IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yanqing Jiang
11/27/2021

Outline

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

Executive Summary

➢Summary of methodologies

- Data Collection & Wrangling
- Data Exploration with Visualization
- Interactive Visualization/Dashboard with Folium and Plotly
- Machine Learning (predictive analysis on classification)

➢Summary of all results

- Best launch site: KSC LC-39A
- Best ML model: Decision Tree
- Correlation between launch success rate and number of launches

Introduction

- ➢ Project Background and context
  - Falcon 9 first stage landing prediction
  - Competition from other providers
  - Cost of launch and success rate of launch will determine SpaceX's pricing and bidding
- ➢ Problems that need answers
  - What are the factors that related to success launches?
  - What are the sites that perform best with launches?
  - What are the factors that determine successful landings?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - **Direct**: SpaceX API
  - **Indirect**: Wikipedia Web scrapping
- Perform data wrangling
  - Data normalization, grouping
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification model
  - Run through 4 machine learning models to find best one

Data Collection – SpaceX API

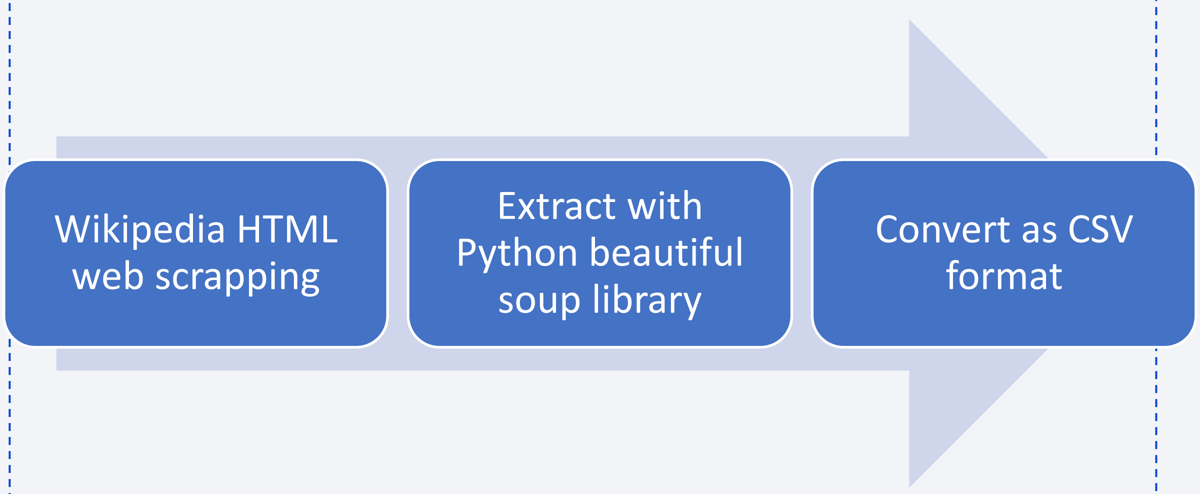
- Data collection with SpaceX REST calls using key phrases and flowcharts

- [GitHub URL of the completed SpaceX API calls]

Data Collection - Scraping

- Web scraping process using key phrases and flowcharts

- [GitHub URL of the completed web scraping notebook]



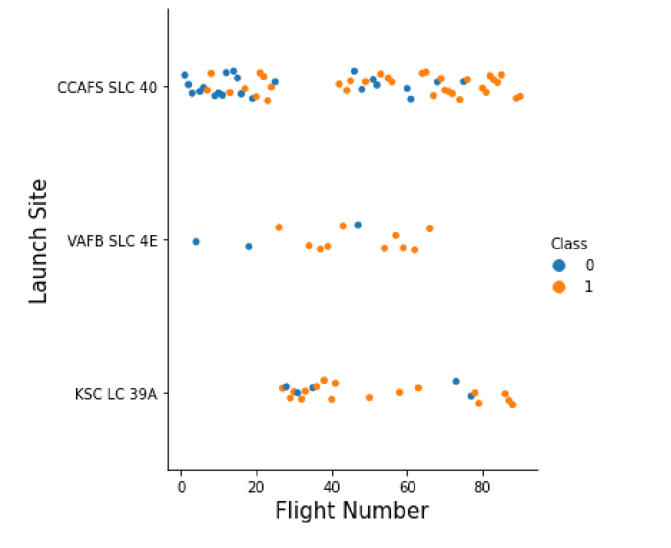
Data Wrangling

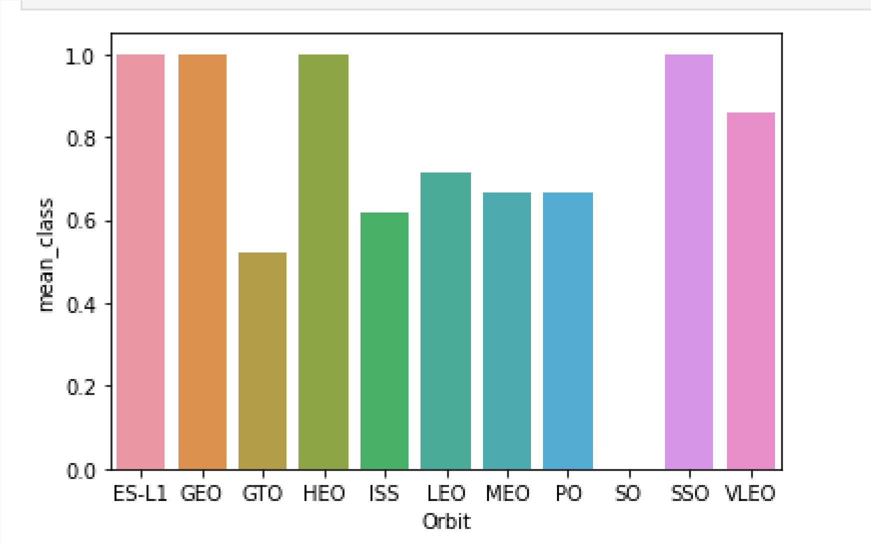
- Data wrangling process using key phrases and flowcharts
- GitHub URL

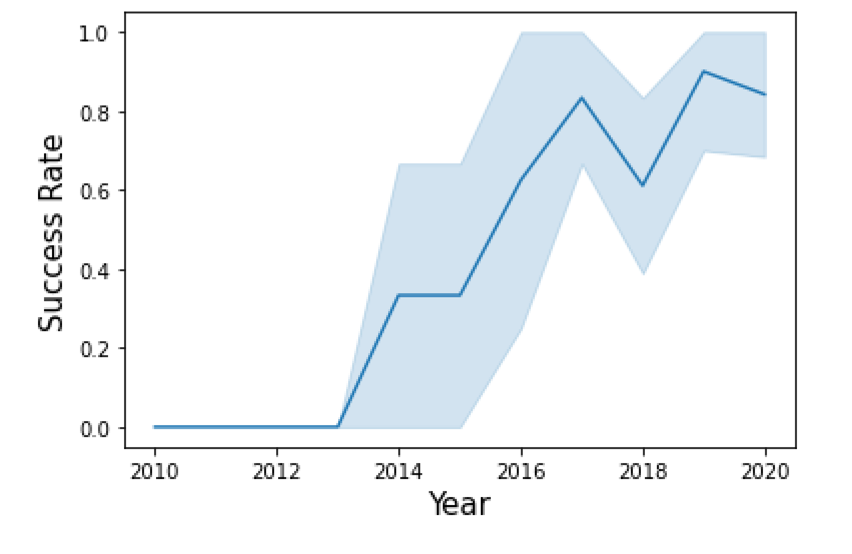
- Number of launches at each site
- Number and occurrence of each orbit
- Number and occurrence of mission outcome per orbit type
- Create a landing outcome label from outcome column

EDA with Data Visualization

- Scatter plot: easy to tell dependencies between variables
- Bar chart: on categorical values
- Line chart: clear on trends with time
- [GitHub URL]()







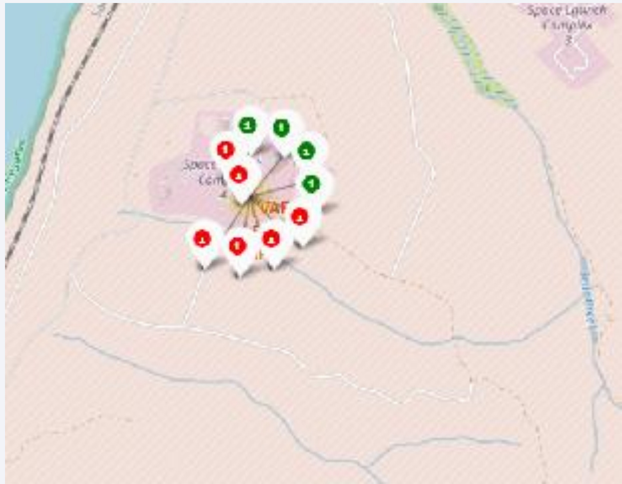
EDA with SQL

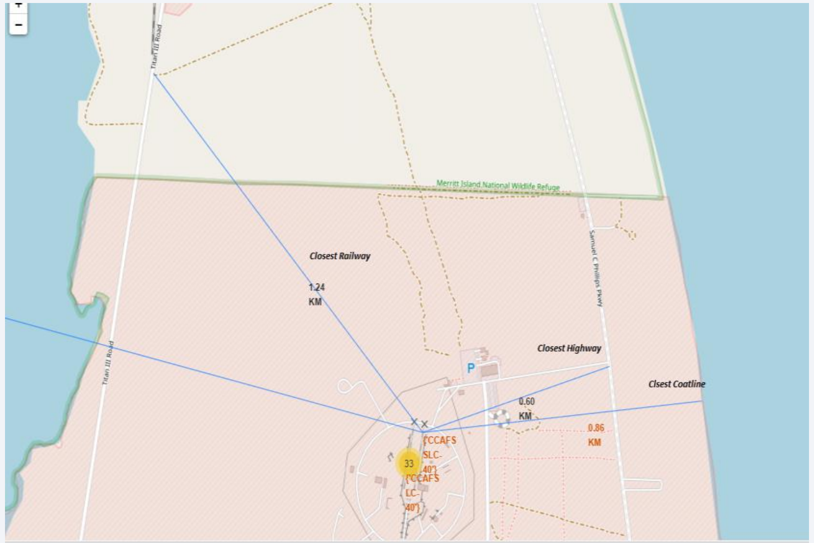
SQL queries performed

- **Names of the unique launch sites in the space mission**
- **Display 5 records where launch sites begin with the string 'CCA'**
- **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**
- **Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000**
- **The total number of successful and failure mission outcomes**
- **The names of the booster_versions which have carried the maximum payload mass. Use a subquery**
- GitHub URL

Build an Interactive Map with Folium

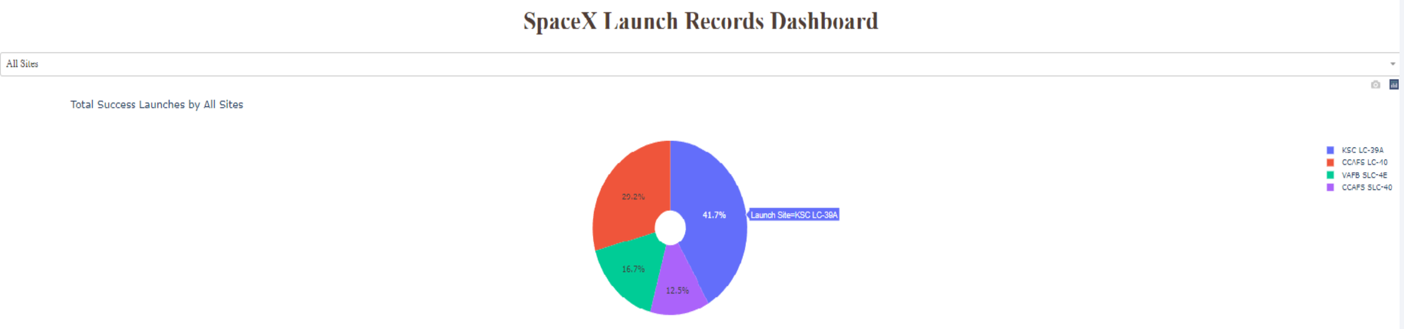
- Map objects such as markers, circles, lines are added to a folium map by Longitude and Latitude
- Markers with green and red color identified success and failures launches
- Lines are to display the distance of a launch site to nearest railway, highway, coastline and cities.
- [GitHub URL]()





Build a Dashboard with Plotly Dash

- Interactive pie charts to display success rate on all sites and a selector to narrow down to individual sites



- Scatter plot to display success and failures on all payload range with adjustable range selector

- [GitHub URL]()

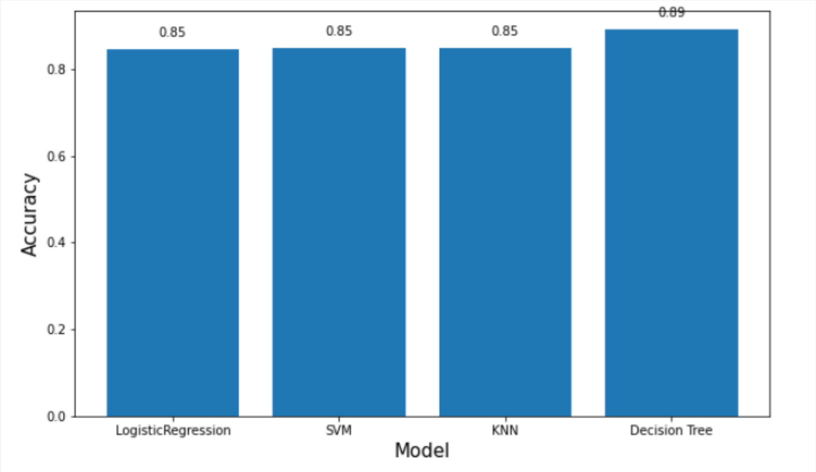
Predictive Analysis (Classification)

Building Model
- Load & Transform using Numpy and Pandas
- Train_Test split dataset
- Build ML Algorithms

Evaluating ML Models
- Check accuracy for each model
- Plot Confusion Matrix

Improving ML Models
- Feature Engineering
- Algorithm Tuning

Find best ML model



- GitHub URL

Results

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

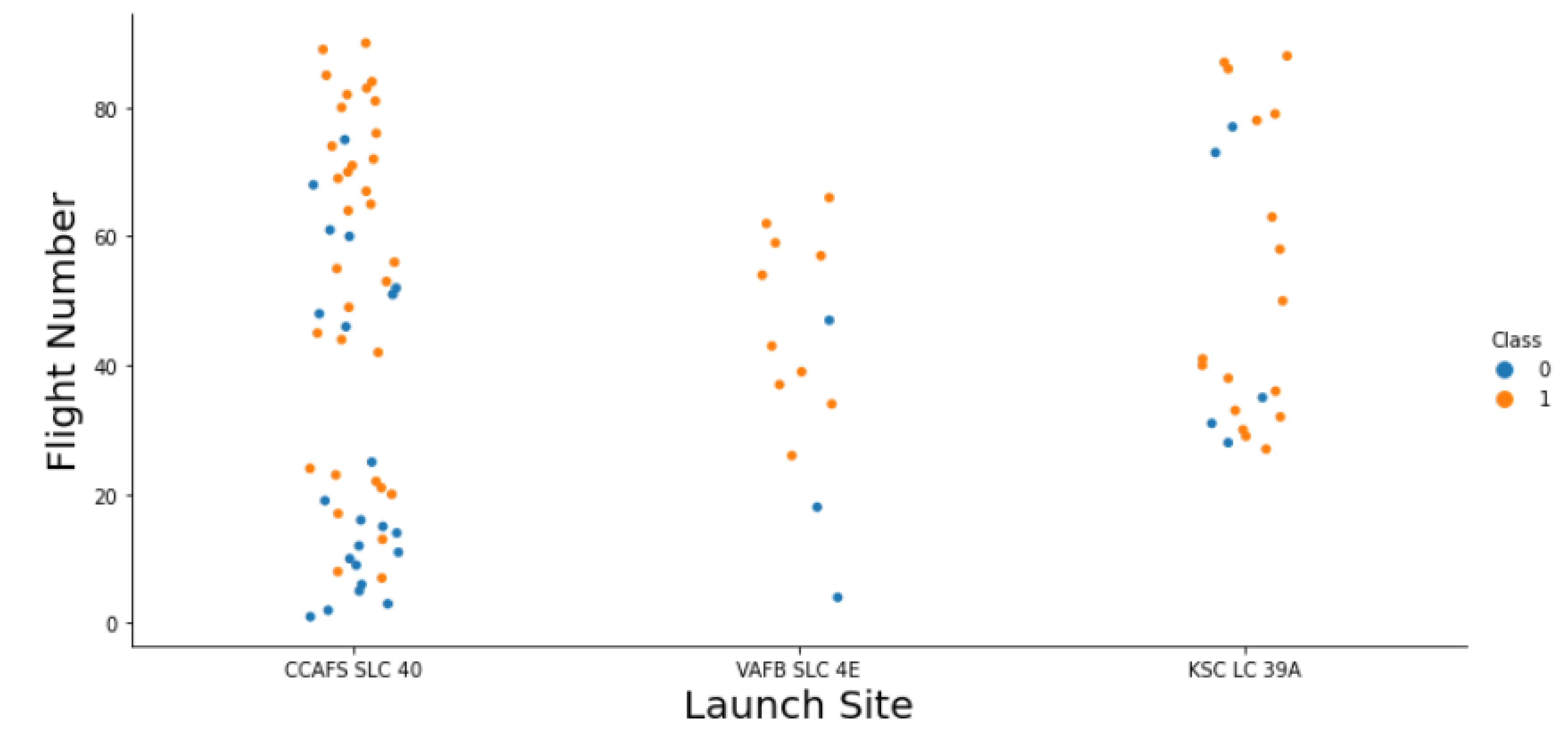
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Class=0 Launch fail
- Class=1 Launch success

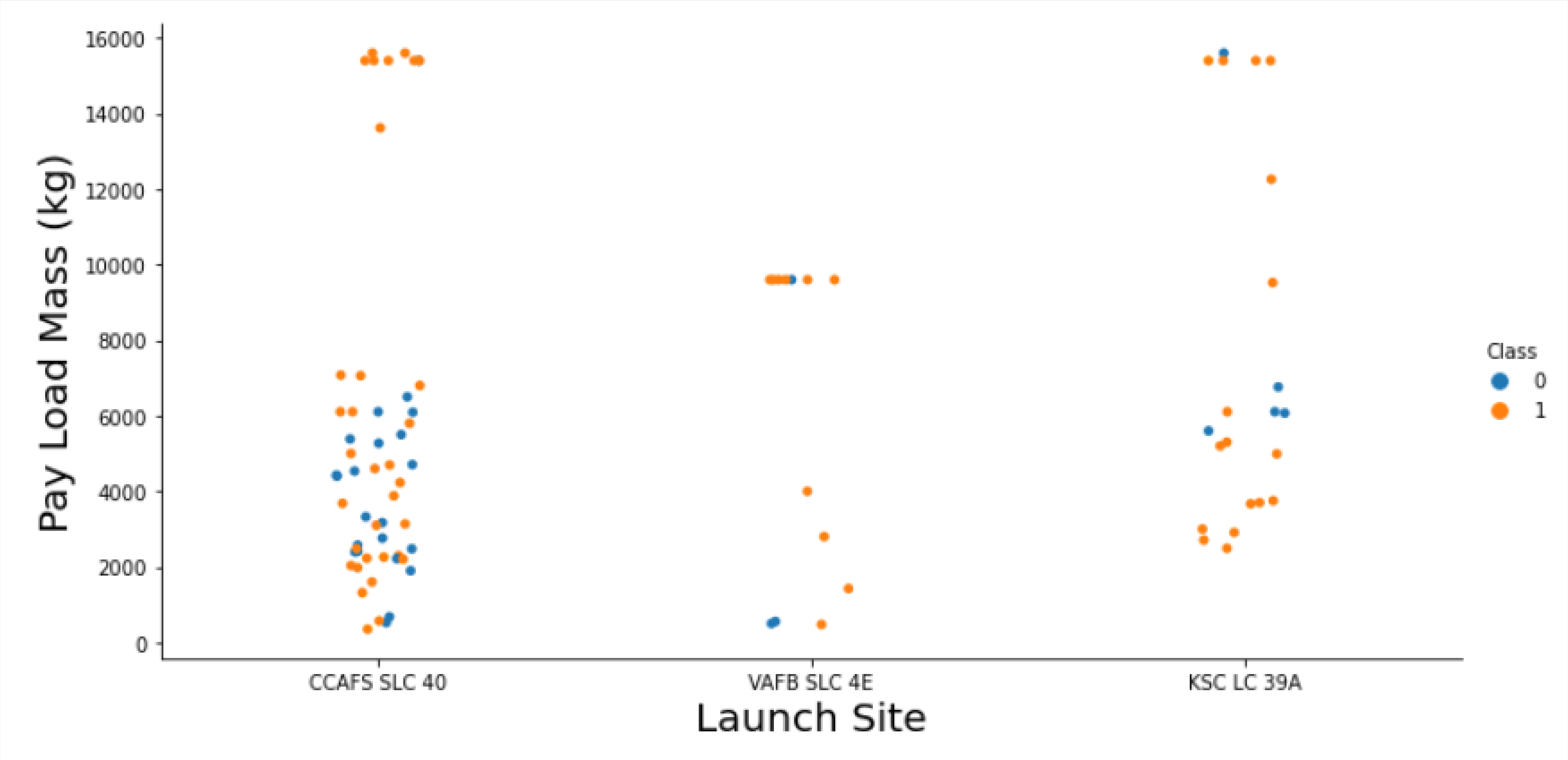
- Site with more launches tend to have more successful launches



Payload vs. Launch Site

- Class=0 Launch fail
- Class=1 Launch success

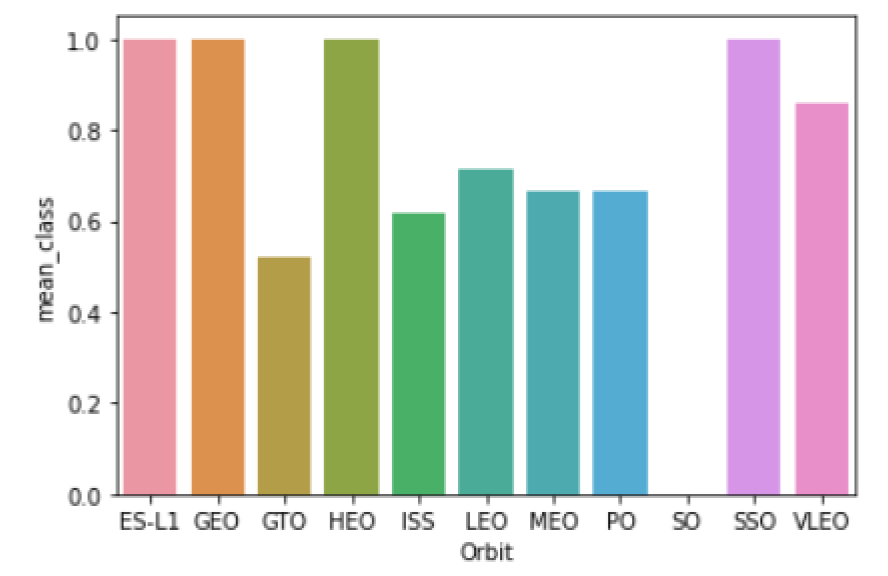
- Site CCAFS SLC 40 has more successful rate on heavy pay load launches



Success Rate vs. Orbit Type

- The more mean_class close to 1 the more success launches with a orbit

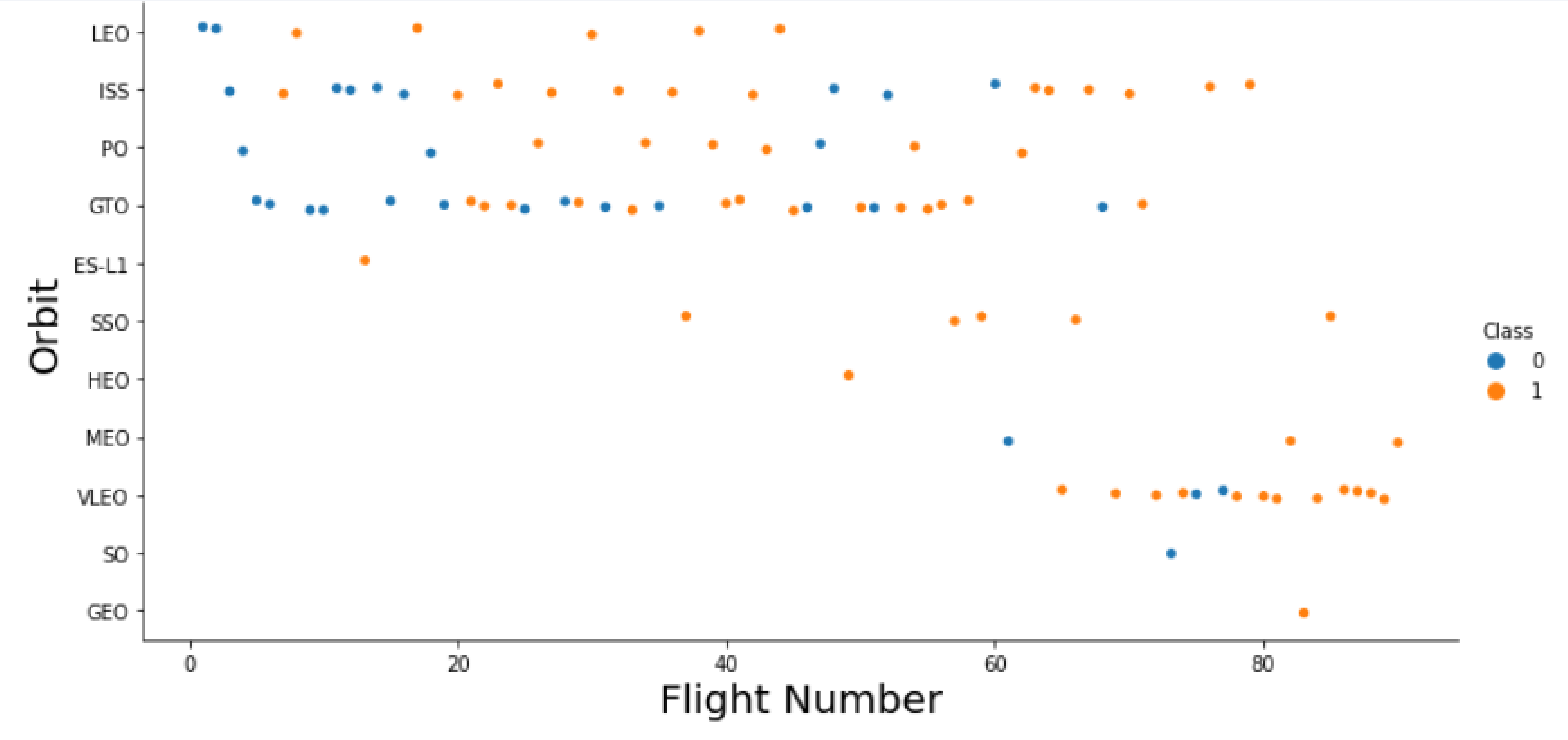
- ES-L1,GEO, HEO and SSO have the best success rates



Flight Number vs. Orbit Type

- Class=0 Launch fail
- Class=1 Launch success

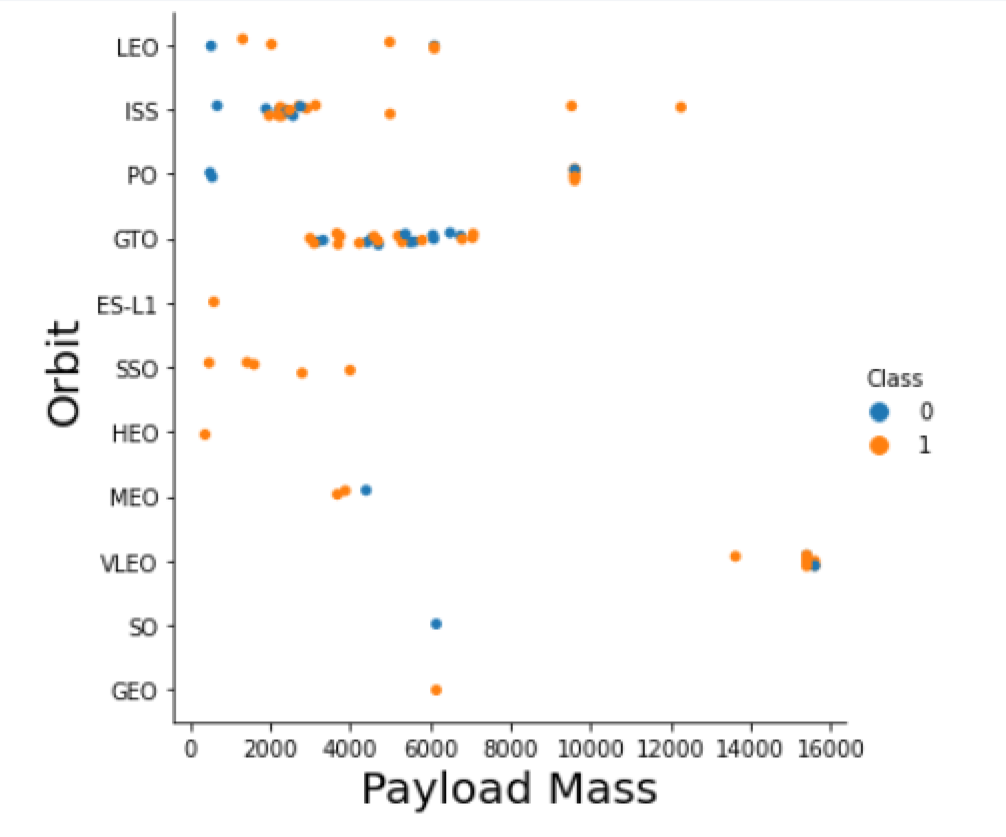
- The higher the flight number go, the higher success rates are for launches (60-80)



Payload vs. Orbit Type

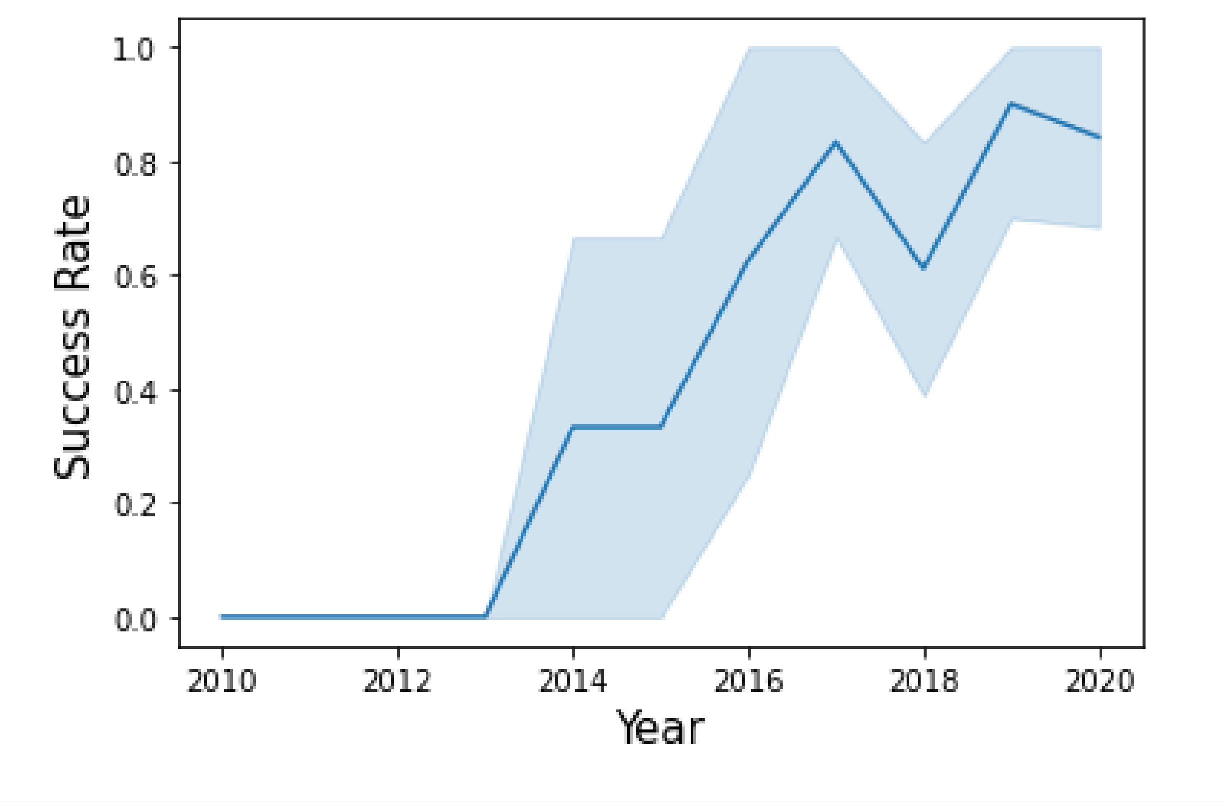
- Class=0 Launch fail
- Class=1 Launch success

- Launch failure mainly happened on under 8000 KG and GTO Orbit



Launch Success Yearly Trend

- With time, Space X's launch success rate grow exponentially
- 2018 Space X has a small set back on launch success rate



All Launch Site Names

- Below SQL query are showing all unique launch site names
- %sql is using SQL Magic method

```
%sql select distinct launch_site from SPACEXDATASET
```

```
 * ibm_db_sa://mjl90806:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with ‘CCA’
- %sql is using SQL Magic method

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

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

```
 * ibm_db_sa://mjl90806:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

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

- Calculate the total payload carried by boosters from NASA
- Confirm SQL number with Python Pandas calculation

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

```
%sql select sum(payload_mass__kg_) as total_payload from SPACEXDATASET where customer='NASA (CRS)'
```

```
 * ibm_db_sa://mjl90806:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

total_payload
45596

```
#Pandas solution
df[df["CUSTOMER"]=="NASA (CRS)"].PAYLOAD_MASS__KG_.sum()
```

```
45596
```

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Confirm SQL result with Python Pandas result

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

```
%sql select avg(payload_mass__kg_) as Average_Payload_Mass from SPACEXDATASET where booster_version='F9 v1.1'
```

```
 * ibm_db_sa://mjl90806:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

average_payload_mass
2928

```
#Pandas solution
df[df["BOOSTER_VERSION"]=="F9 v1.1"].PAYLOAD_MASS__KG_.mean()
```

```
2928.4
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Confirm SQL result with Python Pandas result

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

Hint:Use min function

```
%sql select min(DATE) as First_Date from SPACEXDATASET where landing__outcome='Success'
```

```
 * ibm_db_sa://mjl90806:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

first_date
2018-07-22

```
#Pandas solution
df[df["LANDING__OUTCOME"]=="Success"].DATE.min()
```

```
datetime.date(2018, 7, 22)
```

Successful Drone Ship Landing with Payload between 4000 and 6000

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

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

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

```
 * ibm_db_sa://mjl90806:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Because there are multiple Success and Failure type, this statement shows all outcomes

List the total number of successful and failure mission outcomes

```
%sql select mission_outcome, count(*) as result from SPACEXDATASET group by mission_outcome
```

```
 * ibm_db_sa://mjl90806:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Sub query extract max payload number for each booster_version and match with main query
- Main query returns unique booster versions

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

```
%sql select distinct booster_version from SPACEXDATASET x1 where booster_version=(select booster_version from SPACEXDATASET x2 w
here x1.booster_version=x2.booster_version order by x2.payload_mass__kg_ DESC limit 1)
```

```
 * ibm_db_sa://mjl90806:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

2015 Launch Records

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

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

```
%sql select booster_version,launch_site,landing__outcome from SPACEXDATASET where landing__outcome ='Failure (drone ship)' and y
ear(DATE)='2015'
```

```
 * ibm_db_sa://mjl90806:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

booster_version	launch_site	landing__outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

```
%sql select landing__outcome,count(*) as rank from SPACEXDATASET where date between '2010-06-04' and '2017-03-20' group by landi
ng__outcome order by rank desc
```

```
 * ibm_db_sa://mjl90806:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

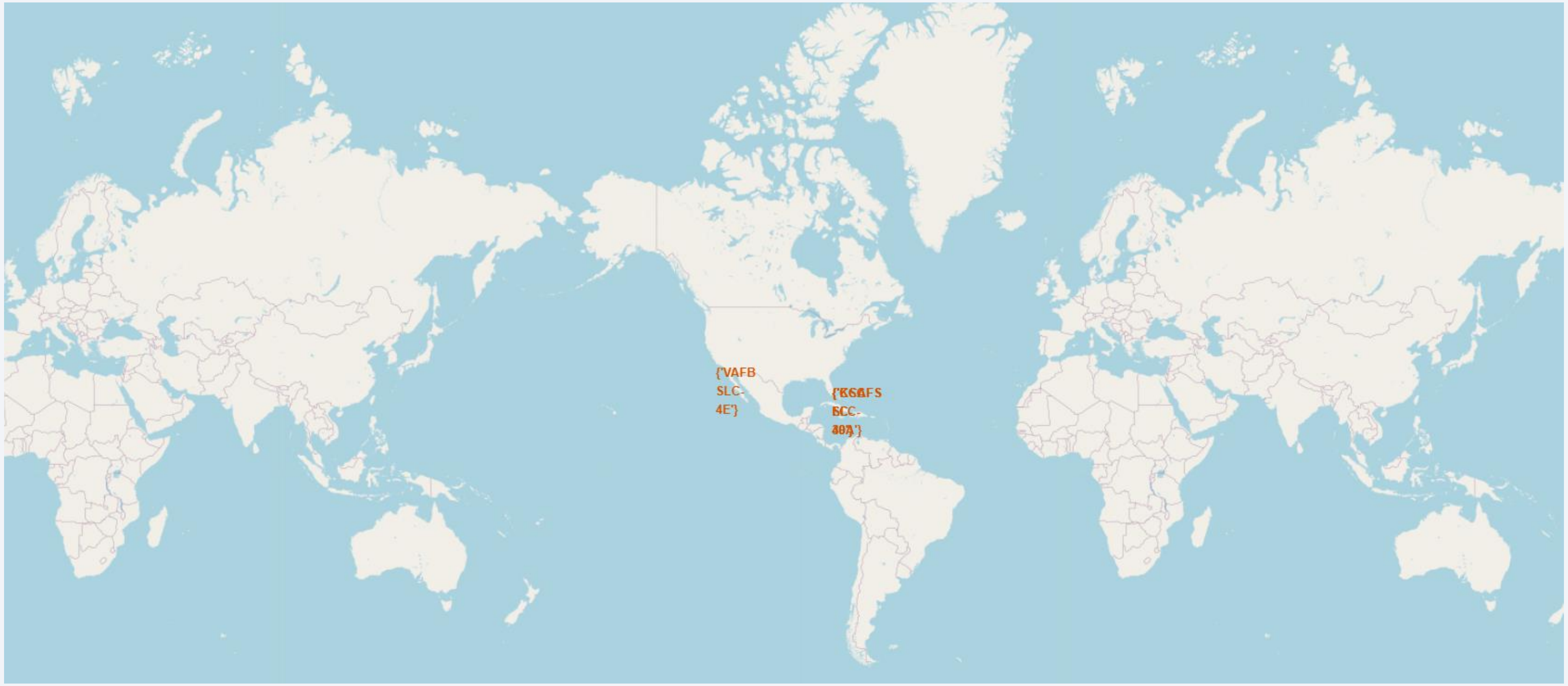
landing__outcome	RANK
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

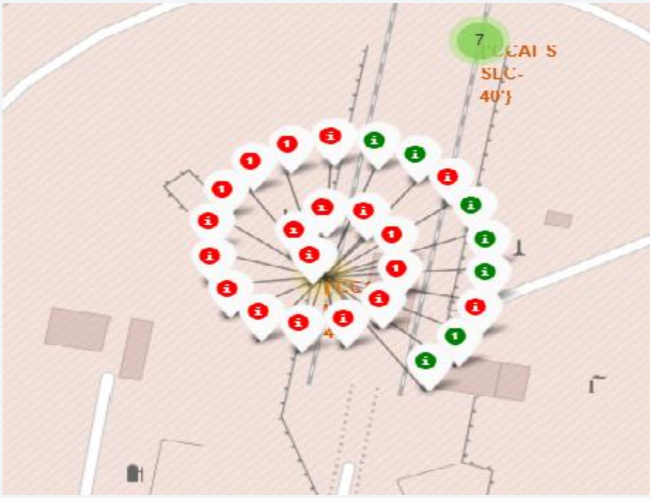
Launch Sites Proximities Analysis

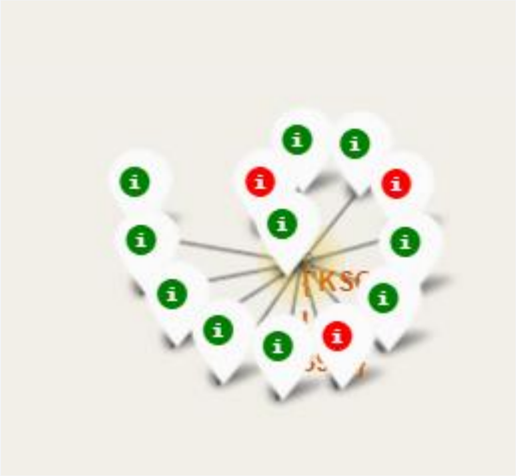
Launch site on a global map

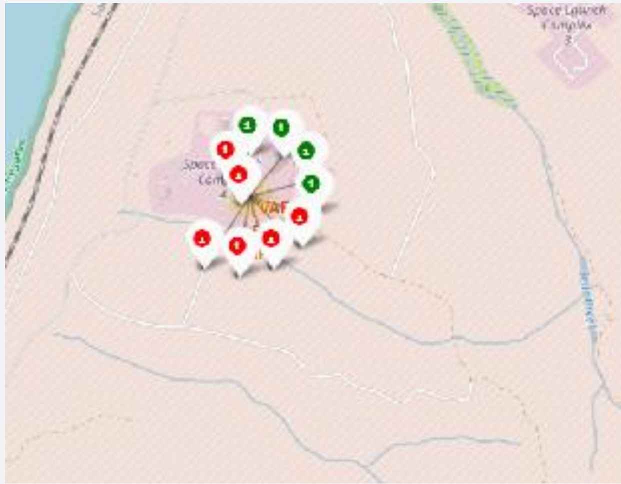
- All SpaceX launch site are in the U.S. continental states (Florida, California)



Map with colored marker







CCAFS LC-40 CCAFS SLC-40 KSC LC-39A

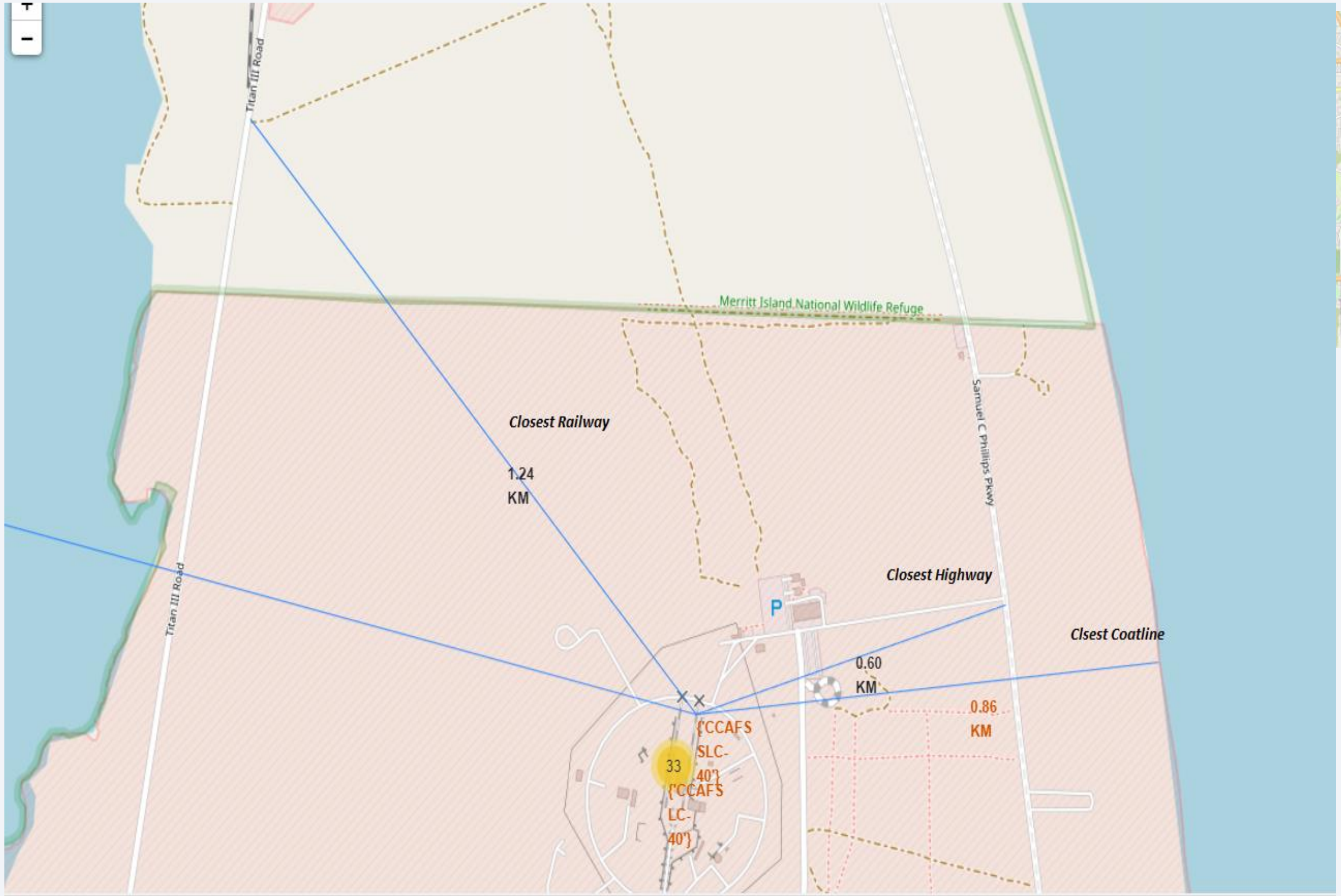
Florida Launch Sites

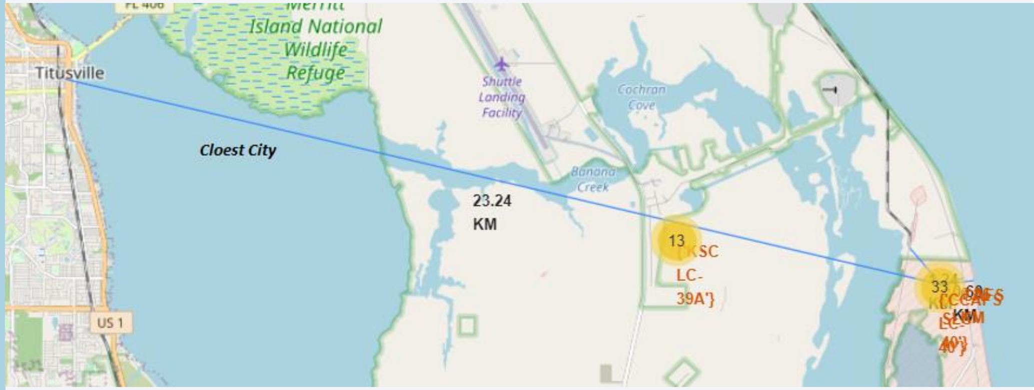
VAFB SLC-4E

California Launch Site

- Green marker is a successful launch
- Red marker is a failed launch

Proximities Map





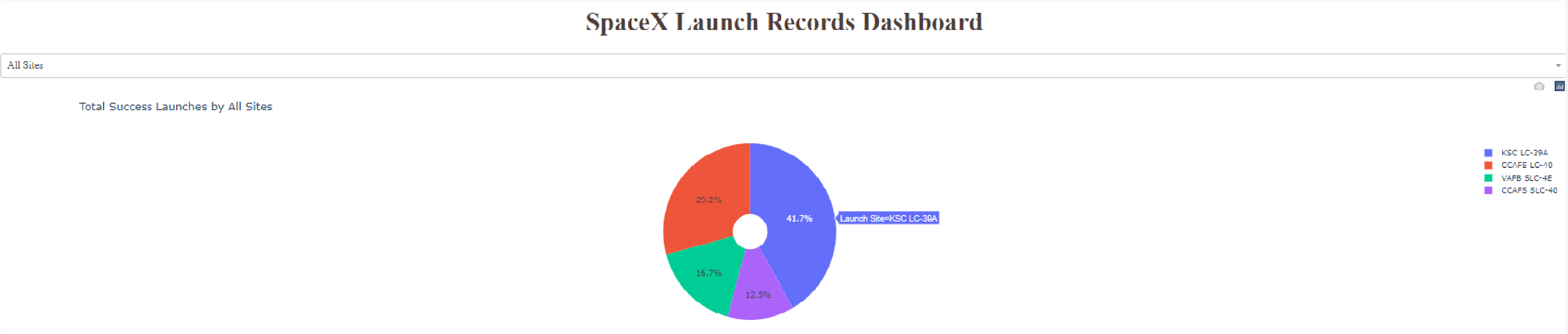
This launch site

- In close proximity to a railway (1.24km)
- In close proximity to a highway (0.6km)
- In close proximity to a coastline (0.86km)
- NOT in close proximity to cities (23.24km)

Section 5

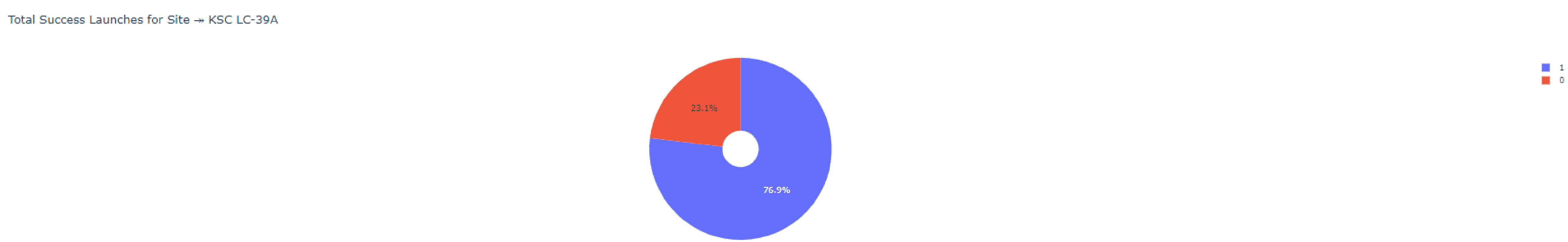
Build a Dashboard with Plotly Dash

Total Success Launches by All Sites



- KSC LC-39A has the most successful launches from all sites

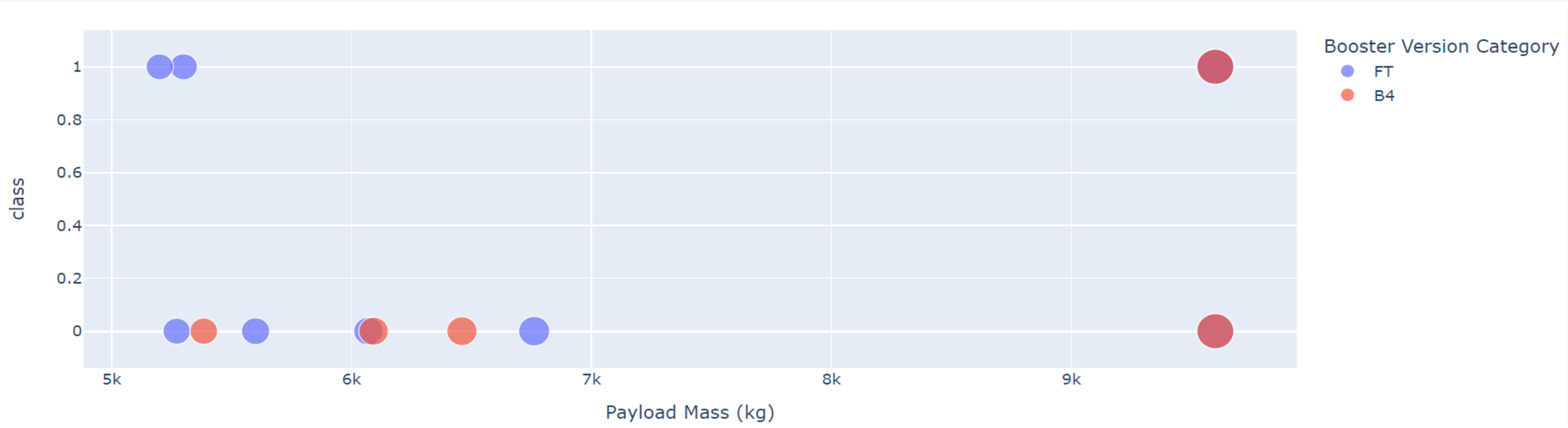
Launch site with highest launch success ratio



- KSC LC-39A has a success rate of 76.9%, which is the highest launch success rate

Payload vs Launch outcome for all sites

High Payload range (5000kg-10000kg) outcome scatter chart



Low Payload range (0 kg-5000kg) outcome scatter chart

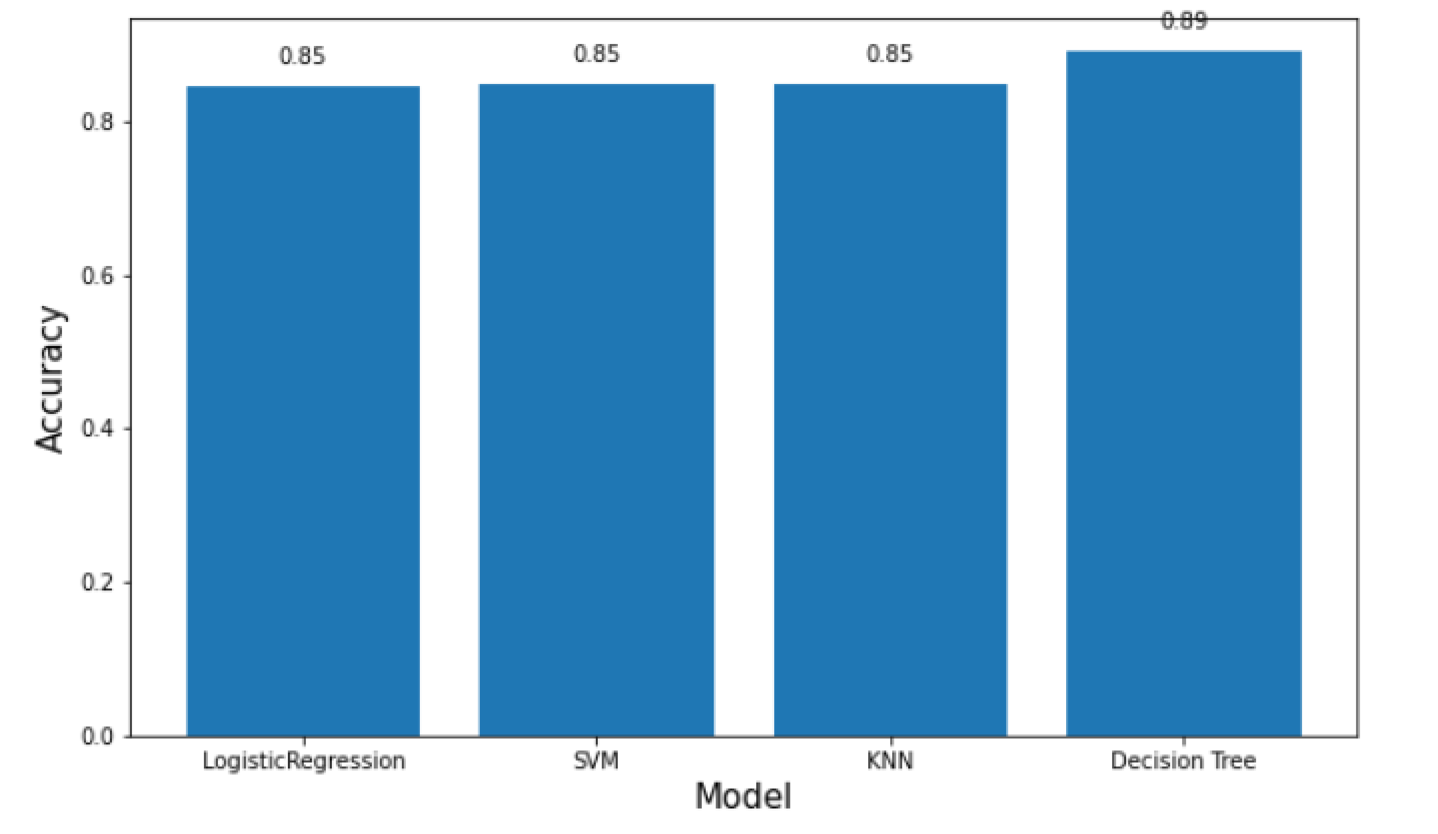
- **Success rates for low payload range is higher than high payload range (more dots on top)**

Section 6

Predictive Analysis (Classification)

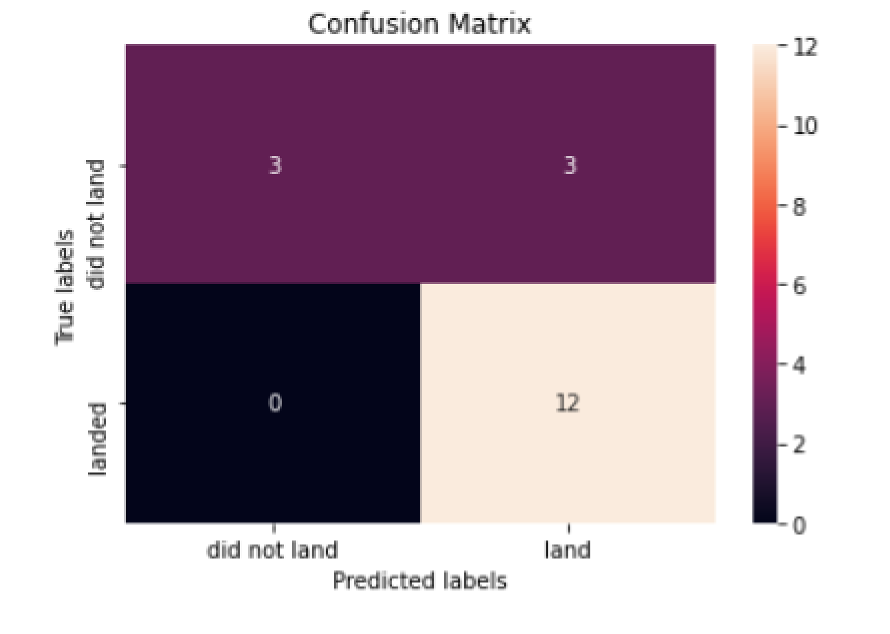
Classification Accuracy

- Best model is Decision Tree with accuracy of 89.11% (83.33% on test data)



Confusion Matrix

- Decision Tree model shows strong positive results for predicted values



Conclusions

- The Decision Tree model is the best machine learning model for this dataset
- KSC LC-39A had the most success launches and higher success launch % compare to other sites
- Site with more launches have better success rate
- SpaceX success rate for launches grow exponentially with time

Appendix

- 2 Python SQL method used:

- SQL Magic:

```
#Method 1 with SQL Magic
%sql ibm_db_sa://mjl90806:Q76g0IIy4Lo3RWZE@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb?security=SSL
```

```
%sql select * from SPACEXDATASET
```

- IBM DB:

```
#Method 2 with ibm db
import pandas as pd
import ibm_db
import ibm_db_dbi
```

```
dsn_hostname="[illegible]"
dsn_uid="[illegible]"
dsn_pwd="[illegible]"

dsn_driver="{IBM DB2 ODBC DRIVER}"
dsn_database="bludb"
dsn_port="30875"
dsn_protocol="TCPIP"
```

```
dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "Security=ssl;").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol, dsn_uid, dsn_pwd)

try:
    conn = ibm_db.connect(dsn, "", "")
    print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)

except:
    print ("Unable to connect: ", ibm_db.conn_errormsg() )
```

```
pd_conn=ibm_db_dbi.Connection(conn)
```

```
QUERY = "select *from SPACEXDATASET"
df=pd.read_sql(QUERY,pd_conn)
df.head()
```

Thank you!