# Winning Space Race with Data Science

Yeltay Zhastay
2022-04-16

# Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

# Executive Summary

Collected information from the open Space API and the SpaceX Wikipedia page. A column of names "class" has been made, in which valid receipts are classified. I studied information using SQL, visualization, folio maps and dashboards. Important columns have been collected to be used as highlights. Changed all categorical coefficients to double using one hot encoding. Standardized information and the use of GridSearchCV to determine the best parameters for machine learning models. Visualize an estimate of the accuracy of all models.

Four machine learning models were presented: Computed Recurrence, Vector Support Machine, Choice Tree Classifier and K Nearest Neighbors. All created images were obtained with an accuracy of approximately 83.33%. All models exceeded the expected fruitful receipts. More information is needed to better demonstrate confidence 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

# Methodology

OVERVIEW OF DATA COLLECTION, WRANGLING, VISUALIZATION, DASHBOARD,  AND  MODEL  METHODS

# Methodology

- 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, Time

# Data Collection – SpaceX API

Request (Space X APIs)

.JSON file + Lists(Launch Site, Booster Version, Payload Data)

Json_normalize to DataFrame data from JSON

Dictionary relevant data

Cast dictionary to a DataFrame

Filter data to only include Falcon 9 launches

Imputate missing PayloadMass values with mean

Github url: https://github.com/yeltayzhastay/IBMDataScienceProfessionalCertification/blob/main/c10/w1/jupyter-labs-spacex-data-collection-api.ipynb

# Data Collection - Scraping

Request Wikipedia html

BeautifulSoup html5lib Parser

Find launch info  html table

Cast dictionary to  DataFrame

Iterate through  table cells to  extract data to  dictionary

Create dictionary

Github url:
https://github.com/yeltayzhastay/IBMDataScienceProfessionalCertification/blob/main/c10/w1/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/yeltayzhastay/IBMDataScienceProfessionalCertification/blob/main/c10/w1/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/yeltayzhastay/IBMDataScienceProfessionalCertification/blob/main/c10/w2/jupyter-labs-eda-dataviz.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 url:
https://github.com/yeltayzhastay/IBMDataScienceProfessionalCertification/blob/main/c10/w2/jupyter-labs-eda-sql-coursera.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/yeltayzhastay/IBMDataScienceProfessionalCertification/blob/main/c10/w3/lab_jupyter_launch_site_location.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/yeltayzhastay/IBMDataScienceProfessionalCertification/blob/main/c10/w3/spacex_dash_app.py

# Predictive Analysis (Classification)

Split label column ‘Class’ from dataset

Fit and Transform Features using Standard Scaler

Train_test_split data

Score models on split test set

Use GridSearchCV on LogReg, SVM, Decision Tree, and KNN models

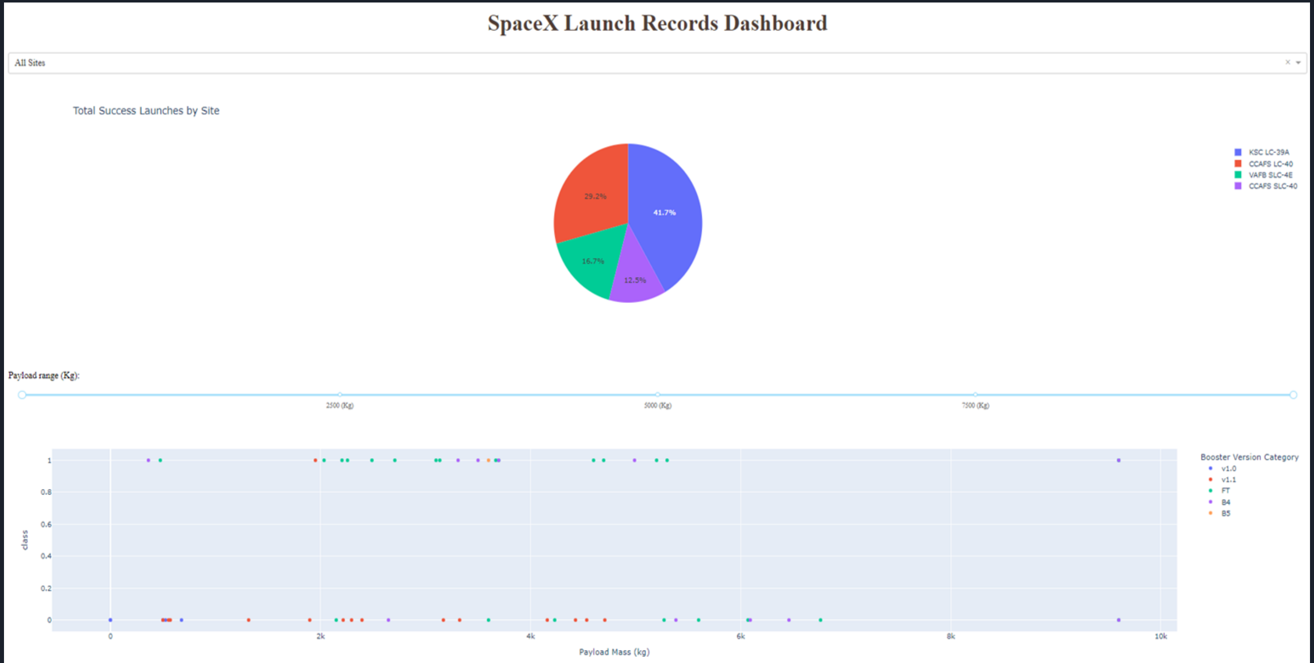
GridSearchCV (cv=10) to find  optimal parameters

Confusion Matrix for all models

Barplot to compare  scores of models

Github url: https://github.com/yeltayzhastay/IBMDataScienceProfessionalCertification/blob/main/c10/w4/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

# Results



SpaceX Launch Records Dashboard

All Sites

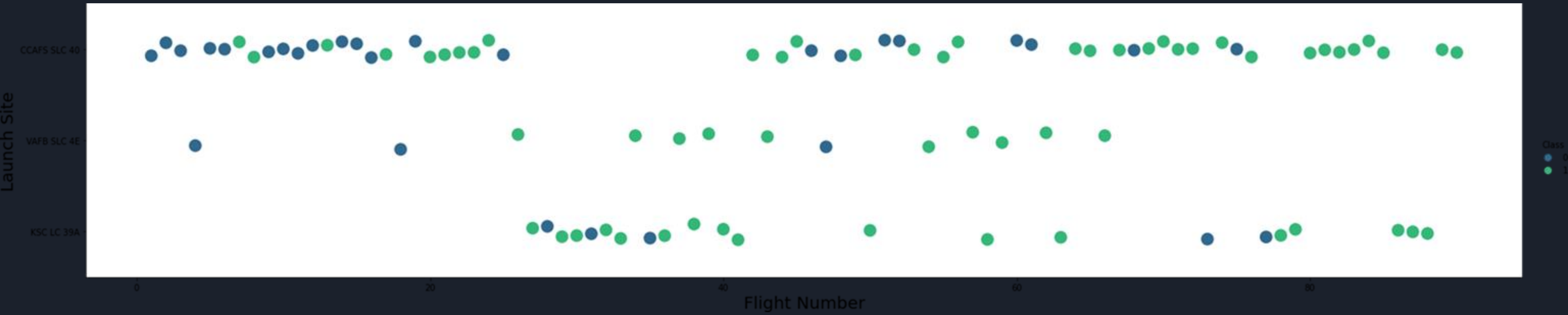
Payload range (Kg):

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.

# EDA with Visualization

EXPLORATORY DATA ANALYSIS WITH SEABORN PLOTS

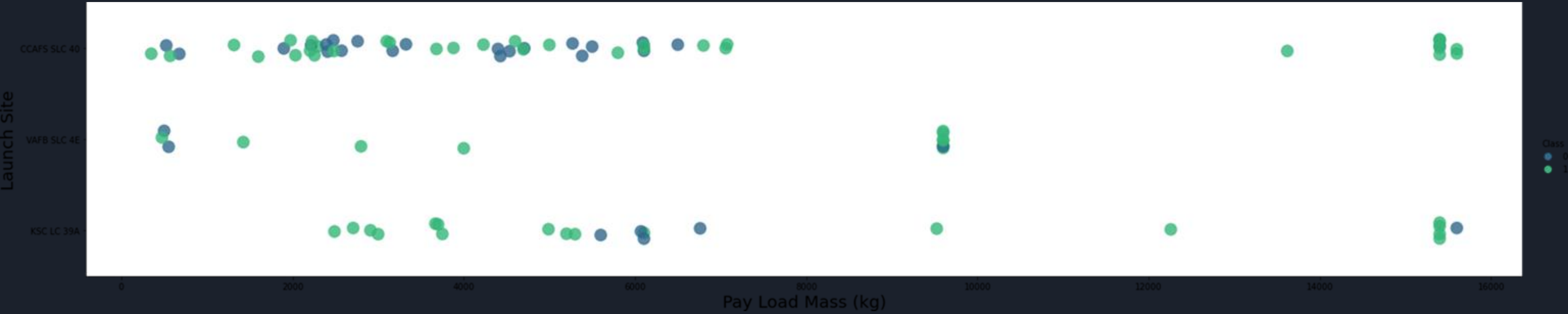
# Flight Number vs. Launch Site



Green indicates successful launch; Purple indicates unsuccessful launch.

Graphic suggests 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

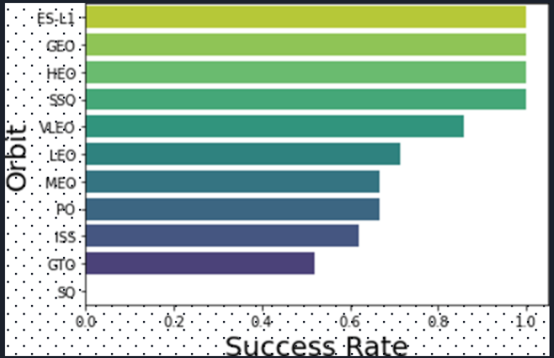


Green indicates successful launch; Purple indicates unsuccessful launch.

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

Success Rate Scale with  0 as 0%
0.6 as 60%  1 as 100%



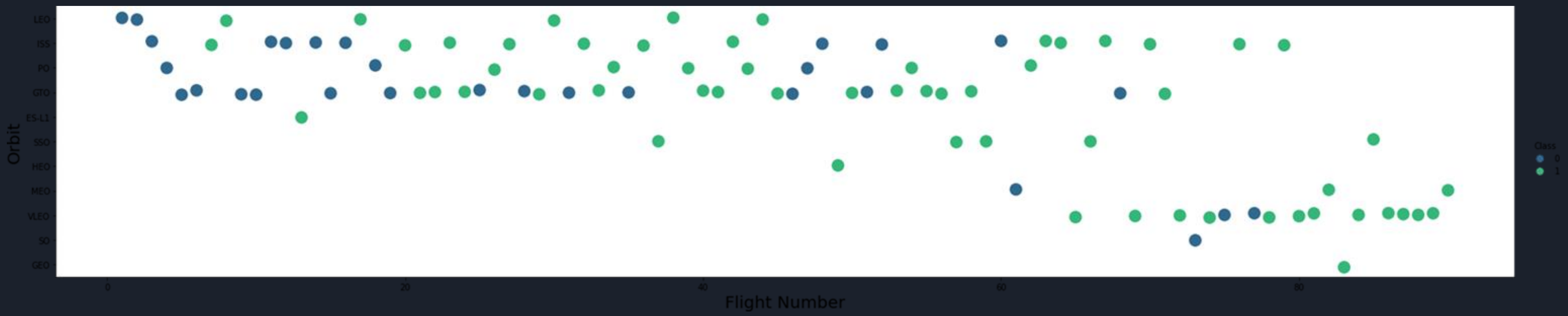
ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis)  SSO (5) has 100% success rate

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

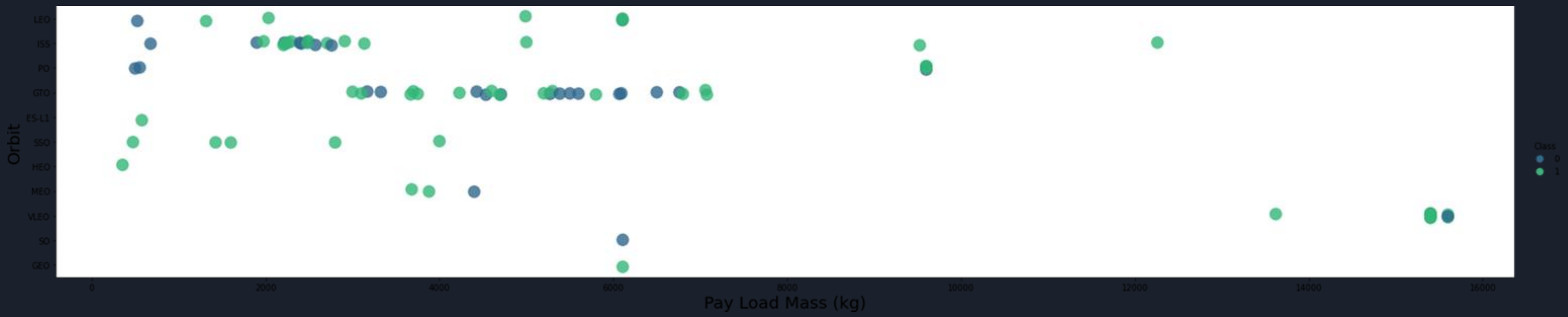


Green indicates successful launch; Purple indicates unsuccessful launch.

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



Green indicates successful launch; Purple indicates unsuccessful launch.

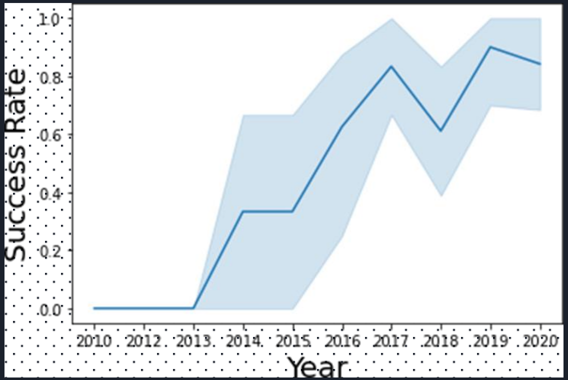
Payload mass seems to correlate with orbit

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

95% confidence interval  (light blue shading)



Success generally increases over time since 2013 with a slight dip in 2018

Success in recent years at around 80%

# EDA with SQL

EXPLORATORY DATA ANALYSIS WITH SQL DB2
INTEGRATED IN PYTHON WITH SQLALCHEMY

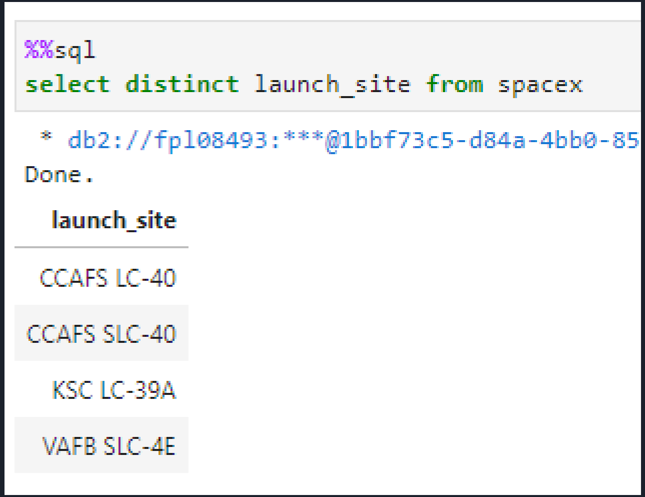
# All Launch Site Names

Query unique launch site names from database.

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



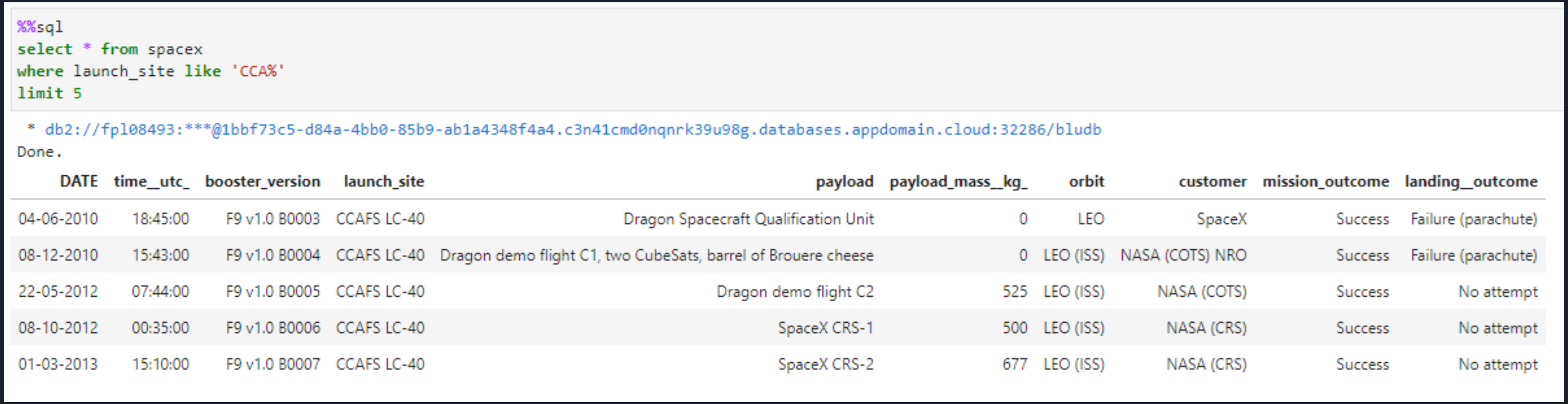
```
%%sql
select distinct launch_site from spacex
```

```
 * db2://fpl08493:***@1bbf73c5-d84a-4bb0-85
Done.
```

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

# Launch Site Names Beginning with `CCA`

First five entries in database with Launch Site name beginning with CCA.



```
%%sql
select * from spacex
where launch_site like 'CCA%'
limit 5
```

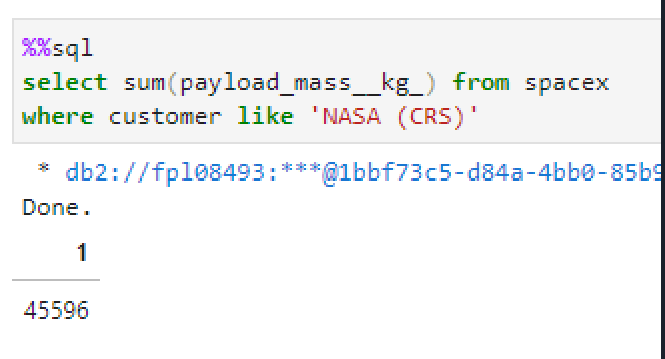
```
 * db2://fpl08493:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/bludb
Done.
```

| DATE | time__utc_ | booster_version | launch_site | payload | payload_mass__kg_ | orbit | customer | mission_outcome | landing__outcome |
|---|---|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 08-12-2010 | 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) |
| 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 |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass from NASA

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).



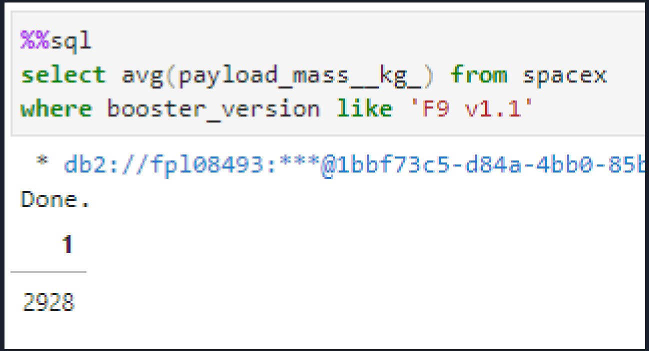
```
%%sql
select sum(payload_mass__kg_) from spacex
where customer like 'NASA (CRS)'
```

```
 * db2://fpl08493:***@1bbf73c5-d84a-4bb0-85b9
Done.
    1
45596
```

# Average Payload Mass by F9 v1.1

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



```
%%sql
select avg(payload_mass__kg_) from spacex
where booster_version like 'F9 v1.1'
```

```
 * db2://fpl08493:***@1bbf73c5-d84a-4bb0-85b
Done.
   1
2928
```

# First Successful Ground Pad Landing Date

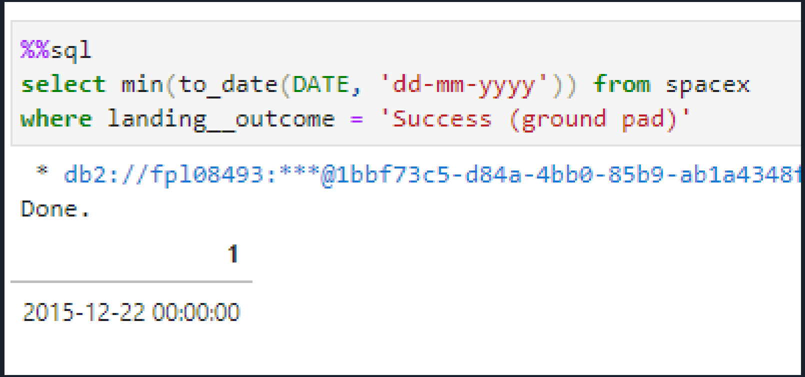
This query returns the first  successful ground pad landing  date.

First ground pad landing wasn’t

until the end of 2015.

Successful landings in general

appear starting 2014.

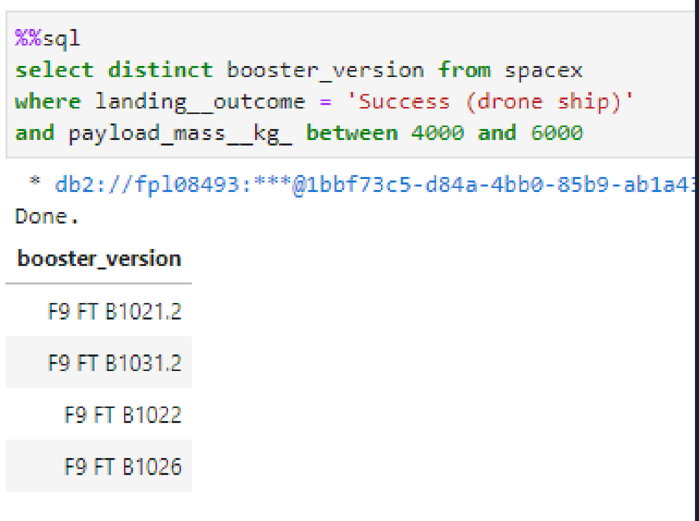


```
%%sql
select min(to_date(DATE, 'dd-mm-yyyy')) from spacex
where landing__outcome = 'Success (ground pad)'
```

```
 * db2://fpl08493:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f
Done.
                  1
2015-12-22 00:00:00
```

# Successful Drone Ship Landing with Payload Between 4000 and 6000

This query returns the four  booster versions that had  successful drone ship landings  and a payload mass between  4000 and 6000 noninclusively.

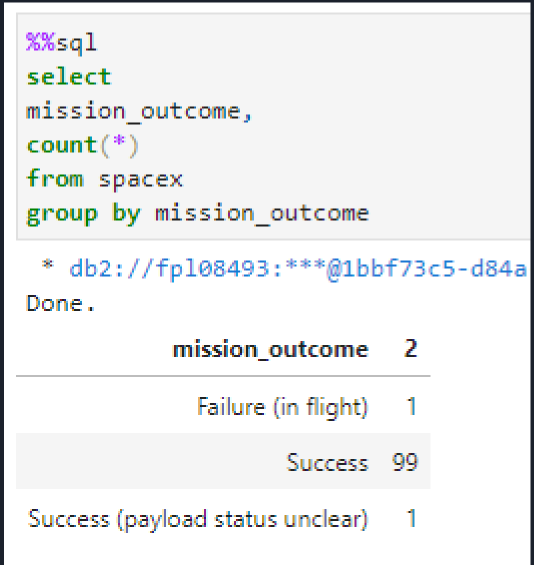


```
%%sql
select distinct booster_version from spacex
where landing__outcome = 'Success (drone ship)'
and payload_mass__kg_ between 4000 and 6000
```

```
 * db2://fpl08493:***@1bbf73c5-d84a-4bb0-85b9-ab1a4
Done.
```

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

# Total Number of Each Mission Outcome



```
%%sql
select
mission_outcome,
count(*)
from spacex
group by mission_outcome
```

```
 * db2://fpl08493:***@1bbf73c5-d84a
Done.
```

| mission_outcome | 2 |
|---:|---:|
| Failure (in flight) | 1 |
| Success | 99 |
| Success (payload status unclear) | 1 |

This query returns a count of each mission outcome.

SpaceX appears to achieve its mission outcome nearly 99% of the time. This means that most of the landing failures are intended.

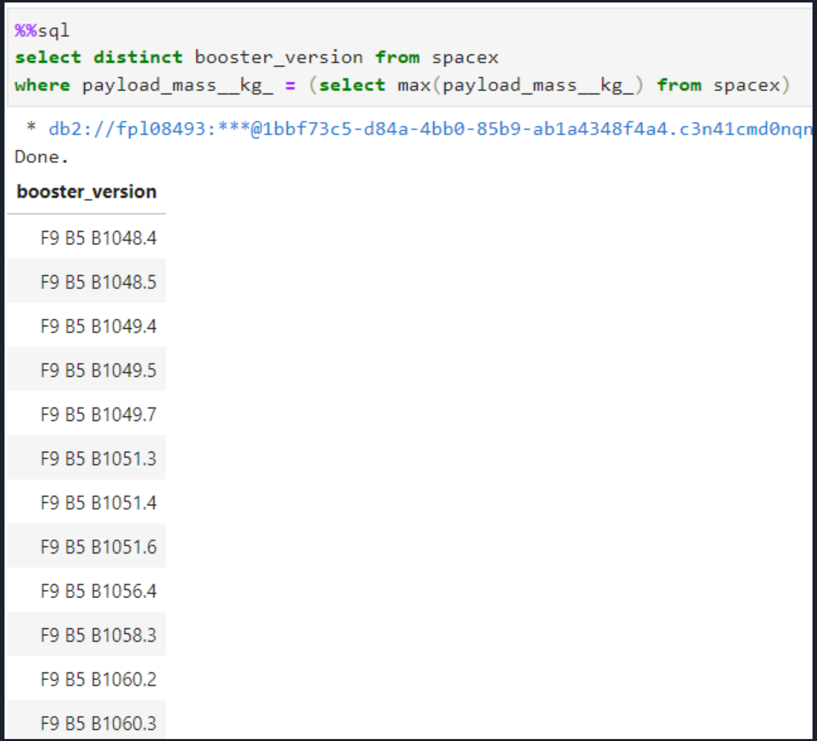
Interestingly, one launch has an unclear payload status and unfortunately one failed in flight.

# Boosters that Carried Maximum Payload

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.



```
%%sql
select distinct booster_version from spacex
where payload_mass__kg_ = (select max(payload_mass__kg_) from spacex)
```

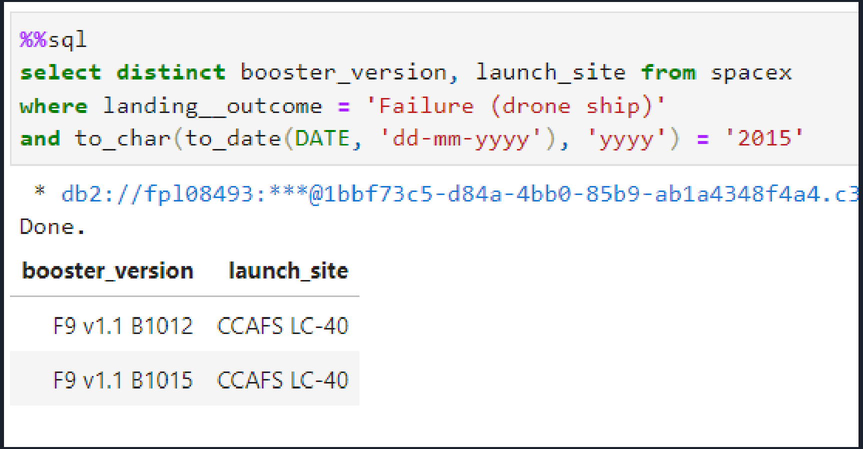
```
 * db2://fpl08493:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqn
Done.
```

| 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 Failed Drone Ship Landing Records

This query returns the Booster Version, Payload Mass (kg), and Launch site of 2015 launches where stage 1 failed to land on a drone ship.

There were two such occurrences.

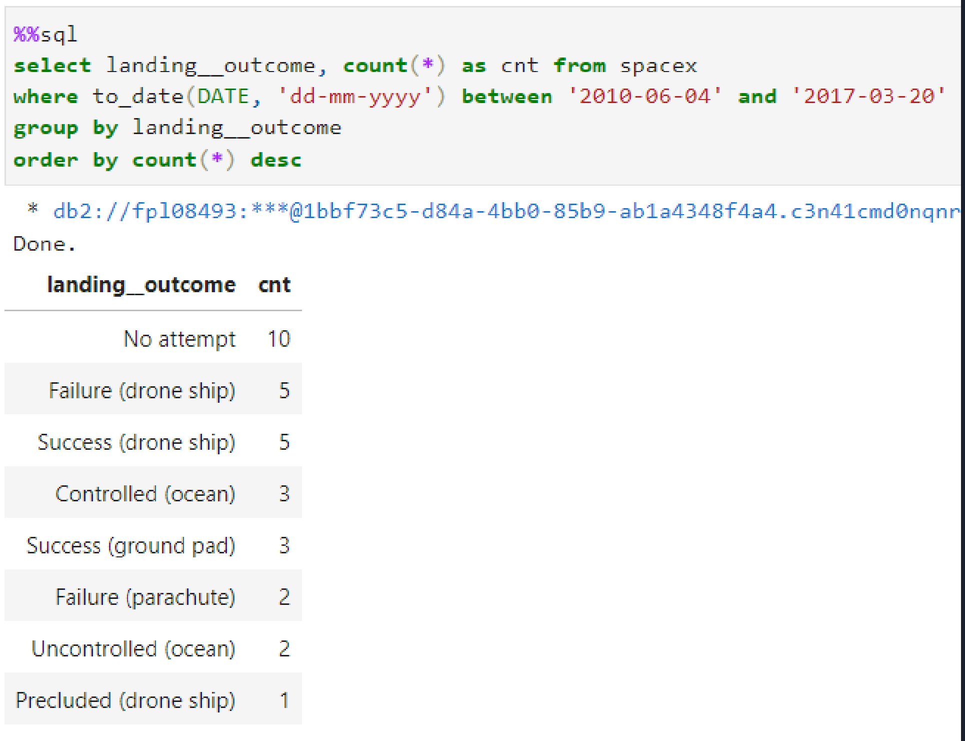


```
%%sql
select distinct booster_version, launch_site from spacex
where landing__outcome = 'Failure (drone ship)'
and to_char(to_date(DATE, 'dd-mm-yyyy'), 'yyyy') = '2015'
```

```
 * db2://fpl08493:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3
Done.
```

| booster_version | launch_site |
|---|---|
| F9 v1.1 B1012 | CCAFS LC-40 |
| F9 v1.1 B1015 | CCAFS LC-40 |

# Ranking Counts of Successful Landings Between 2010-06-04 and 2017-03-20



```
%%sql
select landing__outcome, count(*) as cnt from spacex
where to_date(DATE, 'dd-mm-yyyy') between '2010-06-04' and '2017-03-20'
group by landing__outcome
order by count(*) desc
```

```
 * db2://fpl08493:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnr
Done.
```

| landing__outcome | cnt |
|---:|---:|
| 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 |

This query returns a list of successful landings and between 2010-06-04 and 2017-03-20 inclusively.

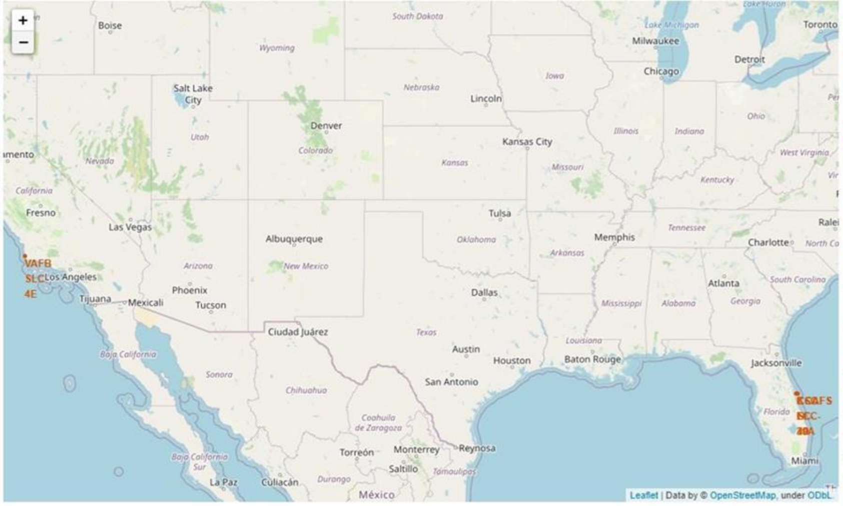
There are two types of successful landing outcomes: drone ship and ground pad landings.

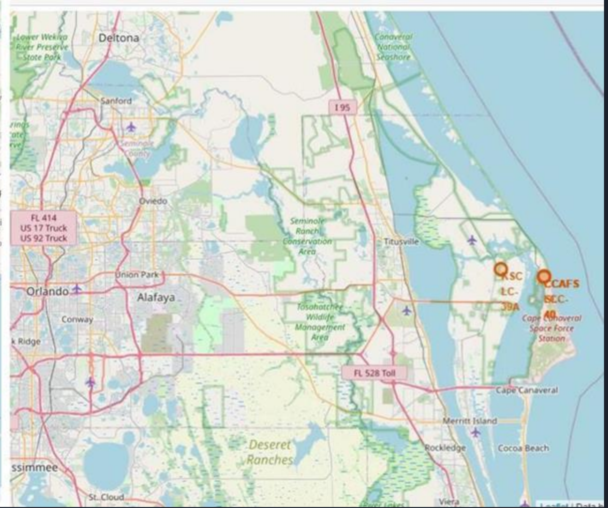
There were 8 successful landings in total during this time period

# Interactive Map with Folium

# 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.

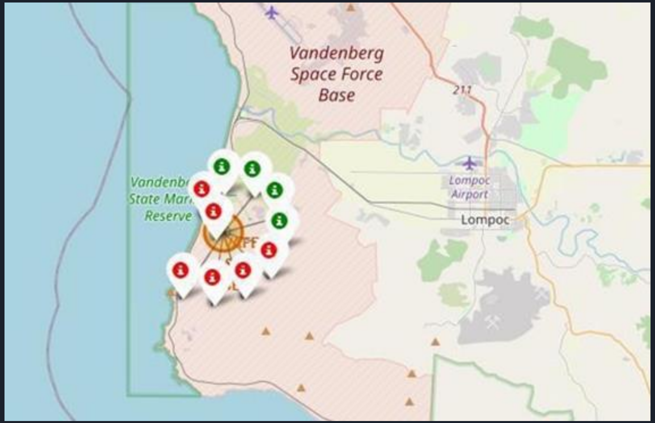




# Color-Coded Launch Markers

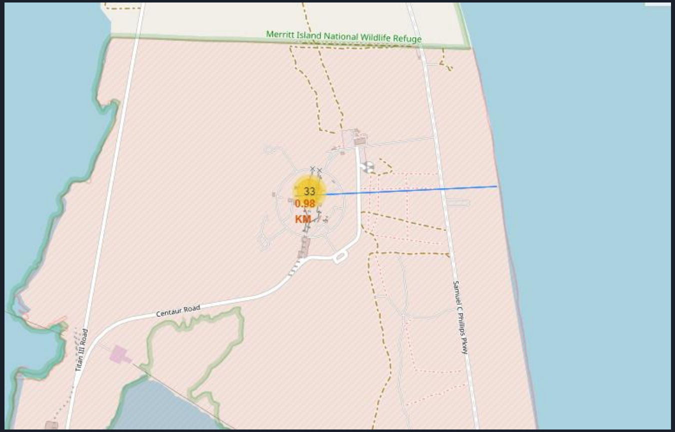
Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed

landing (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failed landings.



# Key Location Proximities

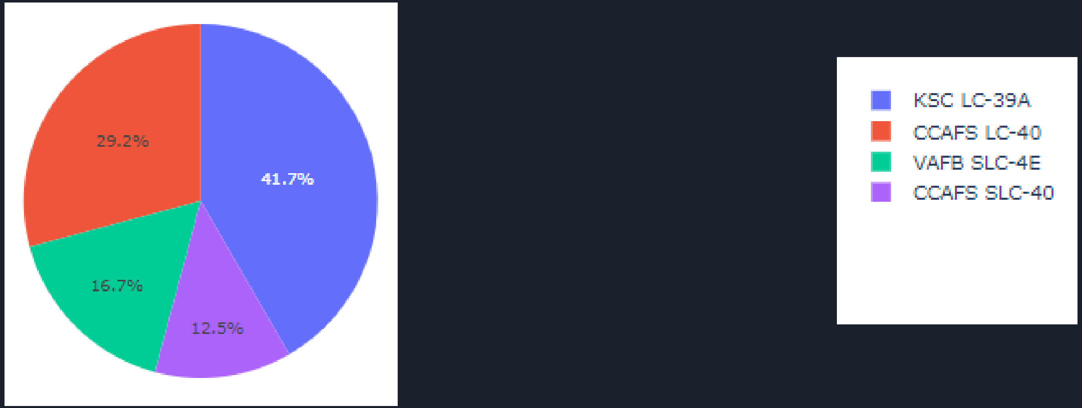
Using Launch site as an example, launch sites are very close to railways for large part and supply transportation. Launch sites are close to highways for human and supply transport. Launch sites are also close to coasts and relatively far from cities so that launch failures can land in the sea to avoid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



# Build a Dashboard with Plotly Dash

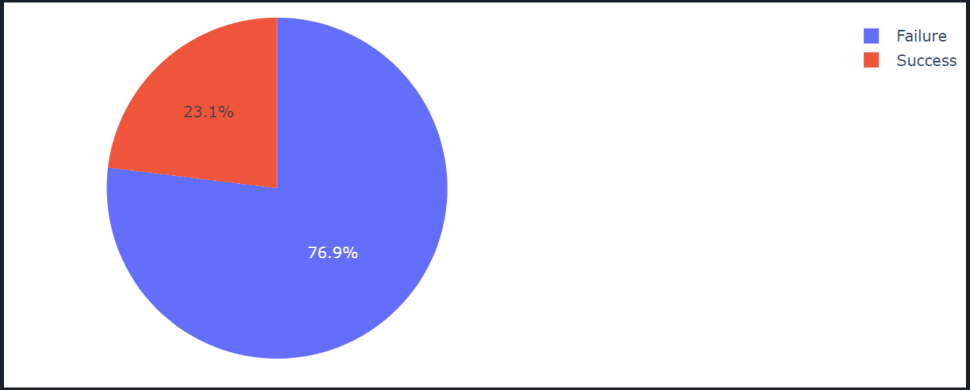
# Successful Launches Across Launch Sites

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.



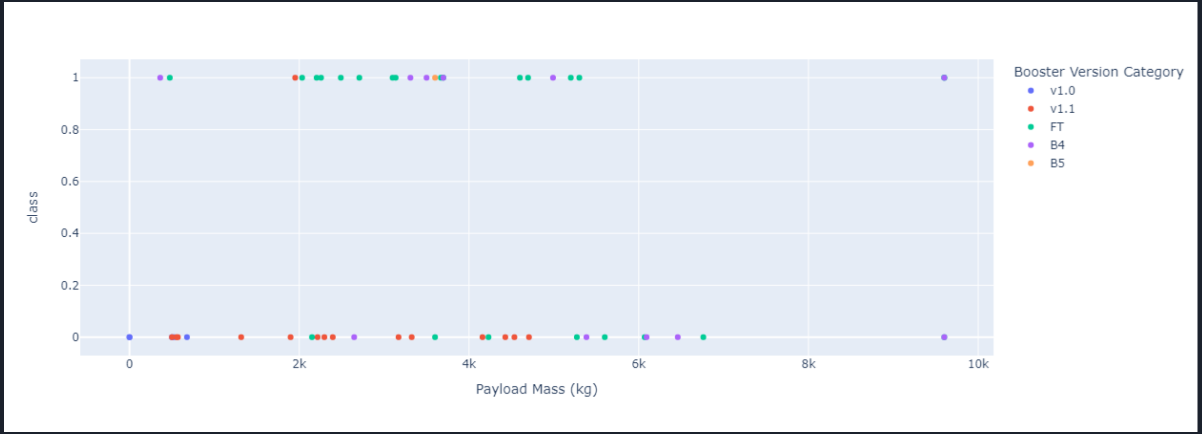
# Highest Success Rate Launch Site

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



# Payload Mass vs. Success vs. Booster Version Category

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.



# Predictive Analysis (Classification)

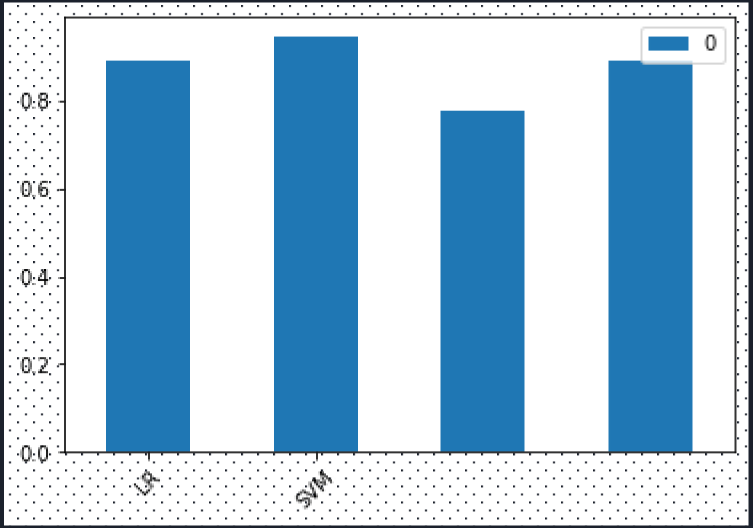
GRIDSEARCHCV(CV=10) ON LOGISTIC REGRESSION, SVM, DECISION
TREE, AND KNN

# Classification Accuracy

All models had virtually the same accuracy on the test set at 94.44% 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.



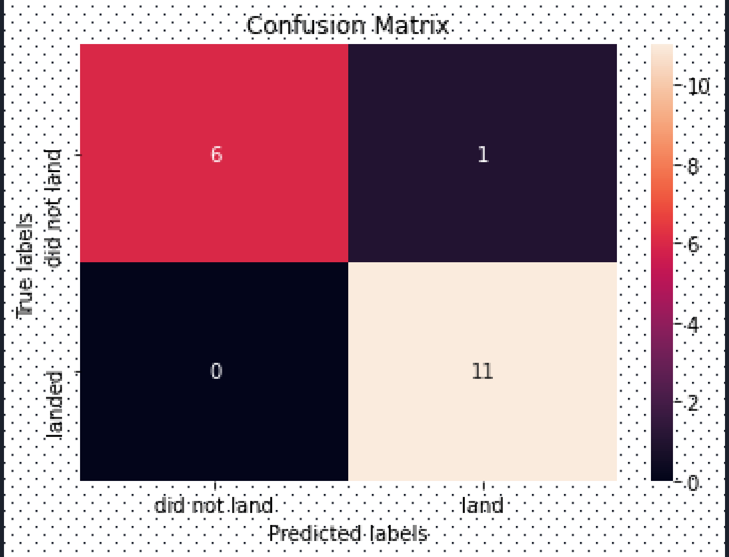
# Confusion Matrix

Correct predictions are  on a diagonal from top  left to bottom right.

Since all models performed the same for the test set, the confusion matrix is the same across all models.  The models predicted 11 successful landings when the true label was successful landing.

The models predicted 1 unsuccessful landings when the true label was unsuccessful landing.

Our models over predict successful landings.



# CONCLUSION

- Our task: to 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

GitHub repository url:

https://github.com/yeltayzhastay/IBMDataScienceProfessionalCertification

Thanks to All Instructors:

🎉🎉😎😎