IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Michael Skovsgaard

09-08-2023

Outline

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

Executive Summary

- Summary of methodologies

- Data Collection via API, Web Scraping and SQL
- Data wrangling
- Interactive maps with Folium
- Predictive analysis for classification models

- Summary of all results

- Best model for Predictive Analysis

Introduction

Space X 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 Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.

In this project we want to create and compare serval classification models to find out which one best predict if the Falcon 9 will land.

Section 1

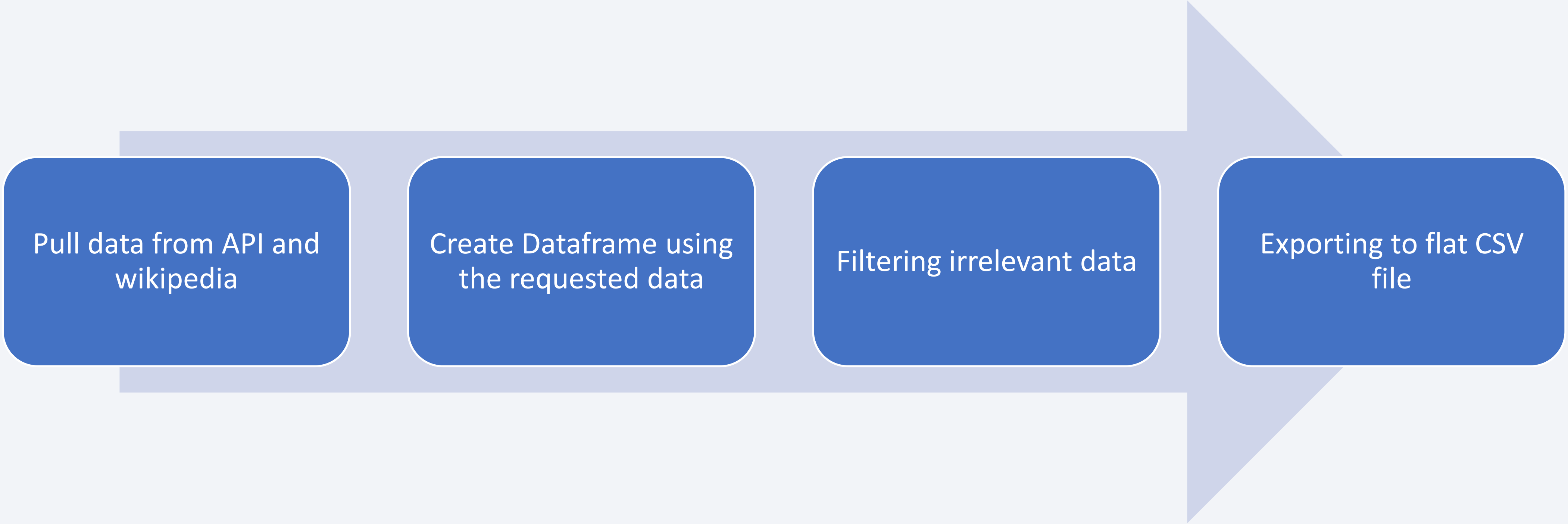
Methodology

Methodology

- Data collection methodology:
  - Via SpaceX Rest API and Web Scraping from Wikipedia
- Perform data wrangling
  - Removing irrelevant columns and missing values.
  - One hot encoding to prepare data for classification.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatterplots and bar graphs to understand patterns in the data.
- Perform interactive visual analytics using Folium and Plotly Dash
  - Using Plotly Dash and Folium for Data visualization.
- Perform predictive analysis using classification models
  - Build and compare classification models (Decision tree, SVM, Logistic Regression and ..)

Data Collection

Data was collected through SpaceX Rest API and Web scraping from Wikipedia.



Data Collection – SpaceX API

- SpaceX get request, normalization and filtering.

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

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

```
# Use json_normalize meethod to convert the json
response.json()

data = pd.json_normalize(response.json())
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
```

```
data_falcon9 = data[data['BoosterVersion']=='Falcon 9']
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

- [Github Url]()

Data Collection - Scraping

- Creating a BeautifulSoup object
- Find all tables in object
- Choosing third table
- Extracting column names from Soup object
- Creating a dictionary using column names
- Creating a Dataframe

Github Url

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

```
response = requests.get(static_url).text
```

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

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

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

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

```
launch_dict= dict.fromkeys(column_names)
```

```
df=pd.DataFrame(launch_dict)
```

Data Wrangling

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

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

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

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

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

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
```

```
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
```

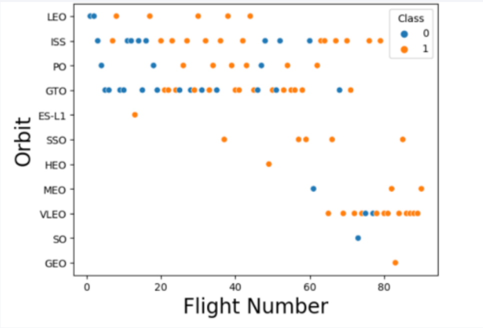
```
df.to_csv("dataset_part_2.csv", index=False)
```

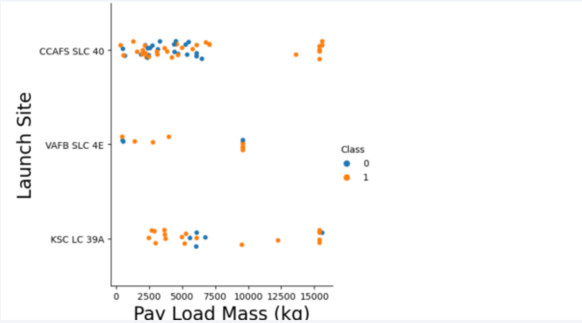
- **Process**
- Checking the count of Launch Site, Orbit and Outcome using value_counts function.
- Assigning the negative outcomes as “bad outcomes”.
- Using “bad outcomes” to create a new variable “landing class” as 0 if the Outcome was a failure and 1 if the Outcome was a success.
- Saving to CSV file.

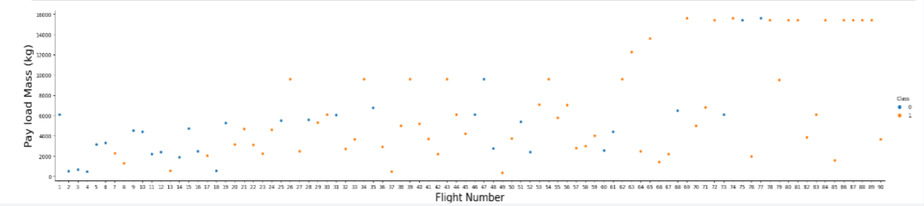
Github Url

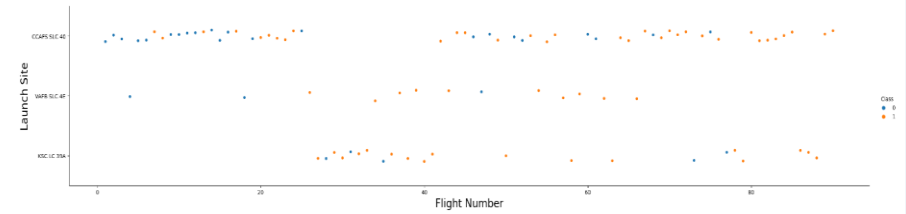
EDA with Data Visualization(1/2)

- Scatterplot with Flight Number vs. Launch Site and Class (Success or Failure)
- Scatterplots revealed the relationship between Launch Site, Payload Mass and Class



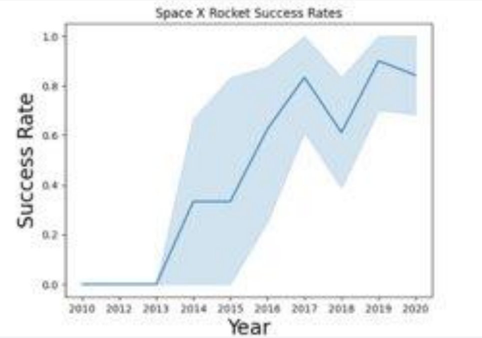




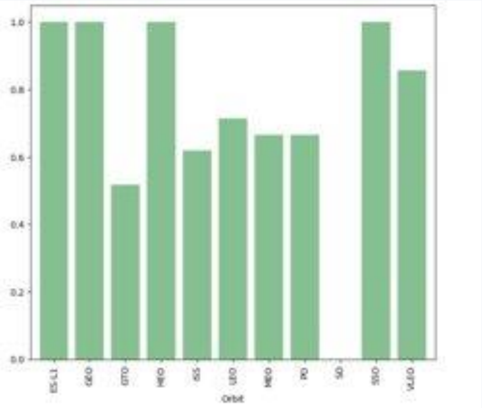


EDA with Data Visualization(2/2)

- Lineplot of Year and Success rate used show trends that can aid predictions.



- Barplot of Success rate vs. Orbit type displays the success rate for each orbit missions.



- [Github Url]

EDA with SQL (1/2)

- %sql SELECT Distinct(Launch_Site) FROM SPACEXTBL
- %sql SELECT * from SPACEXTBL WHERE Launch_Site LIKE 'CCA%'LIMIT 5
- %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER ='NASA (CRS)’
- %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE booster_version = "F9 v1.1“
- %sql SELECT MIN(DATE, landing_outcome FROM SPACEXTBL WHERE landing_outcome = "Success"
- %sql SELECT Booster_Version FROM SPACEXTBL WHERE landing_outcome = "Success (drone ship)" \

AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

- %sql SELECT Mission_Outcome, count(Mission_outcome) AS "Successful Missions" FROM SPACEXTBL \

WHERE Mission_outcome LIKE 'Success%' \

GROUP BY Mission_Outcome

EDA with SQL (2/2)

```
%sql SELECT Booster_Version FROM SPACEXTBL \
where PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

%sql SELECT substr(Date, 4, 2) AS Month, Booster_Version, Launch_Site FROM SPACEXTBL \
where landing_outcome = "Failure (Drone Ship)" AND year(DATE) = substr(Date,7,4)='2015'

%sql SELECT landing_outcome, count(landing_outcome) FROM SPACEXTBL \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
group by landing_outcome\
order by count(landing_outcome) desc
```

Github Url

Building an interactive map with Folium

- Vizualisation of Launch Data in an interactive map using the Longitude and Latitude coordinates for each launch site. Adding a marker for each launch site with label.
- Assigning Green and Red labels to markes using the launch_outcomes dataframe.

Built an interactive dashboard with Flask and Dash

- Building a dashbard with Flask and Dash

- Using piecharts to show total launches by launch site.

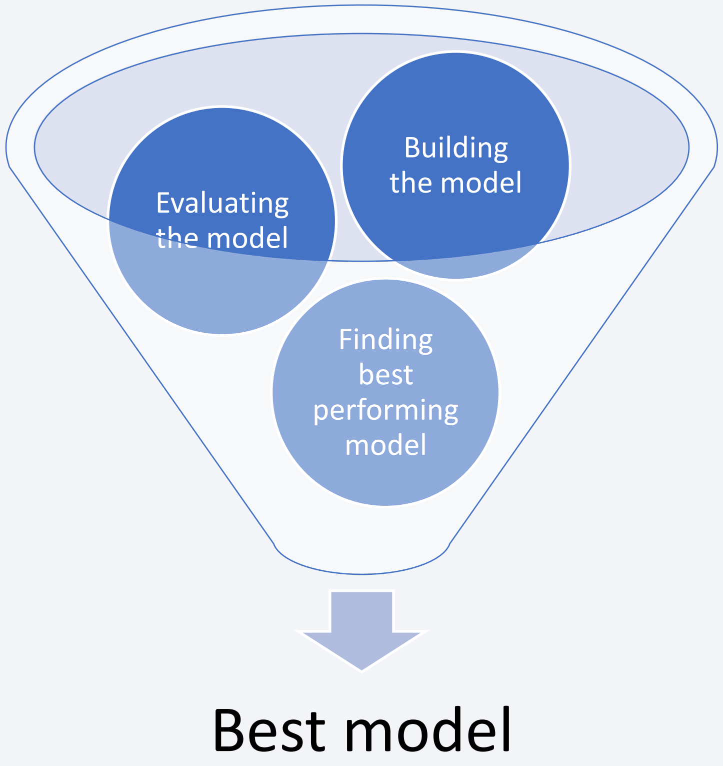
- Scatterplots showing Outcome vs Payload Mass

Predictive Analysis (Classification)

- The process of building the model included loading data and packages, standardizing the data, split the data into training and test data, using GridSearchCV and train our model.

- In the Evaluation phase we checked the accuracy of the model, found the best hyperparameters and plotted the confusion matrix.

- In the comparison phase we listed the accuracy for each model in a Dataframe and listed them according to the accuracy to easily compare which model performed yielded the best results.



Results

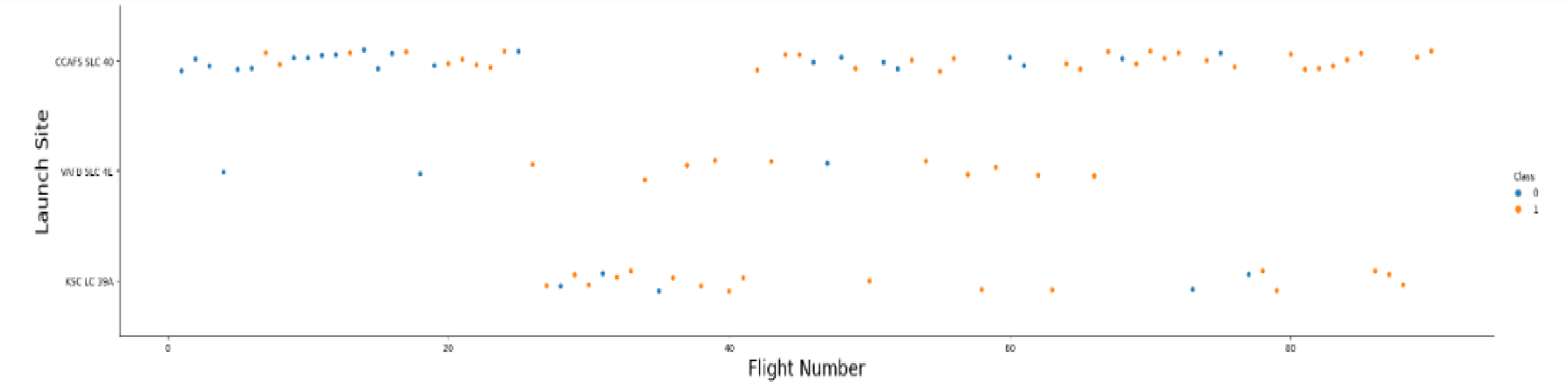
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

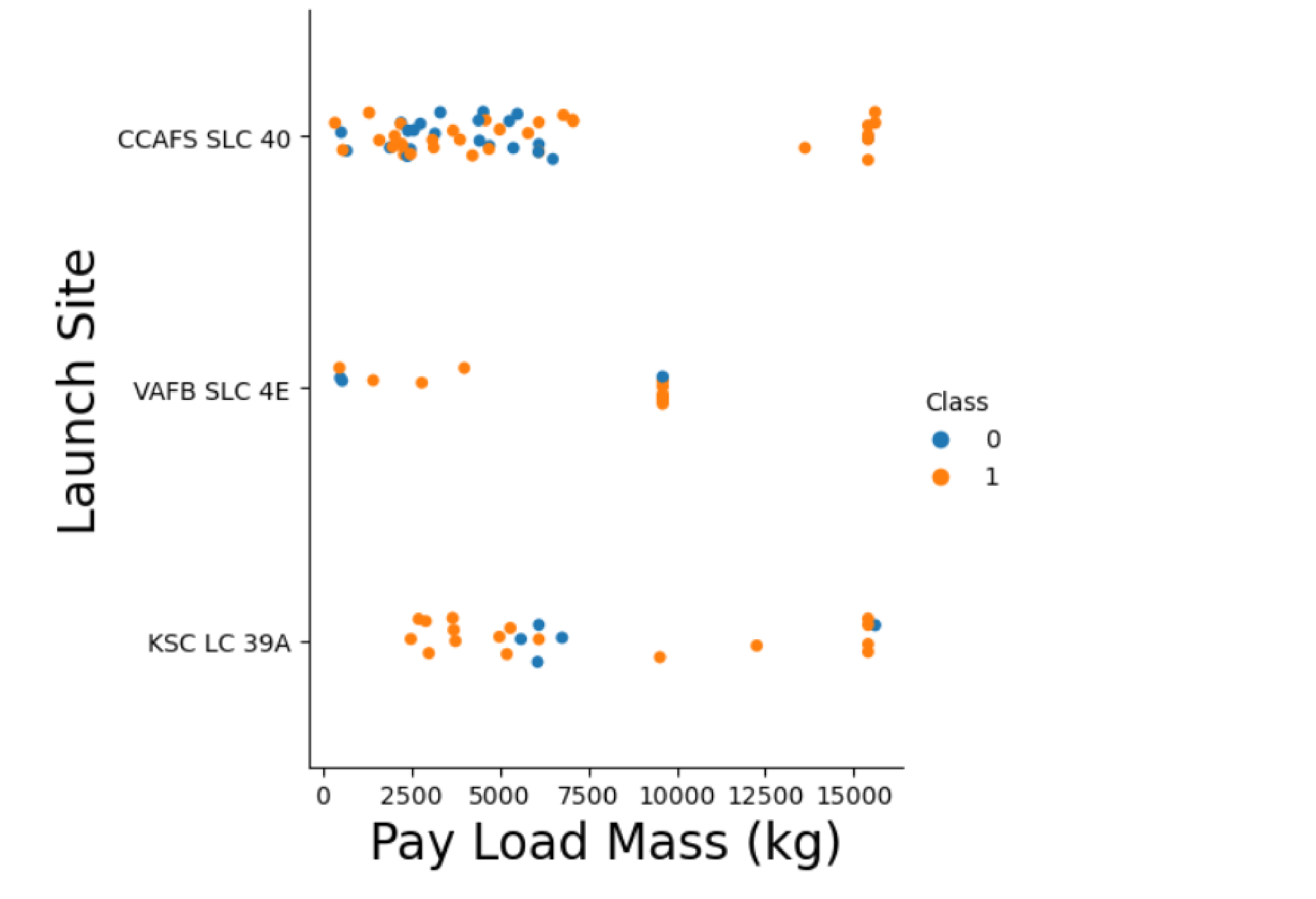
Flight Number vs. Launch Site

- With higher Flight Number the success rate seems to be improving



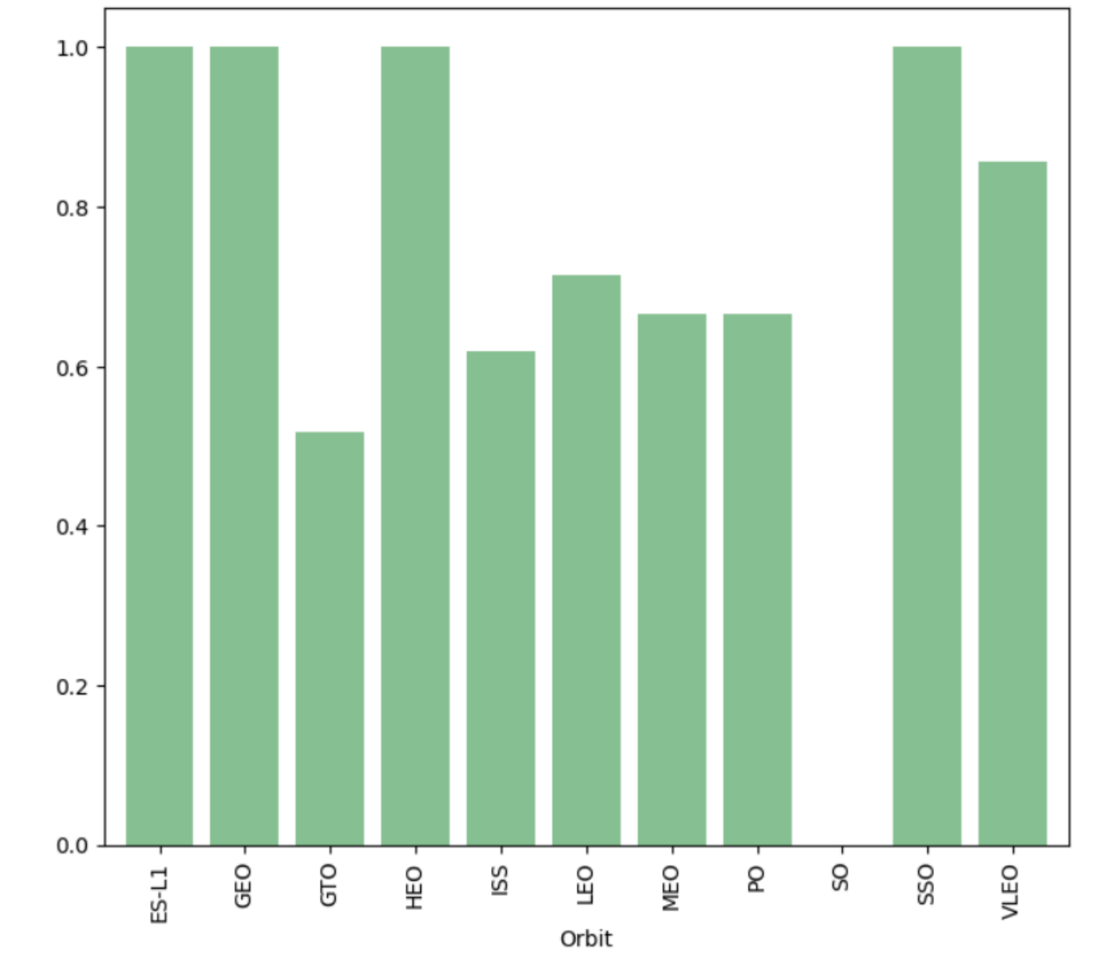
Payload vs. Launch Site

- It seems the higher the PayloadMass the higher chance of success for the rocket.
- However, this plot is not enough to determine a definite relationsship.



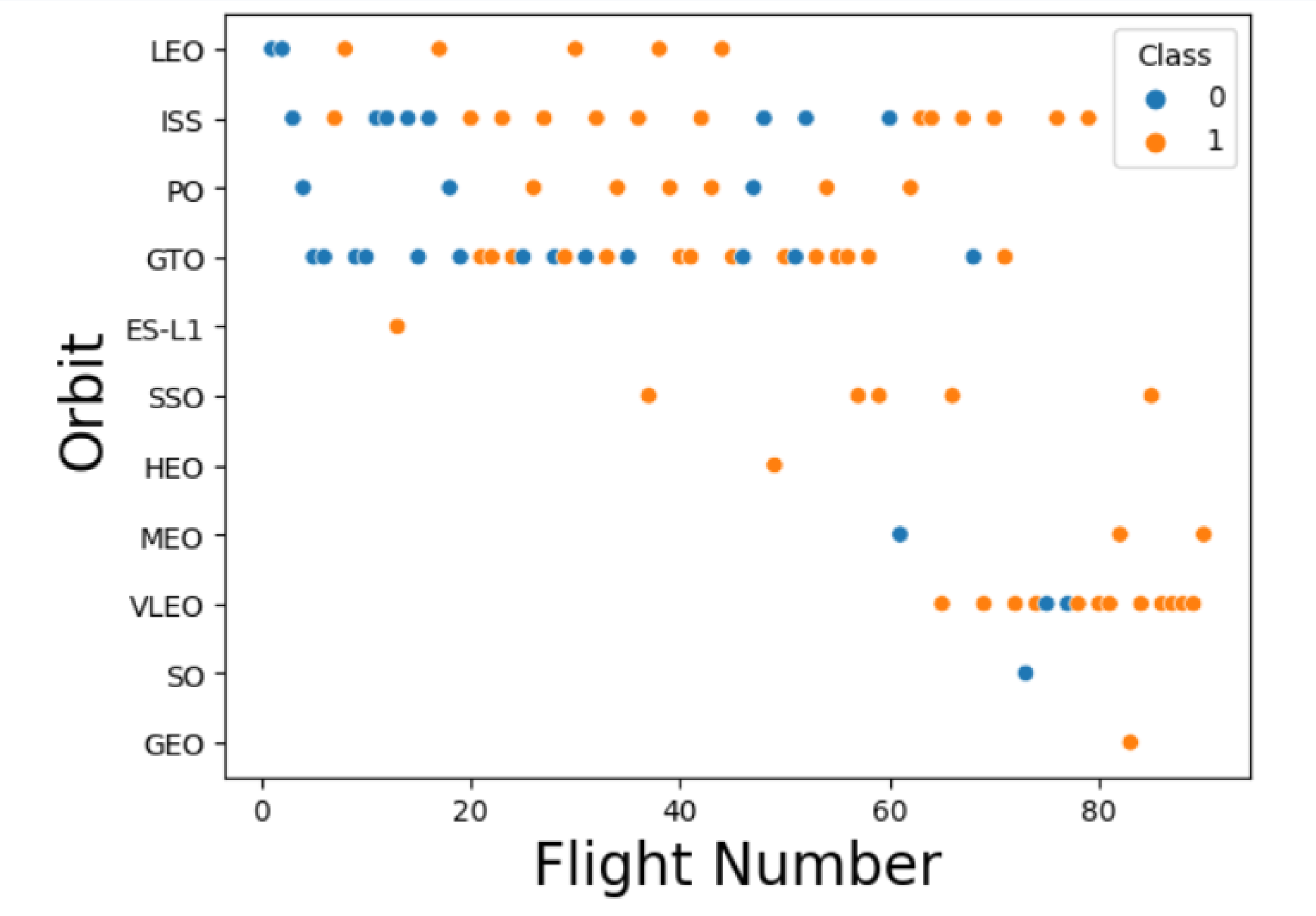
Success Rate vs. Orbit Type

- The Orbit type with the highest successrate is ES-L1, GEO, HEO and SSO.



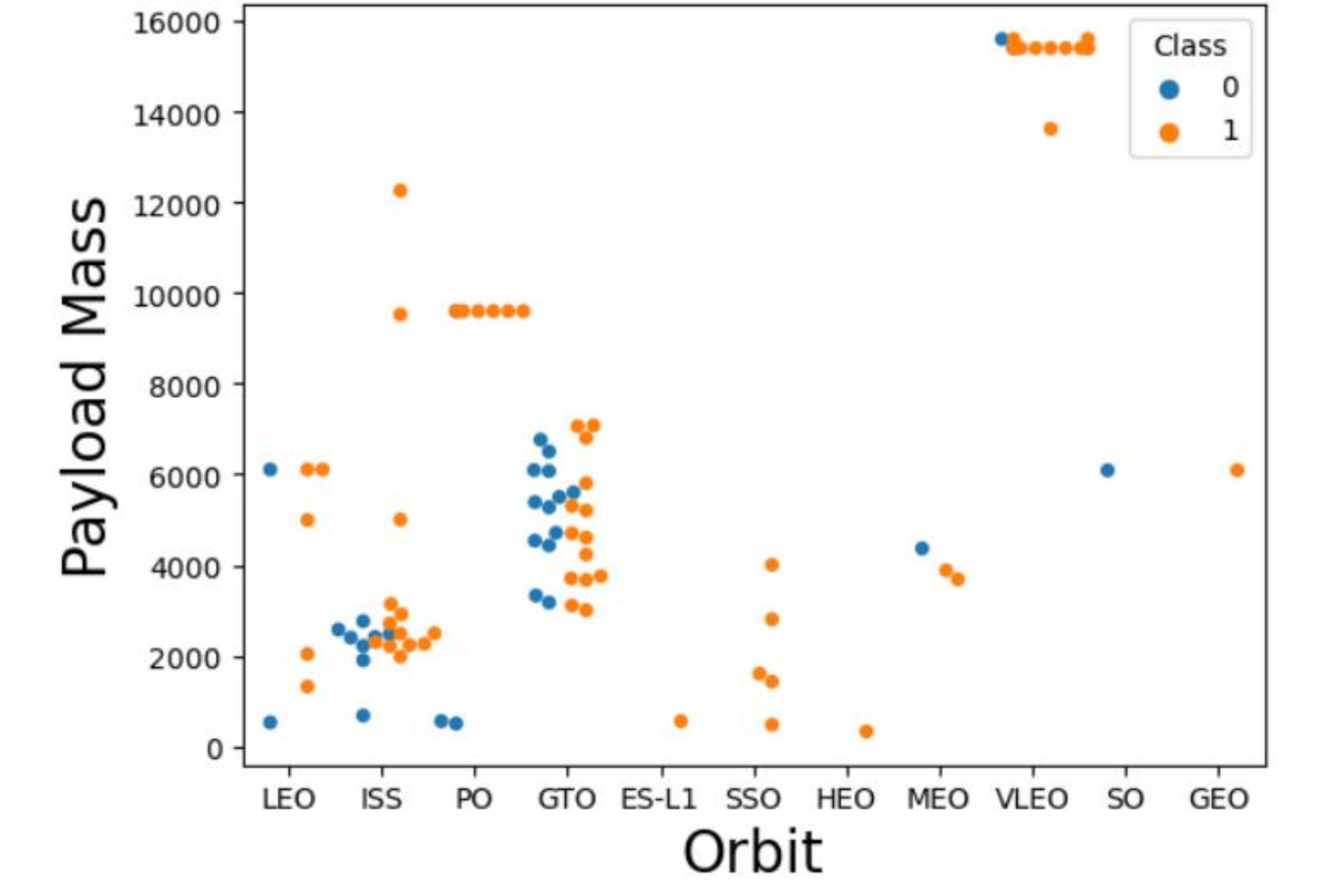
Flight Number vs. Orbit Type

- For LEO, it seems that the successrate increases with flightnumber. However, for GTO there seems to be no relationship between orbit and flightnumber.



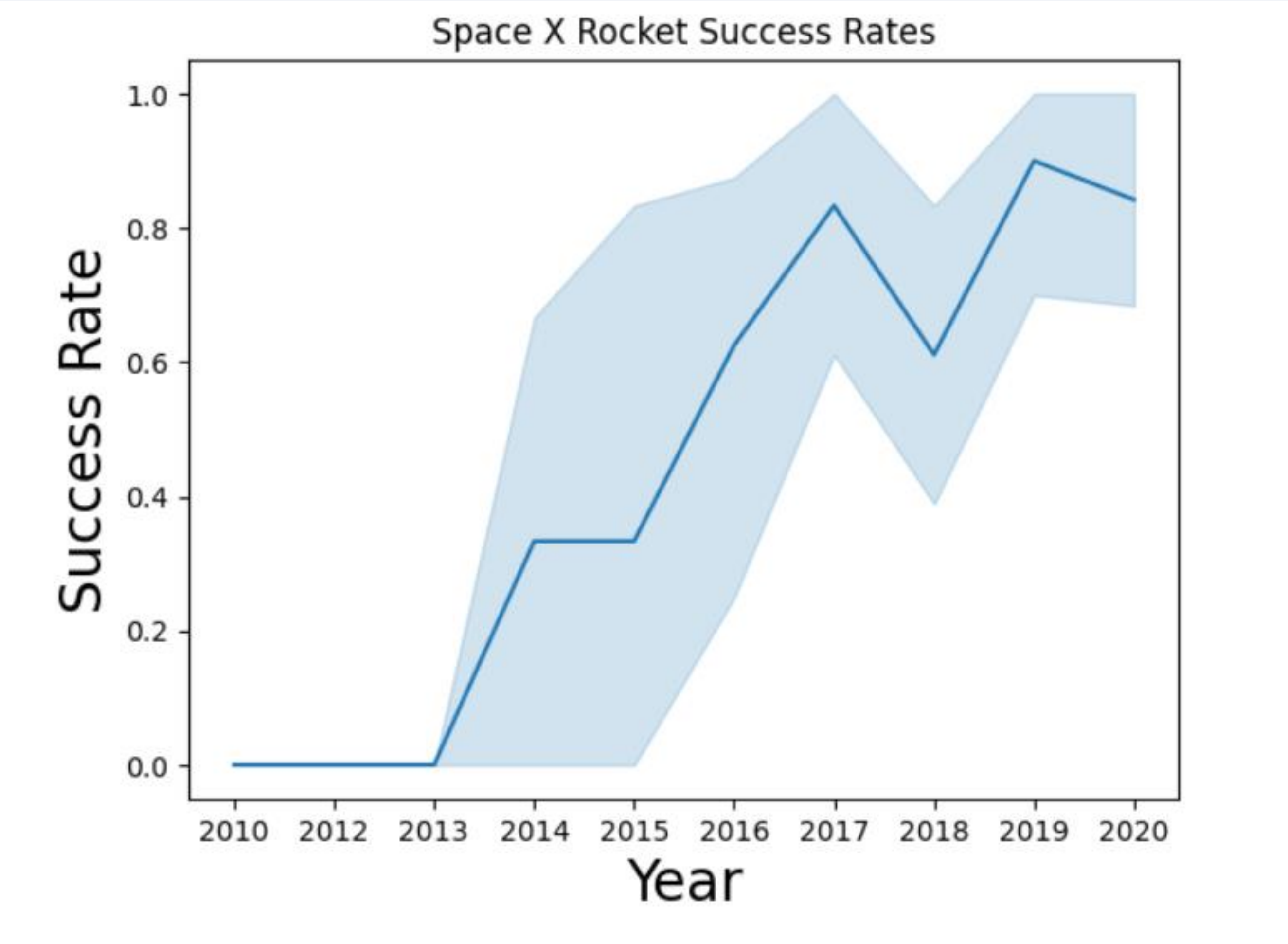
Payload vs. Orbit Type

- From this scatterplot we see that heavy payload mass influence the success of the mission for some Orbits, such as MEO, and VLEO.
- For Orbits an increase in payload mass did not affect the success.



Launch Success Yearly Trend

- This line chart illustrates that as year increases the successrate seems to increase as well.



All Launch Site Names

```
%sql SELECT Distinct(Launch_Site) FROM SPACEXTBL
```

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * from SPACEXTBL WHERE Launch_Site LIKE 'CCA%'LIMIT 5
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT Booster_Version FROM SPACEXTBL 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 Outcomes

```
%sql SELECT Mission_Outcome, count(Mission_outcome) AS "Successful Missions" FROM SPACEXTBL \
WHERE Mission_outcome LIKE 'Success%' \
GROUP BY Mission_Outcome
```

Mission_Outcome	Successful Missions
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
%sql SELECT Booster_Version FROM SPACEXTBL \
where PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

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

```
%sql SELECT landing_outcome, count(landing_outcome) FROM SPACEXTBL \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
group by landing_outcome\
order by count(landing_outcome) desc
```

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

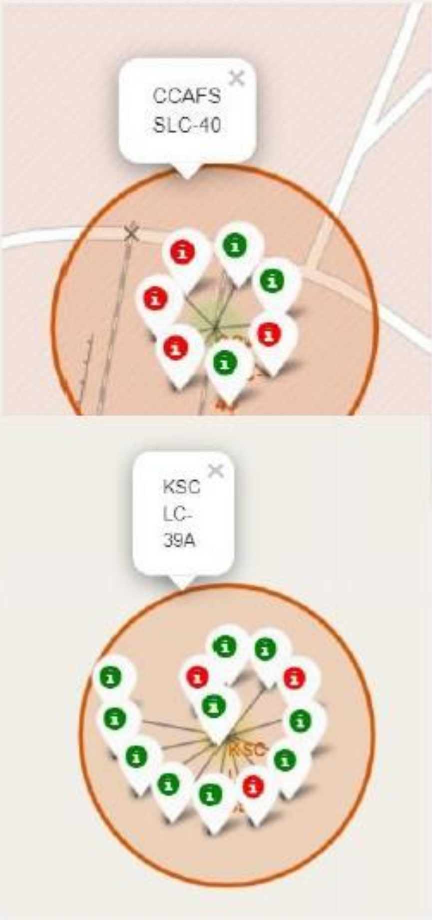
Section 3

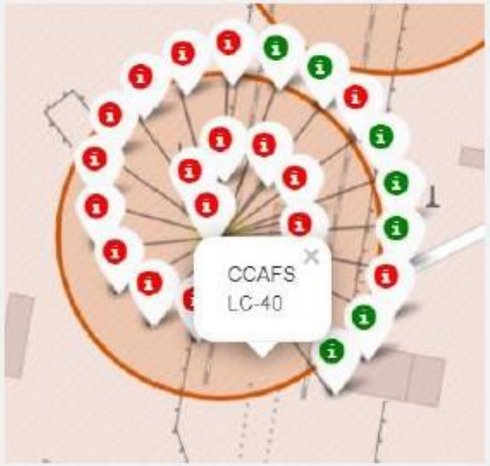
Launch Sites Proximities Analysis

Build an Interactive Map with Folium

- We see that the launch sties are all located in Florida or California in USA

Build an Interactive Map with Folium





Florida Launch Sites

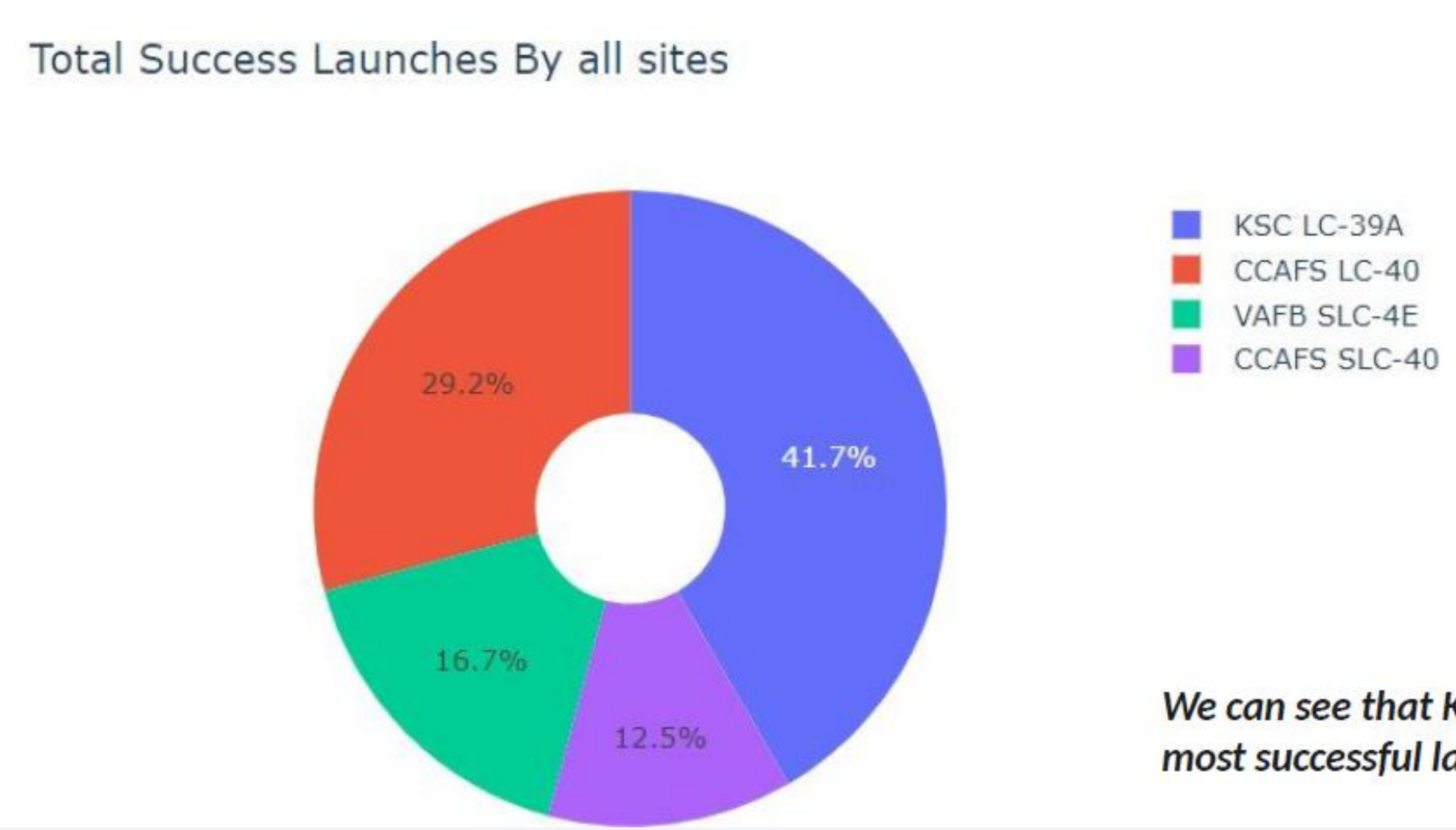
Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

Section 4

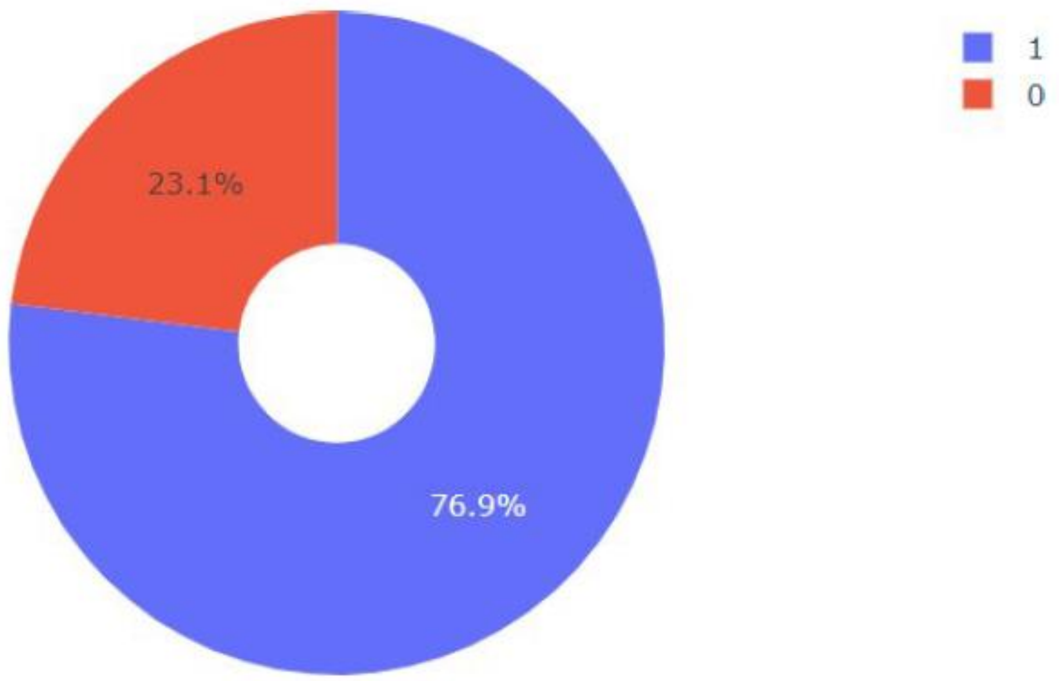
Build a Dashboard with Plotly Dash

Build a Dashboard with Plotly Dash



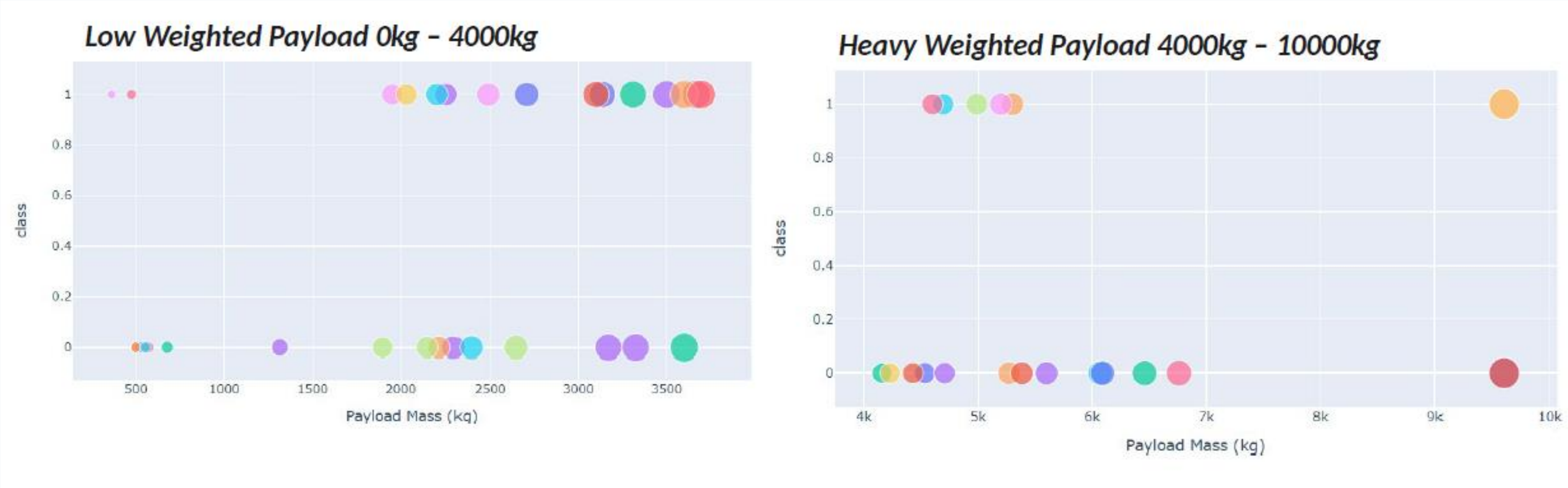
We can see that KSC LC-39A had the most successful launches from all the sites

Build a Dashboard with Plotly Dash



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Build a Dashboard with Plotly Dash



- This illustrates the success rates for low weighted payloads is higher than the heavy weighted payloads

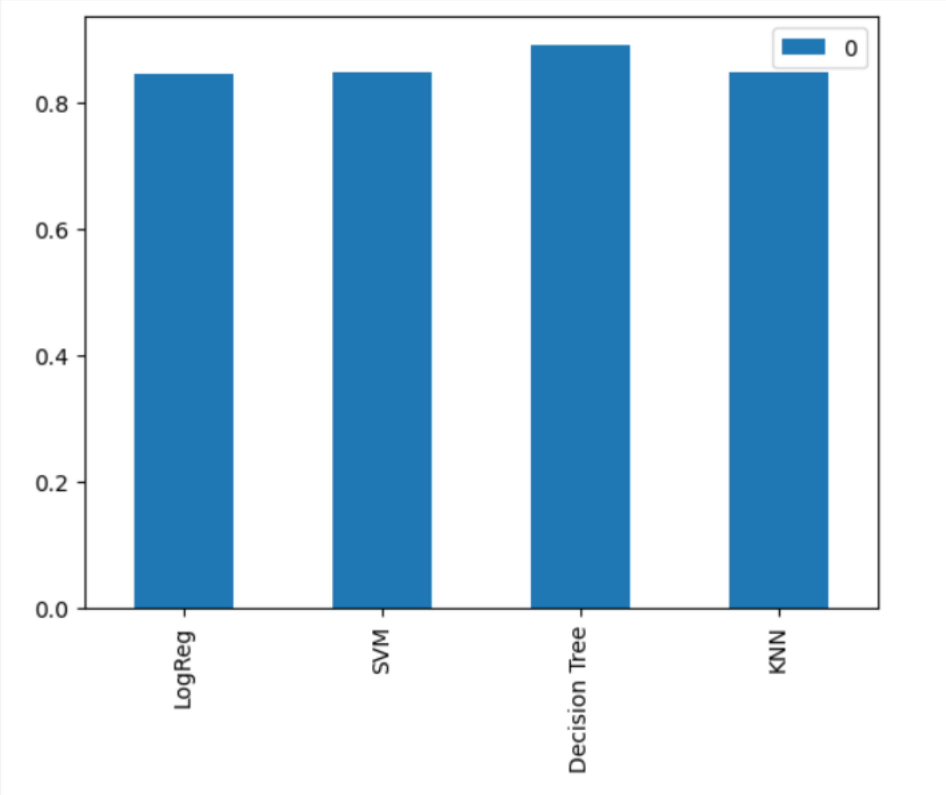
Section 5

Predictive Analysis (Classification)

Classification Accuracy

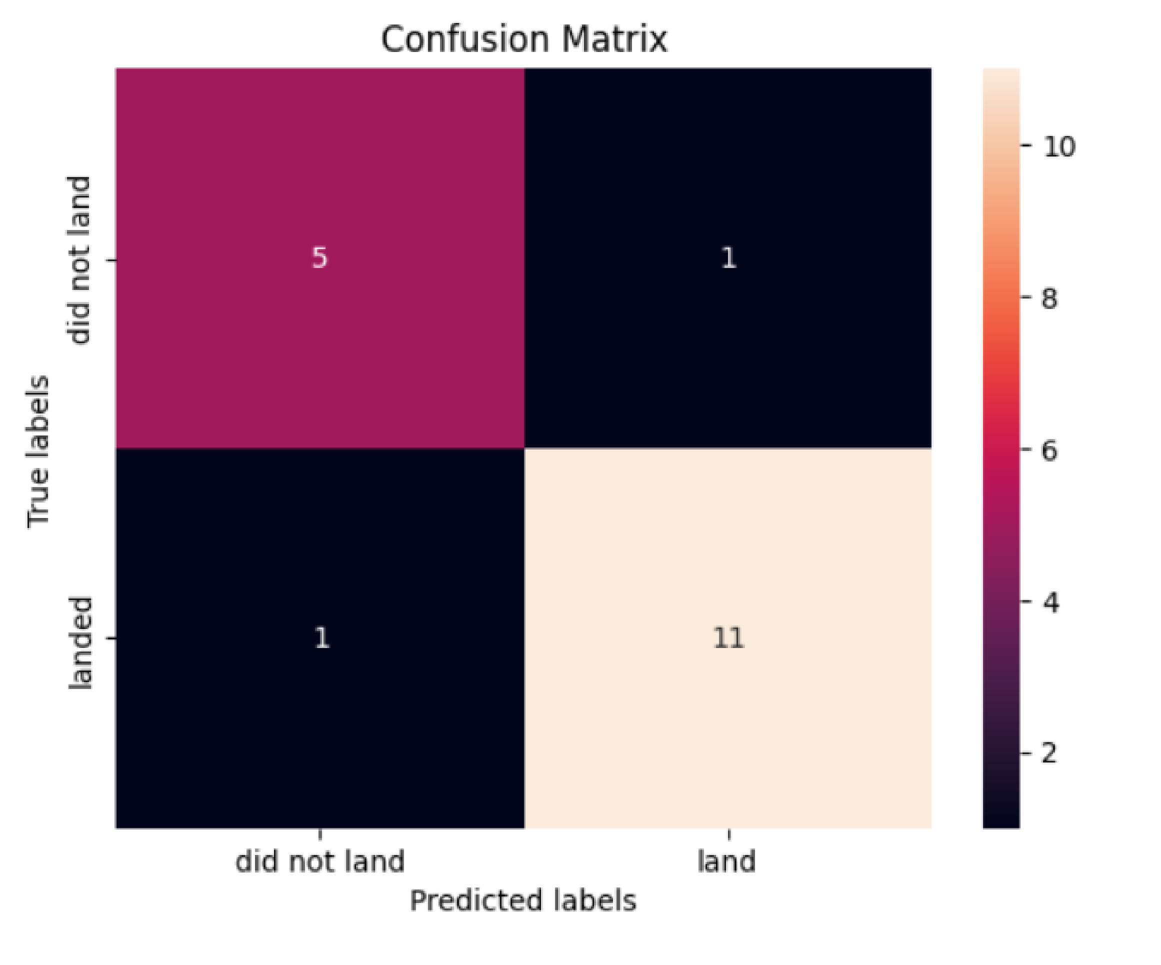
- In this project, A Decision Tree classification model performed a little better than the Logistic Regression, SVM and KNN classification models.

	0
LogReg	0.846429
SVM	0.848214
Decision Tree	0.891071
KNN	0.848214



Confusion Matrix

- The confusion illustrates the prediction from the classification model.
- From this we can see that the model correctly predicted 11 landings and misclassified 1 landing as a failure.
- Furthermore, we can see that the model correctly predicted 5 failures and misclassified 1 failure as a landing.



Conclusions

- The Decision Tree Classifier Algorithm performed better than SVM, Logistic Regression and KNN for this dataset.
- KSC LC-39A has had the most successful launches among the launch sites.
- Orbits GEO, HEO, ES-L1 and SSO has highest sucessrate.
- …

Thank you!