IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

Executive Summary

- Summary of methodologies

1) Business Understanding and its requirements
2) Data Collection using API of SpaceX
3) Wikipedia web scraping information using BeautifulSoup
4) Preliminary Data Visualization of features using plots and histograms
5) Data Wrangling/Cleanup
6) Exploratory Data Analysis using SQL
7) Exploratory Data Analysis using Data Viz
8) Folium map analytics
9) Machine Learning Predictions using various models

- Summary of all results

1. Exploratory data analysis results
2. Interactive analytics demo in screenshots
3. Predictive analysis results

Introduction

- Project background and context

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, much of the savings is because SpaceX can reuse the first stage. The vital requirement of this project is predicting if the first stage of the SpaceX Falcon 9 rocket will land successfully.

- Problems you want to find answers

1) What are the best features in terms of launch sites, booster versions, orbit etc. that can be chosen to increase the chances of Stage 1 landing successfully

2) How each feature influence the success rate of the Stage 1 landing.

3) The relationship between features i.e correlation.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The API endpoints were gathered for launches etc. and using python requests GET method, the data was collected
  - The Wikipedia data was scraped using Beautiful Soup package
- Perform data wrangling
  - Removing the unnecessary features
  - Converting Categorical variables into numerical using one-hot encoding.
  - Filling the null values in dataset using mean method.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Data is split into train and test, train data is used to fit the model and once the model inputs the train data and trains on it, the test data is used as an input and model predicts output. The predicted output and actual output are used to evaluate the model using techniques like Rsquare, F-1 score, LogLoss, Accuracy, Confusion Matrix etc.

Data Collection

- Describe how data sets were collected.

The API of SpaceX store the data in the form of json format. Using requests and get method, the data is received, normalized and response body is converted to pandas dataframe and fed to data wrangling phase

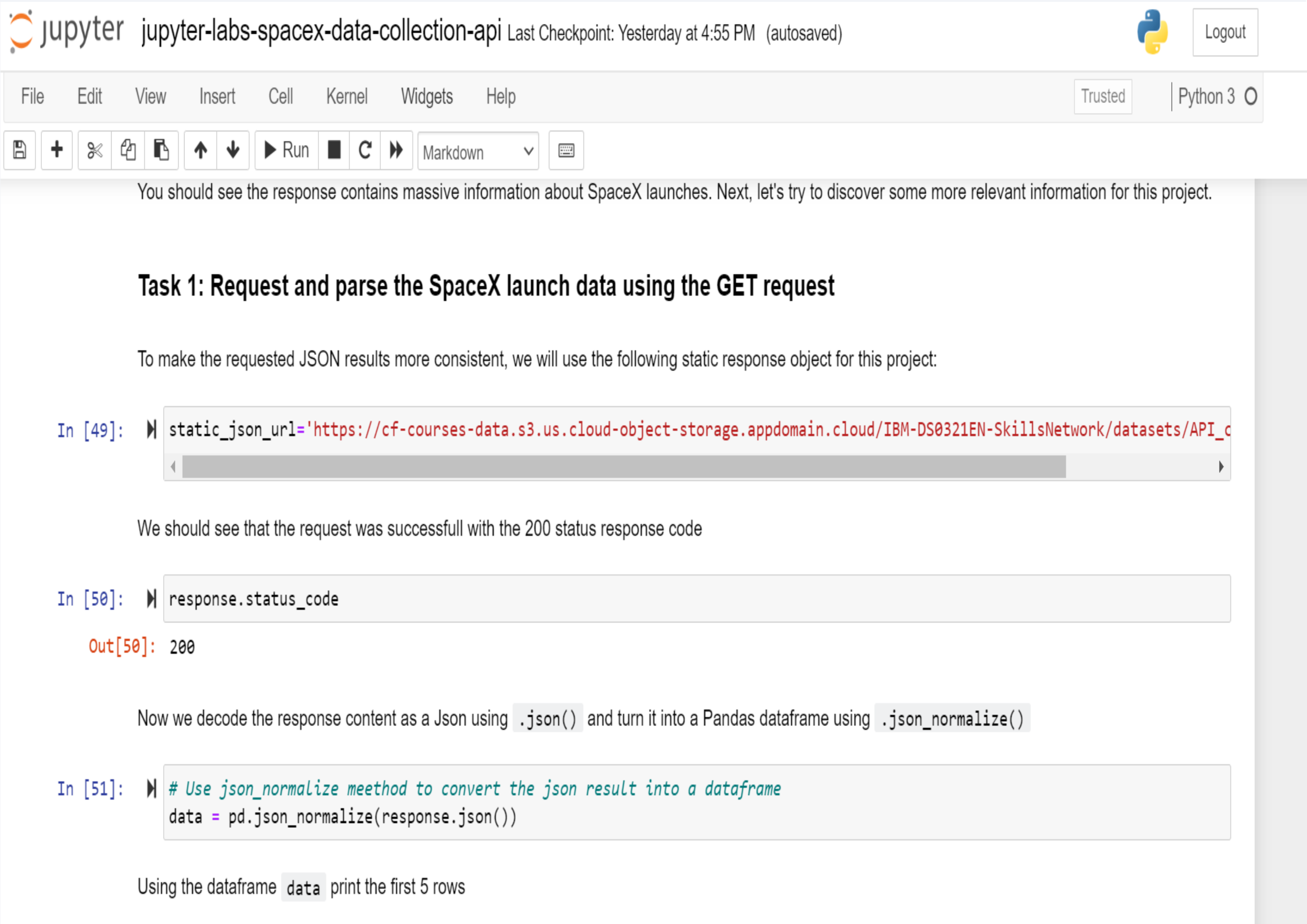
Data from Wikipedia is scraped using requests, BeautifulSoup and the key data points are extracted by using data in html tags.

The missing data was managed by filling in values like mean of the column etc.

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- https://github.com/ShridharMashalkar/helloWorld/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- https://github.com/ShridharMashalkar/helloWorld/blob/main/jupyter-labs-webscraping.ipynb

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.

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

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

In [8]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content)
```

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

the end of this lab

In [13]:
```
# 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')
```

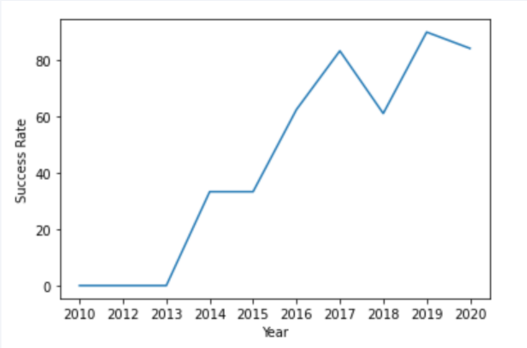
Starting from the third table is our target table contains the actual launch records.

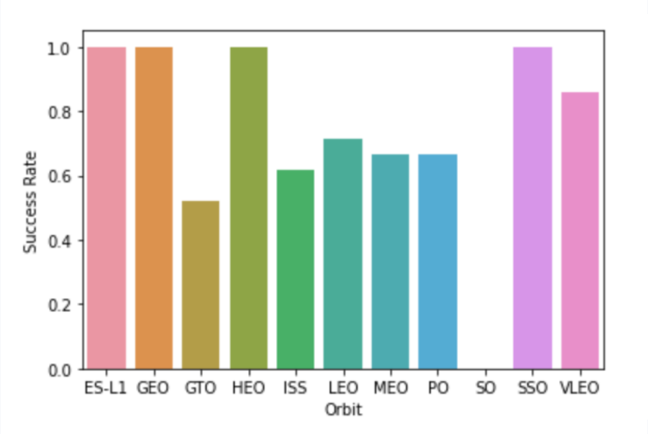
Data Wrangling

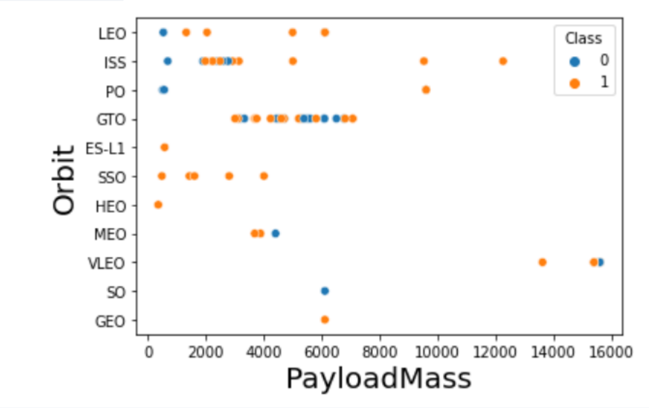
- Describe how data were processed
- The data wrangling included
- cleaning the data by filling missing values,
- feature engineering to remove features which are not required.
- Converting categorical into numerical using one-hot encoding
- Converting the datatype into required format like integer,string etc.
- https://github.com/ShridharMashalkar/helloWorld/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Bar Plot – we use this for categorical data
- Catplot – shows wide range of input in plot
- Pie chart
- Waffle Chart
- Scatter plot
- Line plot
- https://github.com/ShridharMashalkar/helloWorld/blob/main/jupyter-labs-eda-dataviz.ipynb

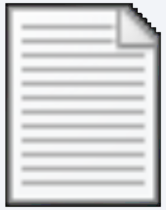






EDA with SQL

- Please refer to this txt file for queries(double click on it to open)



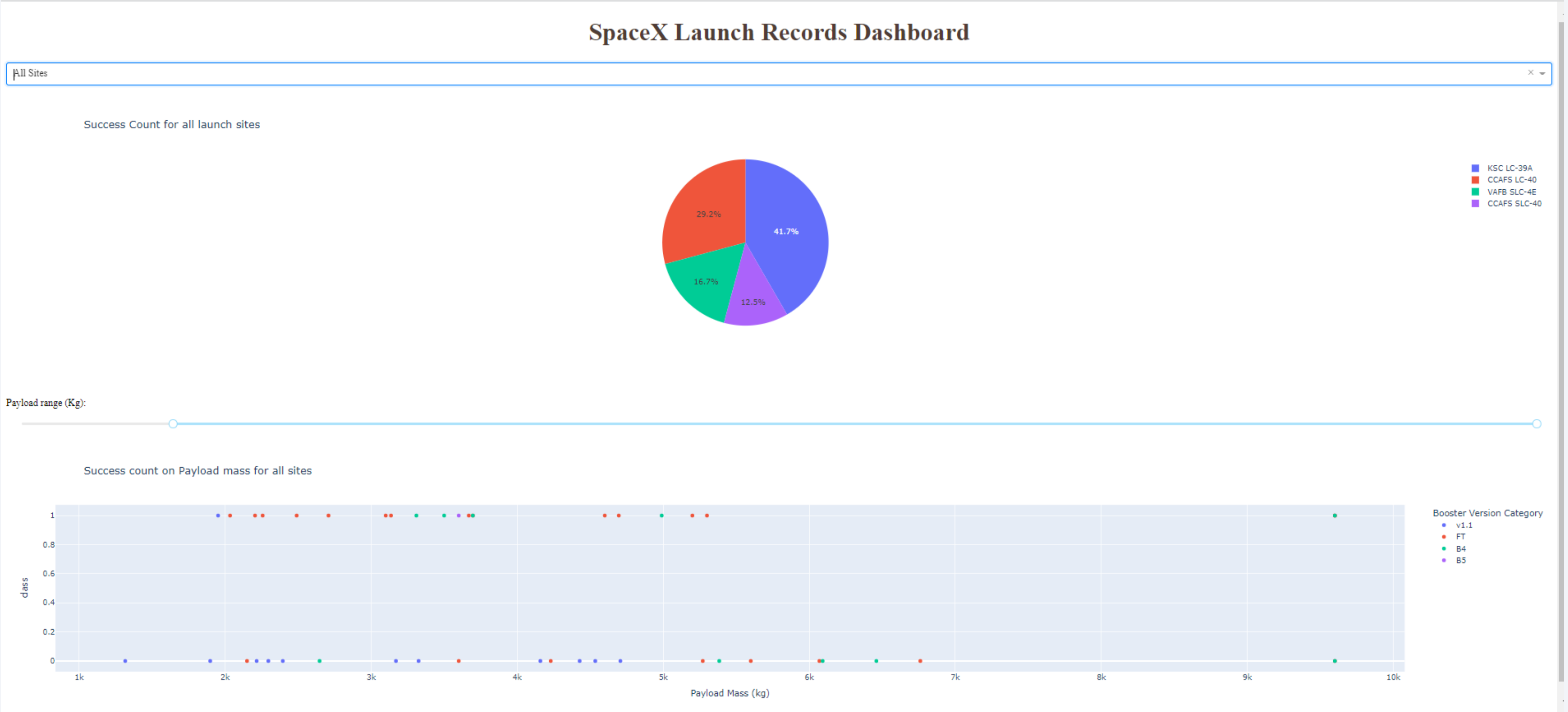
queries.txt

- https://github.com/ShridharMashalkar/helloWorld/blob/main/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- We used markers to mark points on map using latitude and longitude co ordinates
- Circles were used to create an easy identifiable points
- Lines are used to measure the distance from launch sites to other points
- These things helps us to clearly track the geolocations on map and also using colored markers to indicate the number of failures and successful launch.
- https://github.com/ShridharMashalkar/helloWorld/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash



- https://github.com/ShridharMashalkar/helloWorld/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Data is split into train and test, train data is used to fit the model and once the model inputs the train data and trains on it,
- the test data is used as an input and model predicts output.
- The predicted output and actual output are used to evaluate the model using techniques like Rsquare, F-1 score, LogLoss, Accuracy, Confusion Matrix etc.
- The models used are Logistic regression, SVM, KNN, Decision Trees

- https://github.com/ShridharMashalkar/helloWorld/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

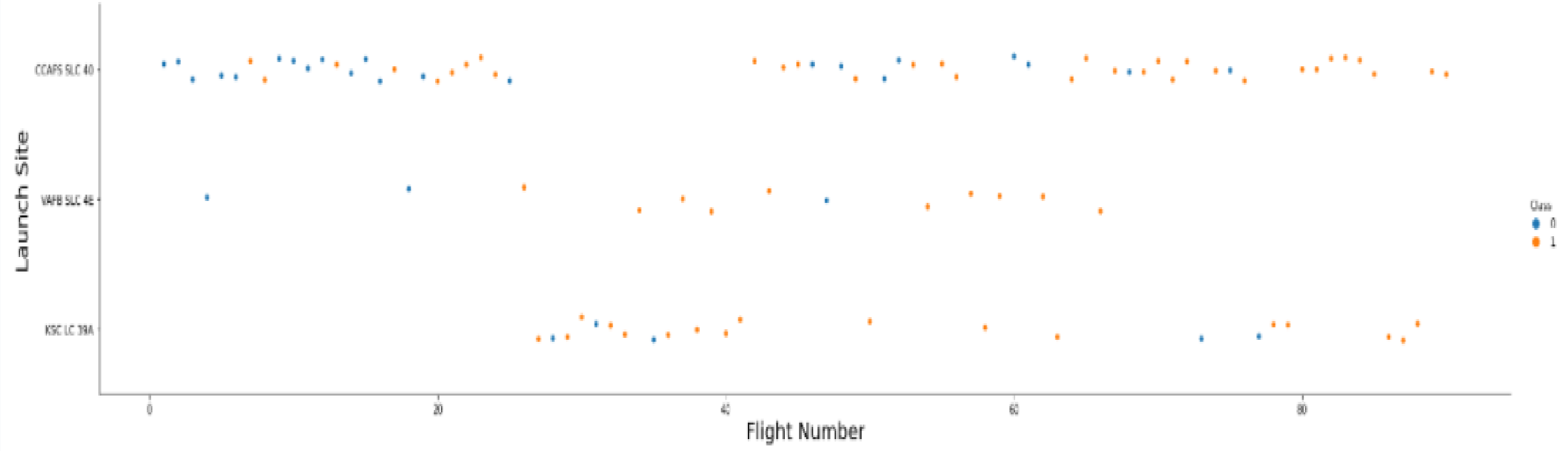
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

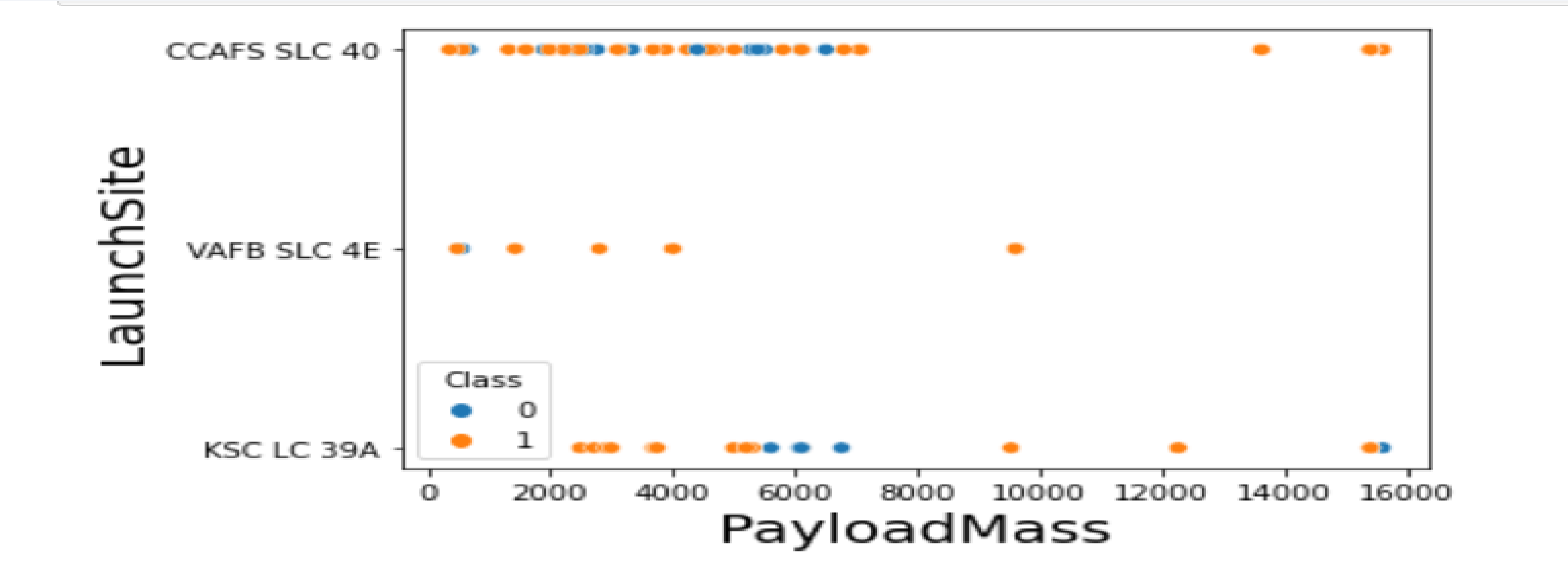
Insights drawn from EDA

Flight Number vs. Launch Site

As you see, with increase in flight number the success is increasing,
meaning there is an improvement in the launches



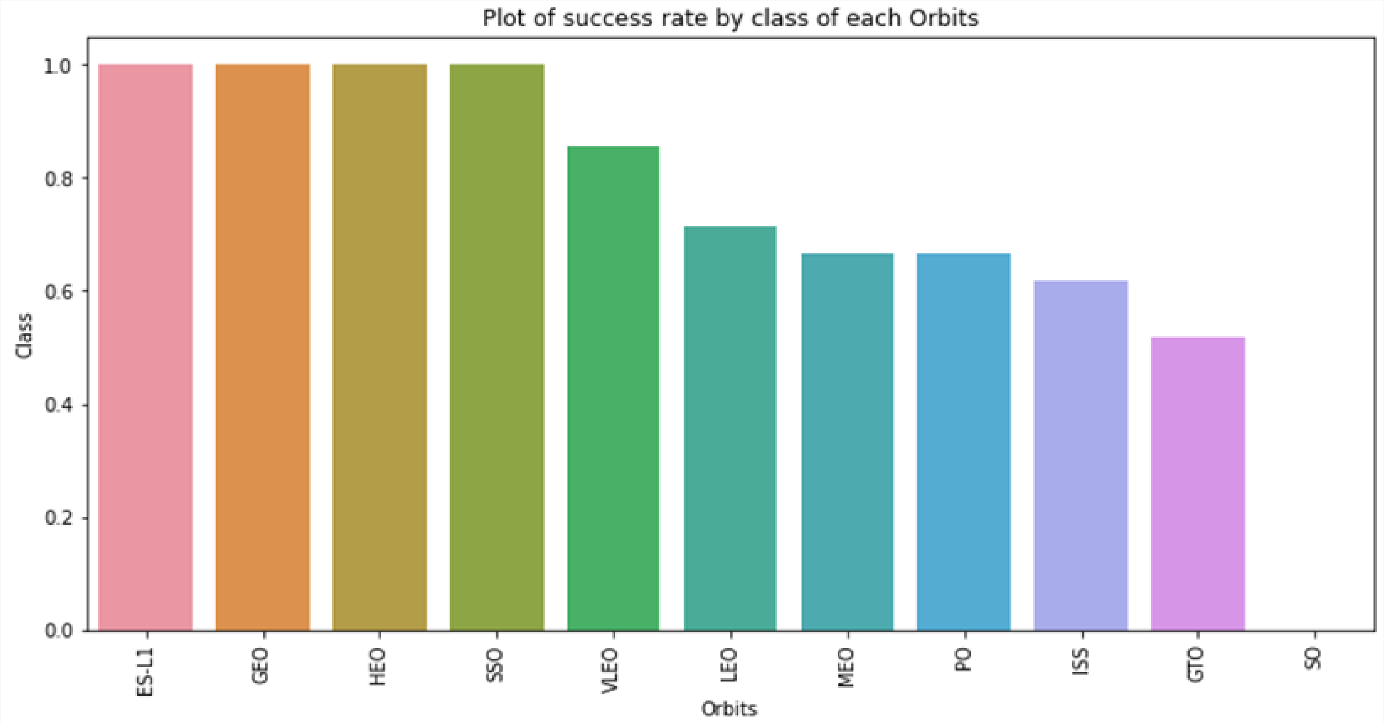
Payload vs. Launch Site



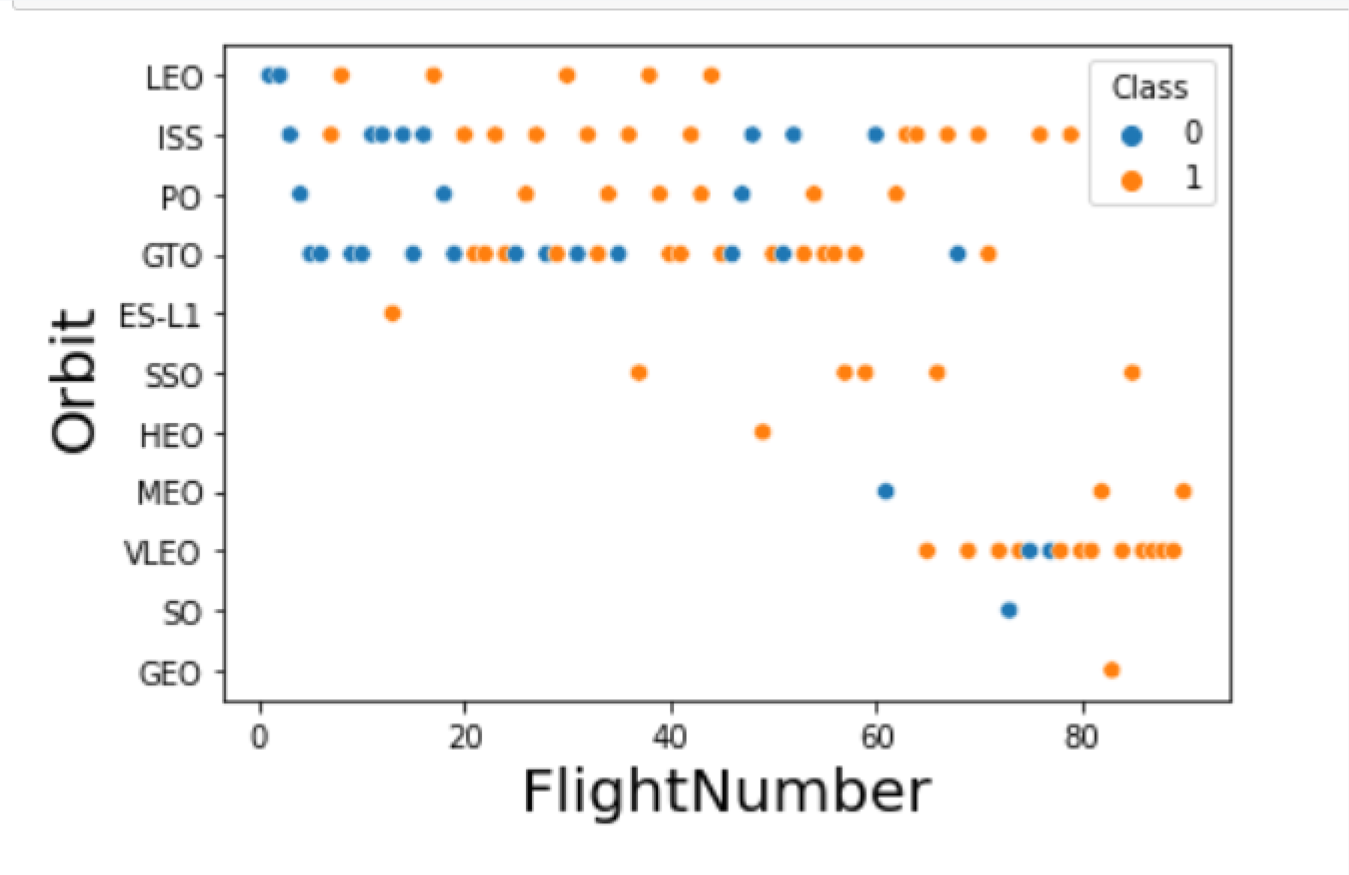
When payload mass is highest, the success rate is close to 100 percent

Success Rate vs. Orbit Type

- ESL1 GEO HEO SSO are the orbits where success rates are the highest.

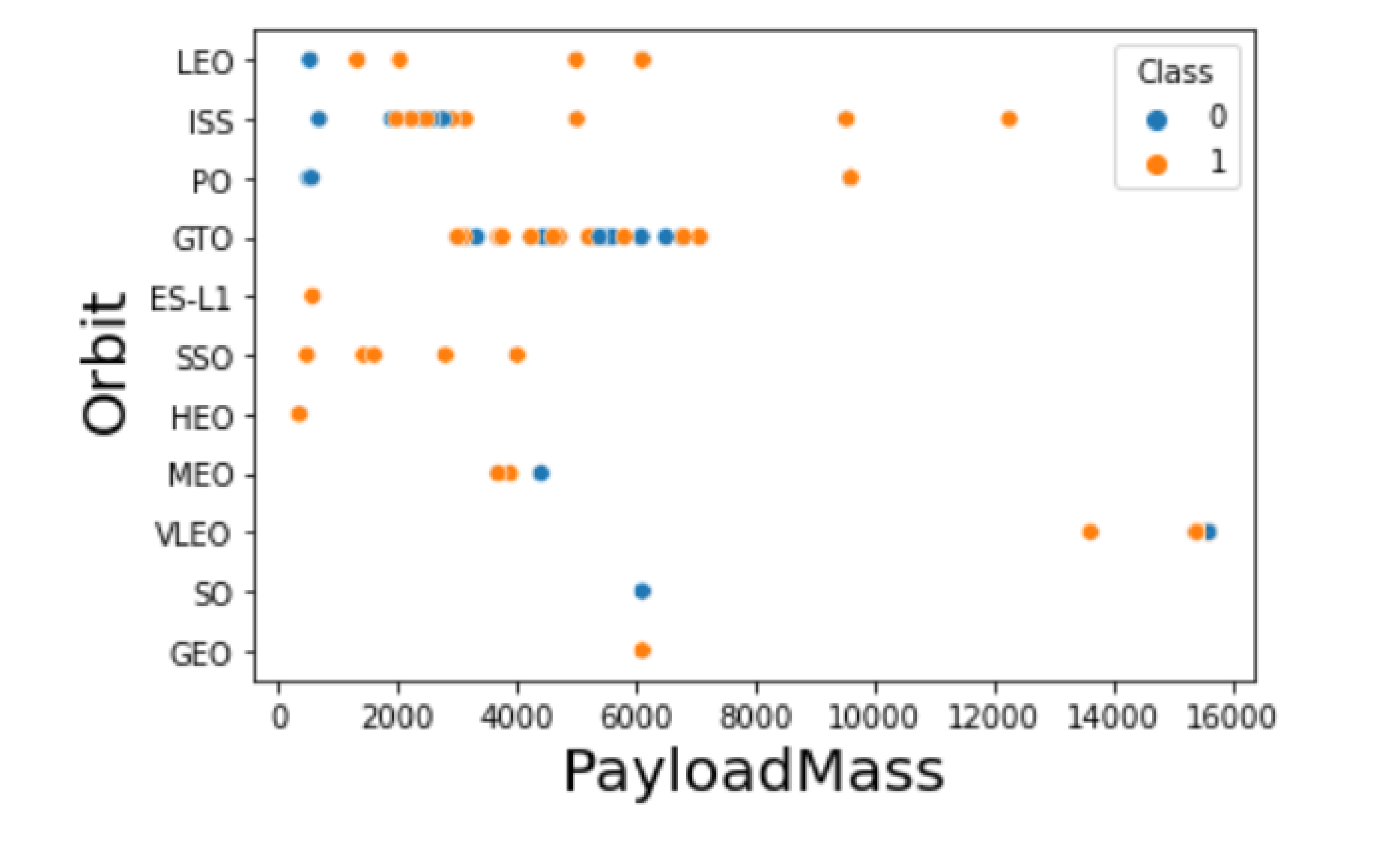


Flight Number vs. Orbit Type



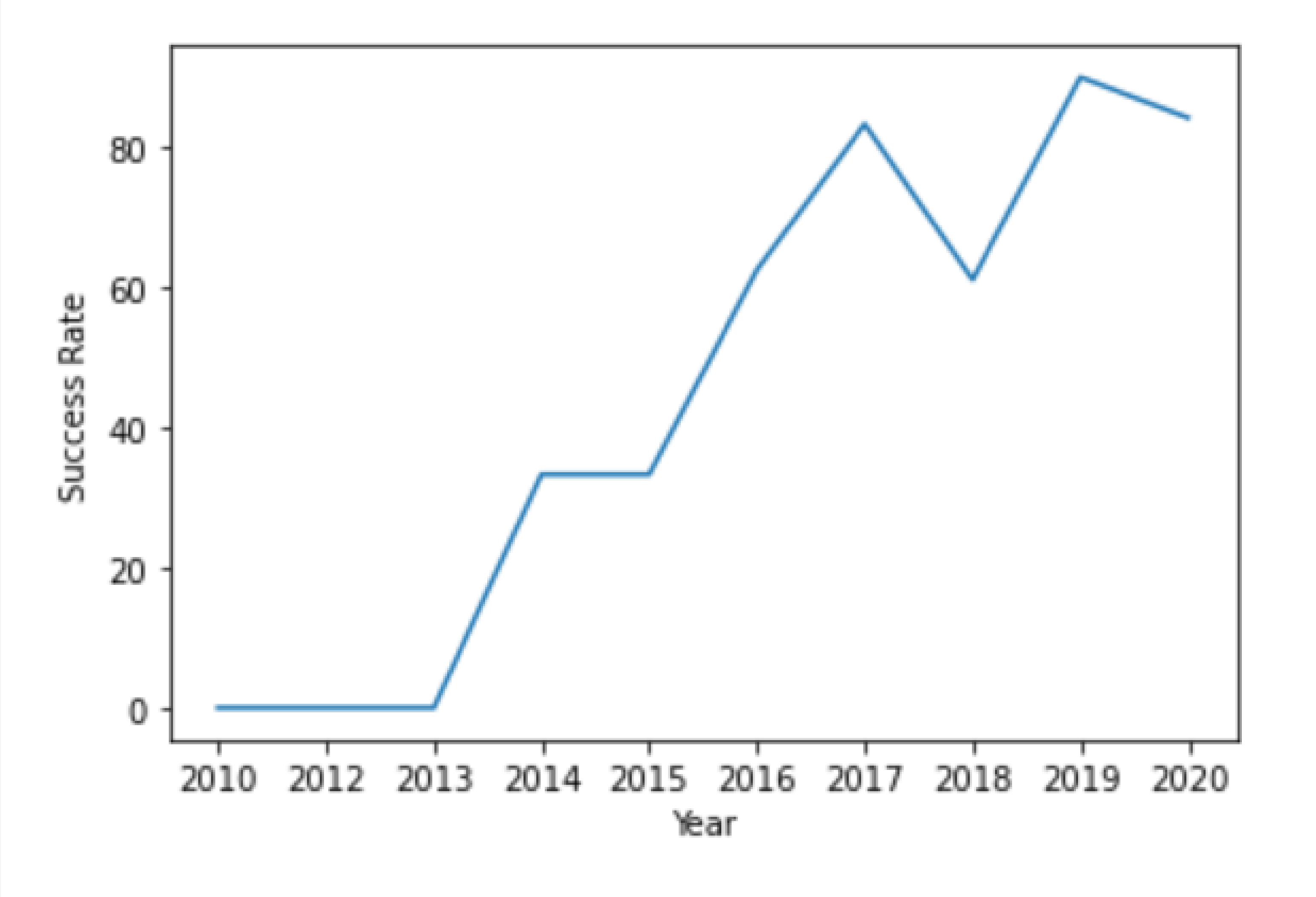
ES-L1, SSO, HEO and GEO orbits have 100 percent success rate

Payload vs. Orbit Type



Orbits ES-L1,SSO,HEO have been fully successful irrespective of the Payload Mass

Launch Success Yearly Trend



The trend says from 2013, the success rate has increased meaning there is a positive improvement

All Launch Site Names

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

In [6]:

```
%%sql

SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

```
 * ibm_db_sa://vfl77824:***@125f9f61-9715-46f9-9399-c8177b21803b.
Done.
```

Out[6]:

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

Launch Site Names Begin with 'CCA'

Task 2

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

In [8]:

```
%%sql

SELECT BOOSTER_VERSION,LAUNCH_SITE,PAYLOAD,ORBIT,CUSTOMER
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5
```

```
 * ibm_db_sa://vfl77824:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.
Done.
```

Out[8]:

booster_version	launch_site	payload	orbit	customer
F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	LEO	SpaceX
F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	LEO (ISS)	NASA (COTS) NRO
F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	LEO (ISS)	NASA (COTS)
F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	LEO (ISS)	NASA (CRS)
F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	LEO (ISS)	NASA (CRS)

Total Payload Mass

Task 3

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

In [9]:

```
%%sql

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

```
 * ibm_db_sa://vfl77824:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[9]:

1
45596

Average Payload Mass by F9 v1.1

Task 4

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

In [10]:

```
%%sql

SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://vf177824:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.
Done.
```

Out[10]:

1
2928

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 [11]:

```
%%sql

SELECT MIN(DATE)
FROM SPACEXTBL
WHERE UCASE(MISSION_OUTCOME) = 'SUCCESS' AND
LANDING__OUTCOME LIKE 'Success (ground pad)'
```

```
 * ibm_db_sa://vfl77824:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[11]:

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

In [13]:

```
%%sql

SELECT BOOSTER_VERSION,PAYLOAD
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

```
 * ibm_db_sa://vfl77824:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

Out[13]:

booster_version	payload
F9 v1.1	AsiaSat 8
F9 v1.1 B1011	AsiaSat 6
F9 v1.1 B1014	ABS-3A Eutelsat 115 West B
F9 v1.1 B1016	Turkmen 52 / MonacoSAT
F9 FT B1020	SES-9
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1030	EchoStar 23

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [16]:

```
%%sql

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

```
 * ibm_db_sa://vf177824:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.
Done.
```

Out[16]:

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

Boosters Carried Maximum Payload

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

In [18]:

```
%%sql

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

```
 * ibm_db_sa://vfl77824:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.a
Done.
```

Out[18]:

booster_version	payload
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0

2015 Launch Records

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

In [20]:

```
%%sql

SELECT BOOSTER_VERSION,PAYLOAD,LAUNCH_SITE,LANDING__OUTCOME,DATE
FROM SPACEXTBL
WHERE UCASE(LANDING__OUTCOME) LIKE '%FAILURE%'
AND YEAR(DATE)=2015
```

```
 * ibm_db_sa://vfl77824:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.
Done.
```

Out[20]:

booster_version	payload	launch_site	landing__outcome	DATE
F9 v1.1 B1012	SpaceX CRS-5	CCAFS LC-40	Failure (drone ship)	2015-01-10
F9 v1.1 B1015	SpaceX CRS-6	CCAFS LC-40	Failure (drone ship)	2015-04-14

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

```
SELECT COUNT(LANDING__OUTCOME),LANDING__OUTCOME
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
```

```
 * ibm_db_sa://vfl77824:***@125f9f61-9715-46f9-9399-c8177b21803b.c
Done.
```

Out[30]:

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

Section 3

Launch Sites Proximities Analysis

Launch Sites on Map

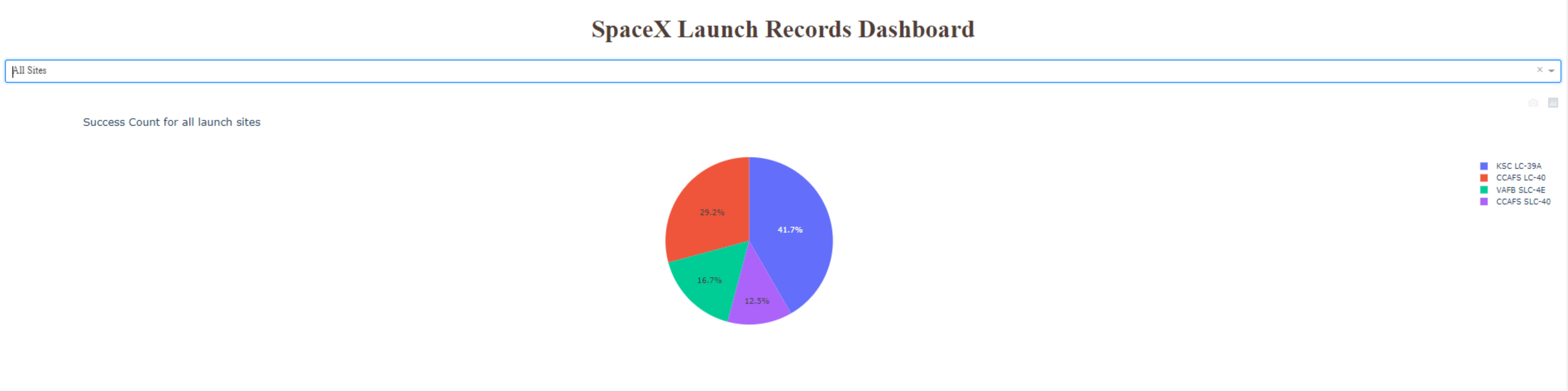
Coloured Outcome

Lines Map

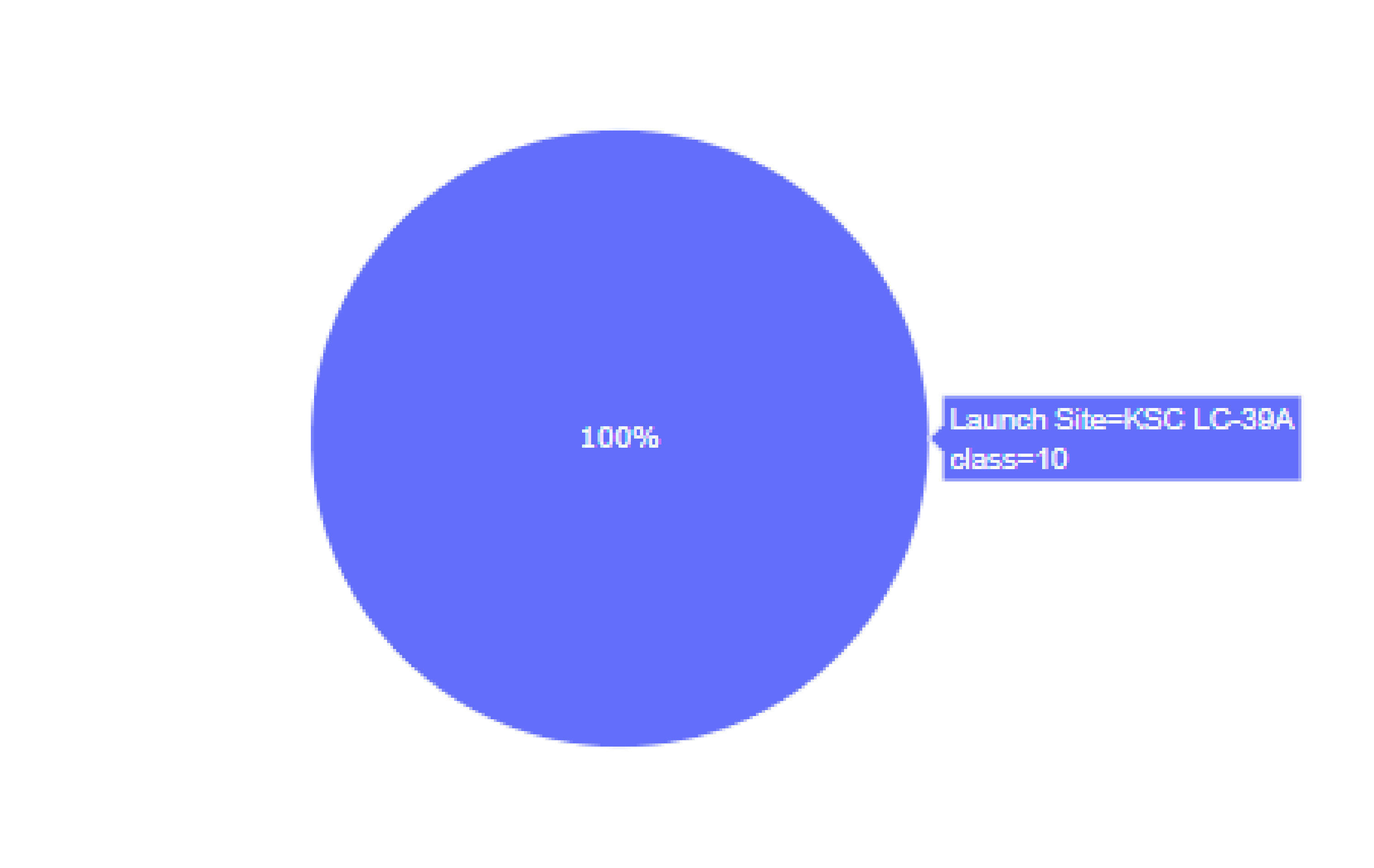
Section 4

Build a Dashboard with Plotly Dash

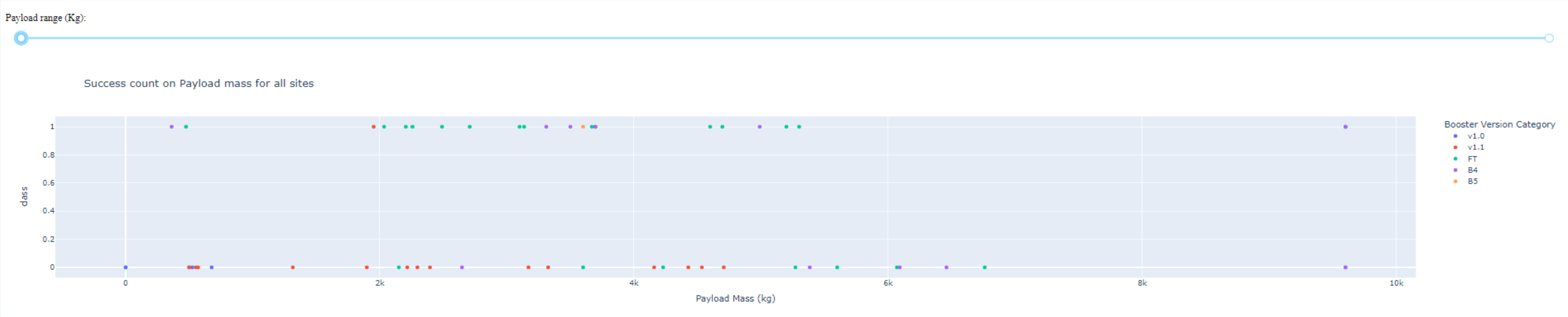
Pie Chart of launch sites success rate



Launch site with Highest success rate



Payload vs Launch outcome for difference sliders

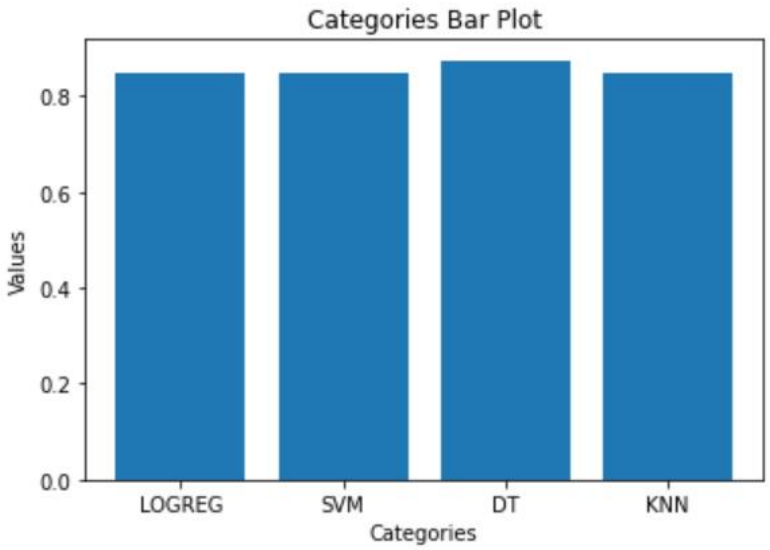


Section 5

Predictive Analysis (Classification)

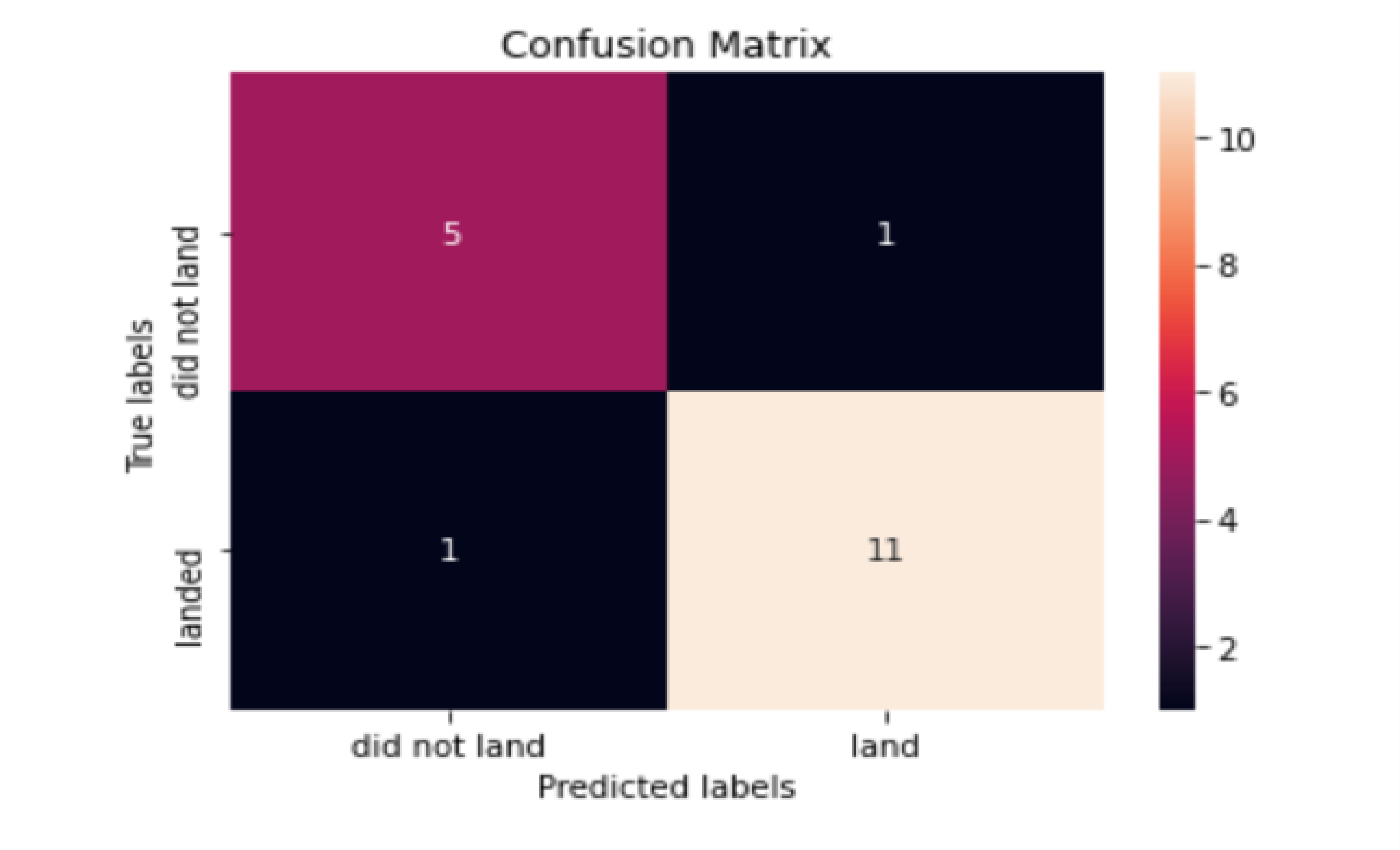
Classification Accuracy

In [42]:
```
yvalues = [logreg_cv.best_score_,svm_cv.best_score_,tree_cv.best_score_,knn_cv.best_score_]
xvalues = ['LOGREG','SVM','DT','KNN']
plt.bar(xvalues,yvalues)
plt.xlabel('Categories')
plt.ylabel("Values")
plt.title('Categories Bar Plot')
plt.show()
```



Model with highest accuracy is Decision trees

Confusion Matrix



Conclusions

- Launch of Rockets success rate started to increase from 2013 to 2020.
- ES-L1, GEO, HEO, SSO, VLEO orbits have been most success rate.
- KSC LC-39A is most successful launch site of any sites.
- The Decision tree classifier is the best machine learning algorithm with highest accuracy score
- If the payload mass is high, the chances of successful landing increases.

Thank you!