IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Clare Zhong>
<Jul 2022>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Machine Learning model for best prediction result

Introduction

- Project background and context

  SpaceX 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 SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - What interactions amongst features effects the outcome.
  - What conditions needs to be meet to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- **Data Collection**: Data collection from SpaceX API.
- **Data Formatting**: Turn it into a pandas DataFrame
- **Data Filtering**: Filter the DataFrame to include only Falcon9 launches
- **Data Wrangling**: Clean data, check for and deal with the missing values.
- **Data Saving**: Save the processed data as CSV file

Data Collection – SpaceX API

- Getting raw data from API
- Converting response to a .json file
- Turning the data into data frame
- Formatting the data only kept the needed features.

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets
```

We should see that the request was successfull with the 200 status response code

```
[10]: response.status_code
[10]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[11]: # Use json_normalize meethod to convert the json result into a dataframe
data = response.json()
data = pd.json_normalize(data)
```

Using the dataframe `data` print the first 5 rows

```
[12]: # Get the head of the dataframe
data.head()
```

[12]:

links.webcast	links.youtube_id	links.article	links.wikipedia	fairings
ttps://www.youtube.com/watch?v=0a_00nJ_Y88	0a_00nJ_Y88	https://www.space.com/2196-spacex-inaugural-falcon-1-rocket-lost-launch.html	https://en.wikipedia.org/wiki/DemoSat	NaN

- GitHub URL of the completed SpaceX API calls notebook: https://github.com/cclaree/IBM-DS-assignemnt-/blob/main/Final%20spaceX/1.%20jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from its URL with BeautifulSoup
- Extract all column/variable names from the HTML table header
- Create a data frame by parsing the launch HTML tables
- Save the data frame into CSV file

- GitHub URL of the completed web scraping notebook: https://github.com/cclaree/IBM-DS-assignemnt-/blob/main/Final%20spaceX/2.%20jupyter-labs-webscraping.ipynb

To keep the lab tasks consistent, you will be asked to scrape the data from a snapshot of the `List of Falcon 9 and Falcon Heavy launches` Wikipage updated on `9th June 2021`

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

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
[5]: # use requests.get() method with the provided static_url
     data=requests.get(static_url).text
     # assign the response to a object
```

Create a `BeautifulSoup` object from the HTML `response`

```
[6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
     soup = BeautifulSoup(data,"html5lib")
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
[10]: # Use soup.title attribute
      tag_titles = soup.title
      tag_titles = tag_titles.string
      tag_titles

[10]: 'List of Falcon 9 and Falcon Heavy launches - Wikipedia'

[17]: len(tag_titles)

[17]: 54
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

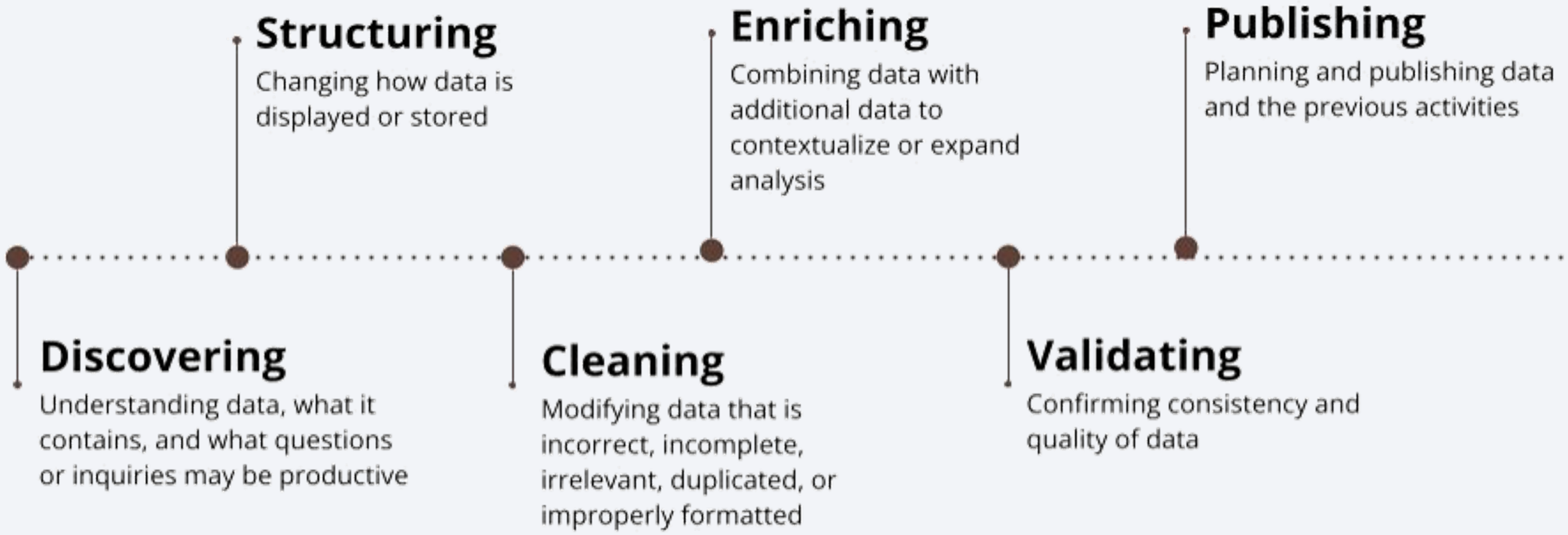
Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
[18]: # Use the find_all function in the BeautifulSoup object, with element type `table`
      # Assign the result to a list called `html_tables`
      html_tables = soup.find_all('table')
      html_tables

[18]: [<table class="multicol" role="presentation" style="border-collapse: collapse; padding: 0; border: 0; background:trans
      parent; width:100%;">

       <tbody><tr>
```

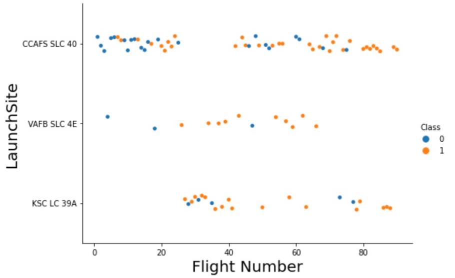
Data Wrangling

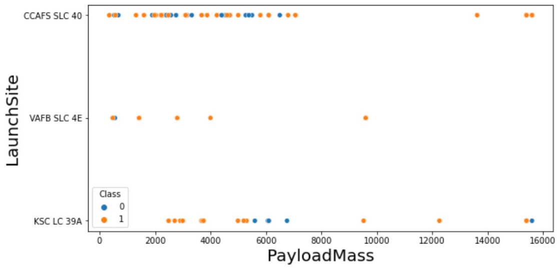


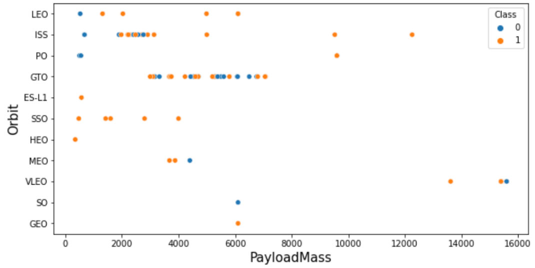
- GitHub URL of the completed notebook: https://github.com/cclaree/IBM-DS-assignemnt-/blob/main/Final%20spaceX/3.labs-jupyter-spacex-Data%20wrangling.ipynb

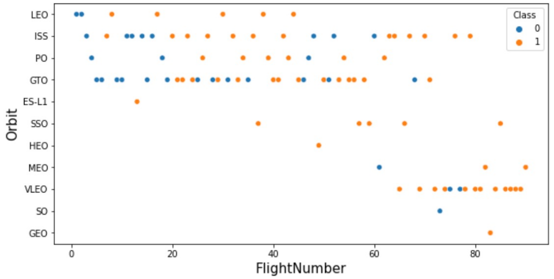
EDA with Data Visualization

- **Visualize the relationship between two features**

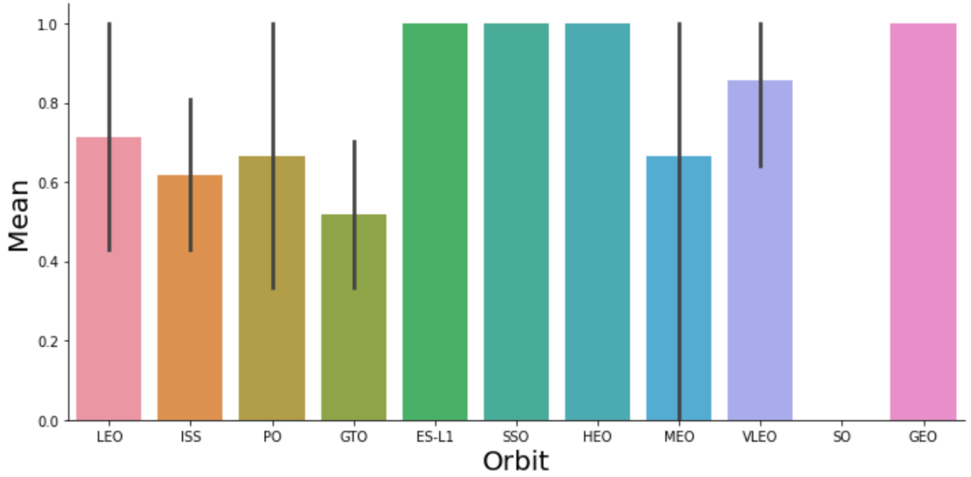




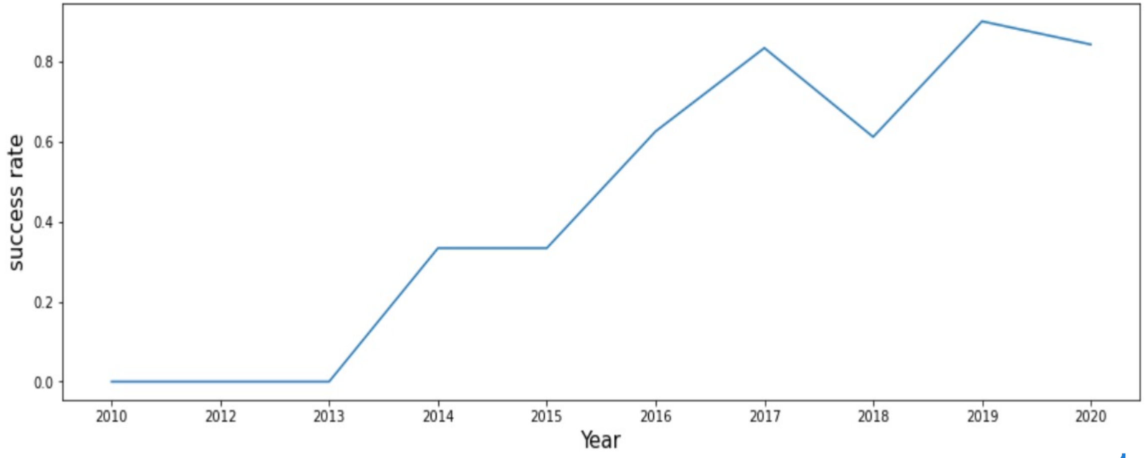




- **Visualize the relationship between success rate of each orbit types**



- **Visualize the launch success yearly trend**



- GitHub URL of the completed EDA with Data Visualization calls notebook: https://github.com/cclaree/IBM-DS-assignemnt-/blob/main/Final%20spaceX/3.labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with SQL

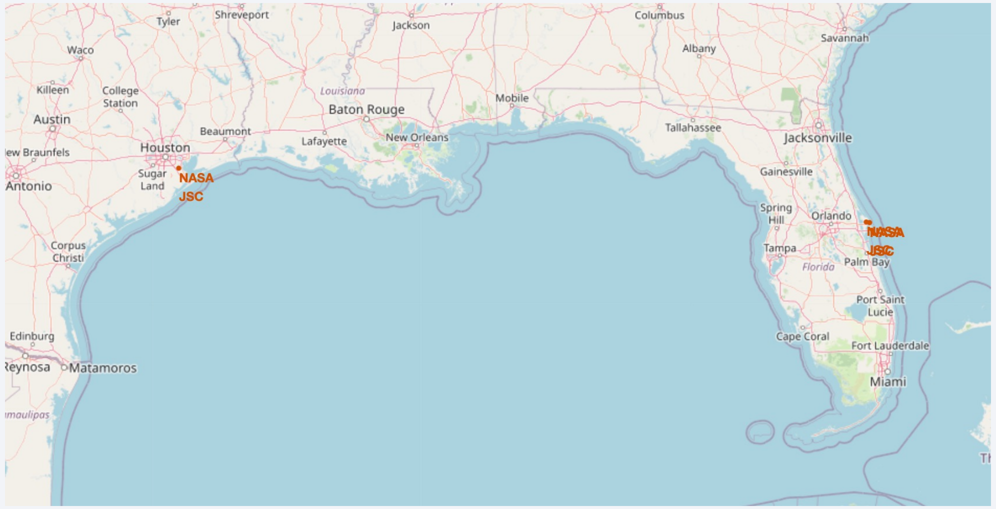
Summary of SQL queries that were used:

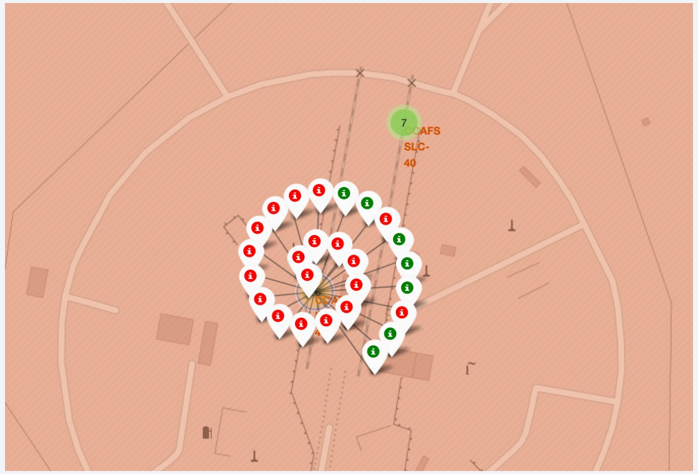
- Display the names of the unique launch sites in the space mission
- Compare the payload mass with boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the total number of successful and failure mission outcomes
- Determine the dates of different landing outcomes

- GitHub URL of the completed notebook: https://github.com/cclaree/IBM-DS-assignemnt-/blob/main/Final%20spaceX/5.jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- Folium Markers were used to show the SpaceX launch sites and their nearest important landmarks like railways, highways, cities and coastlines.
- Polylines were used to connect the launch sites to their nearest land marks.
- Furthermore, Folium Circles were used to highlight circle area of launch sites.
- In order to mark the success/failed launches for each site, marker clusters were used on the map. Whereby red represents rocket launch failures while green represents the successes.

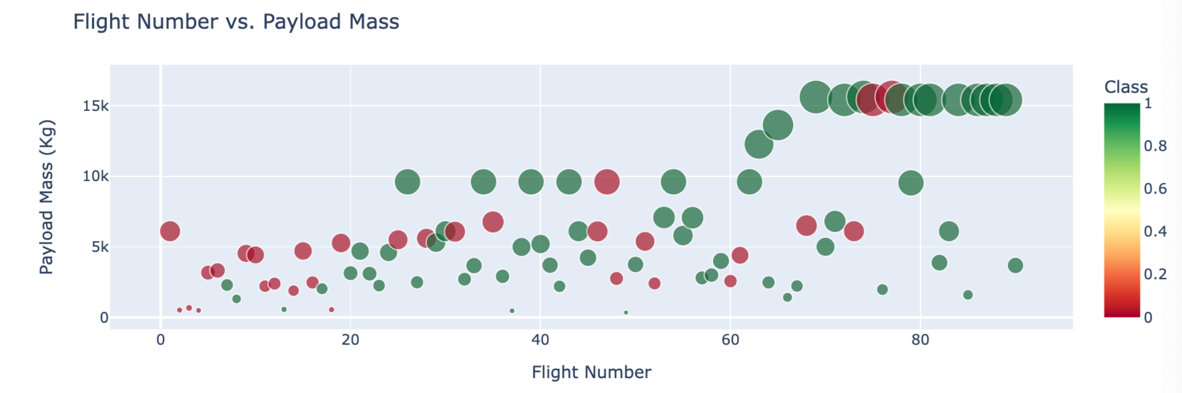


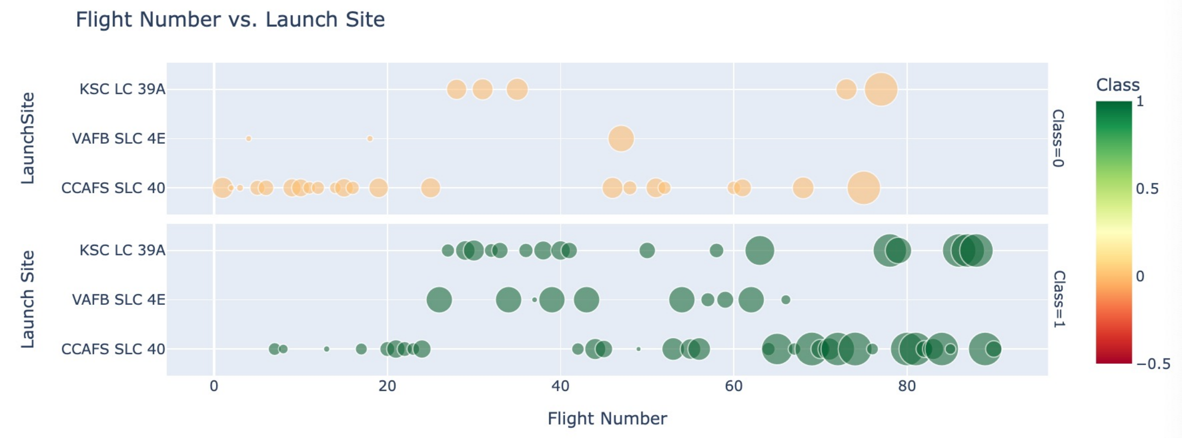


- GitHub URL of the completed notebook: https://github.com/cclaree/IBM-DS-assignemnt-/blob/main/Final%20spaceX/6.lab_jupyter_launch_site_location.ipynb

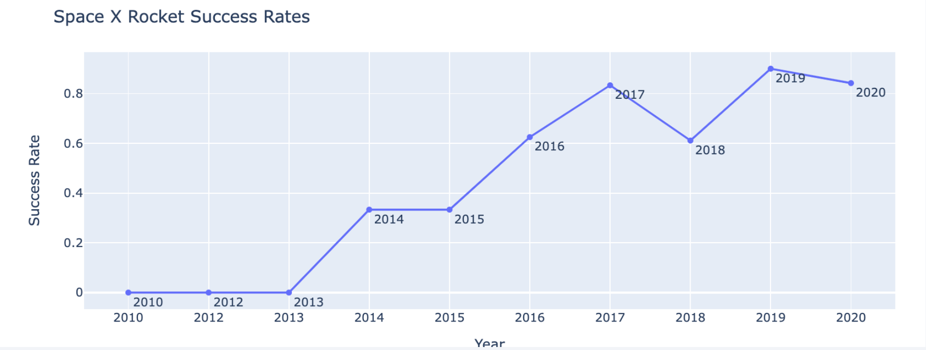
Build a Dashboard with Plotly Dash

- **Scatter chart to visualize the relationship between features with hue = class (launch success or frailer)**

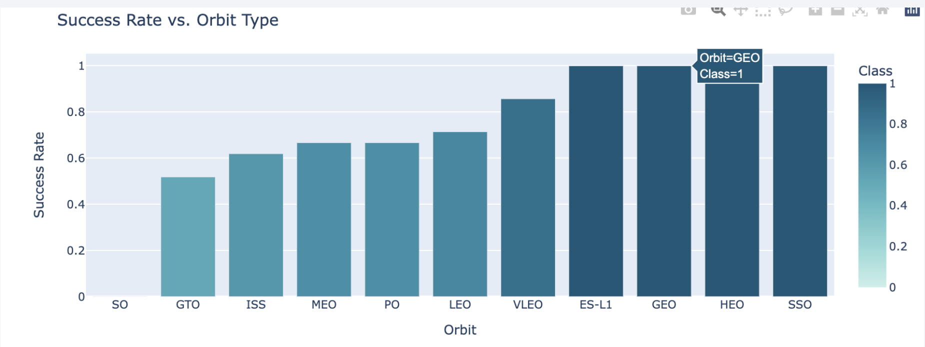




- **Visualize the launch success yearly trend**



- **Visualize the relationship between success rate of each orbit types**



- GitHub URL of the completed notebook: https://github.com/cclaree/IBM-DS-assignemnt-/blob/main/Final%20spaceX/Visualization%20with%20Plotly.ipynb

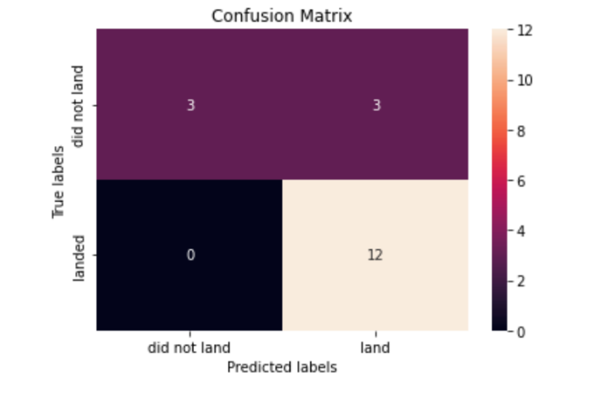
Predictive Analysis (Classification)

- Load data, define X, and y, standardize the data in X, split train test sets.

- Build baseline models with KNN, Decision Tree, Logistic Regression and SVM. Use GridSearchCV to find the best parameters for the model.

- Test and compare the model accuracy with test sets, model.best_score

- Based on the best baseline model and fine tune it to improve model accuracy.



	Algorithm	Accuracy
0	KNN	0.848214
1	Decision Tree	0.875000
2	Logistic Regression	0.846429
3	SVM	0.848214

```
scores = []
for i in range(2,22):
    gs_cv=GridSearchCV(tree,parameters, scoring='accuracy', cv=i)
    tree_cv = gs_cv.fit(X_train, Y_train)
    scores.append(tree_cv.best_score_)
scores
```

```
[0.8888888888888888,
 0.875,
 0.8888888888888888,
 0.8876190476190476,
 0.875,
 0.8896103896103896,
 0.9027777777777777,
 0.875,
 0.8767857142857143,
 0.8722943722943725,
 0.8888888888888892,
 0.8871794871794874,
 0.873809523809524,
 0.8766666666666667,
 0.8875,
 0.9058823529411765,
 0.875,
 0.881578947368421,
 0.9,
 0.8928571428571429]
```

- GitHub URL of the completed notebook: https://github.com/cclaree/IBM-DS-assignemnt-/blob/main/Final%20spaceX/7.%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

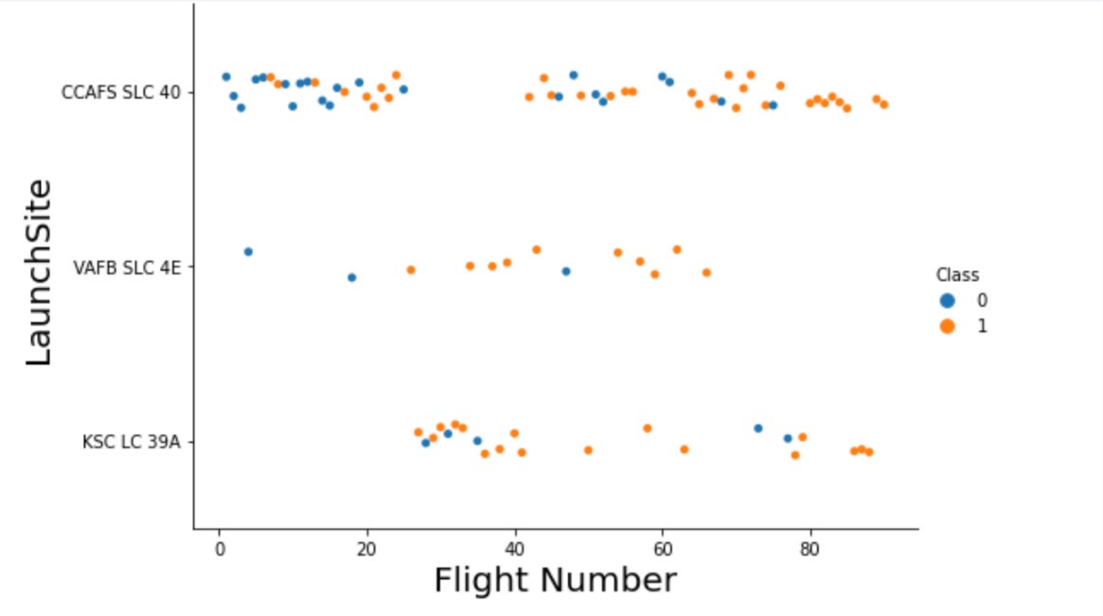
- The exploratory data analysis has shown us that successful landing outcomes are somewhat correlated with flight number. In addition, successful landing outcomes have had a significant increase over time since the year 2013. Some Orbit are significantly favorited (such as VLEO, GTO, ISS with higher flight numbers).
- All launch sites are located near the coast line, not far from railway and highway, significantly distanced from city. This might due to safety and observation reasons, also to ensure the transportation of equipment and research material..
- The machine learning models that were built, were able to predict the landing success of rockets with as high as an accuracy score of 90.58% with decision tree model. This accuracy can be increased in future projects with more data, but need to make sure the model will not be overfitting with the train sets.

Section 2

Insights drawn from EDA

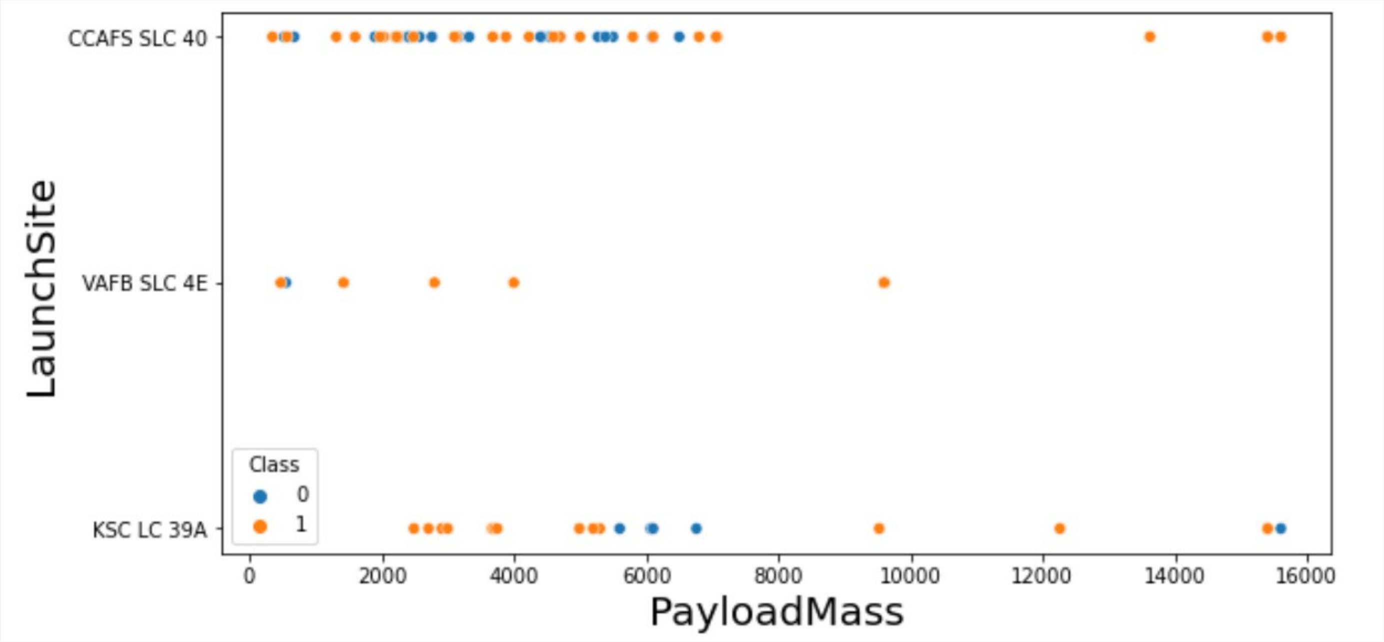
Flight Number vs. Launch Site

- It shows more successful landings as the flight numbers increased. It also seems that launch site **CCAFS SLC 40** had the most number of landing attempts while the site **VAFB SLC 4E** had the least number of attempts.



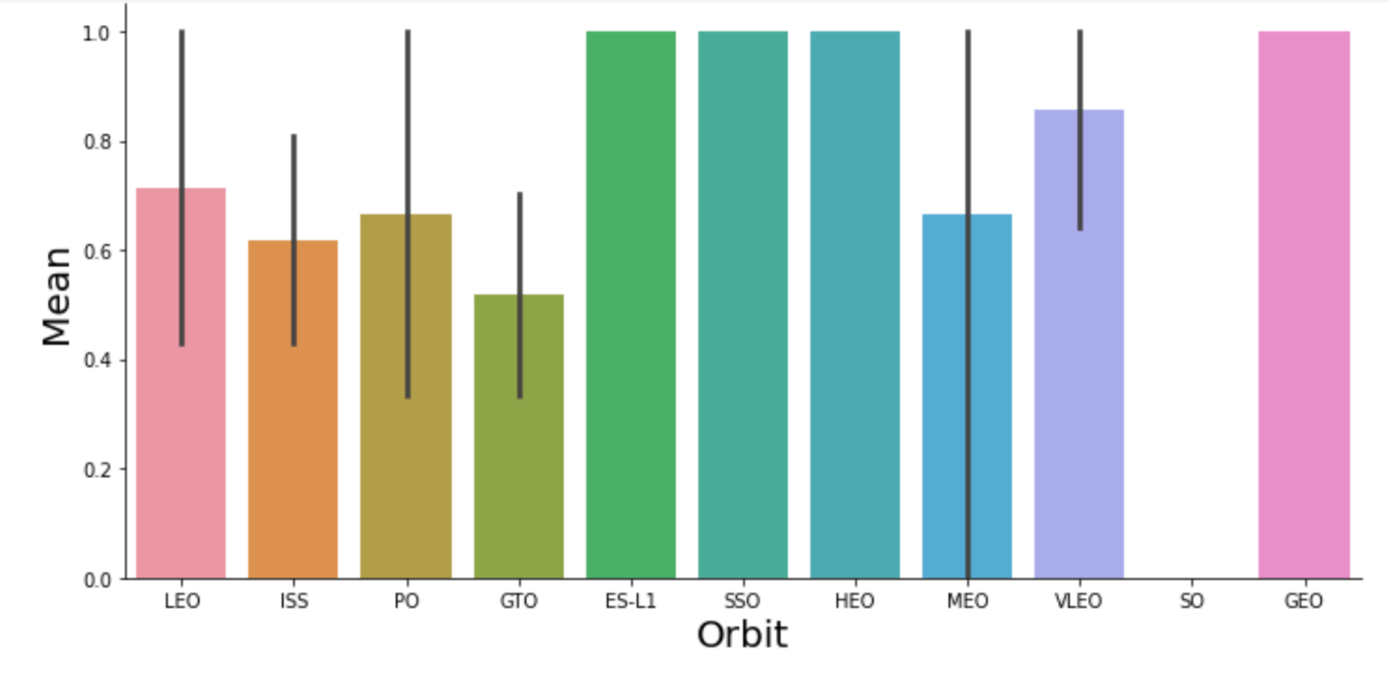
Payload vs. Launch Site

- VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).
- Overall the PayloadMass greater 8000 has significant higher launch success rate, but with much less launches.



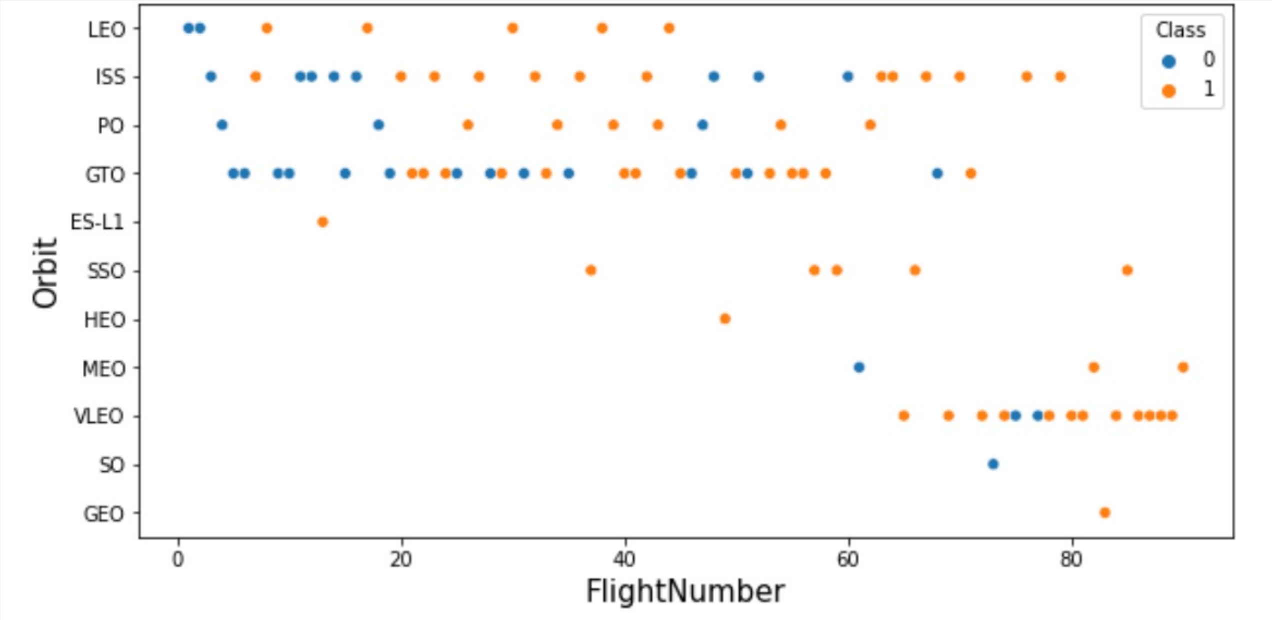
Success Rate vs. Orbit Type

- The orbit types SSO, HEO, GEO and ES-L1 had the highest success rate and based on current data there is no std.
- The max and min of success rate of Orbit type MEO with highest differences.
- Orbit SO with no success launch yet.



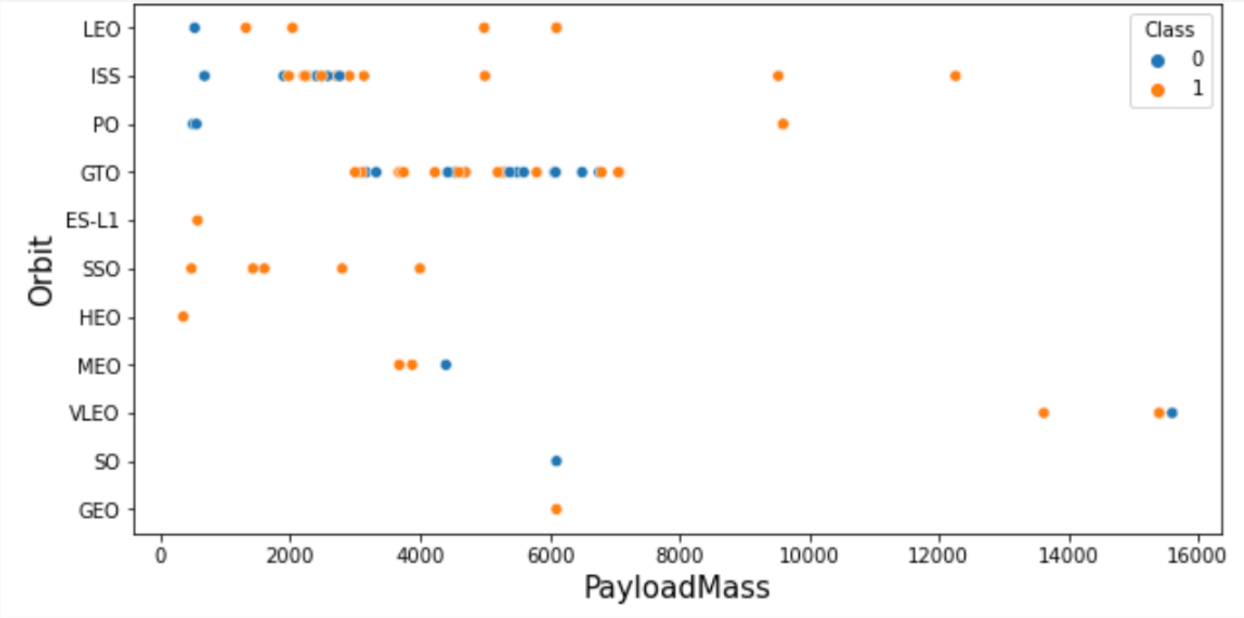
Flight Number vs. Orbit Type

- It is clear that some Orbit type have more flights than others. VLEO only has flight numbers higher than 60. But it sems no clear relation between Orbit type and flight number.



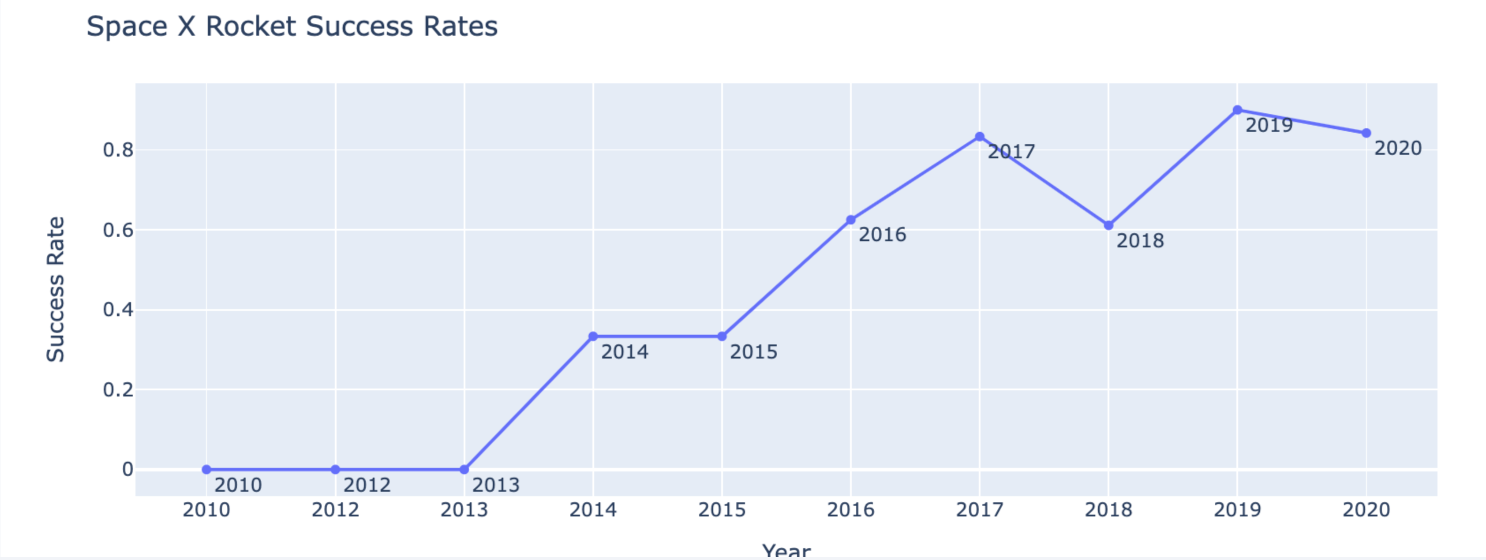
Payload vs. Orbit Type

- Overall, there is not much heavy payload. For Orbit type ISS, LEO,PO heavier the payload is higher the success rate. But for Orbit type MEO and VLEO seems the opposite.
- For the rest cannot see the trend since the launches were not enough or the range of payload is not wide enough.



Launch Success Yearly Trend

- It shows that the success rate were no improve in first 3 years, since 2013 the rate has significantly increasing. From Y2017 to Y2020, the rate is fluctuating between 0.6 and 0.8.



All Launch Site Names

- 3 unique launch site names:
  - CCAFS SLC 40
  - VAFB SLC 4E
  - KSC LC 39A

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b
Done.
```

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

Launch Site Names Begin with 'CCA'

[6]:
```
%sql SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

[6]:

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

- The information in the table displays the total payload mass launched by NASA .

```
[7]: %sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';

 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

[7]:

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

- Below is the average payload mass carried by booster version F9 v1.

```
[8]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEX \
     WHERE BOOSTER_VERSION = 'F9 v1.1';

 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.c
Done.
```

[8]:

Average Payload Mass by Booster Version F9 v1.1
2928

First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad

```
[9]: %sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad" FROM SPACEX \
WHERE LANDING__OUTCOME = 'Success (ground pad)';

 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.dat
Done.
```

[9]:

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.data
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Successful Mission
100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Failure Mission
1

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Total Number of Successful and Failure Mission" FROM SPACEX \
WHERE MISSION_OUTCOME LIKE 'Success%' OR MISSION_OUTCOME LIKE 'Failure%';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Total Number of Successful and Failure Mission
101

```
%sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission", \
    sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" \
FROM SPACEX;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Successful Mission	Failure Mission
100	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mas

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

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX \
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Booster Versions which carried the Maximum Payload Mass
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 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 the in year 2015

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/b
Done.
```

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

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE year(DATE) = '2015' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/b
Done.
```

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

```
%sql SELECT month(DATE) as Month, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE year(DATE) = '2015' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/b
Done.
```

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

```
%sql SELECT {fn MONTHNAME(DATE)} as "Month", BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE year(DATE) = '2015' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/b
Done.
```

Month	booster_version	launch_site
January	F9 v1.1 B1012	CCAFS LC-40
April	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

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 as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Landing Outcome	Total Count
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

```
%sql SELECT COUNT(LANDING__OUTCOME) AS "Rank success count between 2010-06-04 and 2017-03-20" FROM SPACEX \
WHERE LANDING__OUTCOME LIKE '%Success%' AND DATE > '2010-06-04' AND DATE < '2017-03-20' ;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Rank success count between 2010-06-04 and 2017-03-20
8

Section 3

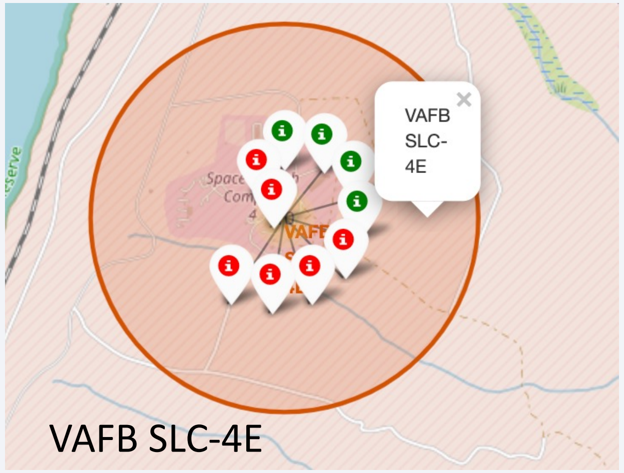
Launch Sites Proximities Analysis

Mark all launch sites on a map

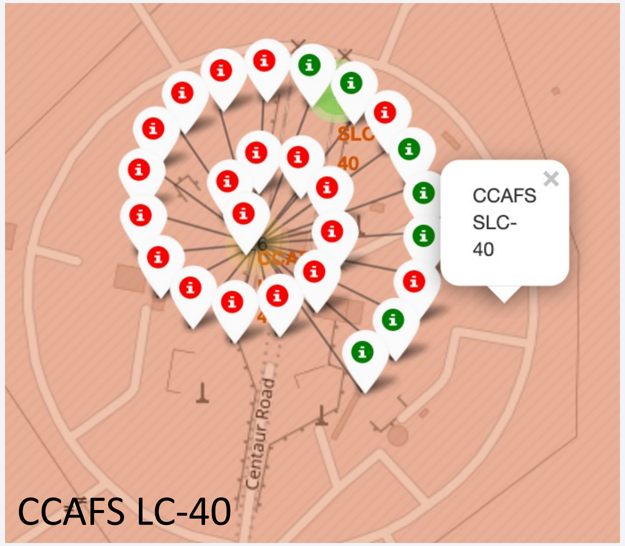
- SpaceX launch sites are near to the US coasts, 3 in Florida and 1in California region.

Mark the success/failed launches for each site on the map

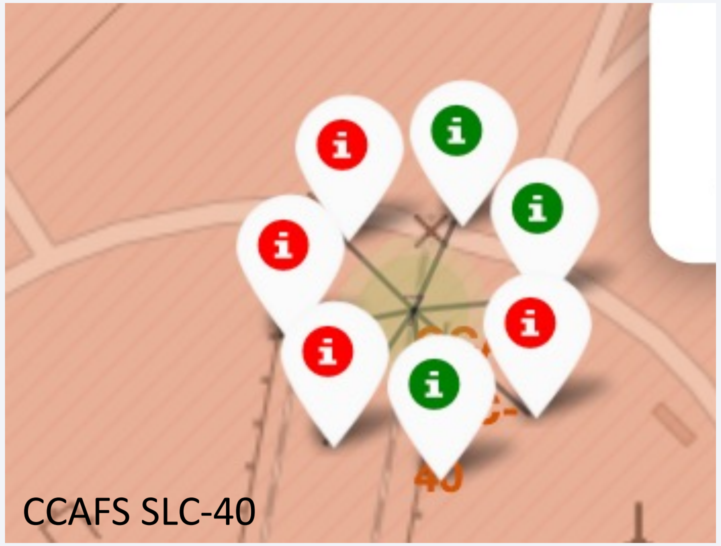
- Green marker means successful launches
- Red marker means failure launches

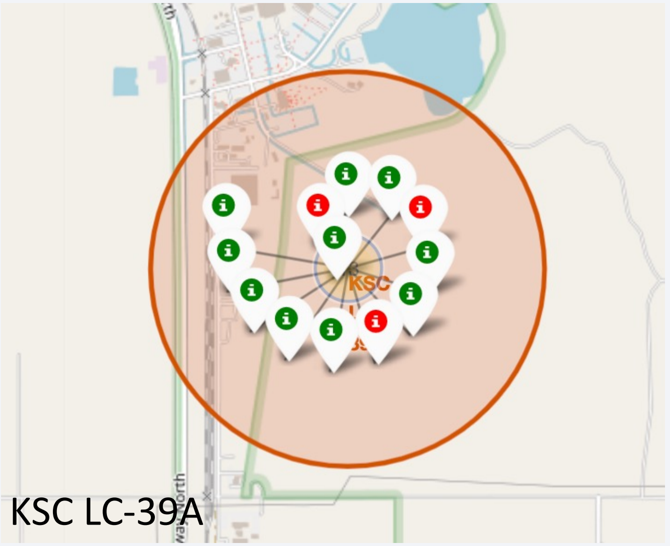


VAFB SLC-4E



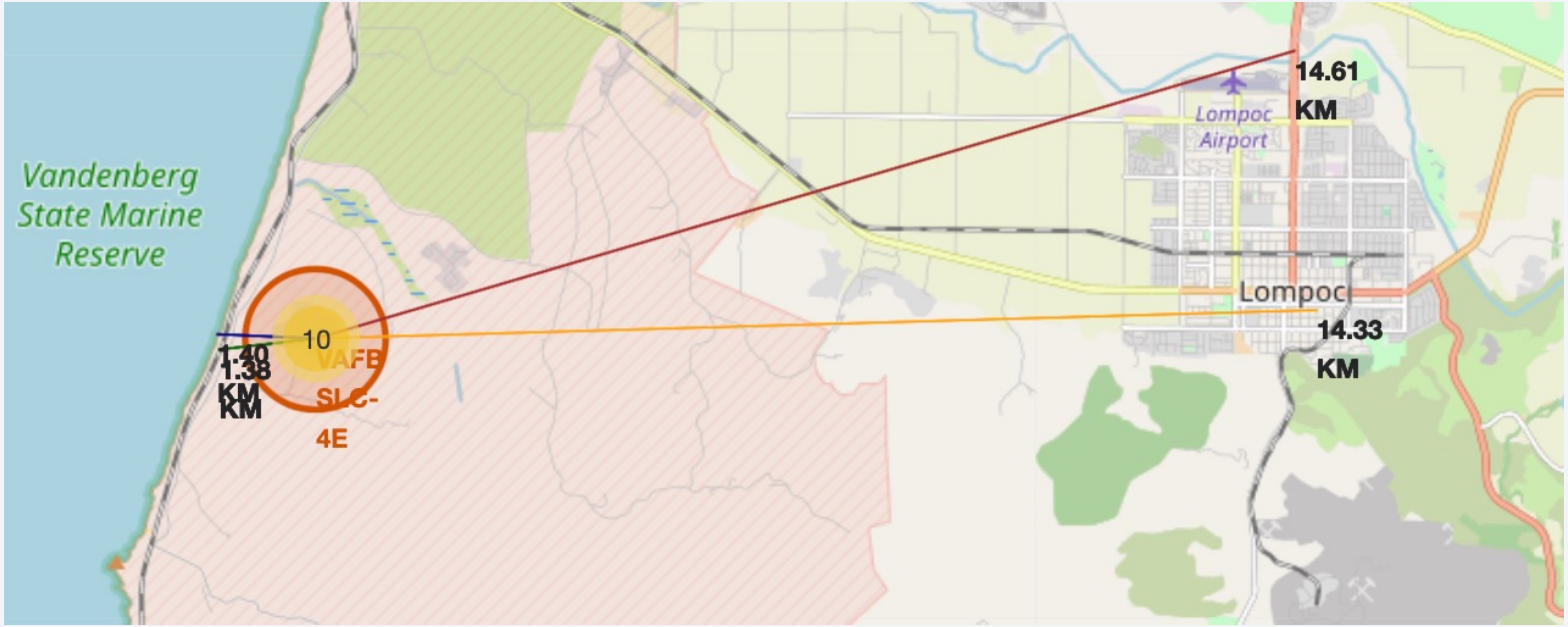
CCAFS LC-40


CCAFS SLC-40


KSC LC-39A

Distances between a launch site to its proximities

- Calculate the distances between a launch site to its proximities: railways (dark-green line), highways(brown line), city(orange line), coastline(dark-blue line). Below is the example for launch site VAFB SLC-4E

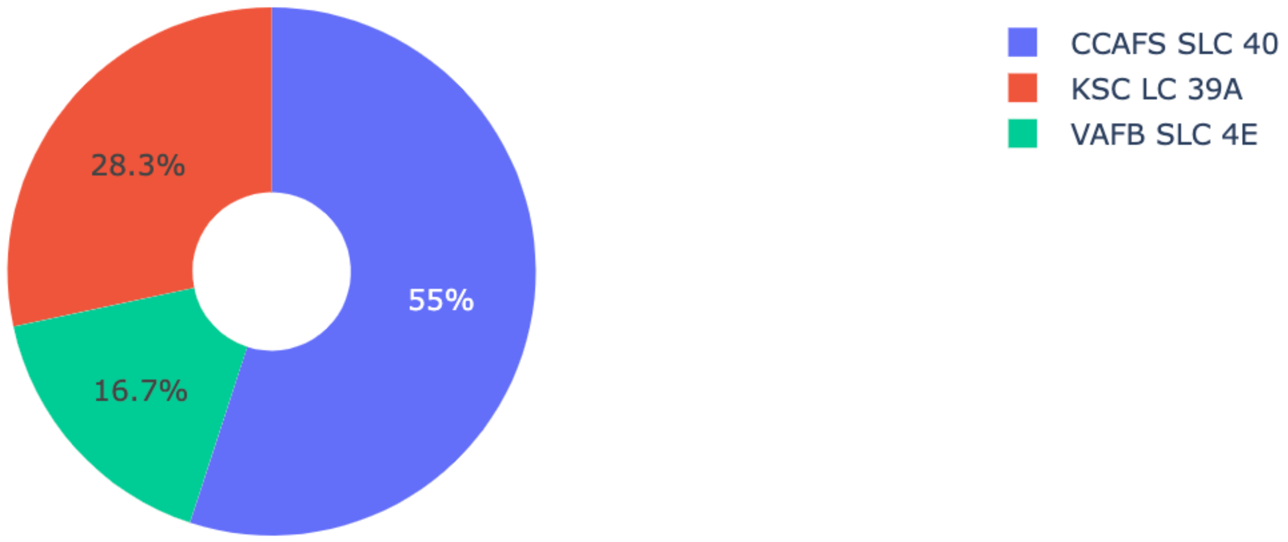


Section 4

Build a Dashboard with Plotly Dash

Launch site launch success ratio

- CCAFS SLC 40 is the launch site with most success launches among all successful launches.
- KSC LC 29A is the launch site with highest launch success ratio of 77.27%



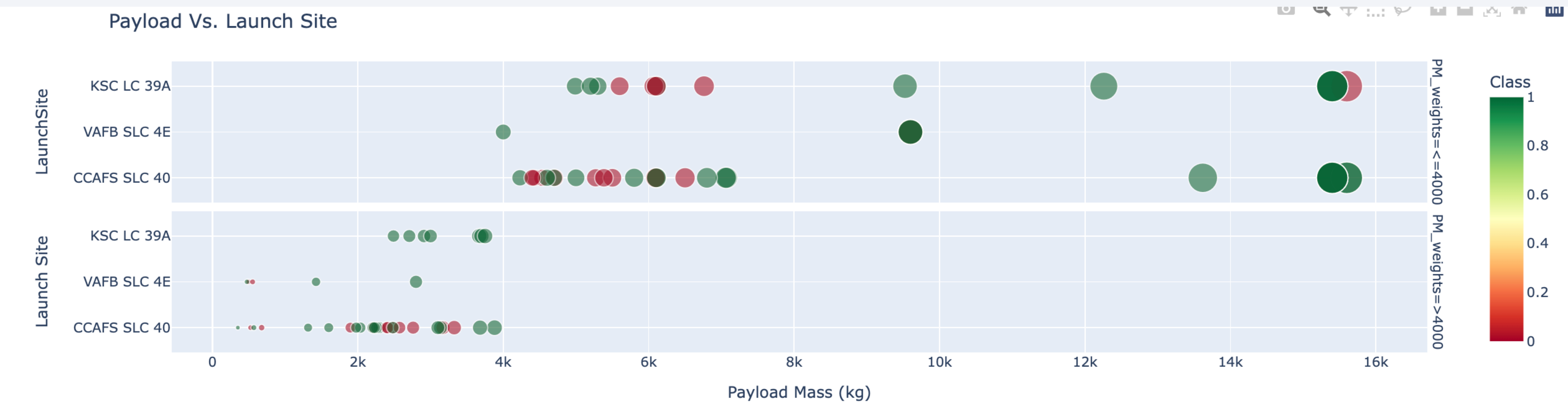

```
[103]: pd.DataFrame(temp/df.groupby('LaunchSite').Class.count() *100).reset_index()
```

[103]:

	LaunchSite	Class
0	CCAFS SLC 40	60.000000
1	KSC LC 39A	77.272727
2	VAFB SLC 4E	76.923077

Payload vs. Launch Outcome scatter plot for all sites

- Screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider, payload of 4000kg as cutoff line.

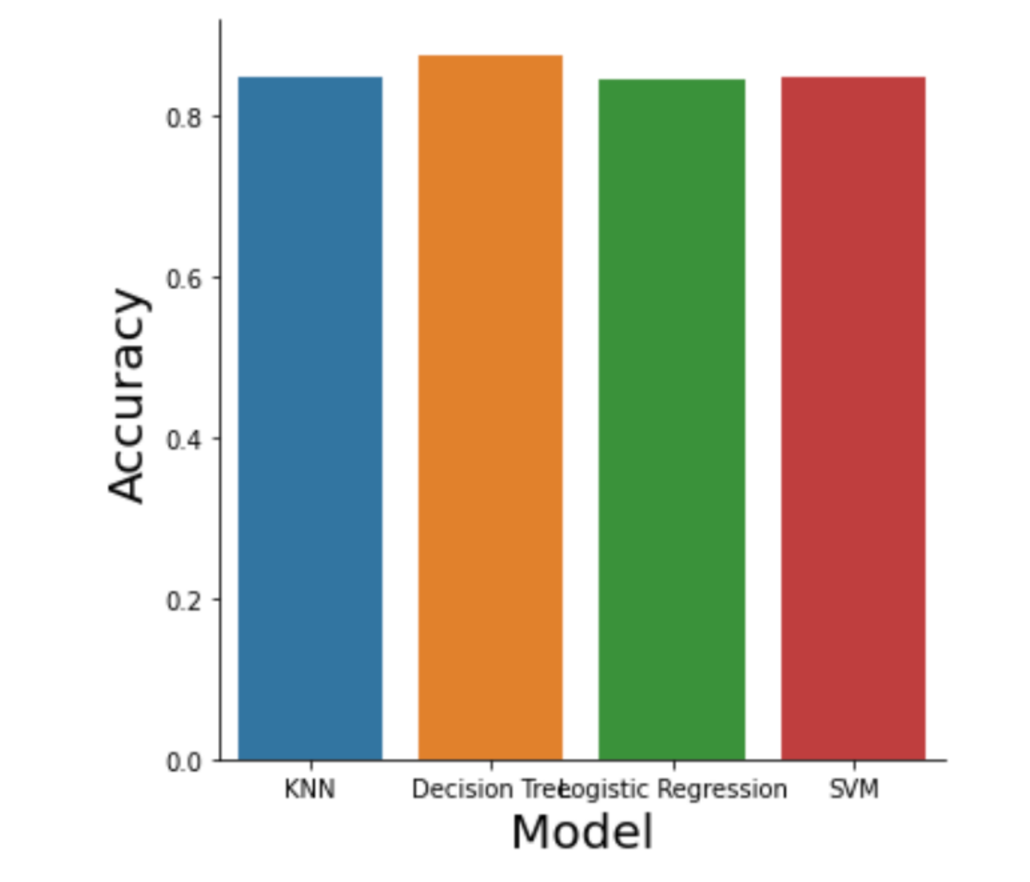


Section 5

Predictive Analysis (Classification)

Classification Accuracy

The best accuracy rate are quite similar, but among these 4 baseline model decision tree model has the highest classification accuracy.

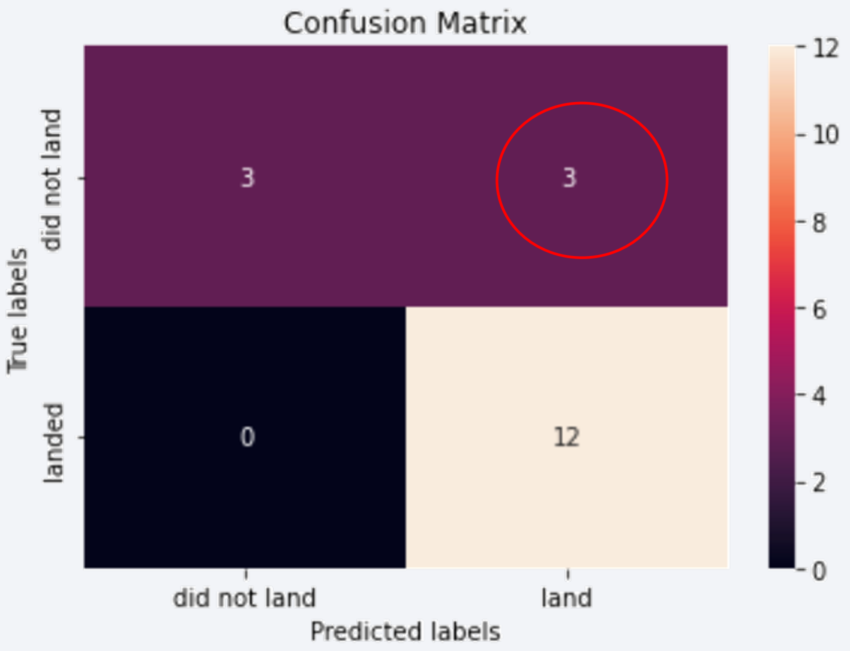


Confusion Matrix

- Confusion matrix of the best performing model see below in right side. Seems model shows false negative, which is type II error.

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)



Conclusions – learnings from SpaceX

- All SpaceX launch sites are located near the coast, away from nearby cities. This enabled to them to test their rocket landings without much interference.
- Site KSC LC-39A had the highest launch success rate out of all the launch sites.
- From 2013 onwards, the success rate of rocket landings significantly increased. It was also apparent that landing success increased with flight number
- Data was used to train machine learning model, with best model outcome around 90% prediction accuracy using a decision tree model.

```
gs_cv=GridSearchCV(tree,parameters, scoring='accuracy', cv=8)
tree_cv = gs_cv.fit(X_train, Y_train)
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_le
af': 2, 'min_samples_split': 5, 'splitter': 'best'}
accuracy : 0.9027777777777777
```

Thank you!