IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Alan Douglas Cohen Tello>
<21-01-2024>

Outline

Executive Summary

- Collected data from public SpaceX API and SpaceX Wikipedia page. Created labels column 'class' which classifies successful landings. Explored data using SQL, visualization, folium maps, and dashboards. Gathered relevant columns to be used as features. Changed all categorical variables to binary using one hot encoding. Standardized data and used GridSearchCV to find best parameters for machine learning models. Visualize accuracy score of all models.

- Four machine learning models were produced: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors. All produced similar results with accuracy rate of about 83.33%. All models over predicted successful landings. More data is needed for better model determination and accuracy.

Introduction

Background:
Commercial Space Age is Here
Space X has best pricing ($62 million vs. $165 million USD)
Largely due to ability to recover part of rocket (Stage 1)
Space Y wants to compete with Space X

Problem:
Space Y tasks us to train a machine learning model to predict successful Stage 1 recovery

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Combined data from SpaceX public API and SpaceX Wikipedia page
- Perform data wrangling
  - Classifying true landings as successful and unsuccessful otherwise
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tuned models using GridSearchCV

Data Collection

- Data collection process involved a combination of API requests from Space X public API and web scraping data from a table in Space X's Wikipedia entry.
  The next slide will show the flowchart of data collection from API and the one after will show the flowchart of data collection from webscraping.

- Space X API Data Columns:
  FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

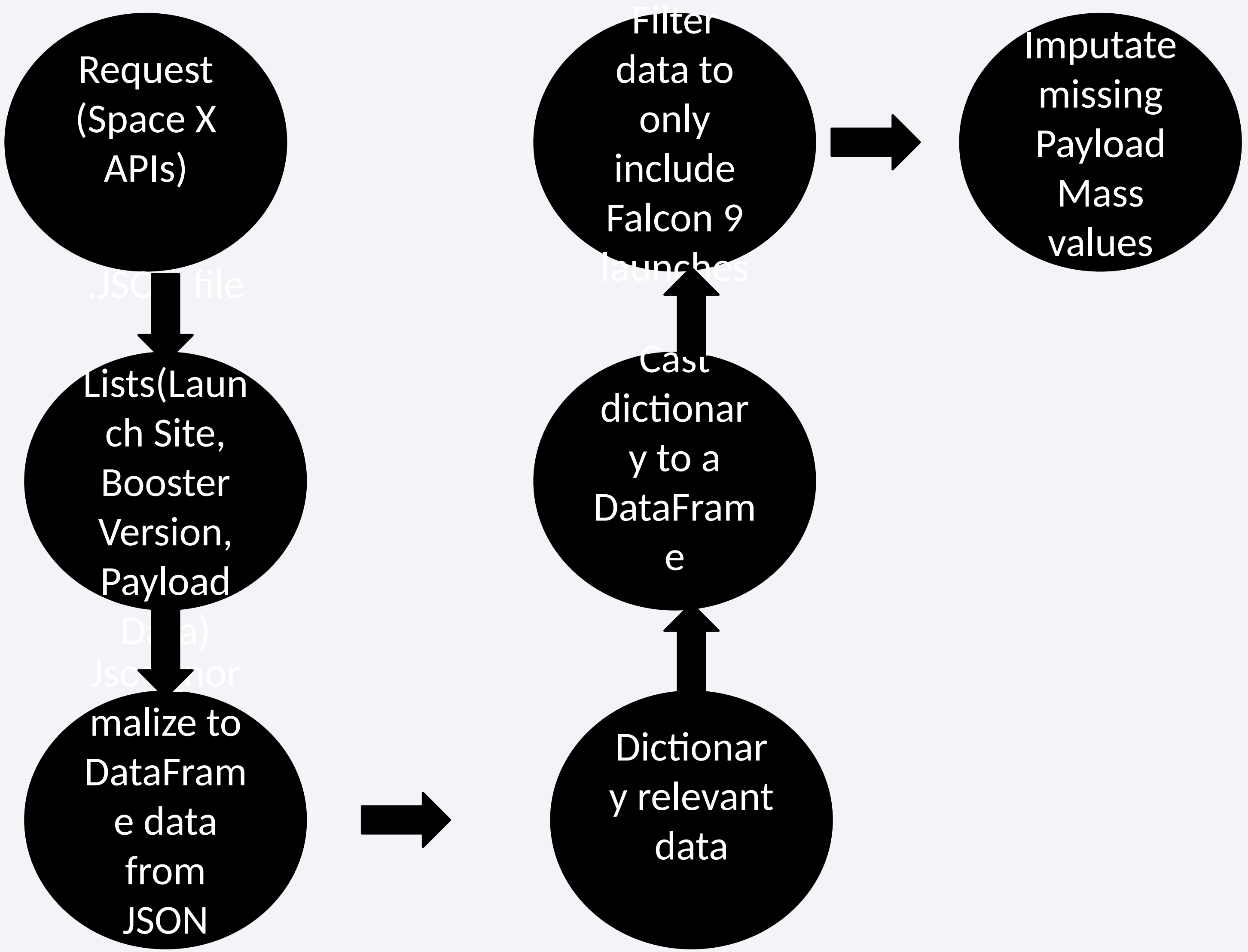
- Wikipedia Webscrape Data Columns:

  Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, TimeS

Data Collection – SpaceX API

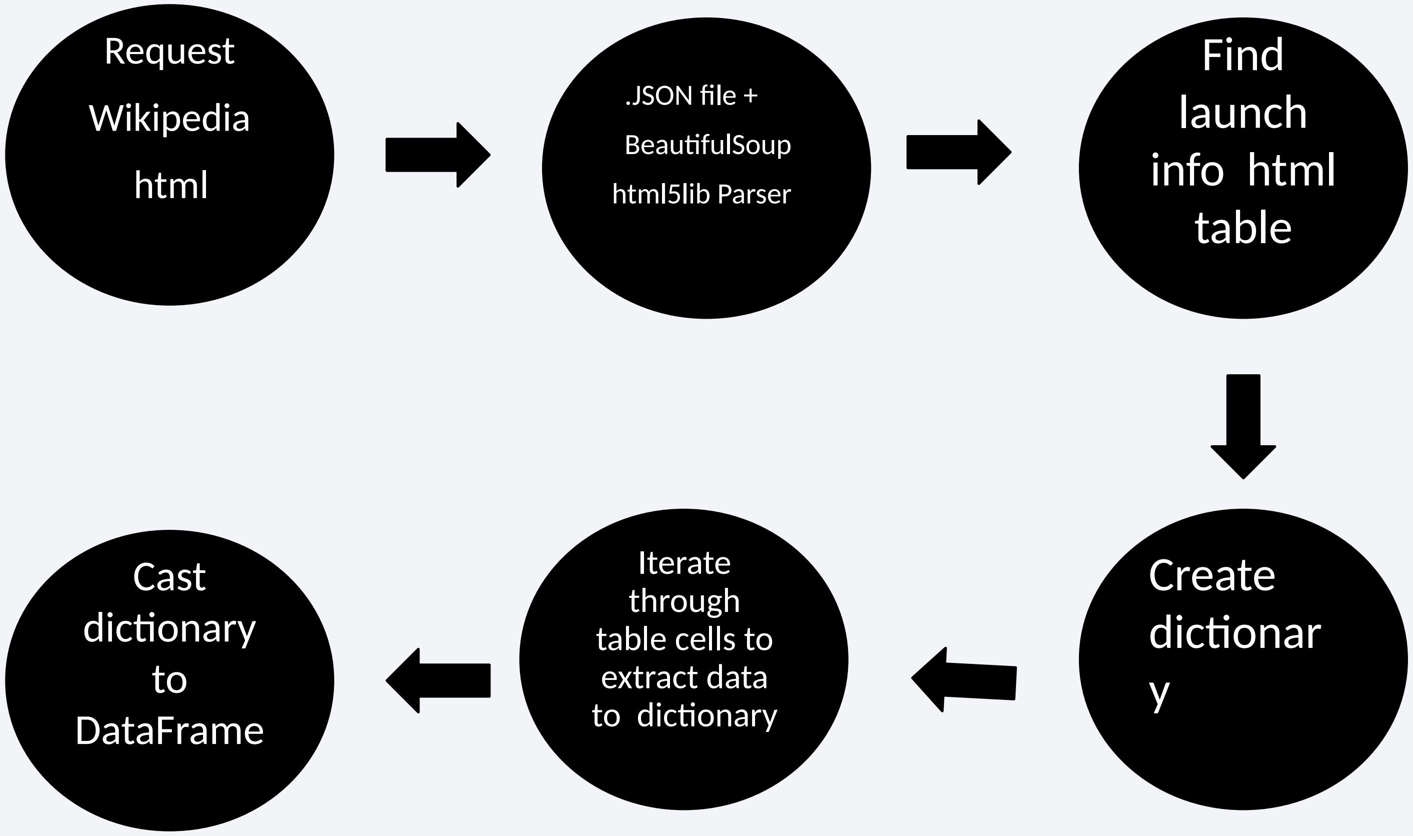
GitHub URL:

https://github.com/alchemistcohen/Applied-Data-Science-Capstone/blob/3f5998eecae6cac22d45bc521c489115a4c4618a/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- GitHub URL : https://github.com/alchemistcohen/Applied-Data-Science-Capstone/blob/3f5998eecae6cac22d45bc521c489115a4c4618a/jupyter-labs-webscraping.ipynb



Data Wrangling

Create a training label with landing outcomes where successful = 1 & failure = 0.
Outcome column has two components: 'Mission Outcome' 'Landing Location'
New training label column 'class' with a value of 1 if 'Mission Outcome' is True and 0 otherwise. Value Mapping:
True ASDS, True RTLS, & True Ocean – set to -> 1
None None, False ASDS, None ASDS, False Ocean, False RTLS – set to -> 0

GitHub url:

https://github.com/alchemistcohen/Applied-Data-Science-Capstone/blob/3f5998eecae6cac22d45bc521c489115a4c4618a/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Exploratory Data Analysis performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class and Year.
Plots Used:
Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit vs. Success Rate, Flight Number vs. Orbit, Payload vs Orbit, and Success Yearly Trend
Scatter plots, line charts, and bar plots were used to compare relationships between variables to decide if a relationship exists so that they could be used in training the machine learning model
GitHub url:
https://github.com/alchemistcohen/Applied-Data-Science-Capstone/blob/3f5998eecae6cac22d45bc521c489115a4c4618a/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

Loaded data set into IBM DB2 Database.

Queried using SQL Python integration.

Queries were made to get a better understanding of the dataset.

Queried information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

GitHub
https://github.com/alchemistcohen/Applied-Data-Science-Capstone/blob/3f5998eecae6cac22d45bc521c489115a4c4618a/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.

This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.

GitHub url:

https://github.com/alchemistcohen/Applied-Data-Science-Capstone/blob/3f5998eecae6cac22d45bc521c489115a4c4618a/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

Dashboard includes a pie chart and a scatter plot.

Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.

Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.

The pie chart is used to visualize launch site success rate.

The scatter plot can help us see how success varies across launch sites, payload mass, and booster version category.

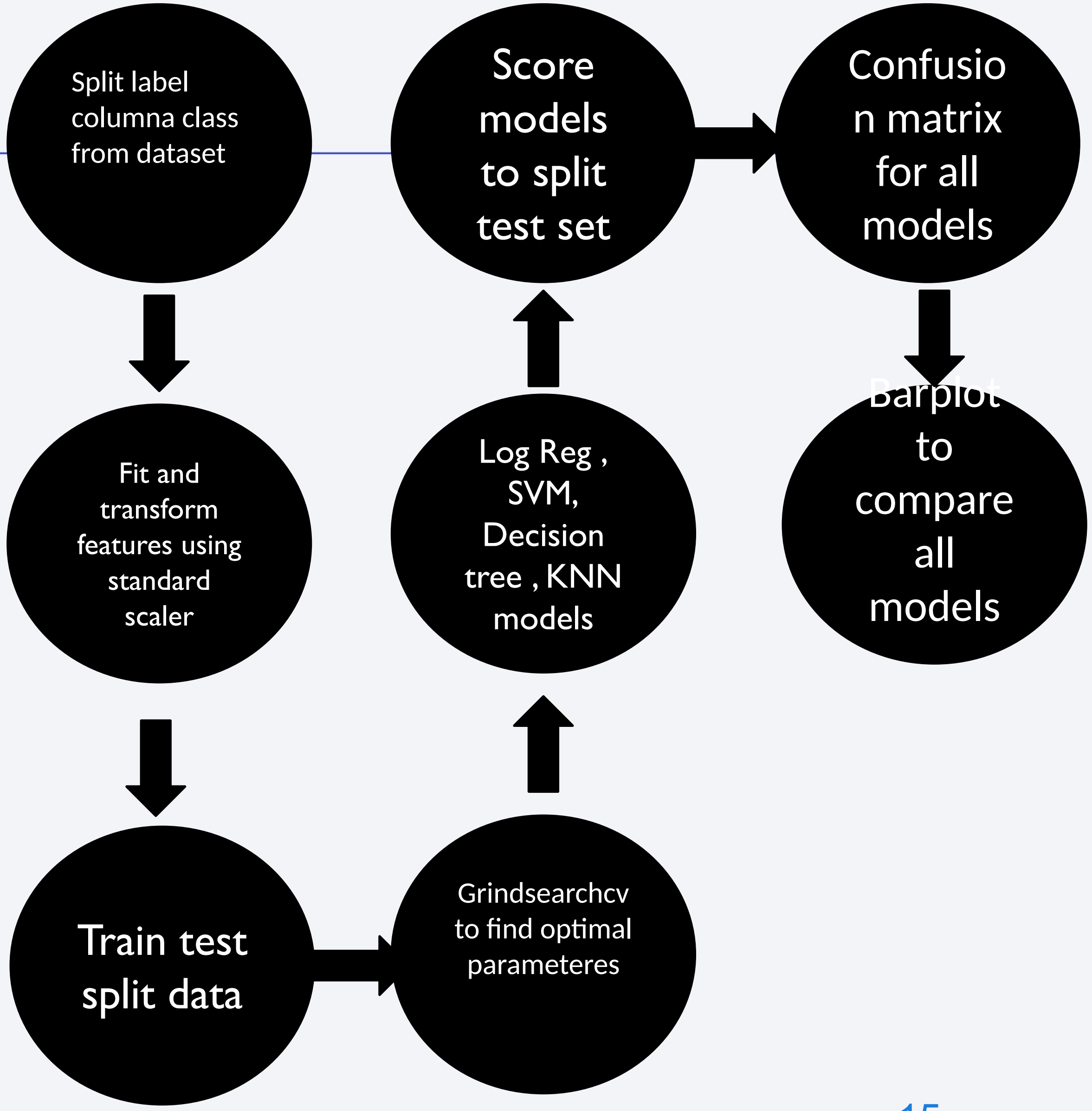
GitHub url:

https://github.com/alchemistcohen/Applied-Data-Science-Capstone/blob/3f5998eecae6cac22d45bc521c489115a4c4618a/spacex_dash_app.py

Predictive Analysis (Classification)

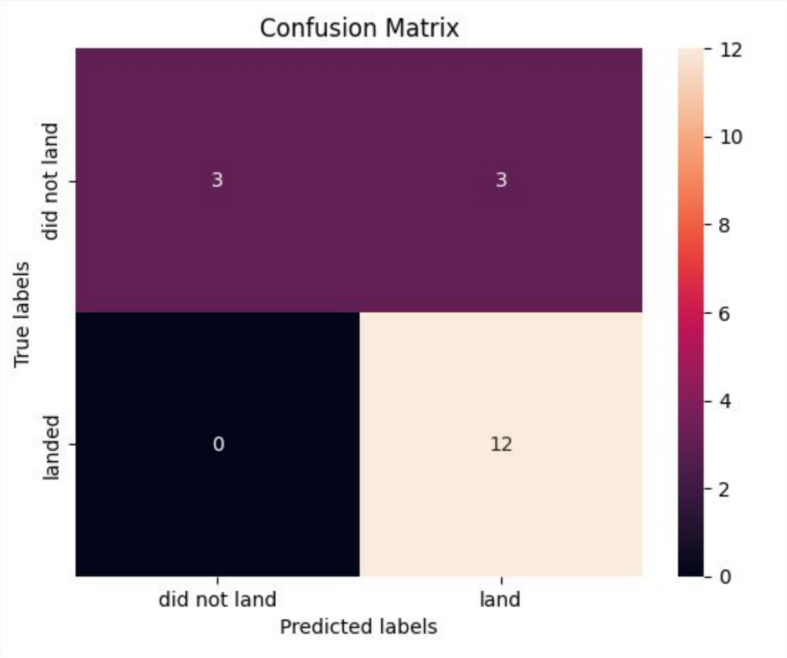
Git Hub URL :

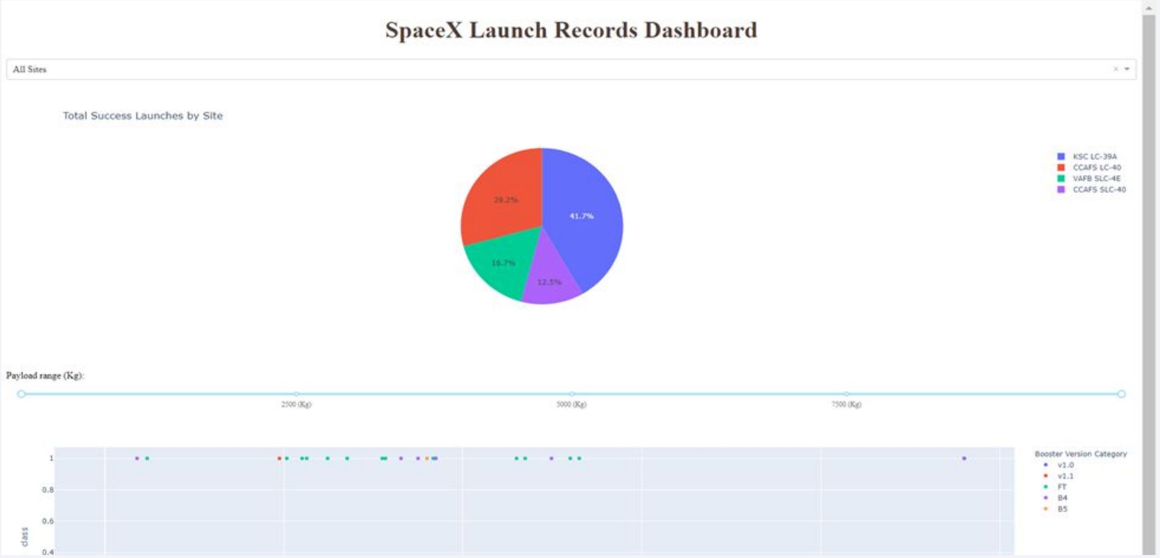
https://github.com/alchemistcohen/Applied-Data-Science-Capstone/blob/3f5998eecae6cac22d45bc521c489115a4c4618a/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



Results

This is a preview of the Plotly dashboard. The following sides will show the results of EDA with visualization, EDA with SQL, Interactive Map with Folium, and finally the results of our model with about 83% accuracy.

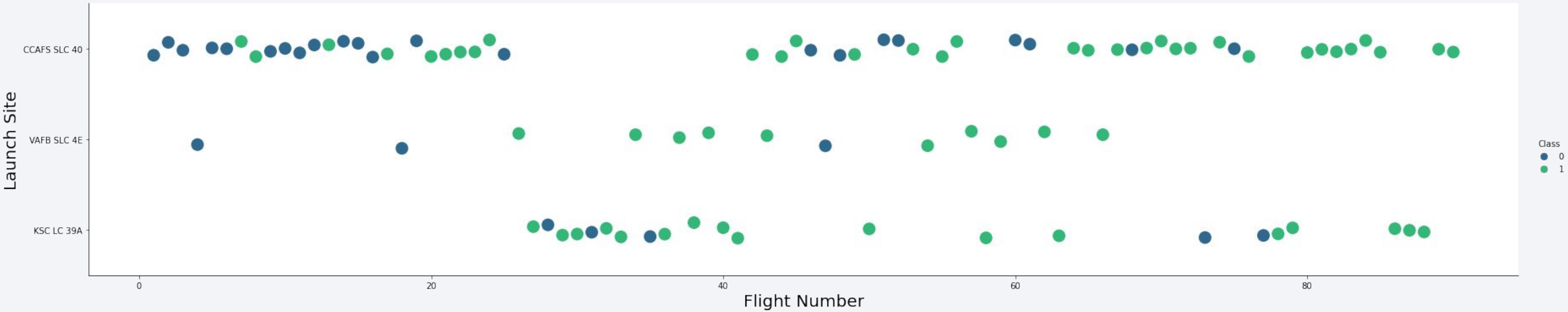




Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



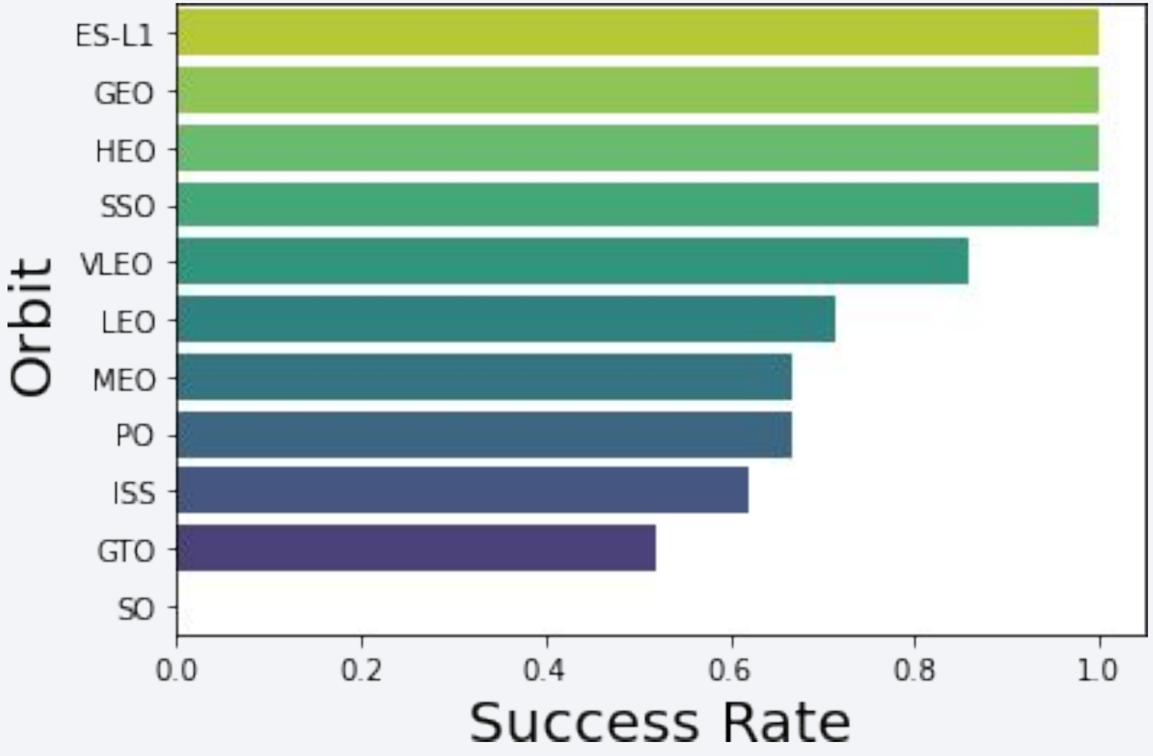
The graphic shows an increase in success rate over time (indicated in Flight Number).
Likely a big breakthrough around flight 20 which significantly increased success rate.
CCAFS appears to be the main launch site as it has the most volume.

Payload vs. Launch Site

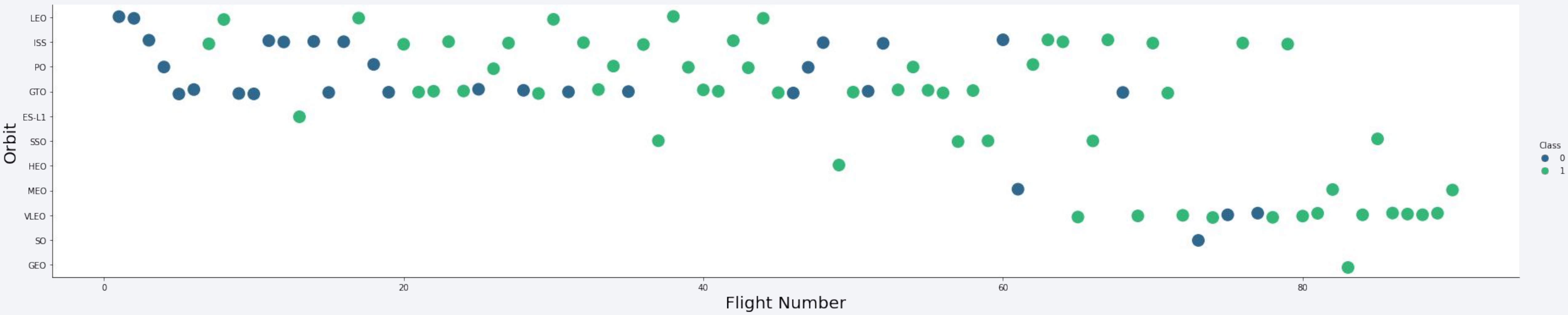
In the visual we can see the Payload mass appears to fall mostly between 0-6000 kg. Different launch sites also seem to use different payload mass.

Success Rate vs. Orbit Type

In possible to deduce in the chart
ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis) SSO (5) has 100% success rate
VLEO (14) has decent success rate and attempts
SO (1) has 0% success rate
GTO (27) has the around 50% success rate but largest sample



Flight Number vs. Orbit Type



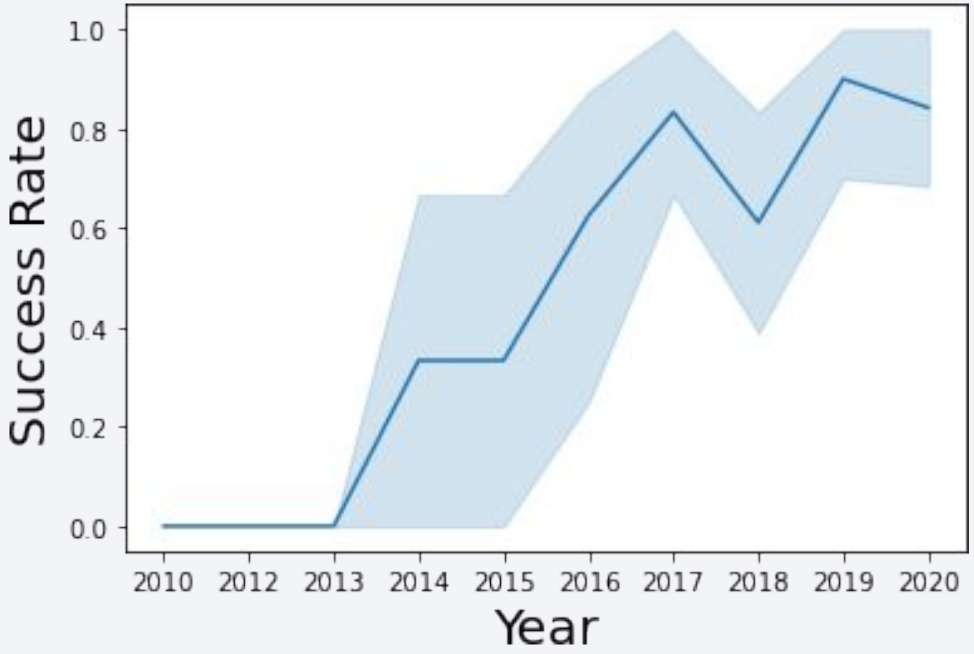
The chart shows Launch Orbit preferences changed over Flight Number. Launch Outcome seems to correlate with this preference.
SpaceX started with LEO orbits which saw moderate success LEO and returned to VLEO in recent launches
SpaceX appears to perform better in lower orbits or Sun-synchronous orbits

Payload vs. Orbit Type

Looking at the plot we can deduce
Payload mass seems to correlate with orbit
LEO and SSO seem to have relatively low payload mass
The other most successful orbit VLEO only has payload mass values in the higher end of the range

Launch Success Yearly Trend

Success generally increases over time since 2013 with a slight dip in 2018
Success in recent years at around 80%



All Launch Site Names

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

```
[28]: %sql select distinct Launch_Site from SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

[28]:

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

CCAFS SLC-40 and CCAFSSLC-40 likely all represent the same
launch site with data entry errors.
CCAFS LC-40 was the previous name. Likely only 3 unique launch_site values: CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E

Launch Site Names Begin with 'CCA'

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

[29]:
```
%sql select *  from SPACEXTABLE where Launch_Site like '%KSC%'limit 5
```

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

[29]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

Task 3

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

```
[30]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where customer like '%NASA (CRS)%'
```

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

[30]:

sum(PAYLOAD_MASS__KG_)
48213

This query sums the total payload mass in kg where NASA was the customer.
CRS stands for Commercial Resupply Services which indicates that these payloads were sent to the International Space Station (ISS).

Average Payload Mass by F9 v1.1

Task 4

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

[32]:

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version like '%F9 v1.1%'
```

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

[32]:

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

This query calculates the average payload mass or launches which used booster version F9 v1.1
Average payload mass of F9 1.1 is on the low end of our payload mass range

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
[34]: %sql select min(date) from SPACEXTABLE where Landing_Outcome like '%ground pad%'
```

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

[34]:

min(date)
2015-12-22

This query returns the first successful ground pad landing date. First ground pad landing wasn't until the end of 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

[37]:
```
%sql select booster_version from SPACEXTABLE where Landing_Outcome like 'Success (ground pad)' AND PAYLOAD_MASS__KG_ <6000 AND PAYLOAD_MA
```

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

[37]:

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

[38]:
```
%sql select Mission_Outcome,count(*) from SPACEXTABLE group by Mission_Outcome
```

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

[38]:

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

This query returns a count of each
mission outcome.
SpaceX appears to achieve its mission outcome nearly 98% of the time.
This means that most of the landing
failures are intended.
Interestingly, one launch has an unclear payload status and unfortunately one failed in flight.

Boosters Carried Maximum Payload

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

[43]:
```
#%sql select distinct Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE
%sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

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

[43]:

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

This query returns the booster versions that carried the highest payload mass of 15600 kg.
These booster versions are very similar and all are of the F9 B5 B10xx.x variety.
This likely indicates payload mass correlates with the booster version that is used.

2015 Launch Records

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT strftime('%m', DATE) AS month, Landing_Outcome, booster_version, launch_site FROM SPACEXTABLE WHERE Landing_Outcome LIKE '
```

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

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

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.

[52]:
```
%sql SELECT Landing_Outcome, COUNT(*) as outcome_count FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing
```

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

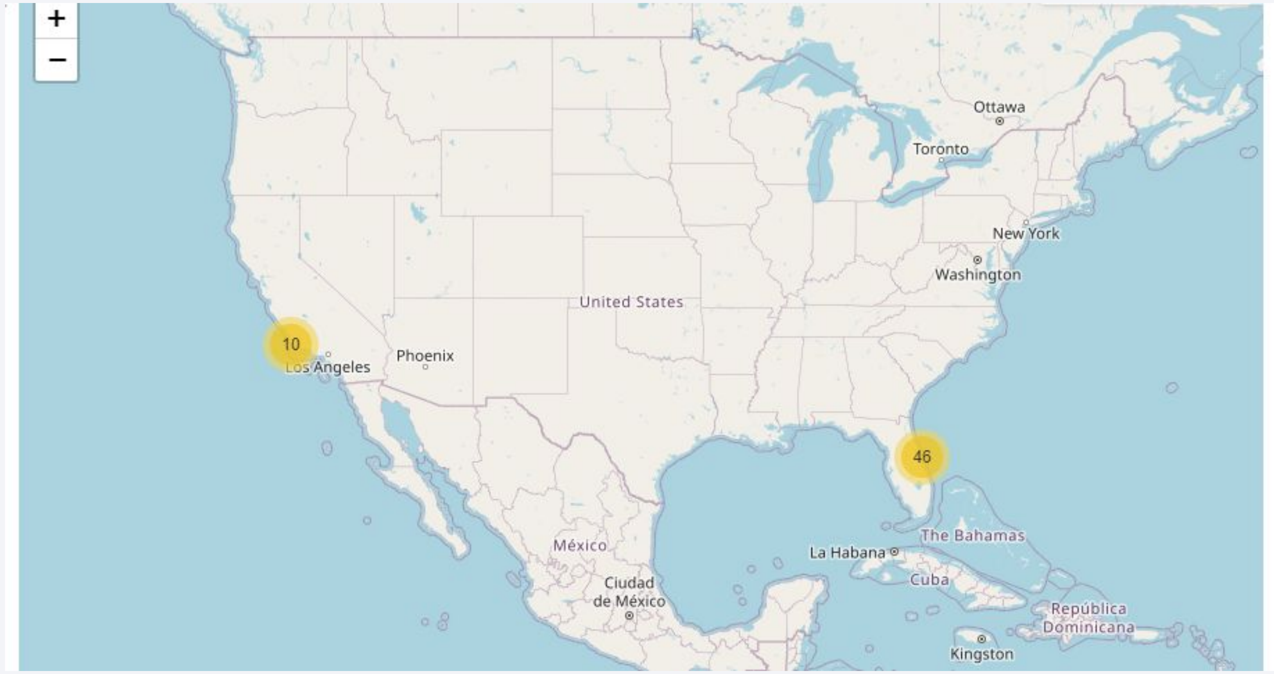
[52]:

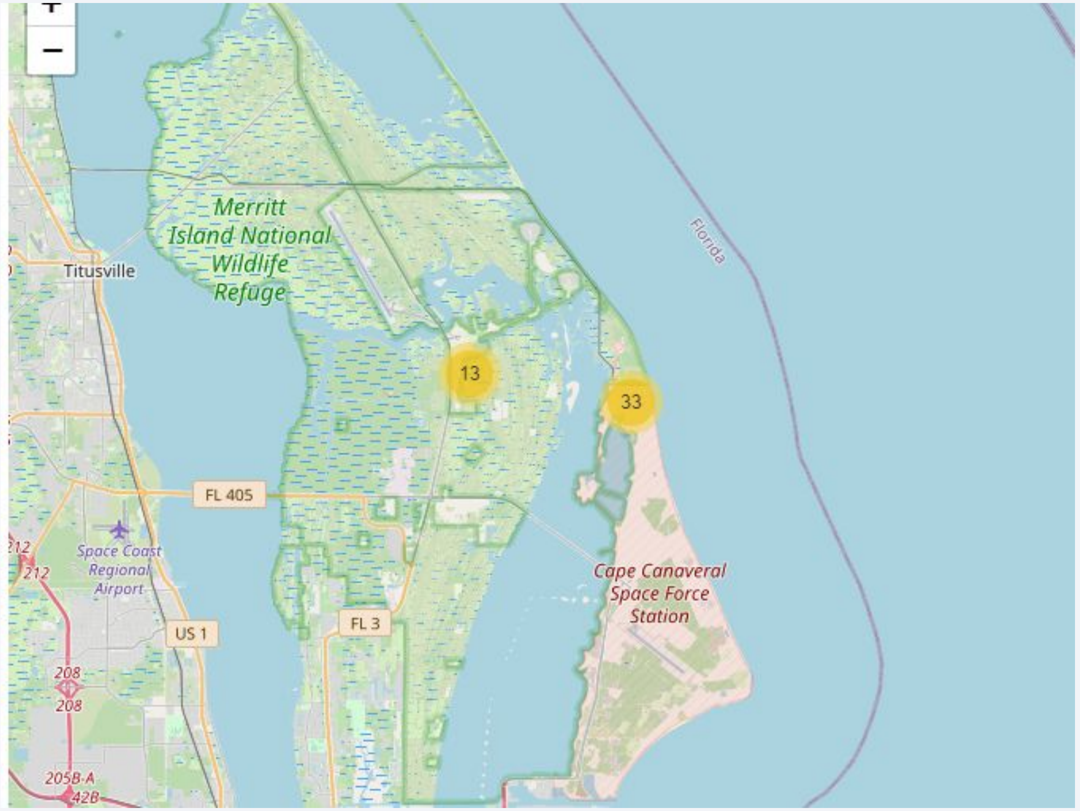
Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

Launch Site Locations

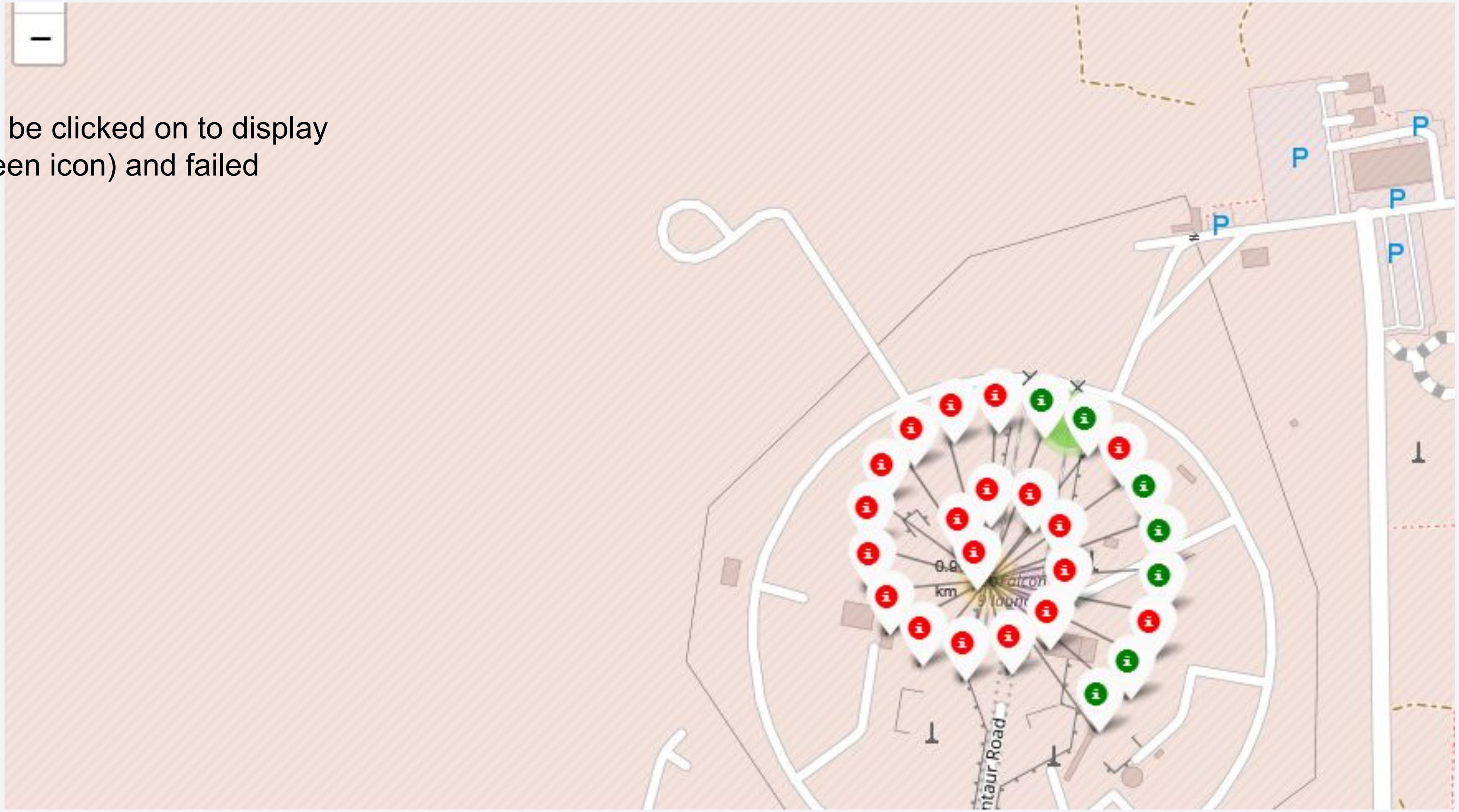




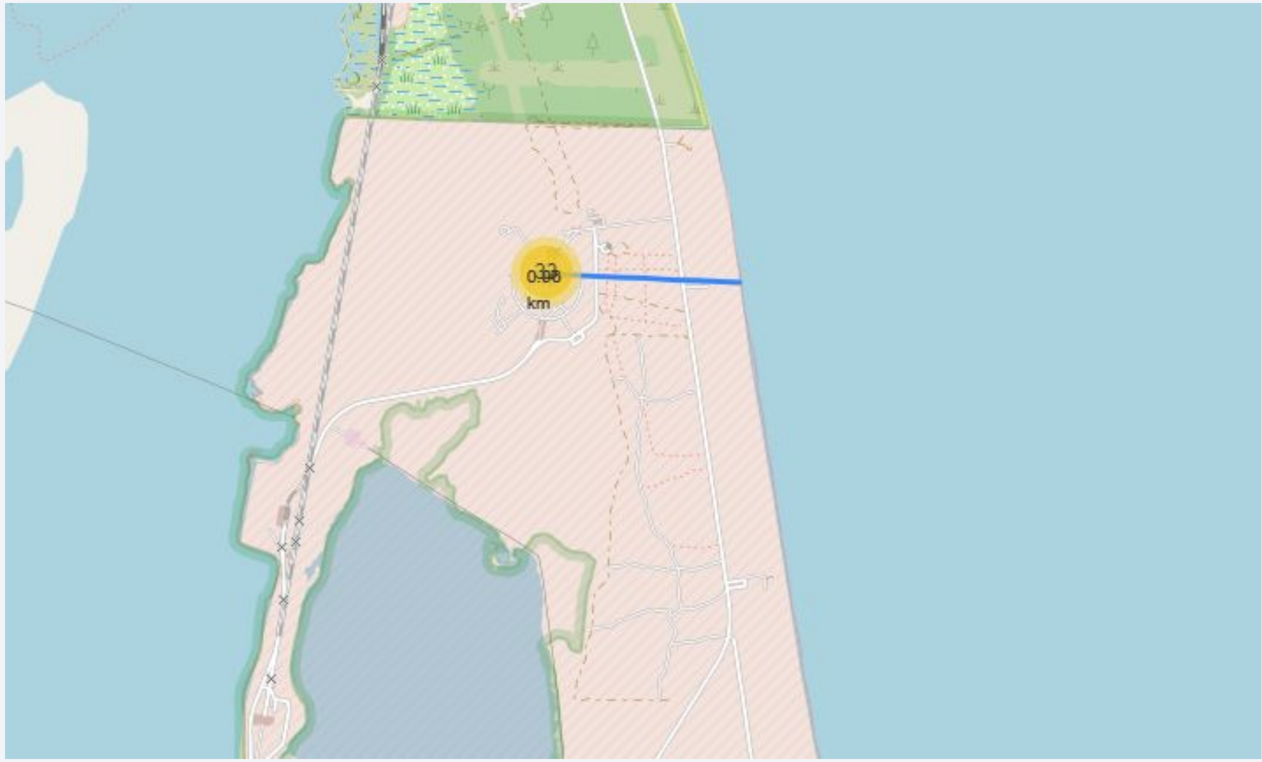
The left map shows all launch sites relative US map. The right map shows the two Florida launch sites since they are very close to each other. All launch sites are near the ocean.

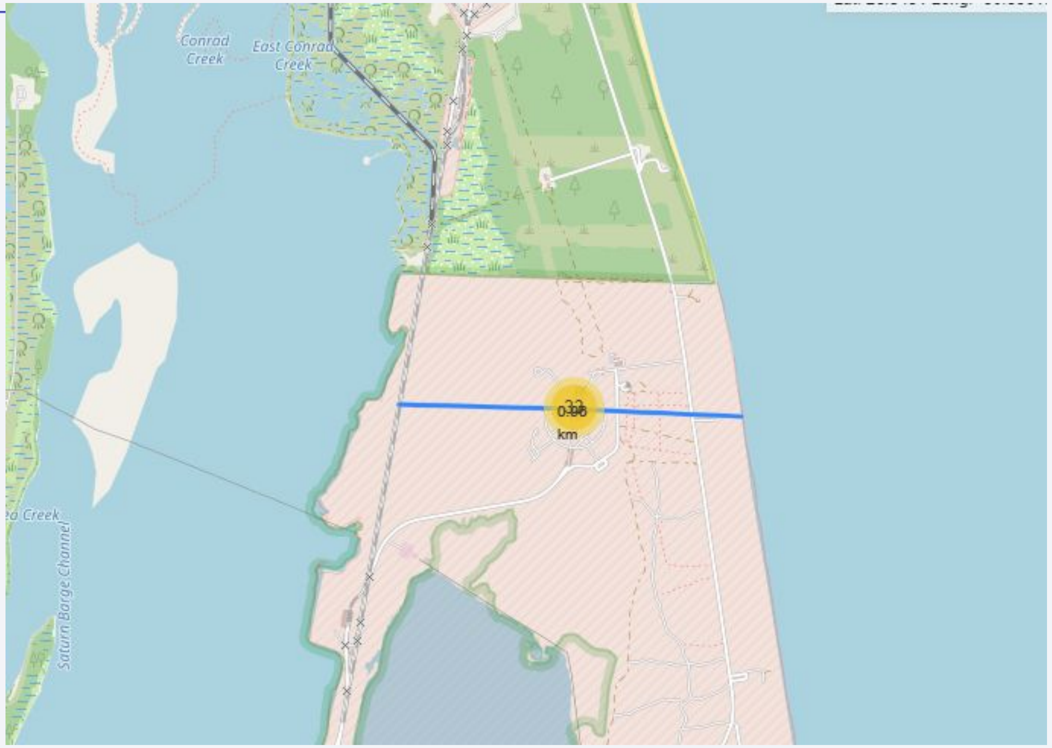
Color-Labeled Launch Markers

Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed landing (red icon).



Key Location Proximities





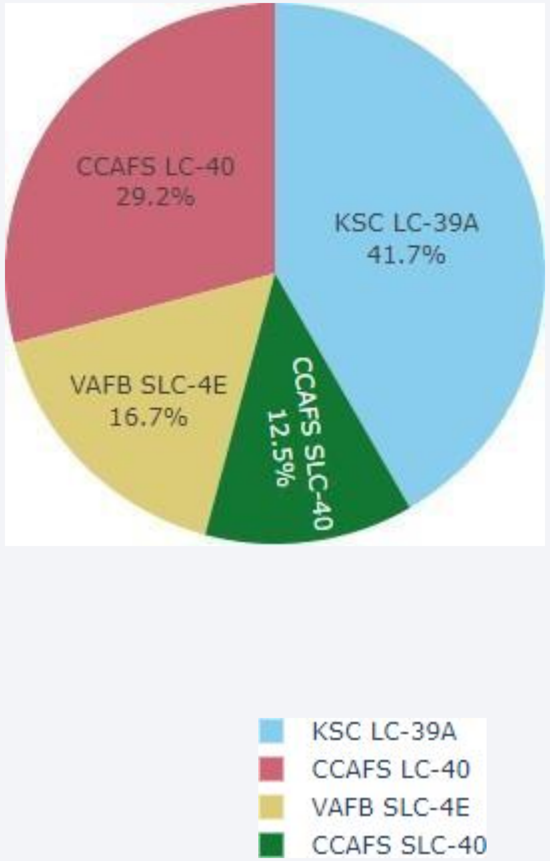
sing CCAFS SLC-40 as an example, launch sites are very close to railways for large part and supply transportation. Launch sites are close to highways for human and supply transport. Launch sites are also close to coasts and relatively far from cities so that launch failures can land in the sea to avoid rockets falling on densely populated areas.

Section 4

Build a Dashboard with Plotly Dash

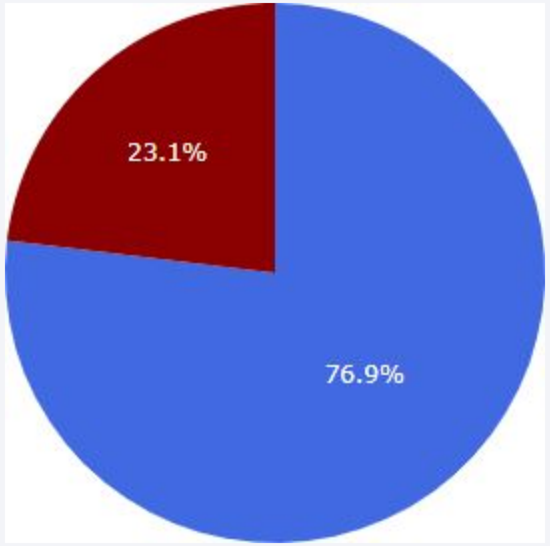
Launch success count for all sites

This is the distribution of successful landings across all launch sites. CCAFS LC-40 is the old name of CCAFS SLC-40 so CCAFS and KSC have the same amount of successful landings, but a majority of the successful landings where performed before the name change. VAFB has the smallest share of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast.



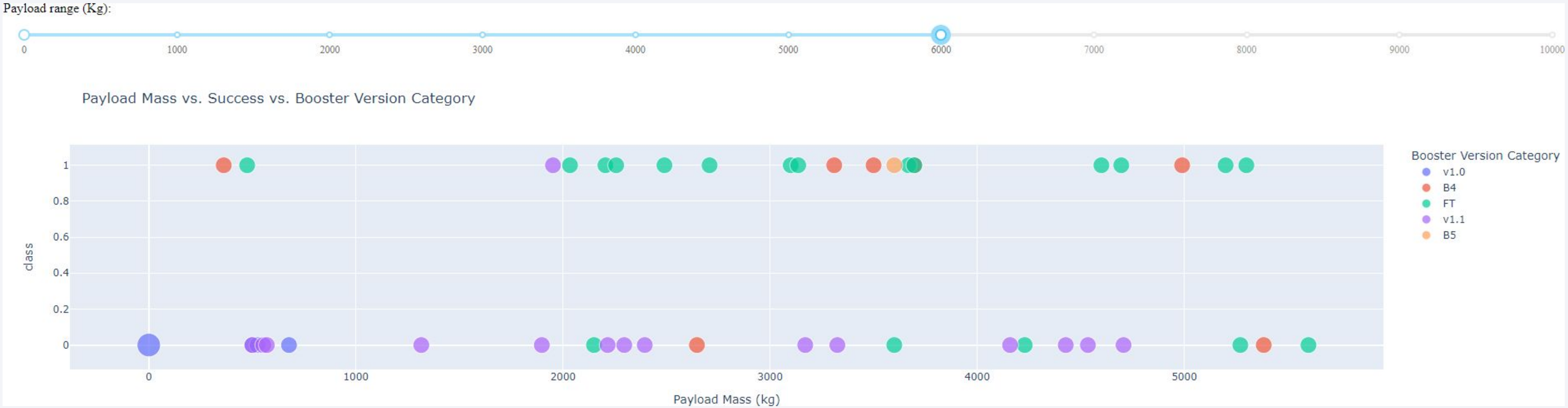
Highest Launch Success Ratio

KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.



KSC LC-39A Success Rate (blue=success)

Payload vs. Launch Outcome

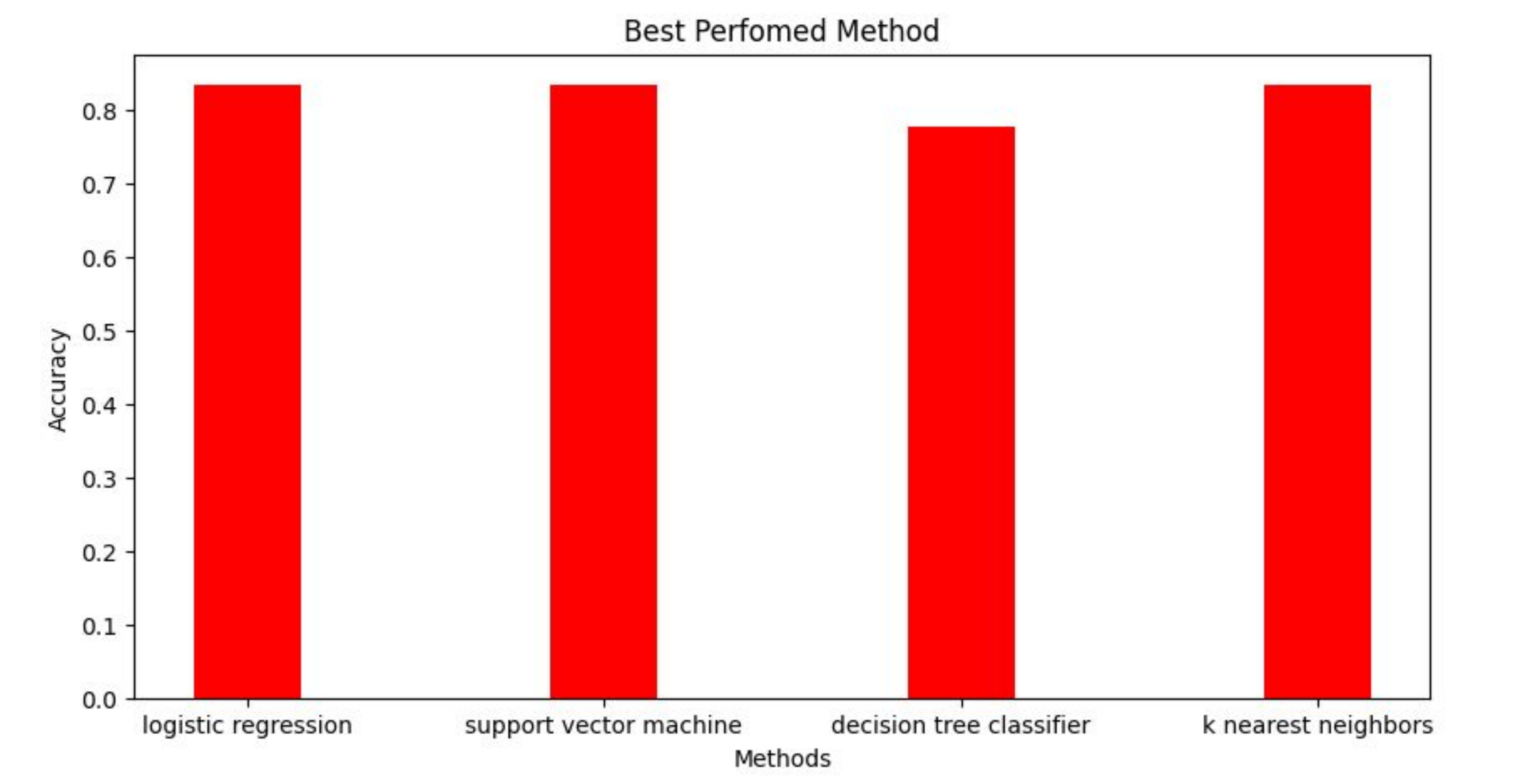


Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size. In this particular range of 0-6000, interestingly there are two failed landings with payloads of zero kg.

Section 5

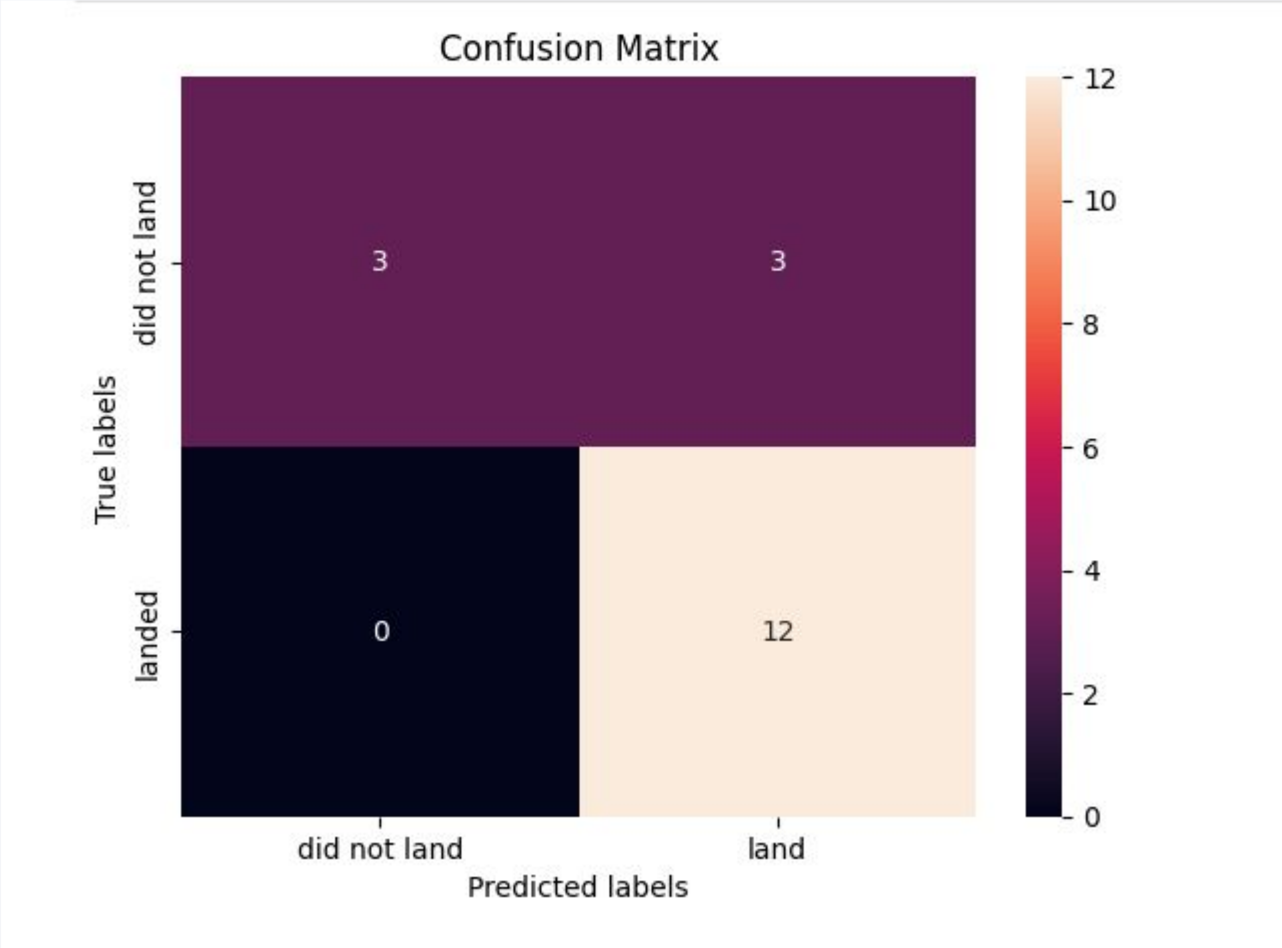
Predictive Analysis (Classification)

Classification Accuracy



All models had virtually the same accuracy on the test set at 83.33% accuracy. It should be noted that test size is small at only sample size of 18.
This can cause large variance in accuracy results, such as those in Decision Tree Classifier model in repeated runs.
We likely need more data to determine the best model.

Confusion Matrix



Since all models performed the same for the test set, the confusion matrix is the same across all models. The models predicted 12 successful landings when the true label was successful landing.
The models predicted 3 unsuccessful landings when the true label was unsuccessful landing.
The models predicted 3 successful landings when the true label was unsuccessful landings (false positives). Our models over predict successful landings.

Conclusions

We develop a machine learning model for Space Y who wants to bid against SpaceX

The goal of model is to predict when Stage 1 will successfully land to save ~$100 million USD

Used data from a public SpaceX API and web scraping SpaceX Wikipedia page

Created data labels and stored data into a DB2 SQL database

Created a dashboard for visualization

We created a machine learning model with an accuracy of 83%

Allon Mask of SpaceY can use this model to predict with relatively high accuracy whether a launch will have a successful Stage 1 landing before launch to determine whether the launch should be made or not
If possible more data should be collected to better determine the best machine learning model and improve accuracy

Appendix

Git Hub Repository

https://github.com/alchemistcohen/Applied-Data-Science-Capstone

Thank you!