
Which refers to Cape Canaveral
Space Launch Complex

```
In [37]: con = sqlite3.connect("my_data1.db")
df=pd.read_sql_query('SELECT Distinct Launch_Site from SPACEXTBL',con)
con.close()
df.head()
```

```
Out[37]:
```

	Launch_Site
0	CCAFS LC-40
1	VAFB SLC-4E
2	KSC LC-39A
3	CCAFS SLC-40

```
con = sqlite3.connect("my_data1.db")
s=pd.read_sql_query(''SELECT * from SPACEXTBL Where Launch_Site LIKE 'CCA%' LIMIT 5'',con)
con.close()
s.head()
```

	Date	Time(UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
0	04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt



SQL EDA – Payload Mass

Total payload mass by NASA Boosters which was found to be 45596

```
con = sqlite3.connect("my_data1.db")
df=pd.read_sql_query("SELECT SUM(PAYLOAD_MASS_KG_) AS Total_PayloadMass FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)'",con)
con.close()
df
```

Total_PayloadMass	
0	45596

While the average payload mass for F9 v1.1 was 2928.4

```
con = sqlite3.connect("my_data1.db")
df=pd.read_sql_query("""
SELECT AVG(PAYLOAD_MASS_KG_) AS Total_PayloadMass
FROM SPACEXTBL
WHERE Booster_Version='F9 v1.1'
""",con)
con.close()
df
```

Total_PayloadMass	
0	2928.4



SQL EDA – First Successful landing on ground pads

First successful landing on ground pads

```
con = sqlite3.connect("my_data1.db")
df=pd.read_sql_query("""
SELECT MIN(Date) AS First_Successfull_landing_date
FROM SPACEXTBL
WHERE Landing_Outcome ='Success (ground pad)'
""",con)
con.close()
df
```

	First_Successfull_landing_date
0	01-05-2017



SQL EDA – Boosters and drone ships

Boosters that had successful landing outcome with drone ships
Having mass in range 4000-6000

```
con = sqlite3.connect("my_data1.db")
df=pd.read_sql_query("""
SELECT Booster_Version AS BOOSTERS_IN_DRONE_SHIPS
FROM SPACEXTBL
WHERE
    Landing_Outcome = 'Success (drone ship)'
    AND PAYLOAD_MASS_KG_ > 4000
    AND PAYLOAD_MASS_KG_ < 6000
""",con)
con.close()
df
```

BOOSTERS_IN_DRONE_SHIPS	
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2



SQL EDA – Total success and max mass by boosters

From the sample of 100, 98 were successful landings

key_0	SUCCESSFUL	UNSUCCESSFUL
0	0	98
3	0	3

List of boosters that carried the maximum mass

```
con = sqlite3.connect("my_data1.db")
df=pd.read_sql_query("""
SELECT Booster_Version,*PAYLOAD_MASS_KG_
FROM SPACEXTBL
WHERE
    PAYLOAD_MASS_KG_ = (SELECT MAX(PAYLOAD_MASS_KG_)
    FROM SPACEXTBL)
ORDER BY Booster_Version
""",con)
con.close()
df
```

	Booster_Version	PAYLOAD_MASS_KG_
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600



SQL EDA – 2015 launches

Launch records from 2015 that had failure in drone ship landings

```
con = sqlite3.connect("my_data1.db")
df=pd.read_sql_query("""
SELECT Booster_Version, Launch_Site, Landing_Outcome, substr(Date, 4, 2) AS MONTH
FROM SPACEXTBL
WHERE Landing_Outcome LIKE 'Failure (drone ship)'
AND substr(Date,7,4)='2015'
""",con)
con.close()
df
```

	Booster_Version	Launch_Site	Landing_Outcome	MONTH
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)	01
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)	04



SQL EDA – Outcomes form 4-6-10 to 20-3-17

```
con = sqlite3.connect("my_data1.db")
df=pd.read_sql_query("""
SELECT  Landing_Outcome AS SUCCESSFUL_LANDINGS, COUNT(Landing_Outcome) AS CNT,substr(Date,7,4) AS YEAR
FROM SPACEXTBL
WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017'
GROUP BY Landing_Outcome
ORDER BY CNT DESC
""",con)
con.close()
df
```

	SUCCESSFUL_LANDINGS	CNT	YEAR
0	Success	20	2018
1	No attempt	10	2012
2	Success (drone ship)	8	2016
3	Success (ground pad)	6	2016
4	Failure (drone ship)	4	2015
5	Failure	3	2018
6	Controlled (ocean)	3	2014
7	Failure (parachute)	2	2010
8	No attempt	1	2019



04-Launch Sites



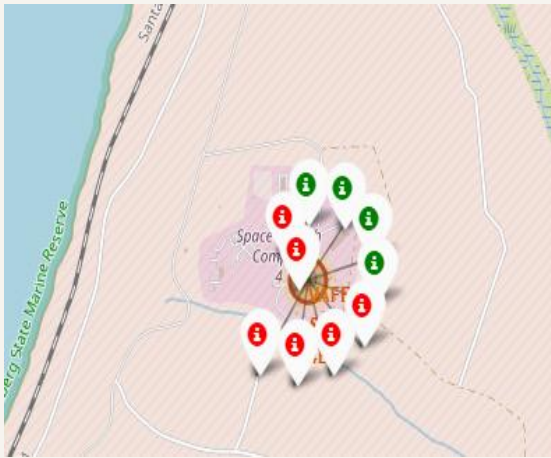
Launch sites

All launch sites were on the east and west coasts of USA



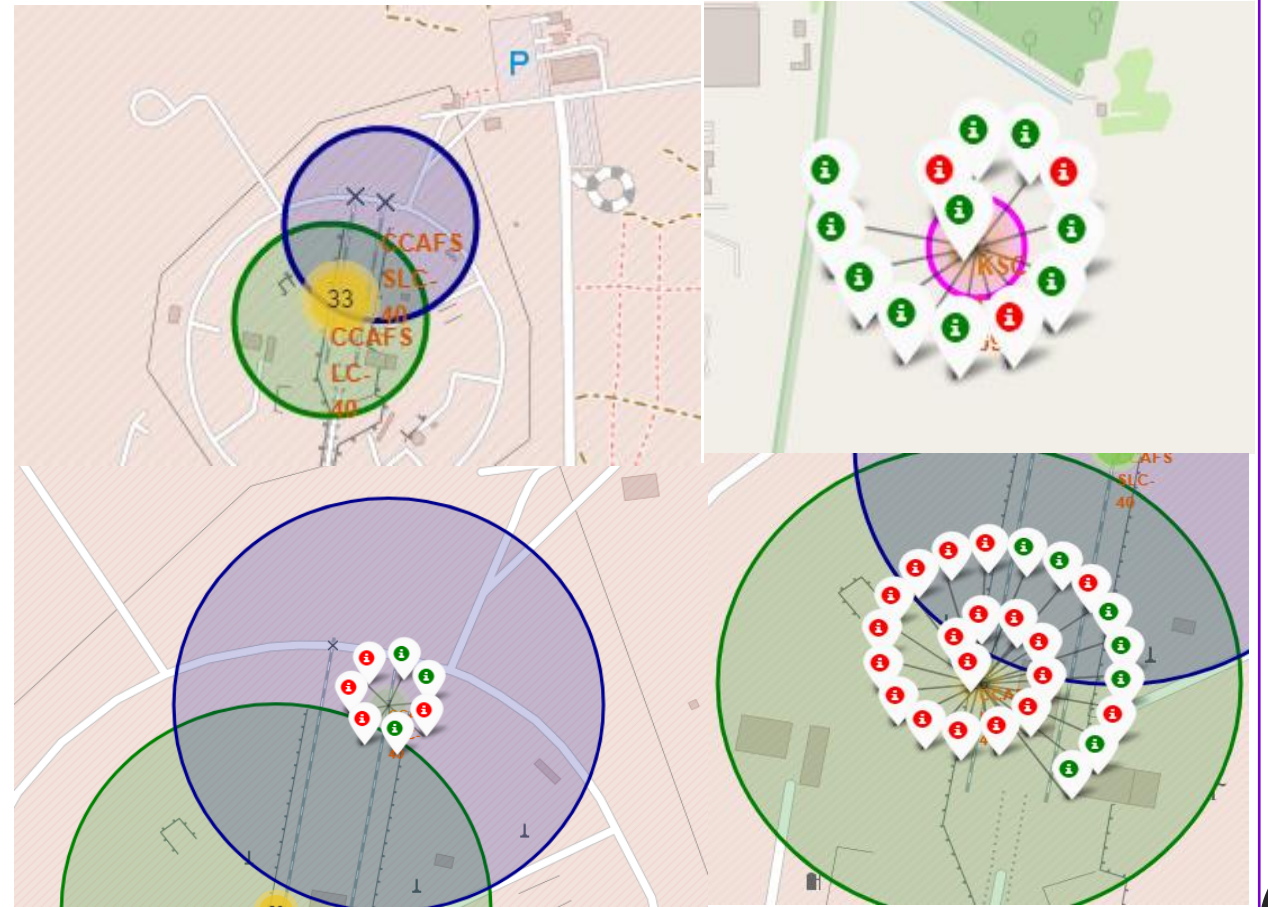
Launch sites

The **red** mark shows failure in launch and the **green** mark shows successful launches

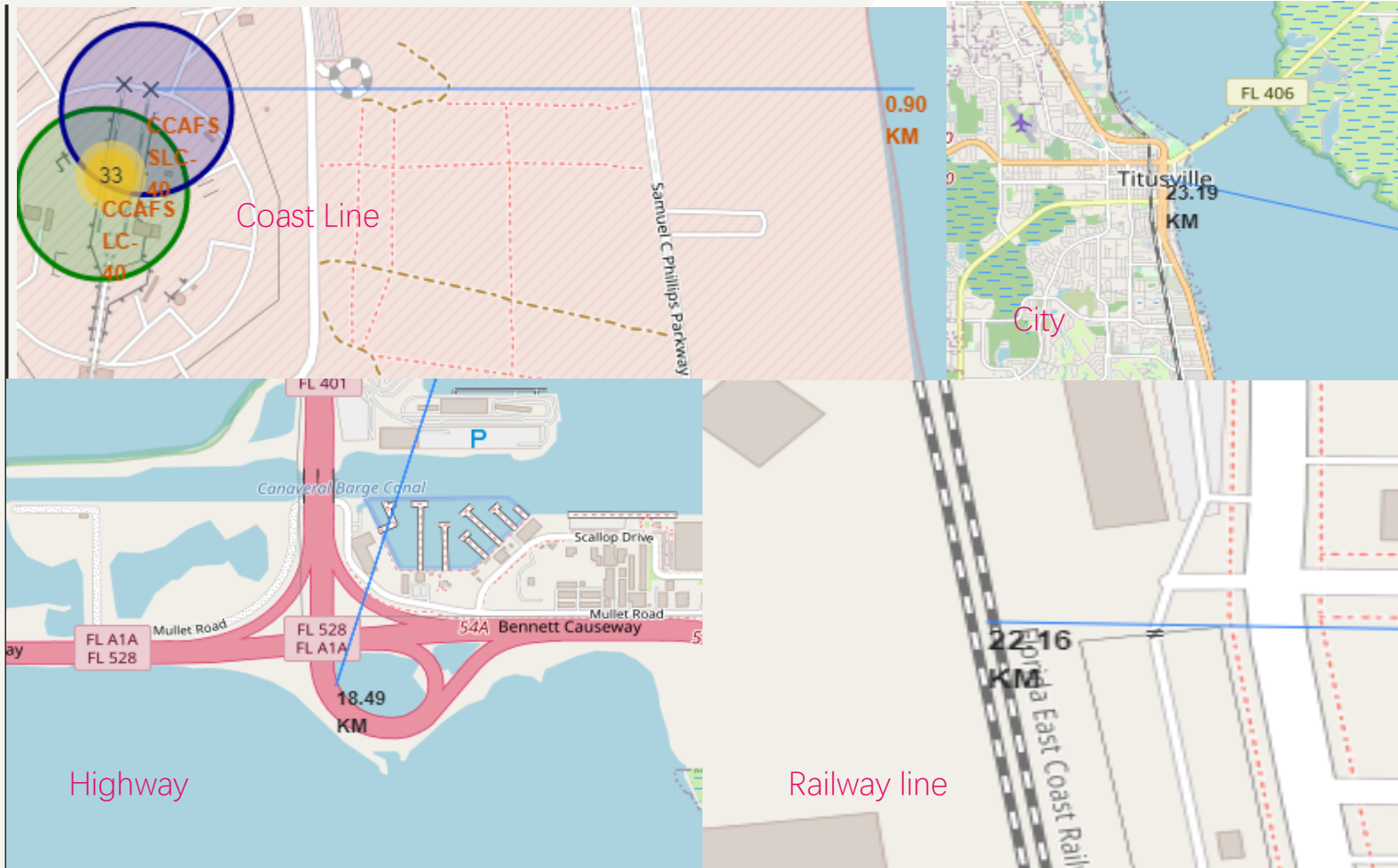


California Site

Florida Sites



Launch sites and distances from landmarks



The sites are not in close proximity To railways, highways and coastline And they are at a safe distance from cities.



05 - Plotly Dashboard

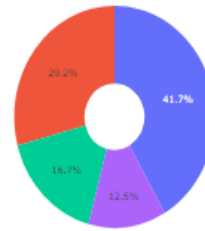


Plotly Dash Dashboard

SpaceX Launch Records Dashboard

All Sites

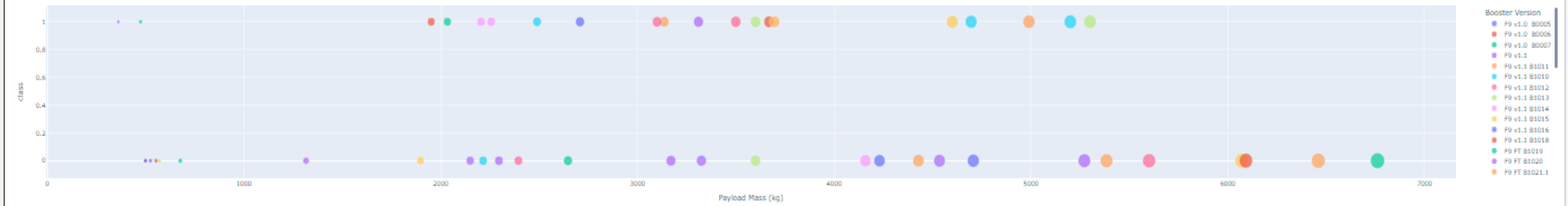
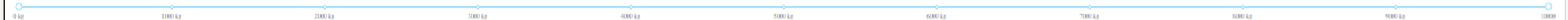
Total Success Launches By all sites



🌐 📄

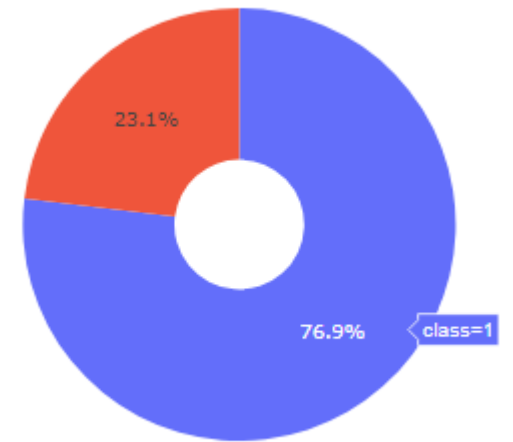
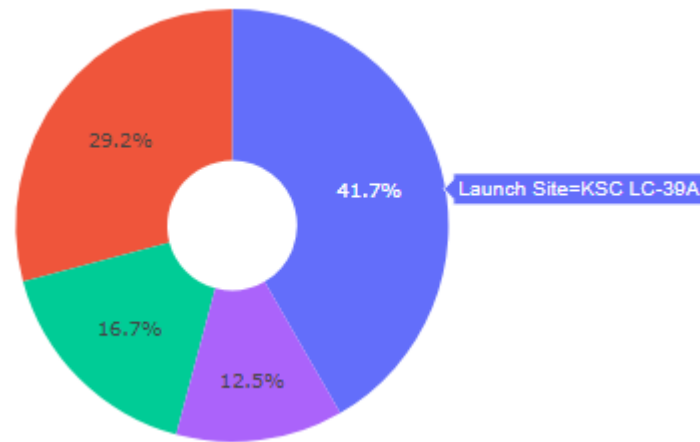
■ KSC LC-39A
■ CCAFS LC-40
■ VAFB SLC-4E
■ CCAFS SLC-40

Payload range (Kg):



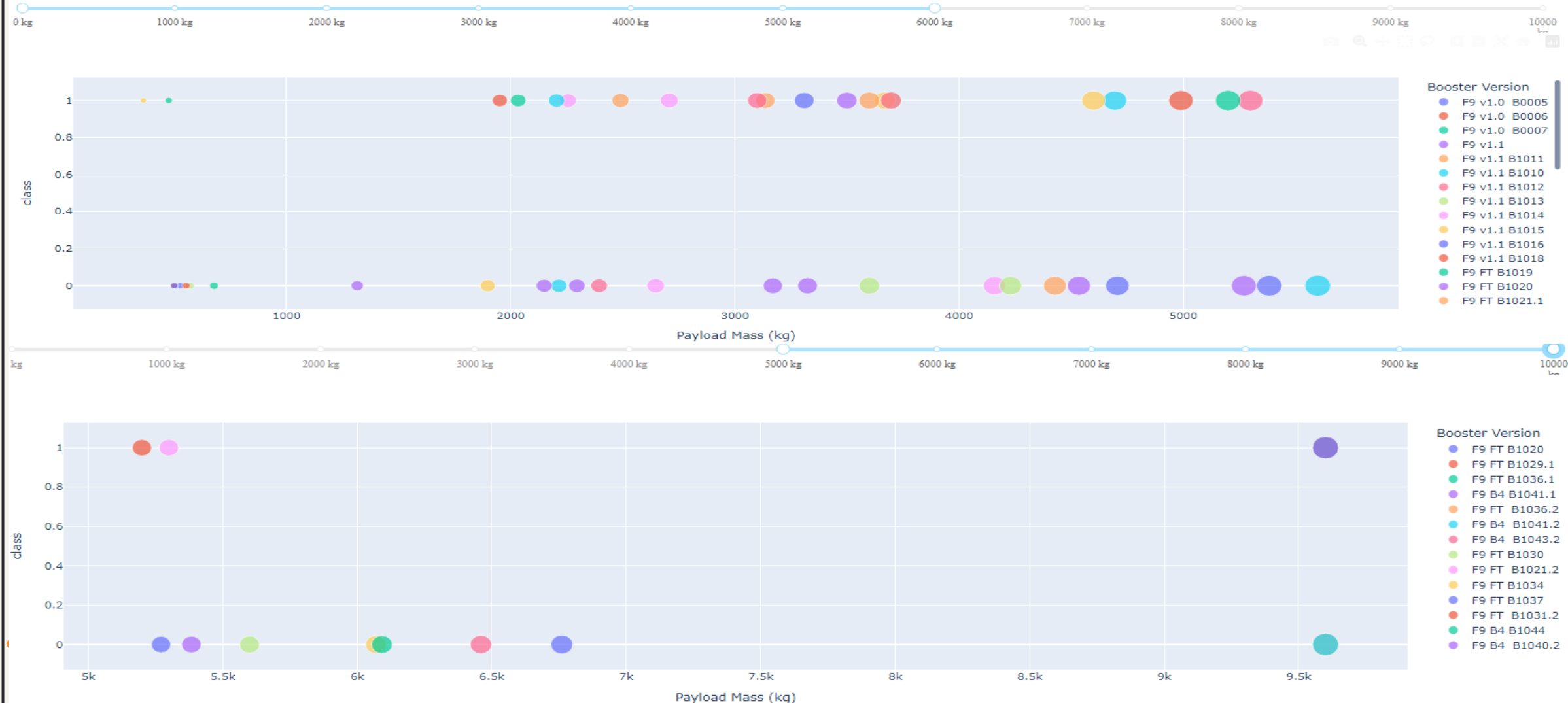
Pie Charts

Out of All launch sites, KSC LC_39A has the most success ratio 76.9% success



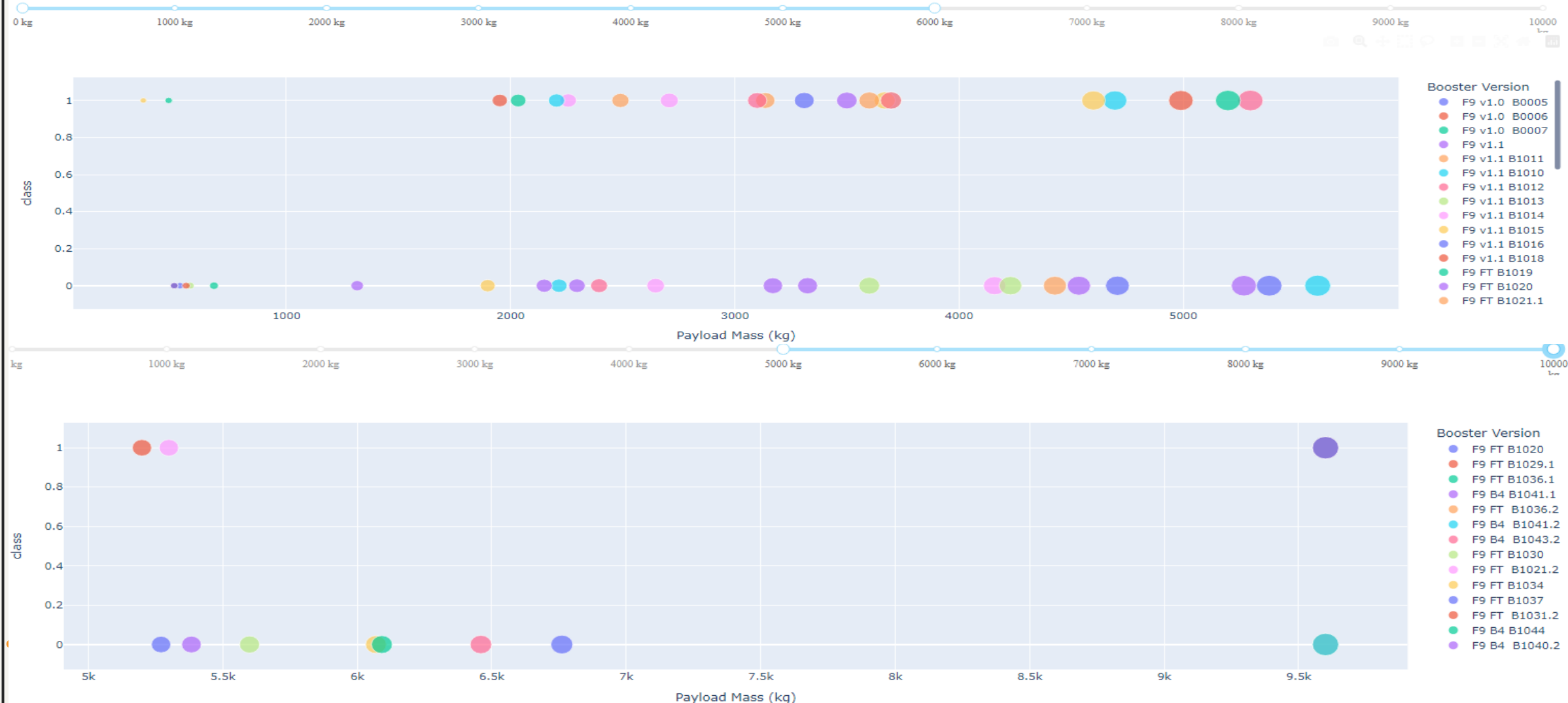
Payload mass and success rate

It was noted that higher(5000-10000) payload mass had a lower success rate in all boosters as compared to lower payload mass(1000-6000)

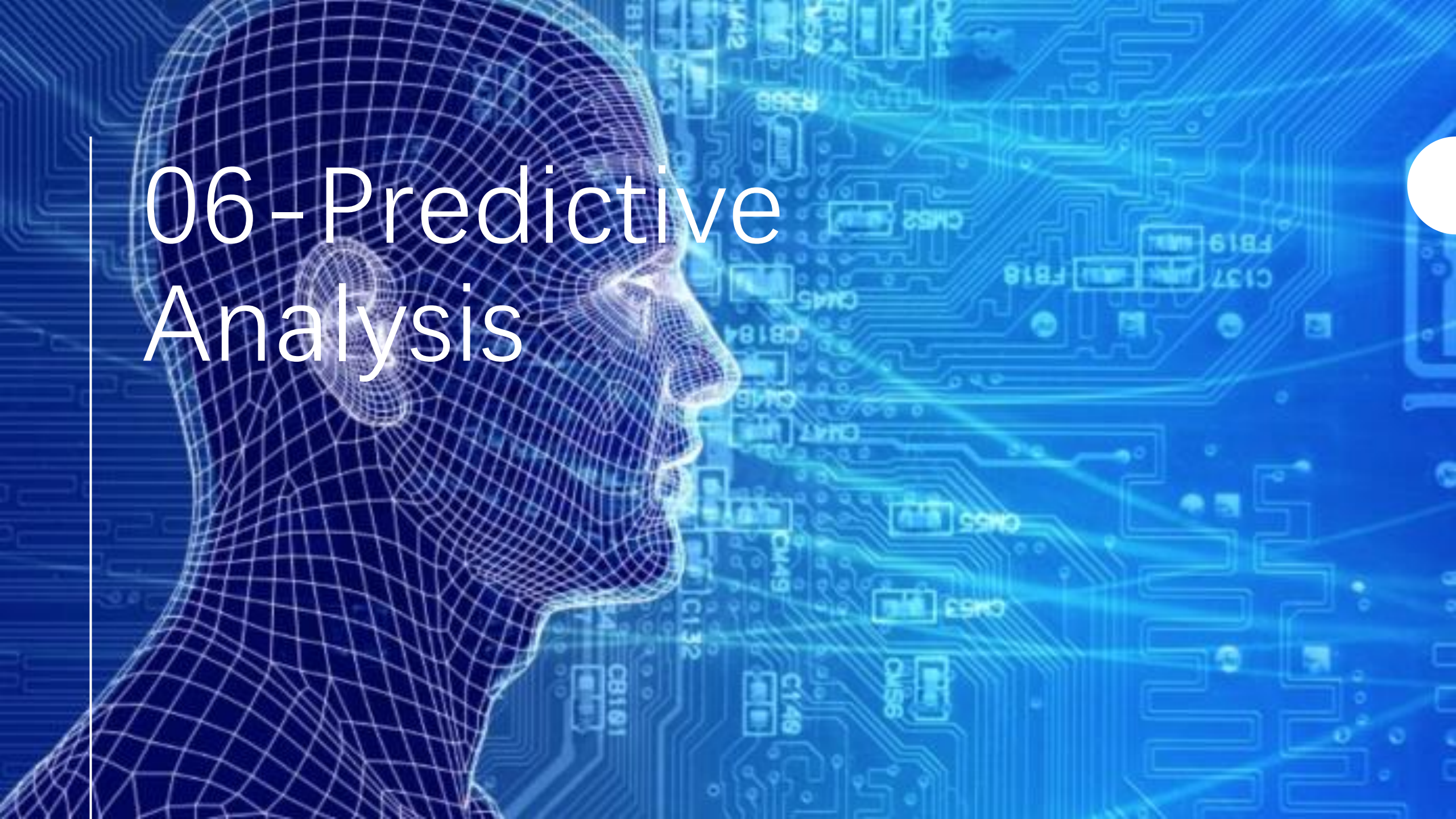


Payload mass and success rate

It was noted that higher(5000-10000) payload mass had a lower success rate in all boosters as compared to lower payload mass(1000-6000)



06 - Predictive Analysis



Predictive Analysis with GridSearchCv

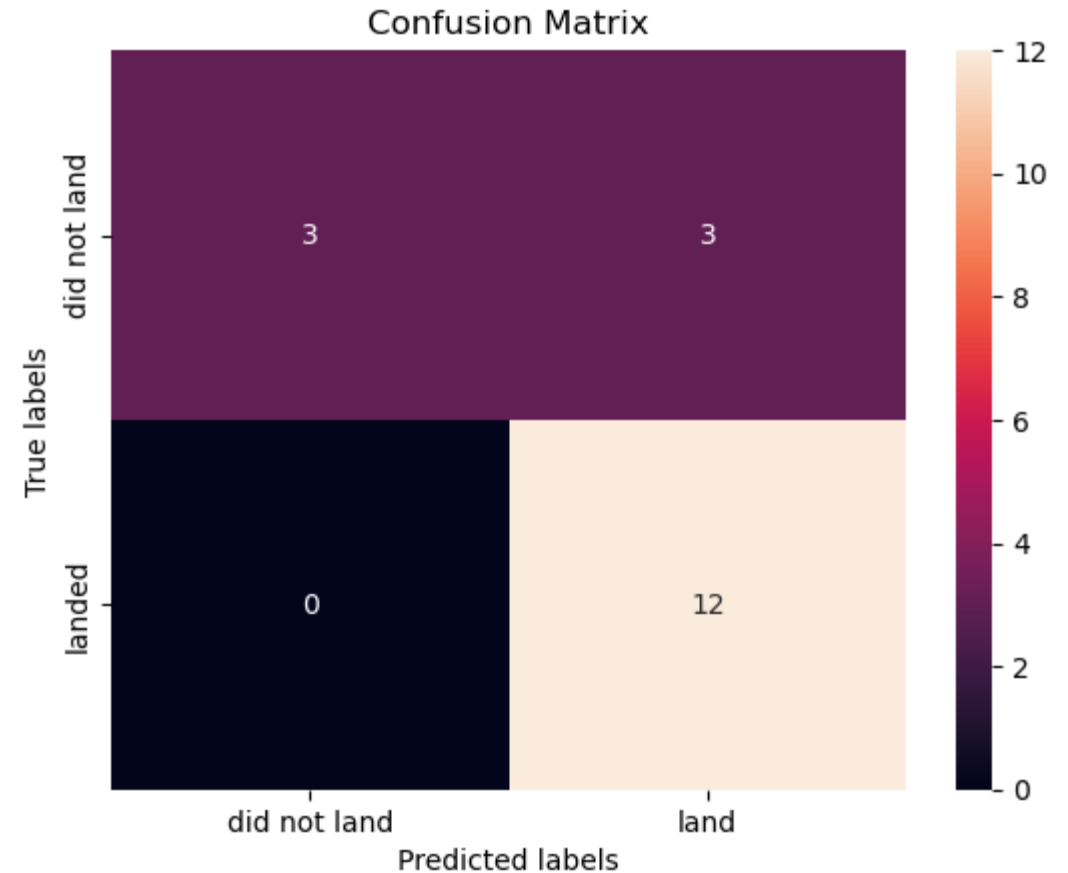
- Data were separated into target variables and dependent variables and standardized using a standard scaler by Scikitlearn, then this data set was split into testing and training sets with a test size of 20% of the total data.
- Then ML models were created using GridSearchCv object with CV=10 and parameters for each algorithm and scoring was set to measure the accuracy of the model.
- Confusion matrix was used for predicted values and test values for each model to Display the number of False positives, True negatives, False negatives, and True negatives.
- The Models used in this analysis were a Decision Tree classifier, K-nearest Neighbor, Logistic regression, and Support Vector Machine.
- Each GridSearchCv object for each model gave us with the tuned hyperparameters and the best score for that model with said parameters.



Heatmap

For all models, the results of the confusion matrix were the same which tells us that the model had 12 True predictions about `success(Class 1)` and 0 False predictions.

3 True predictions about the `failure(Class 0)` and 3 False predictions about the failure.



Hyperparameters and best scores

Logistic Regression:

tuned hyperparameters :(best parameters) {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}

accuracy: 0.8464285714285713

Support Vector machine:

tuned hyperparameters :(best parameters) {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'} accuracy: 0.8482142857142856

Decision Tree Classifier:

tuned hyperparameters :(best parameters) {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}

accuracy: 0.8732142857142856



Hyperparameters and best scores

K-Nearest Neighbor:

tuned hyperparameters :(best parameters) {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}

accuracy : 0.8482142857142858

From all the models' decision tree classifier had the highest score of accuracy hence this algorithm was chosen for the project

	Models	Scores
0	Descision Tree Classifier	0.873214
1	K-Nearest Neighbor	0.848214
2	Support vector Machine	0.848214
3	Logistic Regression	0.846429



Conclusions

We may deduce that:

- The more the number of flights at a launch site, the higher the success rate for a launch site.
- From 2013 through 2020, the success rate rose for successful launches.
- The most successful orbits were ES-L1, GEO, HEO, SSO, and VLEO.
- All launch sites were at a safe distance from landmarks.
- The launch of KSC LC-39A was the most successful of any facility.
- The best machine learning algorithm for this project is the Decision tree classifier.



FIN

Thank You