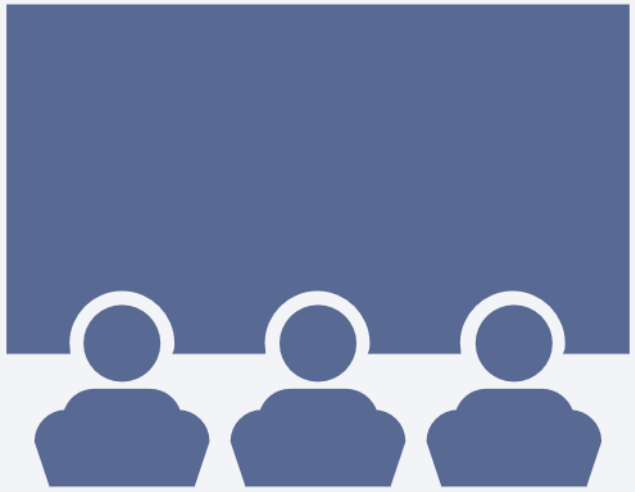
Winning Space Race with Data Science

Gujjari Lakshmi Priyanka

2nd November 2021

Outline



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

Executive Summary

- **Summary of methodologies**
  - Data Collection via API, SQL and Web Scraping
  - Data Wrangling and Analysis
  - Interactive Maps with Folium
  - Prediction Analysis using Classification Models
- **Summary of all results**
  - Data Analysis along with Interactive Visualizations
  - Best model for Prediction Analysis

Introduction

Project background and context

- **SpaceX** is perhaps the most successful space company.
- The reason for SpaceX's success is its inexpensive rocket launches.
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each.
  - The reason for its cost saving is its **reusable first stage**.

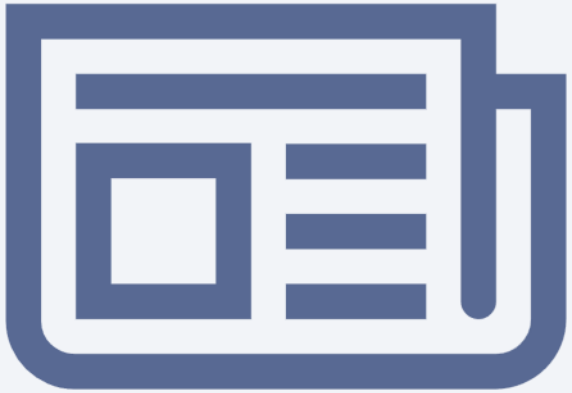
- In this project, we are working for a new rocket company, **SpaceY** that would like to compete with SpaceX.
- Our job as a data scientist is to determine the price of each launch.
- Instead of using rocket science, we are going to build a machine learning model and use public information to predict if SpaceX will reuse the first stage.

- **Problems you want to find answers**
  - What factors play a role in the rocket's successful landing?
  - The effect of the various rocket variables on the outcome
  - Conditions that aid SpaceX to achieve best results.

Section 1

Methodology

Methodology



- **Data Collection Methodology**
  - Using SpaceX REST API
  - Web Scrapping from Wikipedia
- **Data Wrangling**
  - Dropping irrelevant columns
  - One-hot encoding data fields for Model building
- **Exploratory Data Analysis**
  - Using SQL
  - With Data Visualizations
- **Interactive Visual Analytics**
  - Using Folium and Plotly Dash Visualizations
- **Predictive Analysis**
  - Build and evaluate different Classification models

Data Collection – SpaceX API

GitHub: Data Collection

Request rocket launch data from SpaceX API

Apply custom functions to clean data

Filter data frame to include only Falcon9 launches

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
data_falcon9 = launch_df[launch_df['BoosterVersion']!= 'Falcon 1']
```

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1	0	B0003	-80.577366	28.561857
2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1	0	B0005	-80.577366	28.561857
3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1	0	B0007	-80.577366	28.561857
4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1	0	B1003	-120.610829	34.632093
5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1	0	B1004	-80.577366	28.561857

Data Collection – Web Scraping

GitHub: Data Collection with Web Scraping

- Getting Response from HTML
- Creating Beautiful Soup Object
- Finding Tables
- Getting columns
- Creation of dictionary and appending data
- Converting dictionary to dataframe
- Dataframe to .CSV

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
soup = BeautifulSoup(response.content, 'html.parser')
```

```
html_tables = soup.find_all('table')
```

```
first_launch_table = html_tables[2]
```

```
column_names = []

cols = first_launch_table.find_all('th')
for l in cols:
    name = extract_column_from_header(l)
    if (name != None) and (len(name) > 0):
        column_names.append(name)
```

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date
1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010
2	CCAFS	Dragon	0	LEO	NASA (COTS)\nNRO	Success	F9 v1.0B0004.1	Failure	8 December 2010
3	CCAFS	Dragon	525 kg	LEO	NASA (COTS)	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012
4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA (CRS)	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012
5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA (CRS)	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013

Data Wrangling

We are going to convert the mission launch outcomes into Training labels with 1 means booster landed successfully and 0 means unsuccessful landing.

GitHub: Data Wrangling

Calculate number of launches at each site

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

Calculate number and occurrence of each orbit

```
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
SO        1
ES-L1     1
GEO       1
```

Calculate number and occurrence of mission outcome per each orbit

```
landing_outcomes = df['Outcome'].value_counts()
dict(landing_outcomes)
```

```
{'True ASDS': 41,
 'None None': 19,
 'True RTLS': 14,
 'False ASDS': 6,
 'True Ocean': 5,
 'False Ocean': 2,
 'None ASDS': 2,
 'False RTLS': 1}
```

Create landing outcome label from outcome column

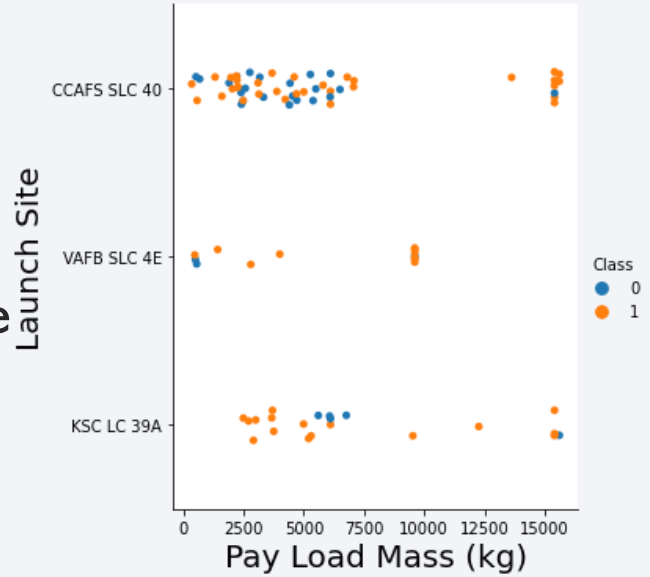
```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
df[['Class']].head(8)
```

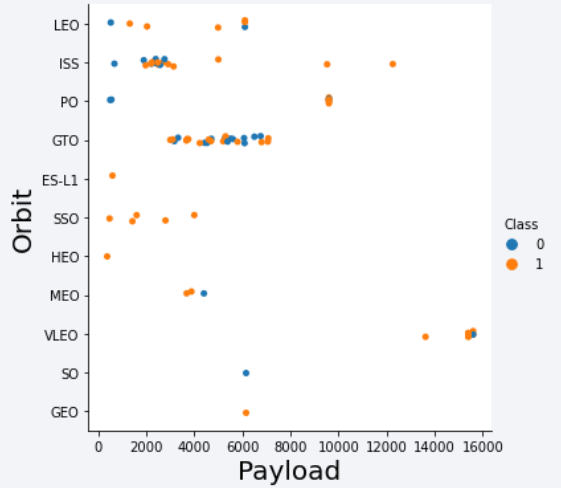
FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0

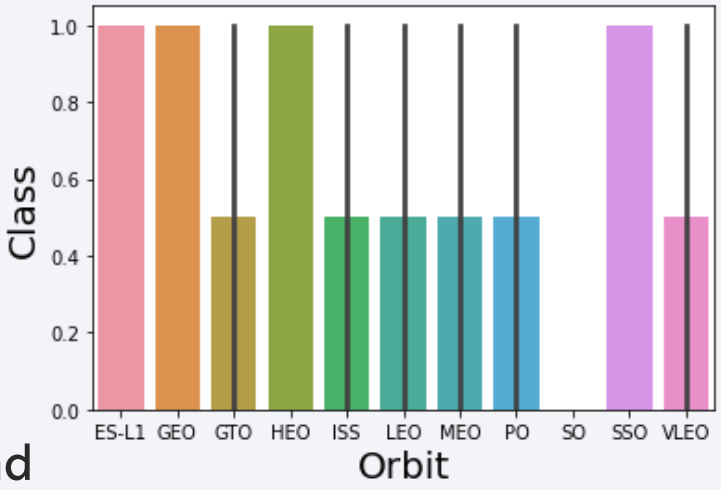
EDA with Data Visualization

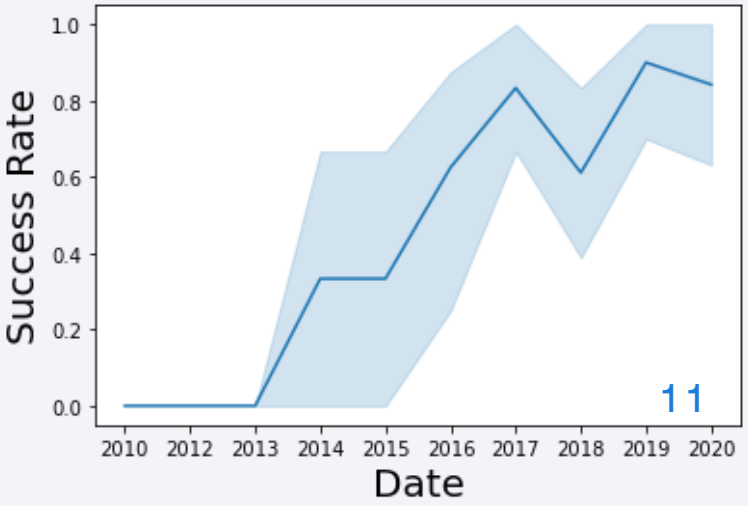
GitHub: EDA with Visualization

- **Scatter Plots Drawn**
  - Payload and Flight Number
  - Flight Number and Launch Site
  - Payload and Launch Site
  - Flight Number and Orbit Type
  - Payload and Orbit Type
- **Bar Graph Drawn**
  - Success Rate vs Orbit Type
- **Line Graph Drawn**
  - Launch Success Yearly Trend









EDA with SQL

Here we use the IBM's Db2 for Cloud, which is a fully managed SQL database provided as a service. GitHub: Data Analysis with SQL

Summary of the SQL queries performed

- 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
- 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

Build an Interactive Map with Folium

GitHub: Data Visualization with Folium

- Folium makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map.
- We used the latitude, longitude of each launch site and added a circle marker around each launch site with a label of its name.
- We also visualized the success and failure of the launches on each site using reg and green markers

MAP OBJECT	CODE	RESULT
MAP MARKER	folium.Marker(	Map object to make a mark on map
ICON MARKER	folium.Icon(	Create an icon on map
CIRCLE MARKER	folium.Circle(	Create a circle where marker is placed
POLYLINE	foluim.PolyLine(	Create a line between points
MARKER CLUSTER OBJECT	MarkerCluster()	To simplify a map with many markers at same point
ANT PATH	folium.plugins.AntPath(	Create an animated line between points

Build a Dashboard with Plotly Dash

Github: Dashboard

- Piechart showing the total success for all sites or by certain launch.
- Scatter Graph showing the correlation between success and payload for all sites or by certain launch site.

Predictive Analysis (Classification)

GitHub: ML Prediction Analysis

Building Model

- Load our feature engineered data into dataframe
- Transform it into Numpy arrays
- Standardize& transform data
- Split data into training and test datasets
- Check how many test samples have been created
- List down machine learning algorithms to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCV object and train the model

Evaluating Model

- Check accuracy of each model : Logistic Regression, SVM and Decision Tree Classifier
- Get best hyperparameters for each algorithm
- Plot confusion matrix

```
Y = data['Class'].to_numpy()
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
Y_test.shape
```

Finding Best Performing Classification model

- The model with the best accuracy score is the best Classification model

Results

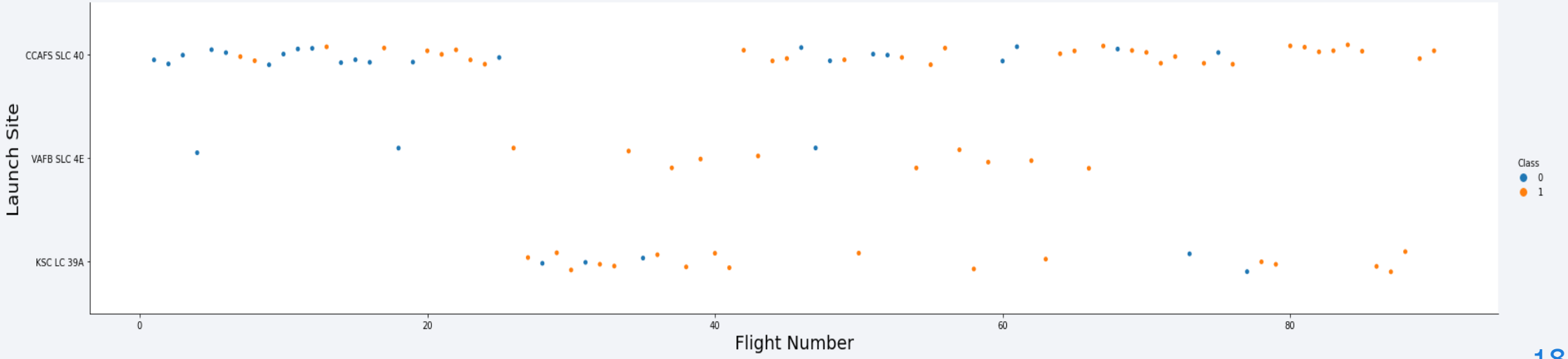
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

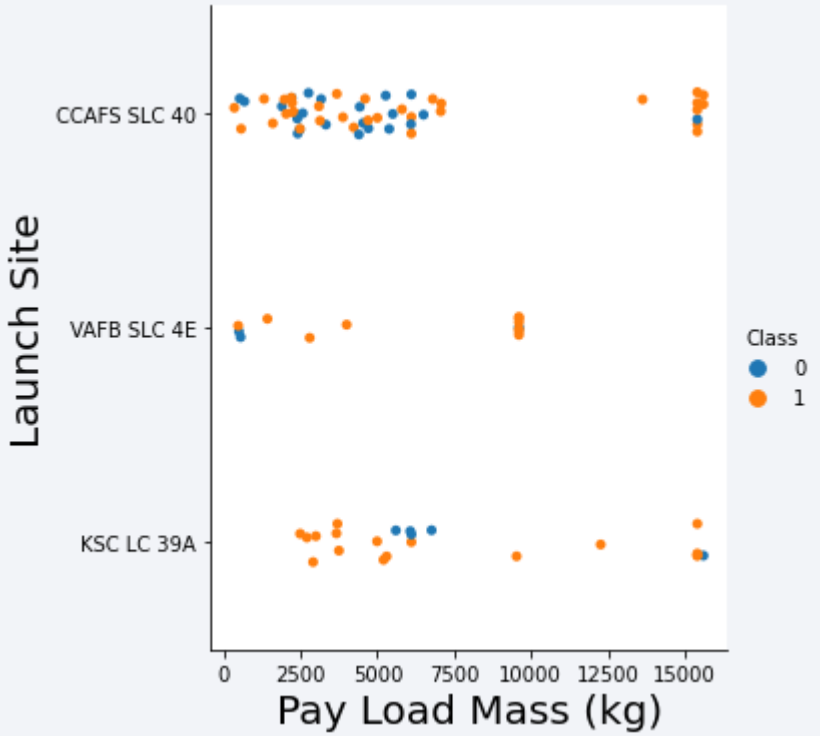
- Scatter plot of Flight Number vs. Launch Site
- With Higher flight numbers (greater than 30) the success rate for the rocket increases



Payload vs. Launch Site

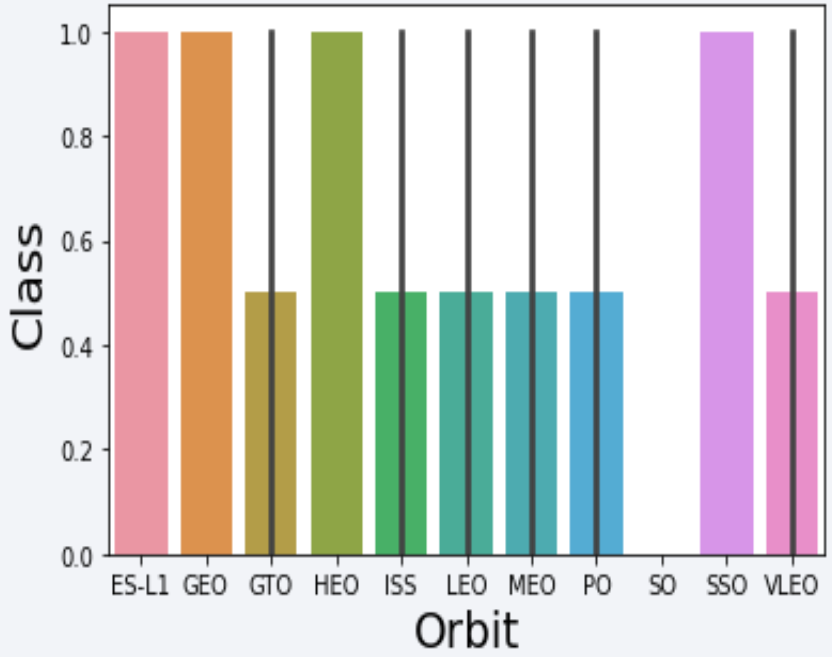
- Scatter plot of Payload vs. Launch Site

- With a greater payload (>7000kg) there is a greater scope for success of the launch.



Success Rate vs. Orbit Type

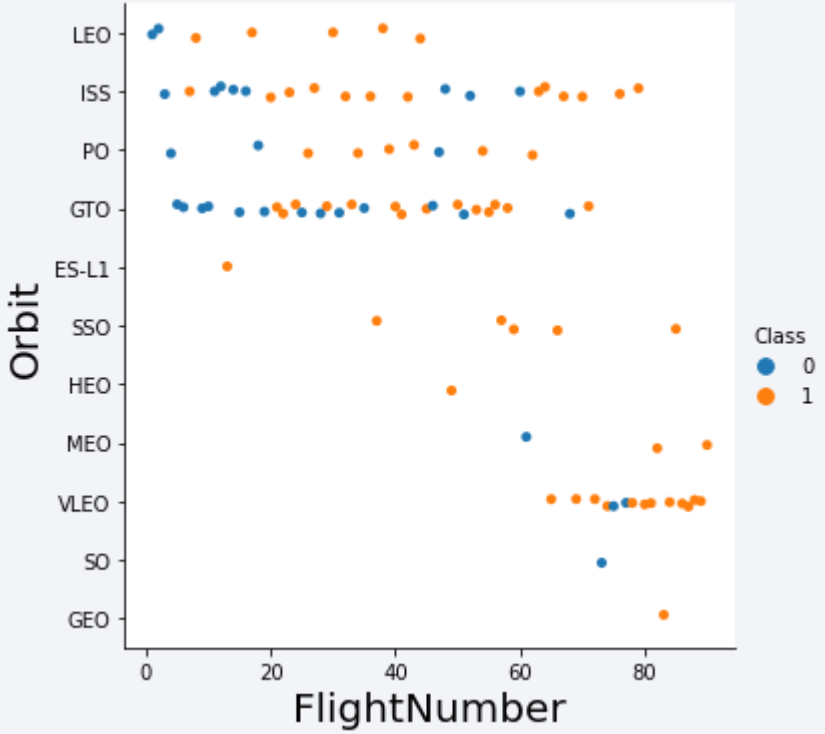
- Bar chart for the success rate of each orbit type
- ES-L1, GEO, HEO, SSO have the highest success rate.



Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type

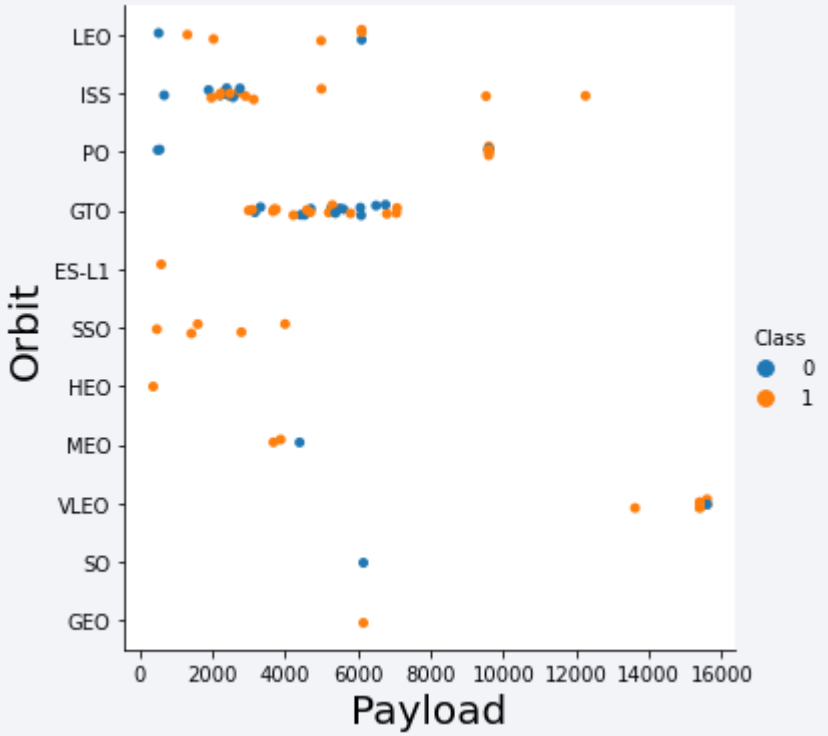
- We see that for LEO orbit success rate increases with the increase in Flight number
- On the other hand, there seems to be no relation between the Flight number and orbit type for GTO orbit



Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

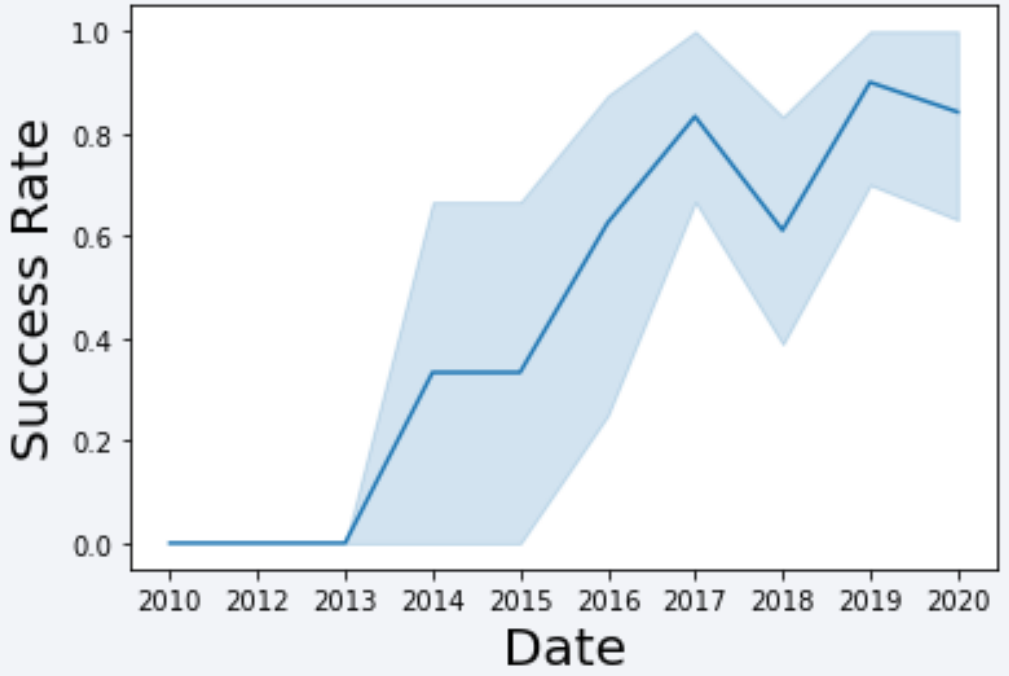
- We observe that heavy loads have a negative impact on MEO, GTO and VLEO orbits.
- Positive impact on ISS and LEO orbits



Launch Success Yearly Trend

- Line chart of yearly average success rate

- The success rate increased relatively since 2013 till 2020 although it took a slight dip in 2019.



All Launch Site Names

- Names of the Launch Sites
- Using the word DISTINCT will pull the unique values of a particular column in the table
- Here the column is `launch_site` and the table is `SPACEXD

```
%sql select distinct(LAUNCH_SITE) from SPACEXDATASET
```

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

```
%sql select * from SPACEXDATASET where LAUNCH_SITE like 'CCA%' limit 5
```

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

Total Payload Mass

- Total payload carried by boosters from NASA
- Using the function SUM calculates the total of the column and WHERE clause filters the data to meet the required condition.
- Here the column is `PAYLOAD_MASS_KG_ and the table is SPACEXDATASET while the clause is NASA

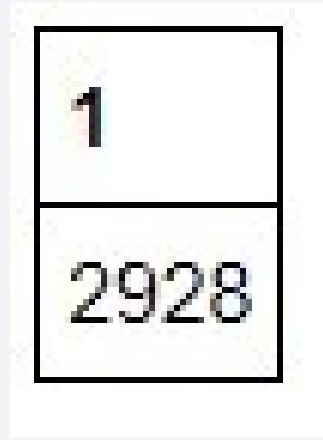
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXDATASET where CUSTOMER = 'NASA (CRS)'
```

1
45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1
- The function AVG calculates the average of the column and the where clause is used to filter the data according to the required condition.
- The column is PAYLOAD_MASS_KG_, table is SPACEXDATASET and the condition is BOOSTER_VERSION F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXDATASET where BOOSTER_VERSION = 'F9 v1.1'
```



First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad
- The function MIN is used to find the minimum of the column in a table and the WHERE clause is used to filter the data according to the given condition

```
%sql select min(DATE) from SPACEXDATASET where Landing__Outcome = 'Success (ground pad)'
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql select BOOSTER_VERSION from SPACEXDATASET where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and
PAYLOAD_MASS__KG_ < 6000
```

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

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes
- The function COUNT is used to numerically count the values according to the conditions satisfied in the WHERE clause

```
%sql select count(MISSION_OUTCOME) from SPACEXDATASET where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in fligh
t)'
```

1
100

Boosters Carried Maximum Payload

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

```
%sql select DISTINCT(BOOSTER_VERSION) from SPACEXDATASET where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXDATASET)
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

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

```
%sql SELECT {fn MONTHNAME(DATE)} as "Month", BOOSTER_VERSION, LAUNCH_SITE from SPACEXDATASET WHERE year(DATE) = '2015' AND \
landing__outcome = 'Failure (drone ship)'
```

Month	booster_version	launch_site
January	F9 v1.1 B1012	CCAFS LC-40
April	F9 v1.1 B1015	CCAFS LC-40

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

- 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

```
%sql select LANDING__OUTCOME, COUNT(LANDING__OUTCOME) as count from SPACEXDATASET where DATE between '2010-06-04' and '2017-03-20'\
GROUP BY LANDING__OUTCOME\
order by COUNT(LANDING__OUTCOME) desc
```

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

Section 4

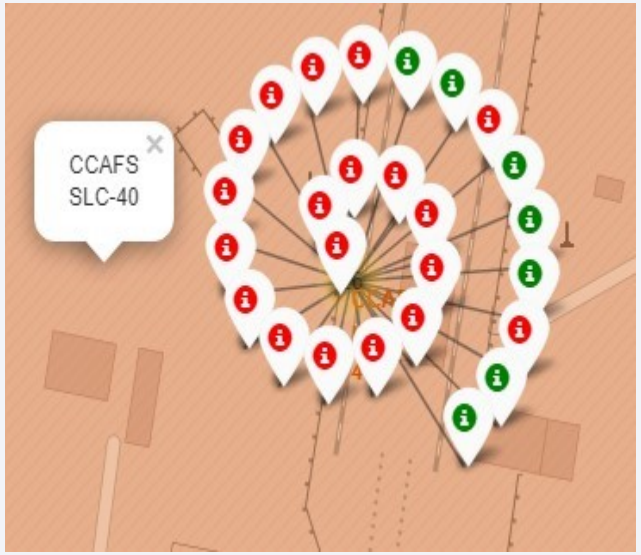
Launch Sites Proximities Analysis

Folium map showing all the launch sites

- Explain the important elements and findings on the screenshot

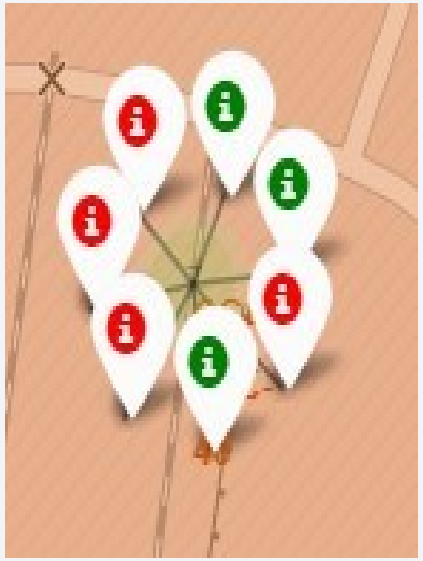
Folium maps showing Success and failure launches at each site

- The green icons indicate success and the red icons indicate failure

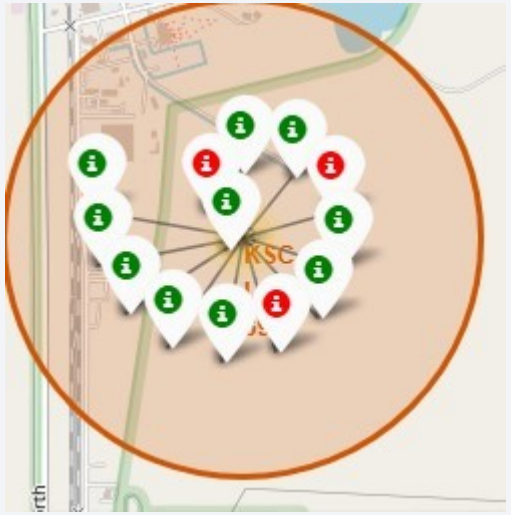


CCAFS LC-40

VAFB SLC-4E



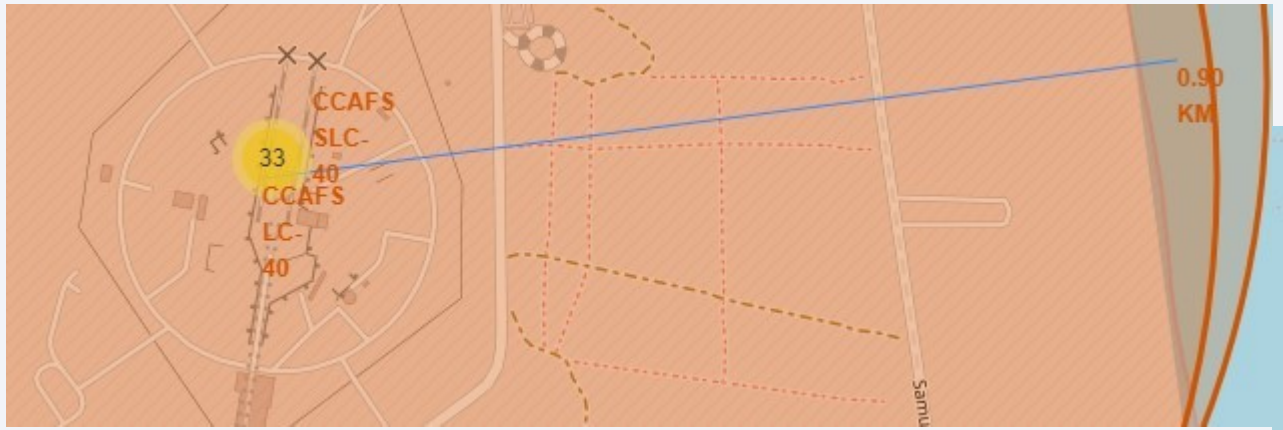
CCAFS SLC-40



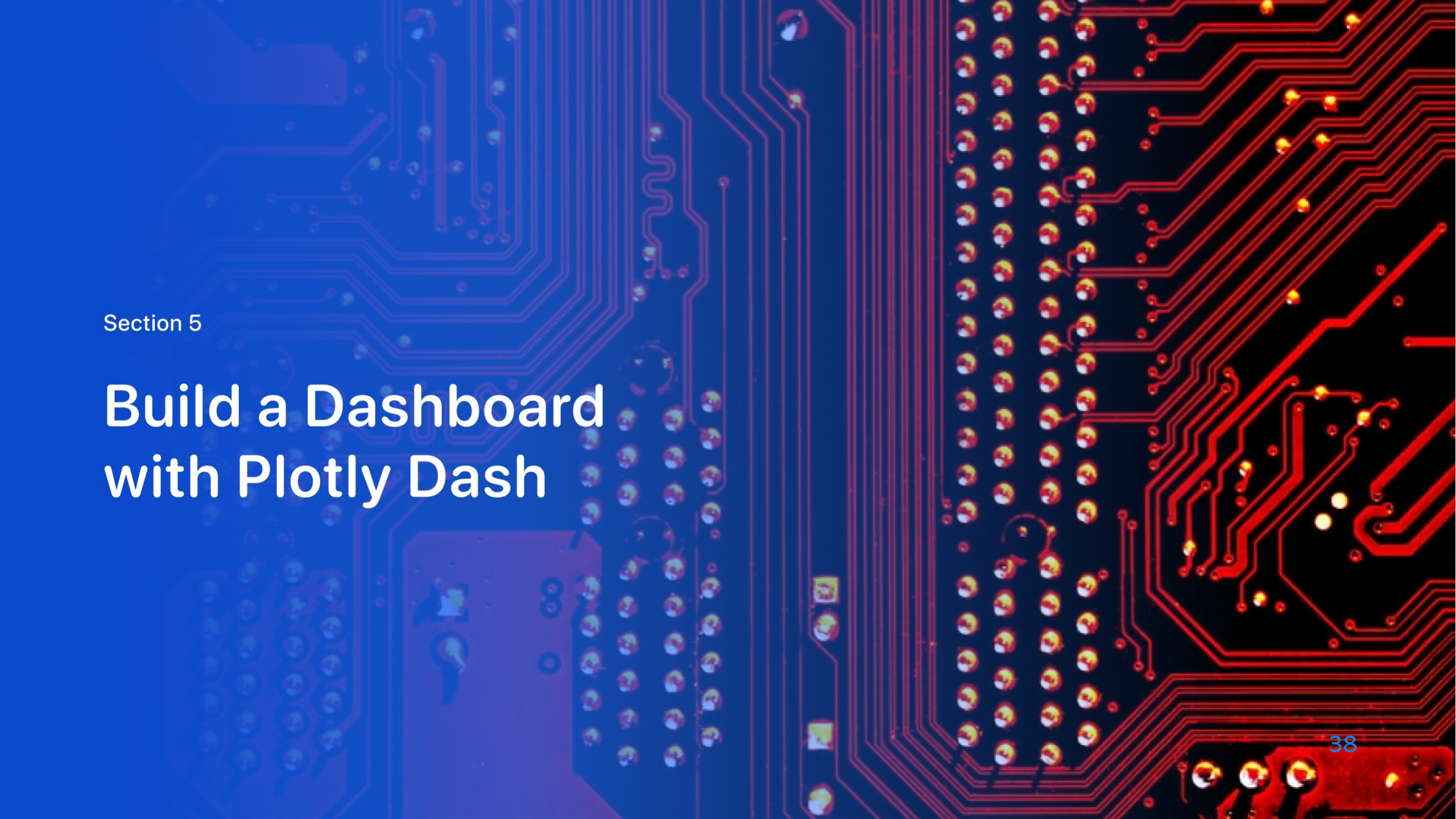
KSC LC-39A

Folium Map showing distance to nearest places

- Folium map showing the distance between the launch site and the coast line



- Folium map showing the distance between the launch site and the nearest railway line- it is about 0.7km

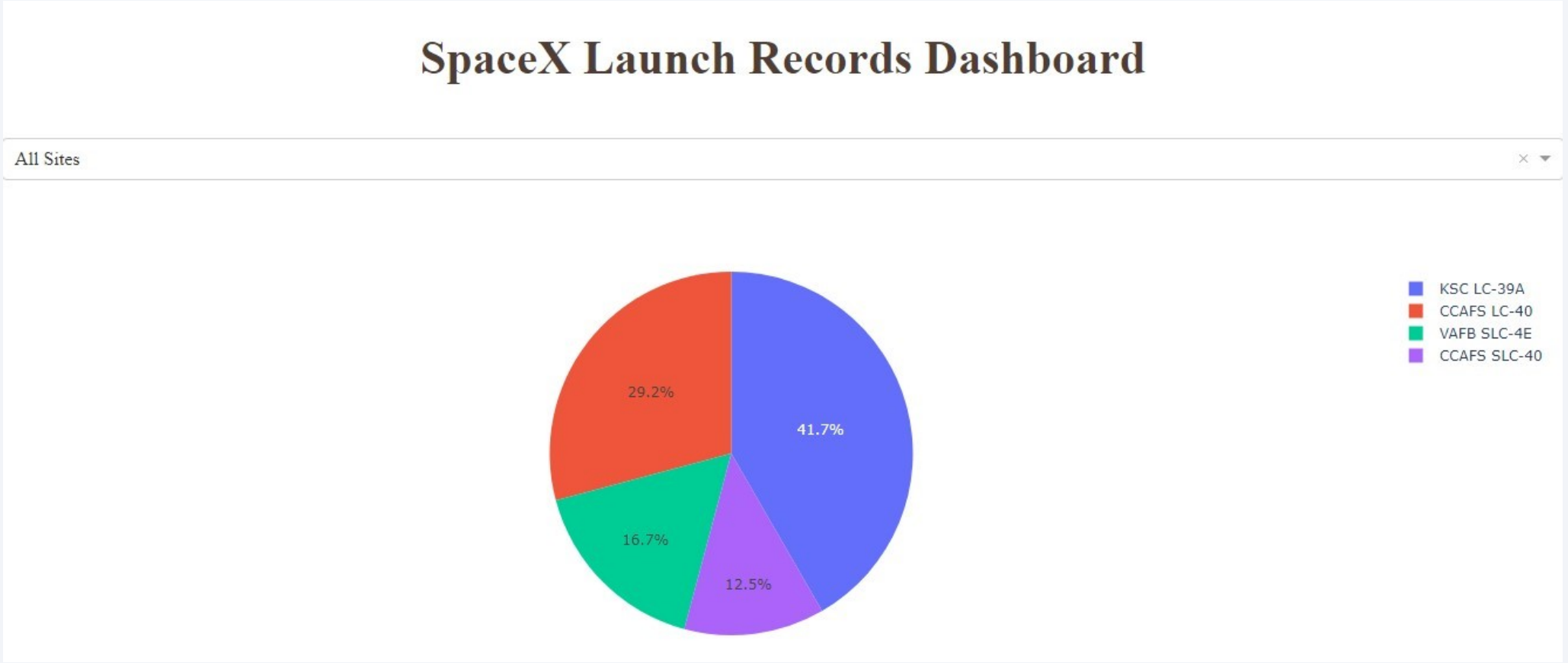


Section 5

Build a Dashboard with Plotly Dash

Launch Success Count of All Sites

- KSC LC-39A is the launch site with the maximum successful launches of all the sites.

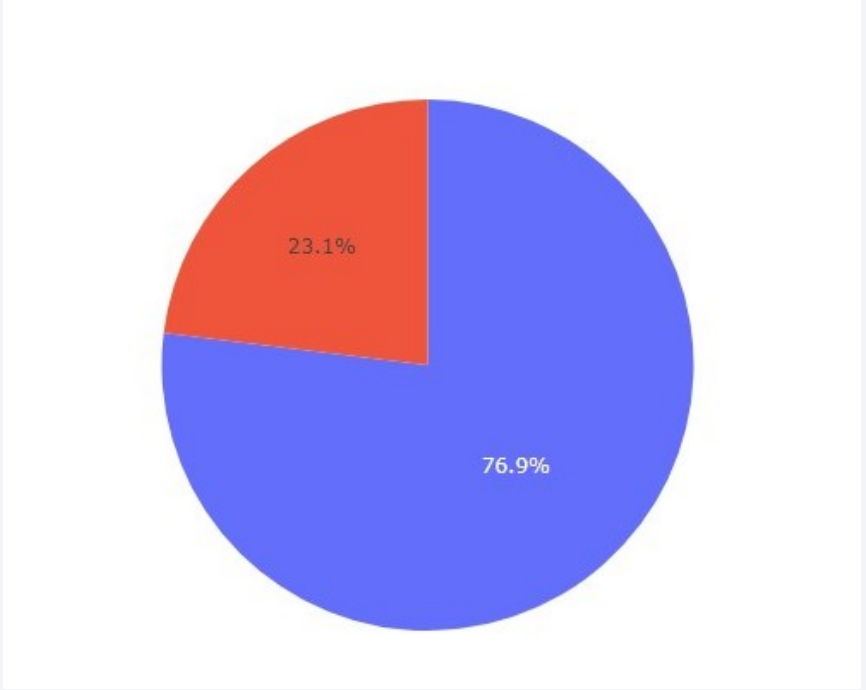


KSC LC-39A Success Rate

- KSC LC-39A achieved a success rate of 76.9%

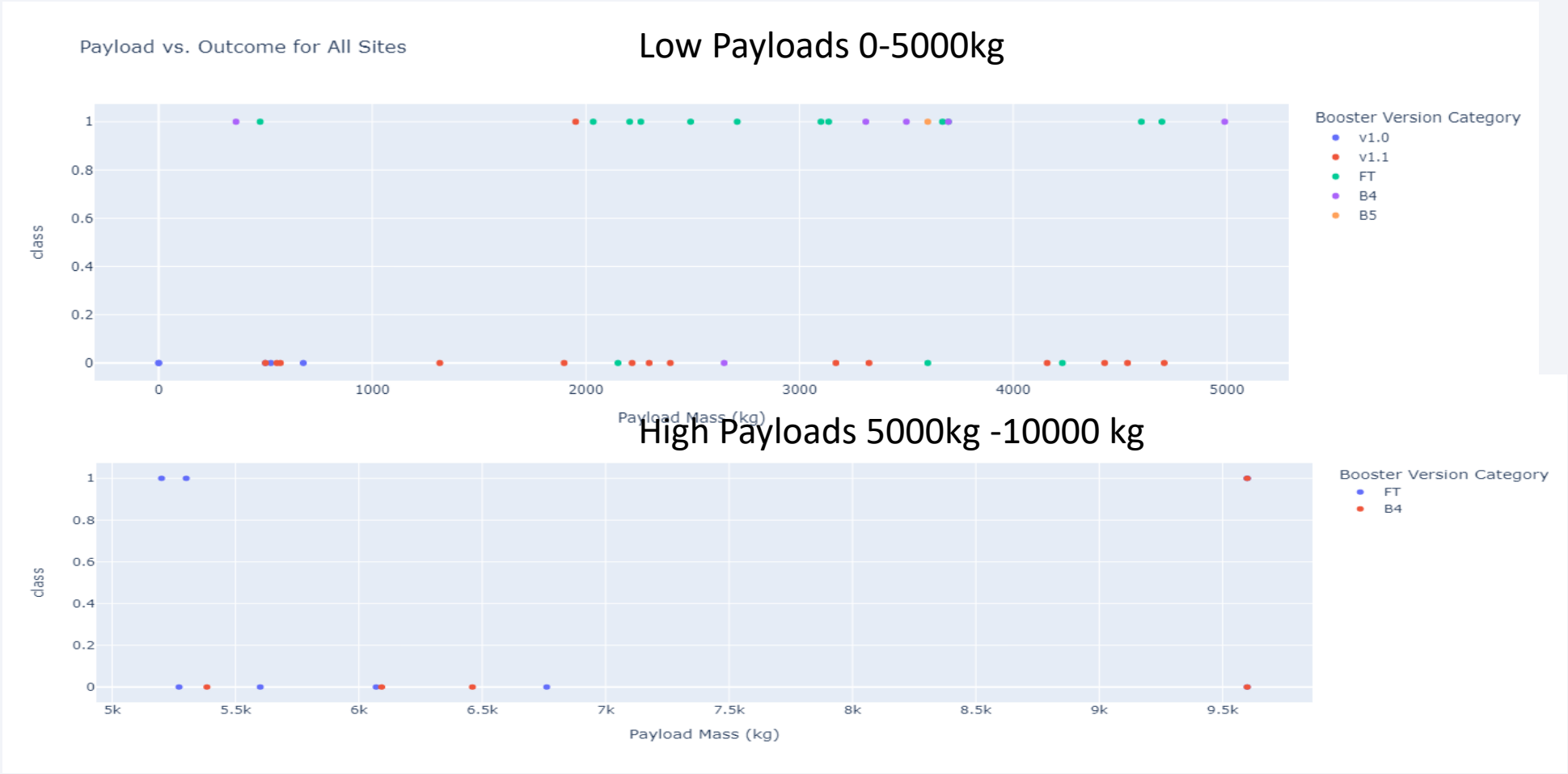
- Insights obtained by visualizing the dashboard:

  1. Which site has the highest launch success rate?
     - KSC LC-39A
  2. Which payload ranges have the highest success rate?
     - 2000kg to 10000kg
  3. Which payload ranges have the lowest success rate?
     - 0 to 2000kg
  4. Which F9 booster version has the highest success rate?
     - FT



Scatter Plots for Payload vs Outcomes

- The success rate for low weighted payloads is higher than for that of higher payloads.



Section 6

Predictive Analysis (Classification)

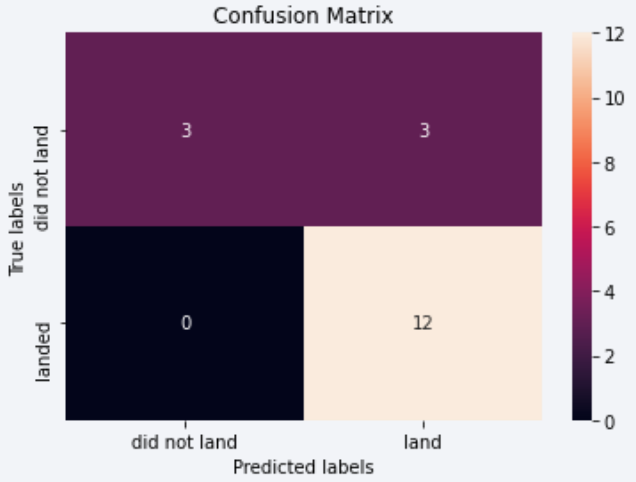
Classification Accuracy

ALGORITHM	ACCURACY	ACCURACY (TEST DATA)	TUNED HYPERPARAMETERS
LOGISTIC REGRESSION	0.84643	0.8334	'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'
SVM	0.84821	0.8334	'C': 1.0, 'gamma': 0.0316227660168379, 'kernel': 'sigmoid'
KNN	0.84821	0.8334	'algorithm': 'auto', 'n_neighbors': 10, 'p': 1
DECISION TREE	0.88571	0.8334	'criterion': 'gini', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'random'

- Almost all the models have same accuracy levels although **Decision Tree model** has a greater accuracy on Training data

Confusion Matrix

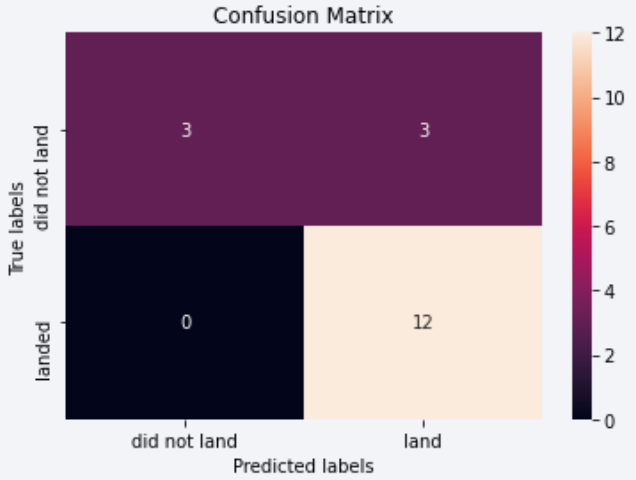
LOGISTIC REGRESSION



DECISION TREE

- All the four models have the same confusion matrix unfortunately.

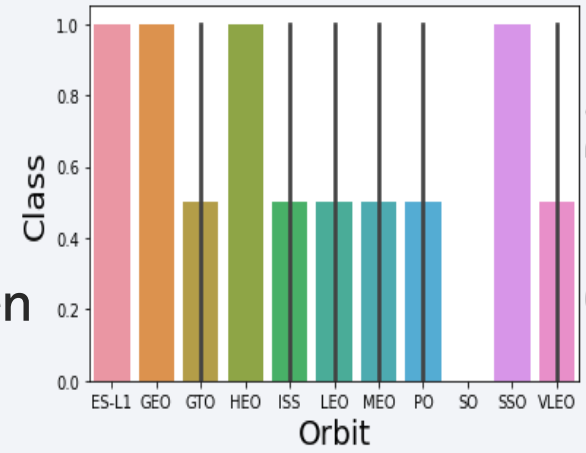
SVM

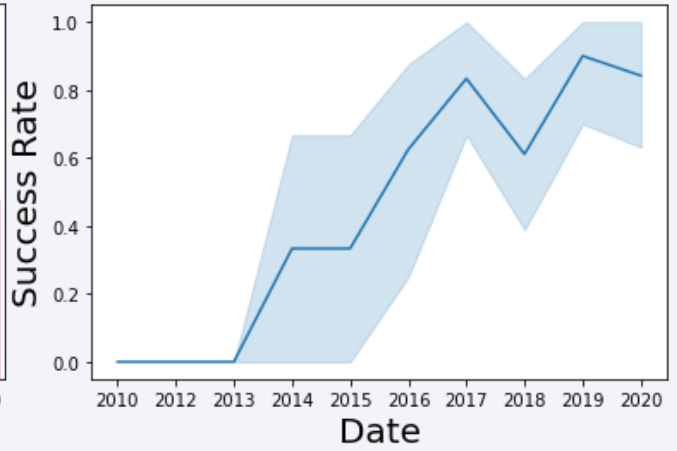


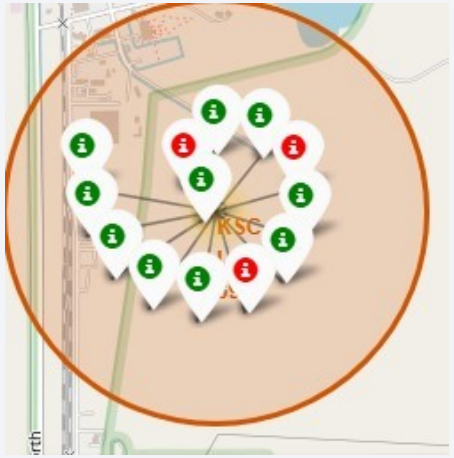
KNN

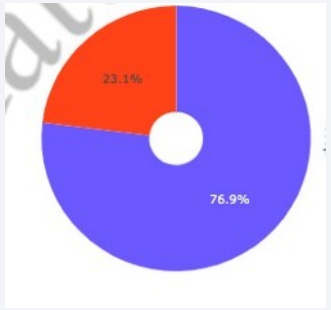
Conclusions

- Orbits ES-L1, GEO, HEO, SSO have higher success rates.
- Success rates for SpaceX launches has been increasing almost consistently and would reach the required target in the future.
- KSC LC-39A had the most successful launches but increasing payload seems to have a negative impact on the success rate.
- Based on the accuracy results, Decision Tree Classification algorithm can be best suited ML model for this dataset









KSC LC-39A

Thank you!