IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Murat Kucukosmanoglu>
<12/28/2021>

Outline

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

Executive Summary

A. Summary of methodologies

1. Data Collection via API and Web Scrapping
2. Exploratory Data Analysis and Data Wrangling
3. Store the dataset in database table using SQL
4. Launch Sites Locations Analysis with Folium
5. Machine Learning Models for Accuracy Analysis

B. Summary of all results

1. We built and tested four machine learning models: k-nearest neighbors, decision tree, logistic regression, and Support-vector machine to predict whether first stage of Falcon 9 will land successfully. They were all accurate to the score of 83.33% on the test data.
2. Based on the accuracy of the train data, the decision tree is predicted to have been the best model.

Introduction

A. Project background and context

1. The commercial space age is arrived, and firms are making space travel accessible to the general public.
2. SpaceX offers Falcon 9 rocket flights at 62 million dollars; other companies charge upwards of 165 million dollars each launch, with much of the savings due to SpaceX's ability to reuse the first stage.
3. As a result, if we can figure out if the first stage will land, we can figure out how much a launch will cost.

B. Problems you want to find answers

1. Train a machine learning model and use public information to predict if SpaceX will reuse the first stage.
2. What factors could influence the success of SpaceX's landing?

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
  - Data Collected from SpaceX rest API and
  - Web Scrapping from Wikipedia page
- **Perform data wrangling**
  - Locate the data in your sources and gain access to it.
  - Combine and clean the edited data for future usage and analysis via machine learning models.
- **Perform exploratory data analysis (EDA) using visualization and SQL**
  - To demonstrate a trend among data, use line and scatter plots as well as bar graphs.

Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune, evaluate classification models

Data Collection

- **Describe how data sets were collected.**

The data was gathered using a combination of API queries from the Space X API and webs craping from Wikipedia page for Space X. The summary of the dataframe for each is shown below.

Space X API

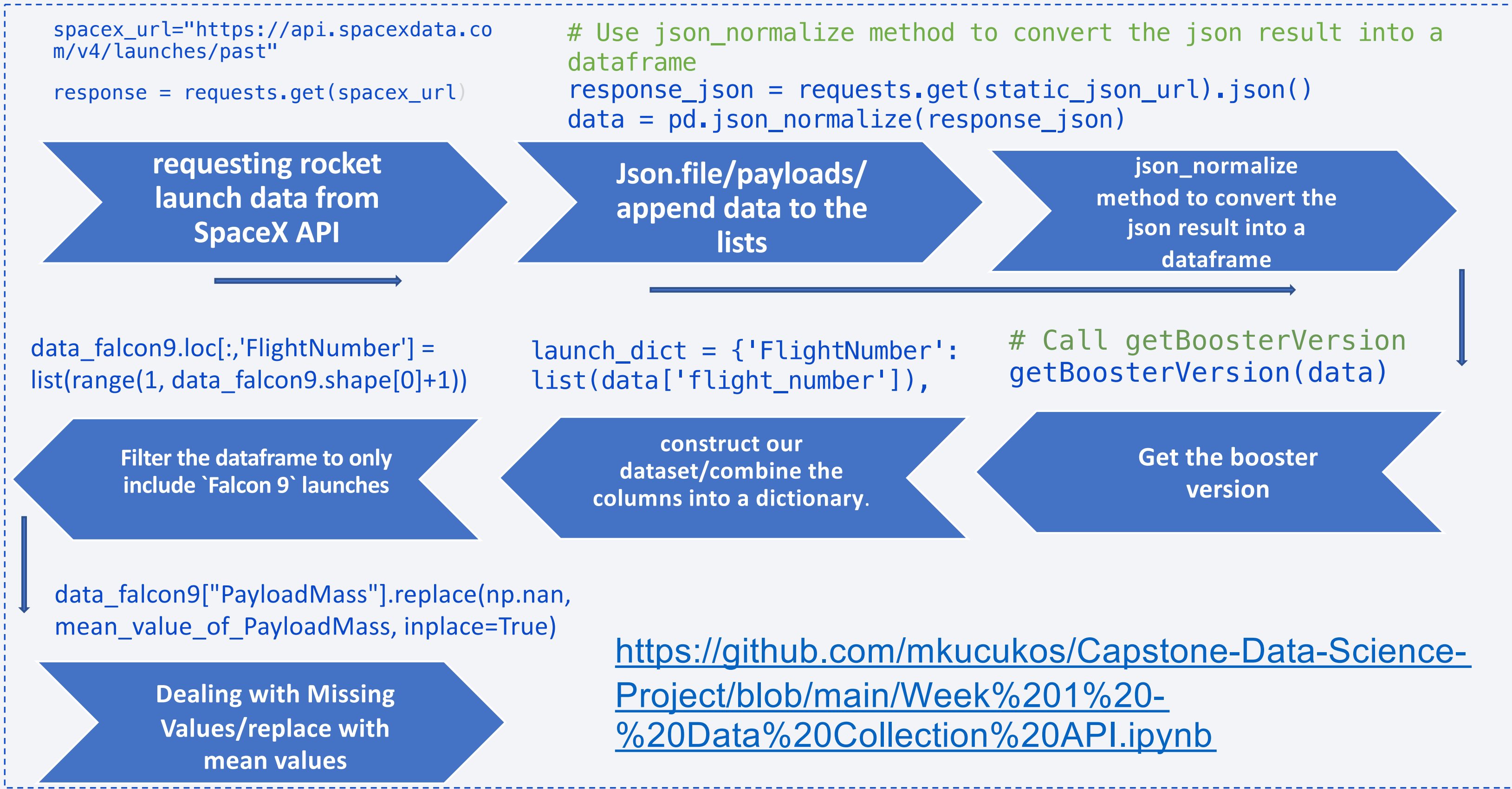
```
[FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude]
```

Webs craping from Wikipedia

```
['Flight No.', 'Date and time ( )', 'Launch site', 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome']
```

Data Collection – SpaceX API

Flowchart of SpaceX API calls



Data Collection - Scraping

Flowchart of Web Scraping

Getting data from HTML page

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

Create a `BeautifulSoup` object

```
soup = BeautifulSoup(page.text, 'html.parser')
```

Create a `HTML tables`

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

Create a dictionary by parsing the launch HTML tables

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

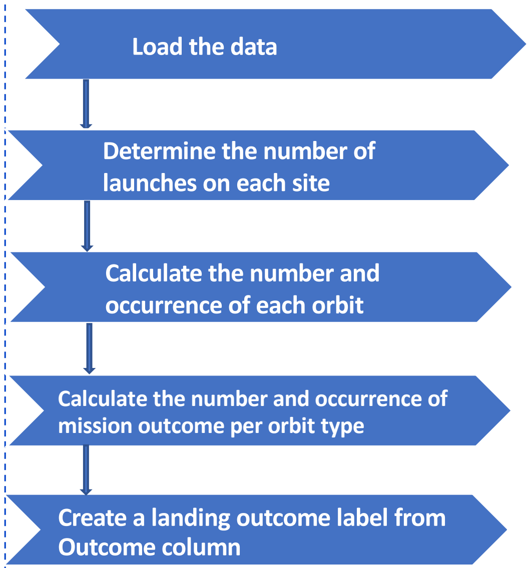
Convert the dictionary to the data frame and save it as csv

```
df=pd.DataFrame(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

https://github.com/mkucukos/Capstone-Data-Science-Project/blob/main/Week%201-webscraping.ipynb

Data Wrangling

Flowchart of Data Wrangling



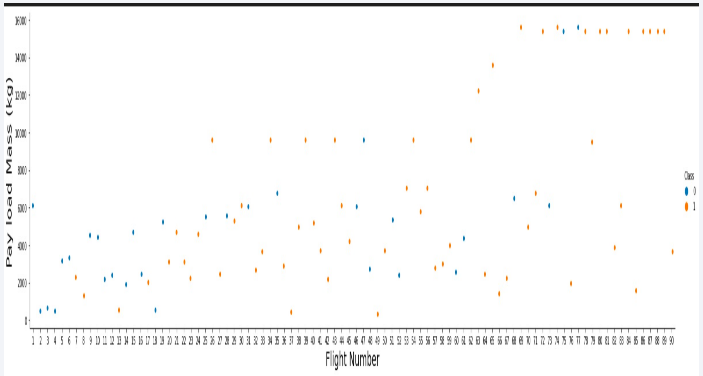

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDS0321ENSkillsNetwork/datasets/dataset_part_1.csv")

df['LaunchSite'].value_counts()

df['Orbit'].value_counts()

landing_outcomes = df["Outcome"].value_counts()

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

https://github.com/mkucukos/Capstone-Data-Science-Project/blob/main/Week-1_Data%20wrangling.ipynb

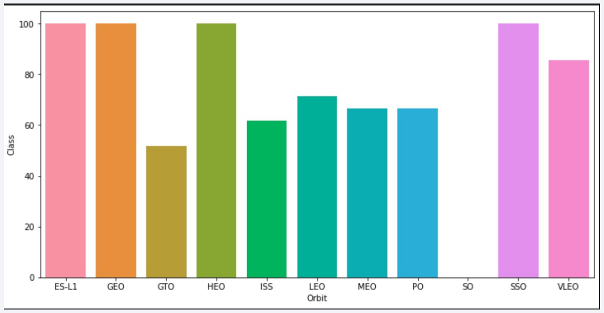
EDA with Data Visualization

- Exploratory Data Analysis performed on variables Launch Site, Flight Number, Payload Mass, Orbit and success rate.

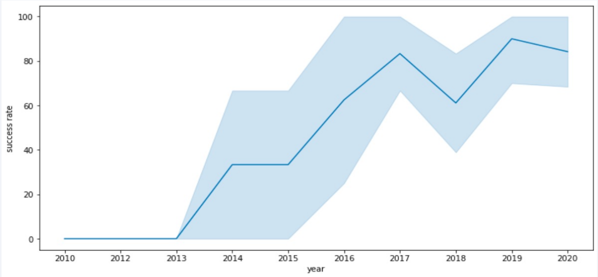
- A scatter plot is a fantastic way to demonstrate the interdependence of factors. We utilized this approach to demonstrate the relationship between Payload Mass (kg) and Flight Number, for example.



BarCharts is very useful to visually check if there are any relationship between success rate and orbit type.



you can observe that the success rate since 2013 kept increasing till 2020 with a line chart.



https://github.com/mkucukos/Capstone-Data-Science-Project/blob/main/Week-2_Exploring_and_preparing_data.ipynb

EDA with SQL

- The data was imported into an IBM DB2 database.
- To complete the assigned tasks, SQL queries were written and executed.
- To gain a better grasp of the dataset, queries were run.
- Some analyses are conducted, for example, on the names of launch sites, average payload mass, landing outcomes, booster names, and mission outcomes.

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

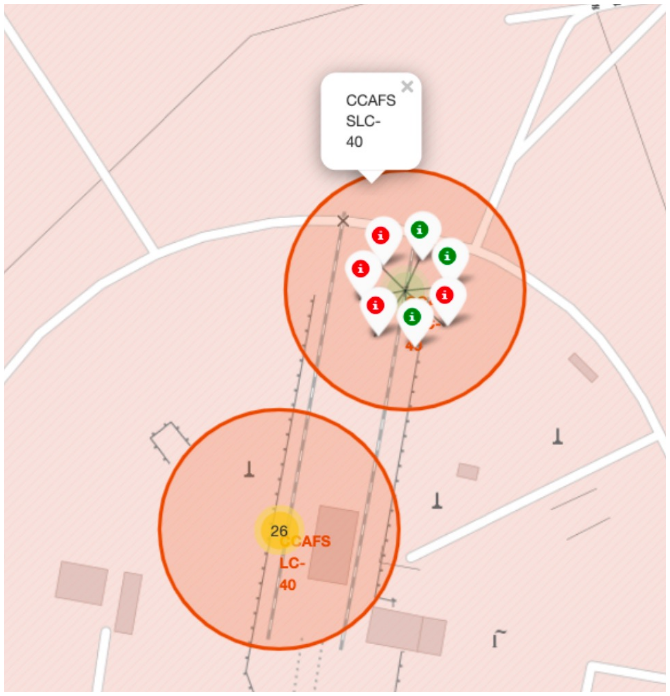
landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

https://github.com/mkucukos/Capstone-Data-Science-Project/blob/main/Week-2-jupyter-labs-eda-sql.ipynb

Build an Interactive Map with Folium

- The success rate of a launch can be affected by a variety of factors, including payload mass, orbit type, and so on. It might also be influenced by the location and proximity of a launch site, as well as the initial trajectory of rockets.
- Many aspects go into determining the best location for a launch site, and maybe we can learn about some of them by evaluating previous launch site sites. This enables us to comprehend why launch sites are situated in the locations.

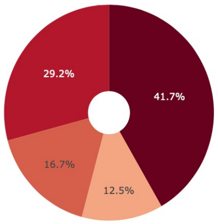
https://github.com/mkucukos/Capstone-Data-Science-Project/blob/main/week3_launch_site_location_analyses_with_folium.ipynb



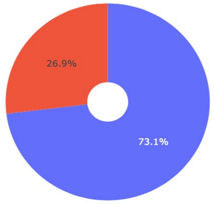
Build a Dashboard with Plotly Dash

- A pie chart and a scatter plot are included on the dashboard.
- The pie chart may be used to indicate the success rates of single launch locations or to show the distribution of successful landings across all launch sites.
- Scatter plots require two inputs: all sites or a single site, as well as a payload mass slider that ranges from 0 to 10000 kg. The success rate of the launch site is displayed via a pie chart. The scatter plot may indicate how success varies by launch location, payload tonnage, and booster version category.

Total Success Launches by All Sites



Total Success Launches for Site = CCAFS LC-40



Predictive Analysis (Classification)

Create a NumPy array from the column Class in data

```
y = data['Class'].to_numpy()
```

Standardize the data and reassign it to x

```
transform = preprocessing.StandardScaler()
X_standardized = transform.fit(X).transform(X)
X = pd.DataFrame(X_standardized)
```

Split the data X and Y into training and test data

```
X_train, X_test, Y_train, Y_test = train_test_split(X,
y, test_size=0.2, random_state=2)
```

create a GridSearchCV object

```
parameters={'C':[0.01,0.1,1],'penalty':['l2'],'solver':['lbfgs
']}
```

Evaluate the models

```
logreg_cv.score(X_test, Y_test)
svm_cv = GridSearchCV(svm, parameters, cv = 10)
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

Find best performing model

https://github.com/mkucukos/Capstone-Data-Science-Project/blob/main/Week-4-SpaceX_Machine%20Learning%20Prediction.ipynb

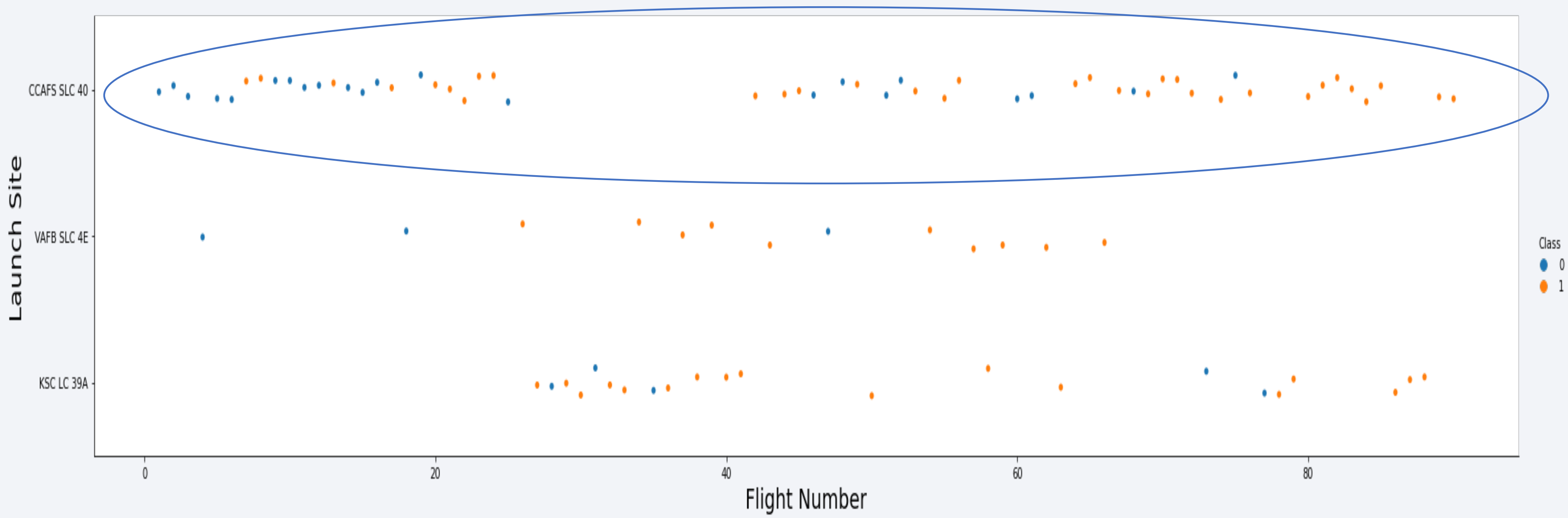
Results

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

Section 2

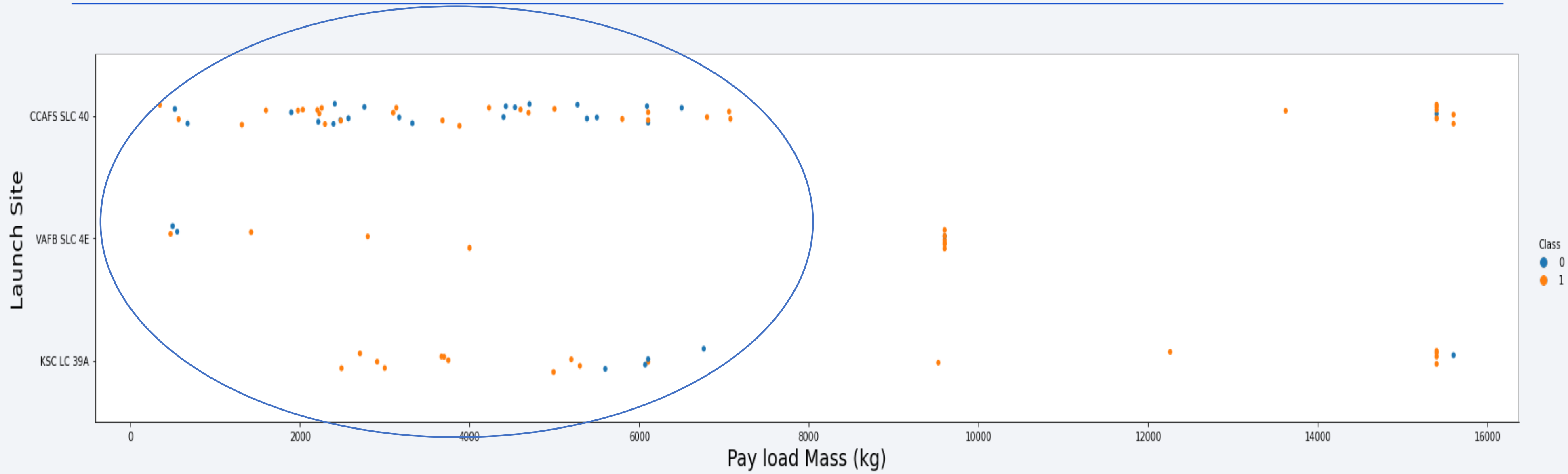
Insights drawn from EDA

Flight Number vs. Launch Site



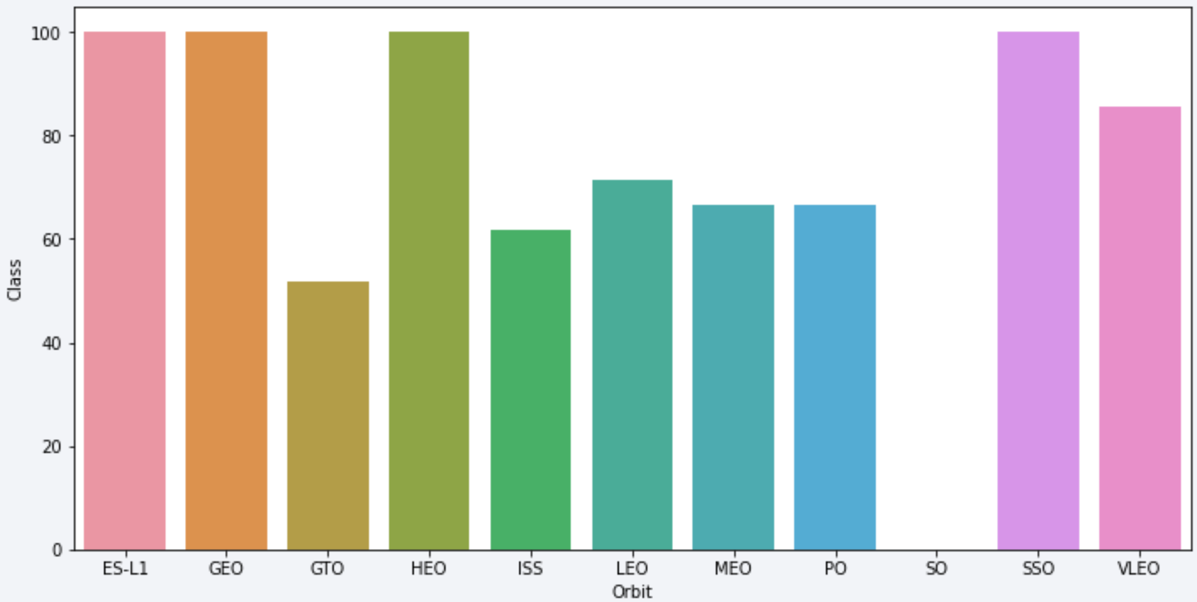
- The relationship between the Flight Number and the Launch Site is depicted in the diagram above. Class 1 denotes a successful launch, whereas Class 0 denotes a failure. Other launch sites have less flights than CCAFS SLC 40.

Payload vs. Launch Site



- The graphic above depicts the relationship between the Pay Load Mass (kg) and the Launch Site. A successful launch is classified as Class 1, whereas a failure is classified as Class 0. Other launch locations have fewer flights than CCAFS SLC 40, and the majority of payload mass (kg) values are less than 800 kg.

Success Rate vs. Orbit Type

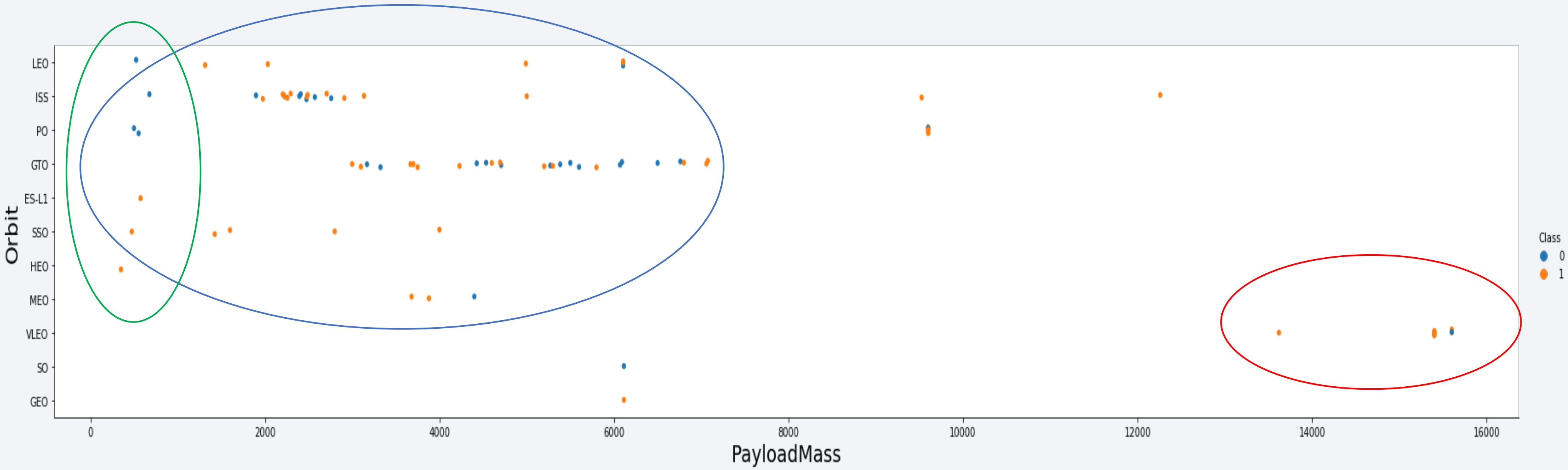


- The success rate for ES-L1, GEO, HEO, and SSO is about % 100.
- The success percentage of VLEO, PO, MEO, LEO, ISS, and GTO ranges between 80 and 50 %.
- SO has a 0% success rate.

Flight Number vs. Orbit Type

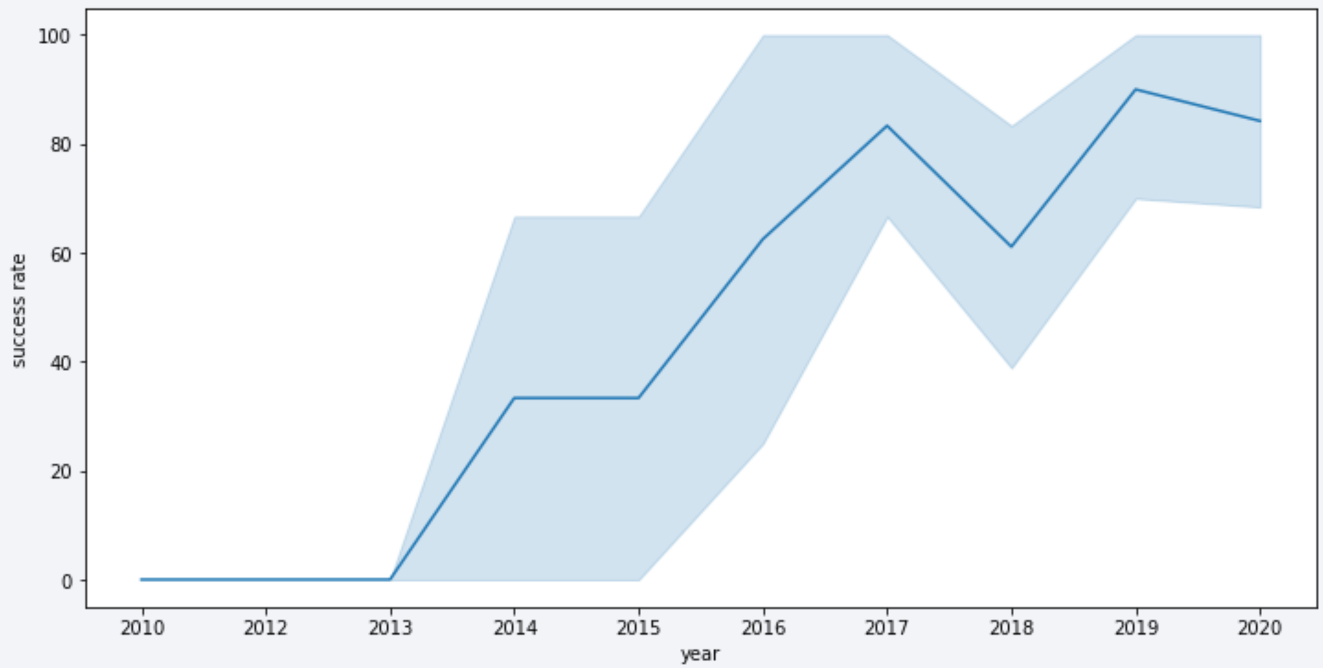
- The link between Flight Number and Orbit Type is seen in the diagram above. With an increase in the frequency of flights around orbit VLEO, more successful launches may be expected.
- Class 1 denotes a successful launch, whereas Class 0 denotes a failure.

Payload vs. Orbit Type



- The link between PayloadMass and Orbit Type is seen in the diagram above. The majority of launches have a PayloadMass of less than 8000 kg (blue circle).
- The PayloadMass of VLEO appears to be higher than that of other orbits (red circle).
- Lower PayloadMass appears to result in more failed launches (green circle).
- Class 1 denotes a successful launch, whereas Class 0 denotes a failure.

Launch Success Yearly Trend



- The success rate improves from 2013 to 2020, with a little drop around 2018. The 95 percent confidence level is shown by the blue shade region.

All Launch Site Names

```
%sql select DISTINCT LAUNCH_SITE from SPACEXTBL
```

```
* ibm_db_sa://wkl91726:***@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

- Using the "distinct" function in the query, we get the unique values of the Launch Site column from SPACESTBL.

Launch Site Names Begin with 'CCA'

```
%sql select * from  SPACEXTBL where launch_site like 'CCA%' limit 5
✓ 0.1s                                                                                       Python
* ibm_db_sa://wkl91726:***@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

- "Like" keyword returns a launch site that begins with CCA. Limit 5 displays the table's first five records.

Total Payload Mass

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

```
* ibm_db_sa://wkl91726:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/BLUDB
Done.
```

SUM
45596

- ”Sum” function calculates total mass carried by boosters launched by NASA (CRS) from SPACESTBL, “where” function filters it

Average Payload Mass by F9 v1.1

Task 4

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

```
%sql select avg(payload_mass__kg_) as "Average" from SPACEXTBL where booster_version like 'F9 v1.1%'
```

```
* ibm_db_sa://wkl91726:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/BLUDB
Done.
```

Average
2534

- ”Avg” function calculates average payload mass carried by booster version F9 v1.1, which “where” function filters it

First Successful Ground Landing Date

```
%sql select min(date) as Date from SPACEXTBL where landing__outcome like 'Success (ground pad)'

 * ibm_db_sa://wkl91726:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/BLUDB
Done.
```

DATE
2015-12-22

The function "min" generates min date, which is the date on which successful landing outcomes occurred. ‘Success (ground pad)' is filtered using the "Where" function.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select booster_version from SPACEXTBL where (mission_outcome like 'Success') AND (payload_mass__kg_ BETWEEN 4000 AND 6000) AND (landing__outcome like 'Success (drone ship)')
```

Python

```
* ibm_db_sa://wkl91726:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/BLUDB
Done.
```

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

Using "where" function, we filter dataset to find successful mission and pay load mass between 4000 and 6000. AND clause connects the conditions.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXTBL GROUP by mission_outcome ORDER BY mission_outcome
```

```
* ibm_db_sa://wkl91726:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/BLUDB
Done.
```

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

“Group by” function groups the mission outcomes, and ”count” function calculates the total number of each mission outcome.

Boosters Carried Maximum Payload

```
max_payload_mass = %sql select max(payload_mass__kg_) from SPACEXTBL
%sql select booster_version from SPACEXTBL where payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXTBL)
```

```
 * ibm_db_sa://wkl91726:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/BLUDB
Done.
 * ibm_db_sa://wkl91726:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/BLUDB
Done.
```

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

The booster version with payloads equal to the maximum value is returned by this "query." 15600 kg is the maximum payload mass.

2015 Launch Records

```
%sql select landing__outcome, booster_version, launch_site from SPACEXTBL where DATE like '2015%' AND landing__outcome like 'Failure (drone ship)'
```

```
* ibm_db_sa://wkl91726:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/BLUDB
Done.
```

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

This query returns the 2015 landing outcome, booster version, and launch site that were unsuccessful. The year=2015 and failure outcome are filtered using the "where" clause.

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

```
%sql select landing__outcome, count(*) as count_of_landing_outcomes from SPACEXTBL \
where date between '2010-06-04' and '2017-03-20' GROUP by landing__outcome order by count desc
```

✓ 0.2s

```
* ibm_db_sa://wkl91726:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/BLUDB
Done.
```

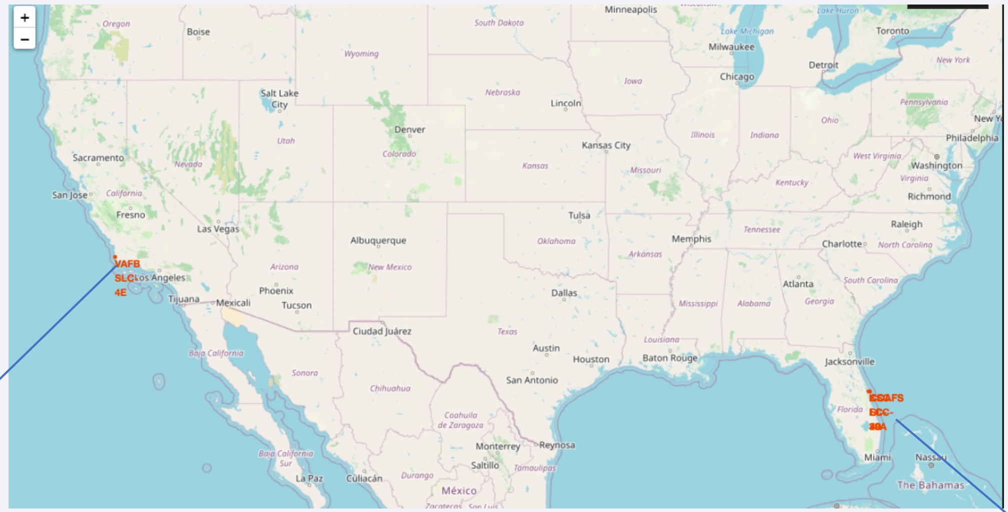
landing__outcome	count_of_landing_outcomes
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Where clause filters the date between 2010-06-04 and 2017-03-20, and group by descending order

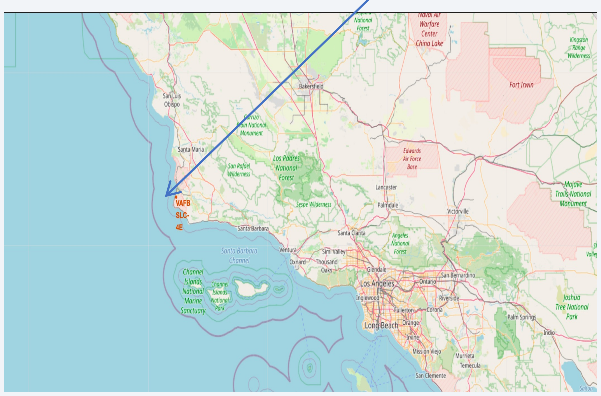
Section 4

Launch Sites Proximities Analysis

Location of the Launch Sites

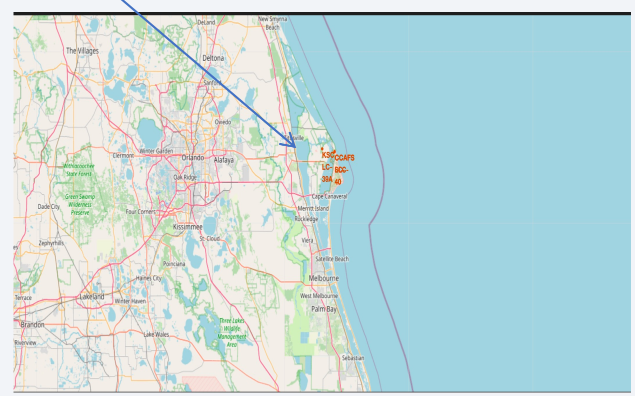


Pacific Ocean



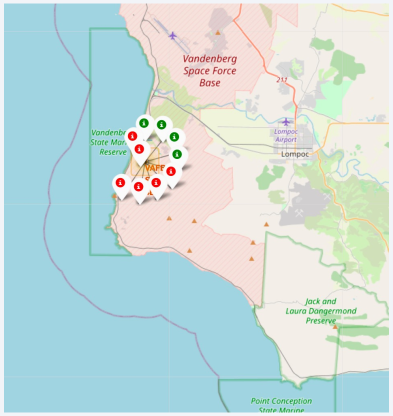
One station is plainly positioned near the Pacific Ocean, and the other is located near the Atlantic Ocean.

Atlantic Ocean

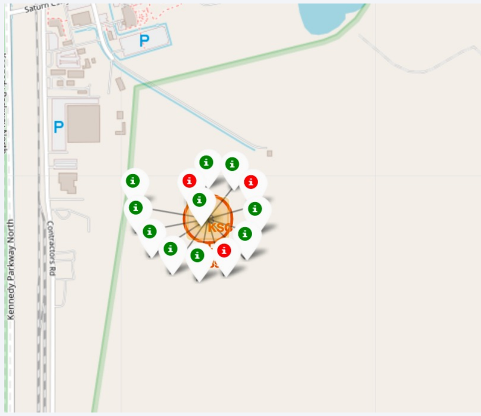


Color Labelled to launch outcome

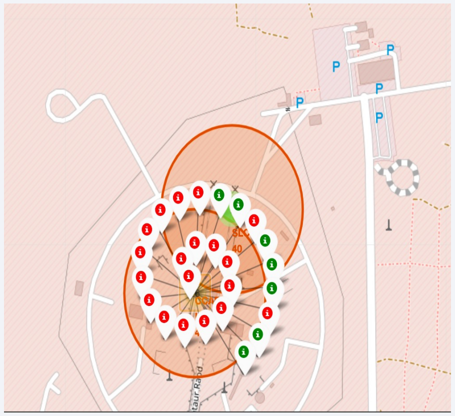
VAFB-SLC-4E



KSC-LC-39A



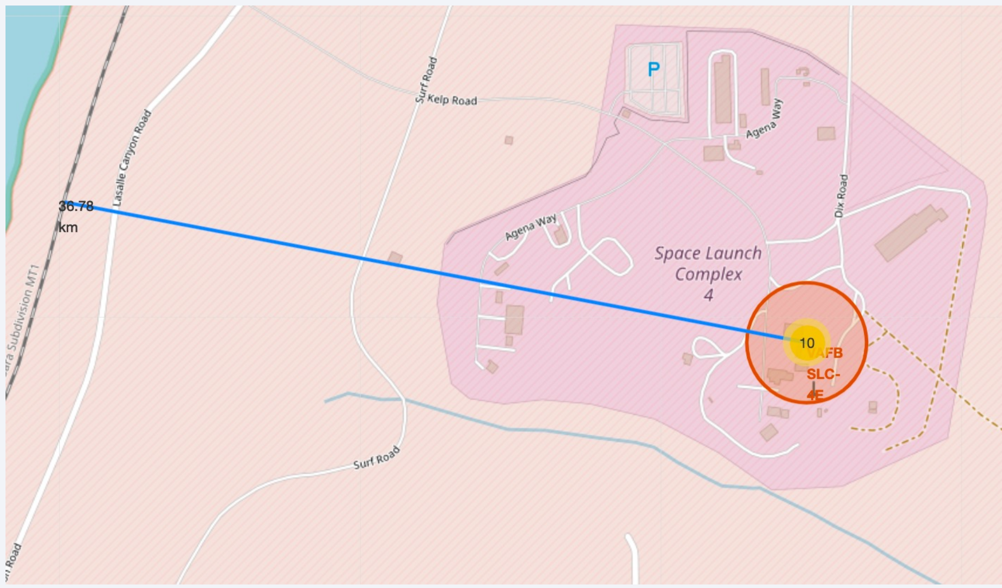
CCAFS-LC-40



CCAFS-SLC-40

KSC-LC-39A has better results, whereas CCAFS-LC-40 has the lowest results.

Railway Point



VAFB-SLC-4E, for example, is too near to a railway. This is also true for other stations. This indicates that the stations are in close proximity to mass transportation.

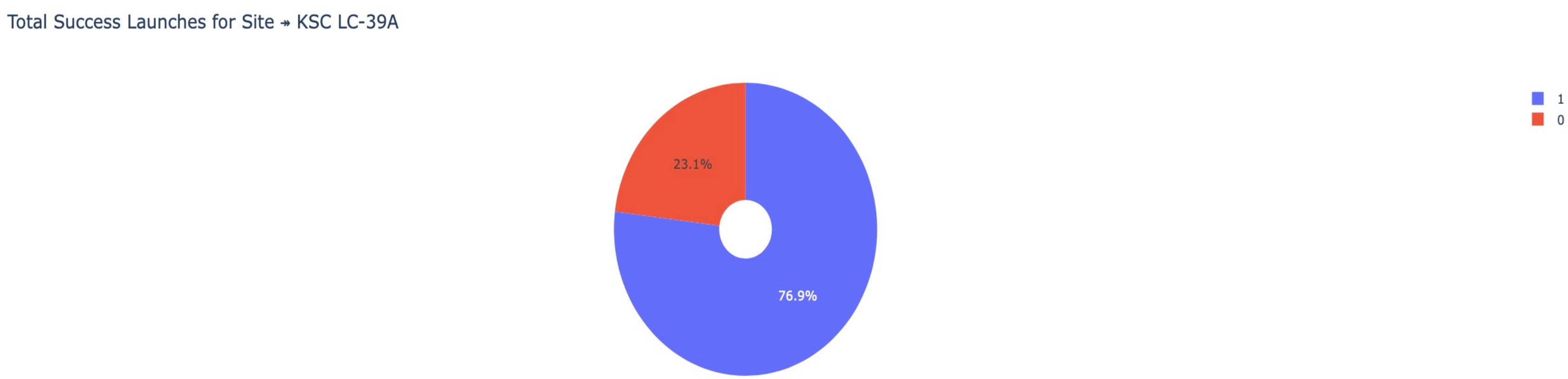
Section 5

Build a Dashboard with Plotly Dash

Space X Total Success Launches by All Sites

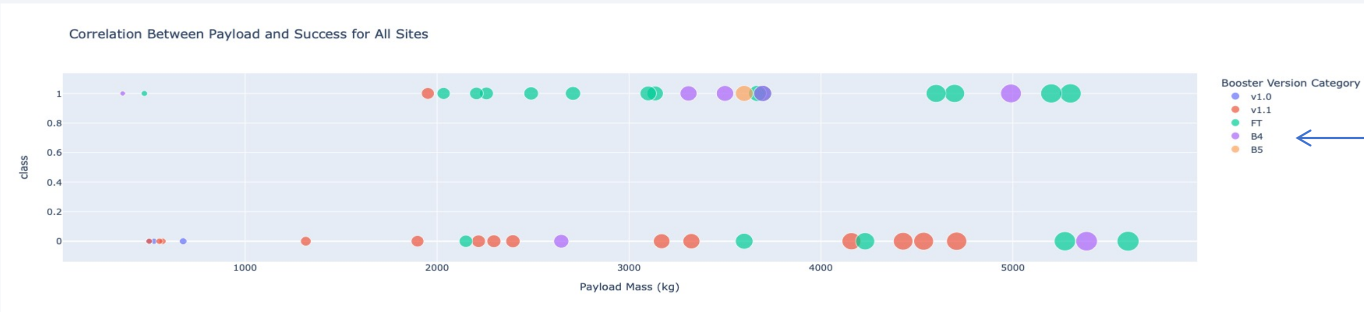
KSC LC-39 has performed better than the other locations, however CCAFS SLC-40 has performed the worst.

Highest launch success (KSC LC 39-A)

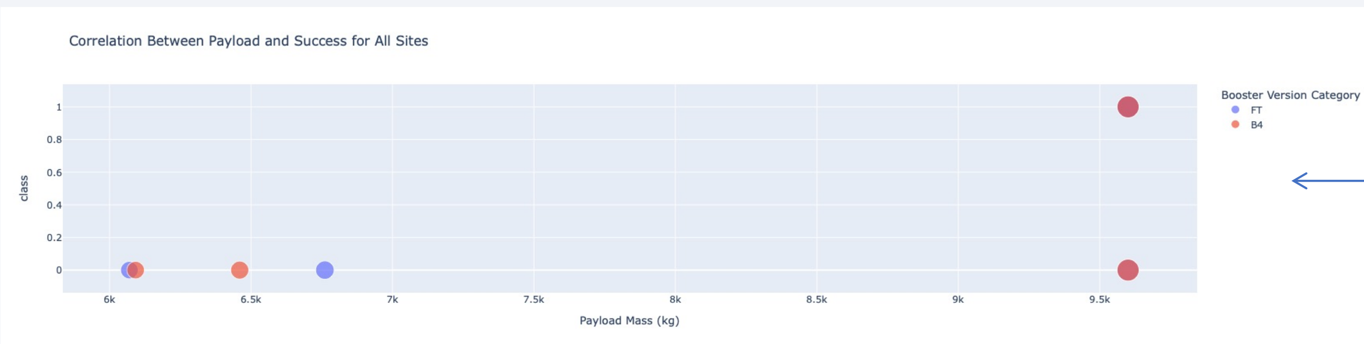


The KSC LC-39 has a success rate of 76.9%. Approximately three out of four launches were successful. Successful launches are represented by blue, whereas failure launches are represented by red.

Correlation between Payload Mass and Success



More successful Launches!!



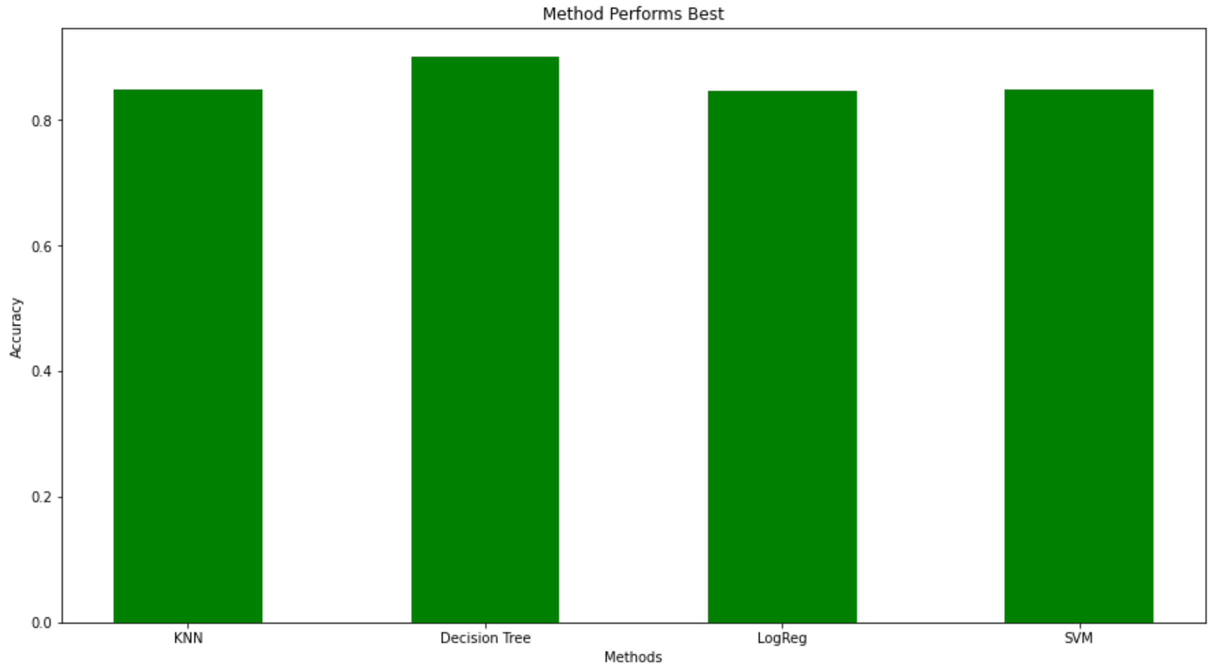
Only one successful launch!

Only two booster here!

Section 6

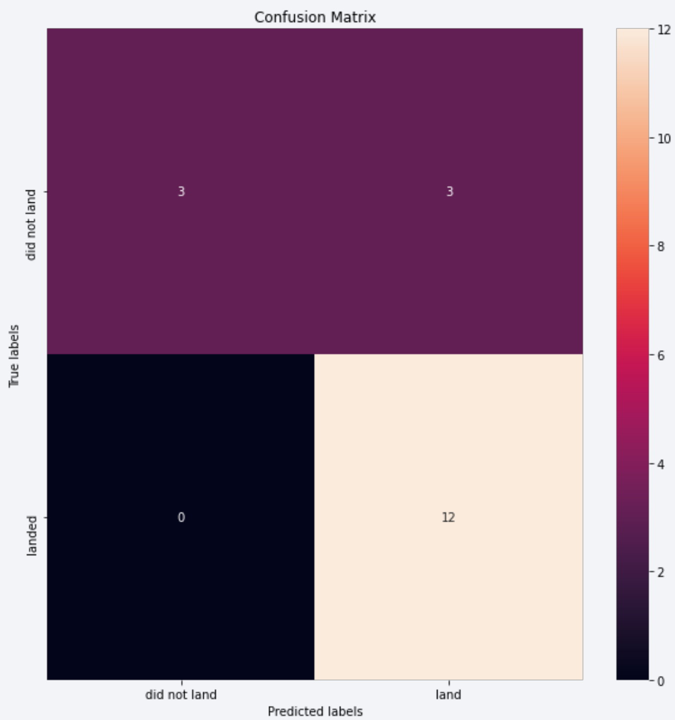
Predictive Analysis (Classification)

Classification Accuracy



The method perform pest is " Decision Tree " which has accuracy of 0.9017

Confusion Matrix



- The models projected 12 successful landings. (TP)
- When the real label was failure landing, the models projected three unsuccessful landings. (TN)
- When the real label was failed landings, the models projected three successful landings (FP)

For all models, the confusion matrix was the same! It's also depicted in the above confusion matrix for the Decision Tree model.

Conclusions

- **The success rate for ES-L1, GEO, HEO, and SSO is about % 100. They're the most successful orbit kinds, according to research.**
- **From 2013 through 2020, launch success is on the rise, with a little dip around 2018.**
- **KSC LC-39 has done better than the others, although CCAFS SLC-40 has done the worst.**
- **Near the Pacific and Atlantic Oceans, stations are located. They're also in close proximity to public transportation.**
- **The method perform pest is " Decision Tree " which has accuracy of 0.9017.**

Appendix

- GitHub url :

https://github.com/mkucukos . It covers the all materials related to this project.

Thanks to IBM Data Science Instructors!


Thank you!