IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Luis M. Jerez
August 4th, 2022

Outline

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

Executive Summary

- Summary of methodologies

➢ Data Collection

➢ 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

➢ EDA results

➢ Interactive analytics demo in screenshots

➢ Predictive analysis

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 lead to rocket successful landings?

➢ The interaction among variables to determine the success rate of successful landings.

➢ To find the best operation conditions that lead to successful landings.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and dropping of irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Plotting : Scatter and Bar Charts to show relationship between variables and show patterns of data.
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune and evaluate classification models

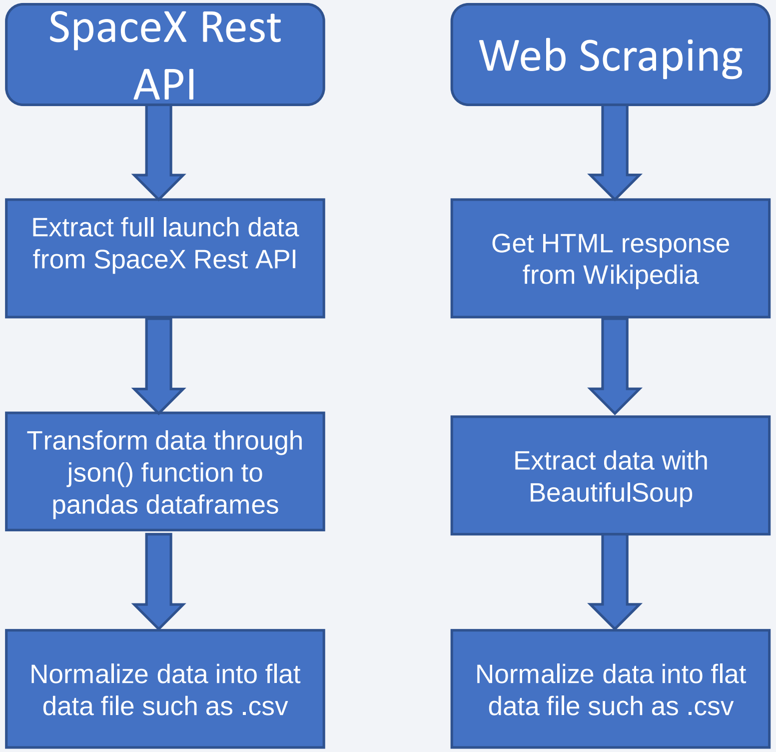
Data Collection

- Data Set Collection process:
- SpaceX Rest API:

1. Extract full launch data from SpaceX Rest API
2. Transform data through json() function to pandas dataframes
3. Normalize data into csv file

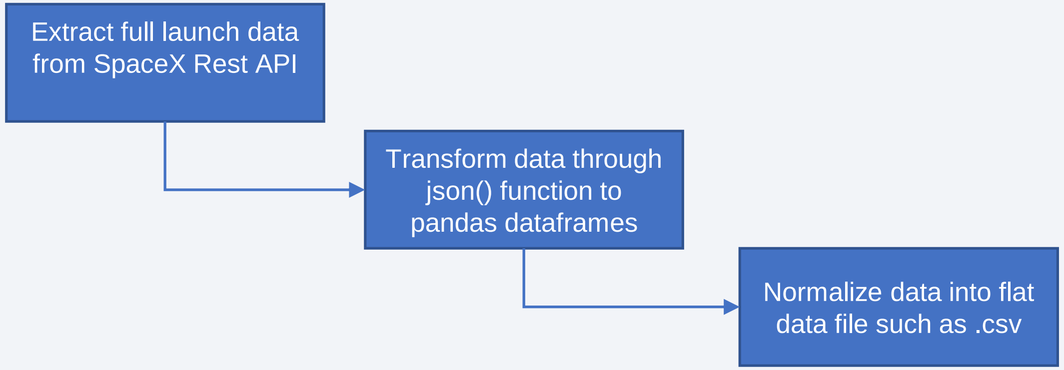
- Web Scraping:

1. Get HTML response from Wikipedia for Falcon 9 launch records
2. Extract data with BeautifulSoup
3. Normalize data into flat data file such as .csv



Data Collection – SpaceX API

- Data Collection with Space Rest API:



- Github URL: [Data Collection with Space Rest API]()

1. Requesting rocket launch data from SpaceX API

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [8]: response = requests.get(spacex_url)
```

2. Transforming response to Json file and turning it to Pandas dataframe

```
In [10]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/dat
```

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

```
In [11]: response.status_code
Out[11]: 200
```

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

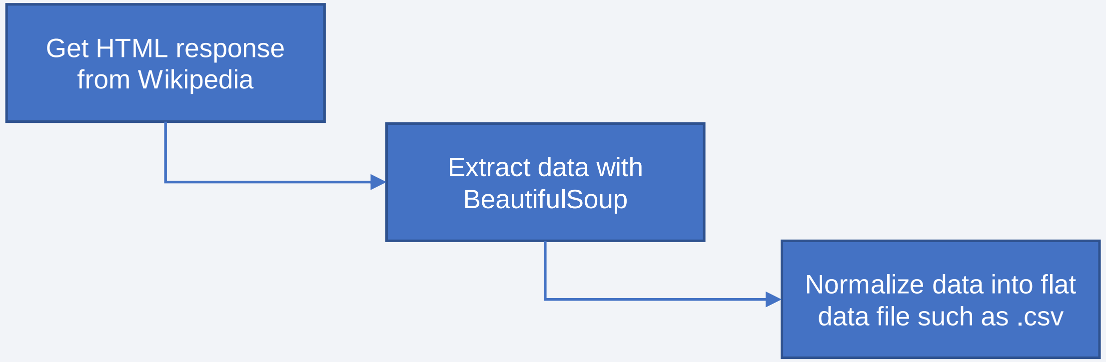
```
In [12]: # Use json_normalize meethod to convert the json result into a dataframe
         data = pd.json_normalize(response.json())
```

3. Filter dataframe and export it to flat file (.csv)

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

Data Collection - Scraping

- Web Scraping



- GitHub URL: [Web Scraping]()

1. Request HTML response from Wikipedia

```
In [5]:
# use requests.get() method with the provided static_url
response = requests.get(static_url)

response.status_code

print(response.url)

# assign the response to a object
x = response
```

```
https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
```

2. Extract data with BeautifulSoup

```
In [10]:
# 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')
```

Starting from the third table is our target table contains the actual launch records.

```
In [11]:
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

3. Create dataframe by parsing HTML tables and export it to flat file (.csv)

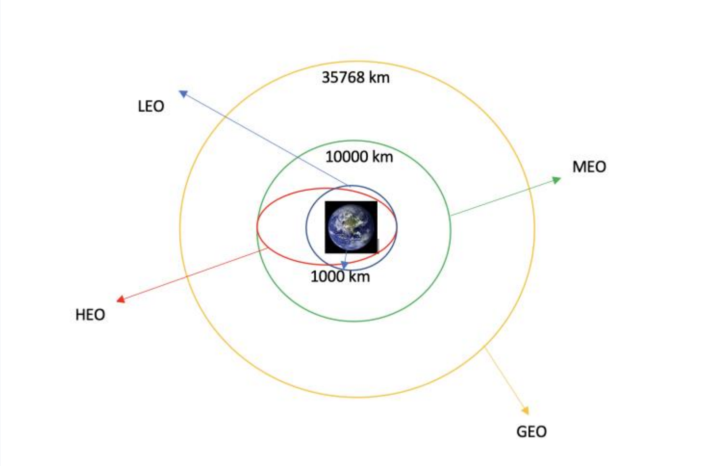
```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

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

Data Wrangling

- Main steps:

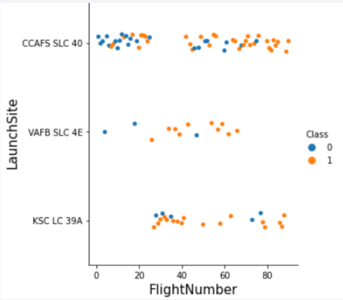
1. Load SpaceX dataset and perform exploratory data analysis (EDA) and determine the training labels
2. Calculate the number of launches from each site and the occurrence of each orbit
3. Create a landing outcome label from Outcome column and calculate general success rate



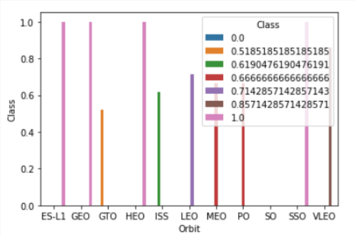
- GitHub URL: [Data Wrangling]

EDA with Data Visualization

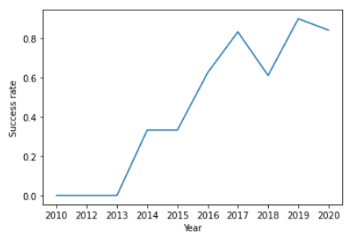
- Scatter Plots: for determining relationship between two variables as independent and dependent.
  - Flight Number vs. Payload Mass
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Orbit vs. Flight Number
  - Payload vs. Orbit
  - Orbit vs. Payload Mass



- Bar charts plots: useful to compare data performance between variables or groups of variables.
  - Success Rate vs. Orbits



- Line Chart: useful to see trends through a timeline and helpful to make predictions.
  - Success Rate vs. Year



Github URL: [EDA with Data Visualization]

EDA with SQL

SQL queries performed:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA‘
- Display the 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.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass.
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

GitHub URL: [EDA with SQL]

Build an Interactive Map with Folium

- All launch sites were marked, and added map objects such as markers, circles and lines to show the success or failure of launches for each site on the folium map. In that way is possible to find out visually patterns or factors to take into account during the whole process.
- Feature launch outcomes (failure or success) were assigned to class 0 and 1, 0 for failure and 1 for success.
- Through color-labeled marker clusters, launch sites with high success rate were identified.
- Distances between a launch site and some points of interest were calculated, as per as follows:

  - railways, highways and coastlines.

  - distance away from cities.

GitHub URL: Interactive Map with Folium

Build a Dashboard with Plotly Dash

Interactive Dashboard with following charts:

- Pie Chart: Total Launches by Site, to see the proportion between different launch sites and corresponding success rates.
- Scatter Plot with Payload selection slider: Outcome vs. Payload mass by booster version, helpful to find some pattern, check best/worst payload ranges and make useful predictions for future projects.

GitHub URL: Interactive Dashboard with Plotly Dash

Predictive Analysis (Classification)

Predictive Analysis (Classification):

- Load data using numpy and pandas, transform and split data into training and testing sets.
- Built of different machine learning models and tune different hyperparameters using GridSearchCV function.
- Method score to calculate the accuracy as the metric for the models.
- Feature engineering and algorithm tuning for improving and finally find best performance classification model.

GitHub URL: Predictive Analysis (Classification)

Results

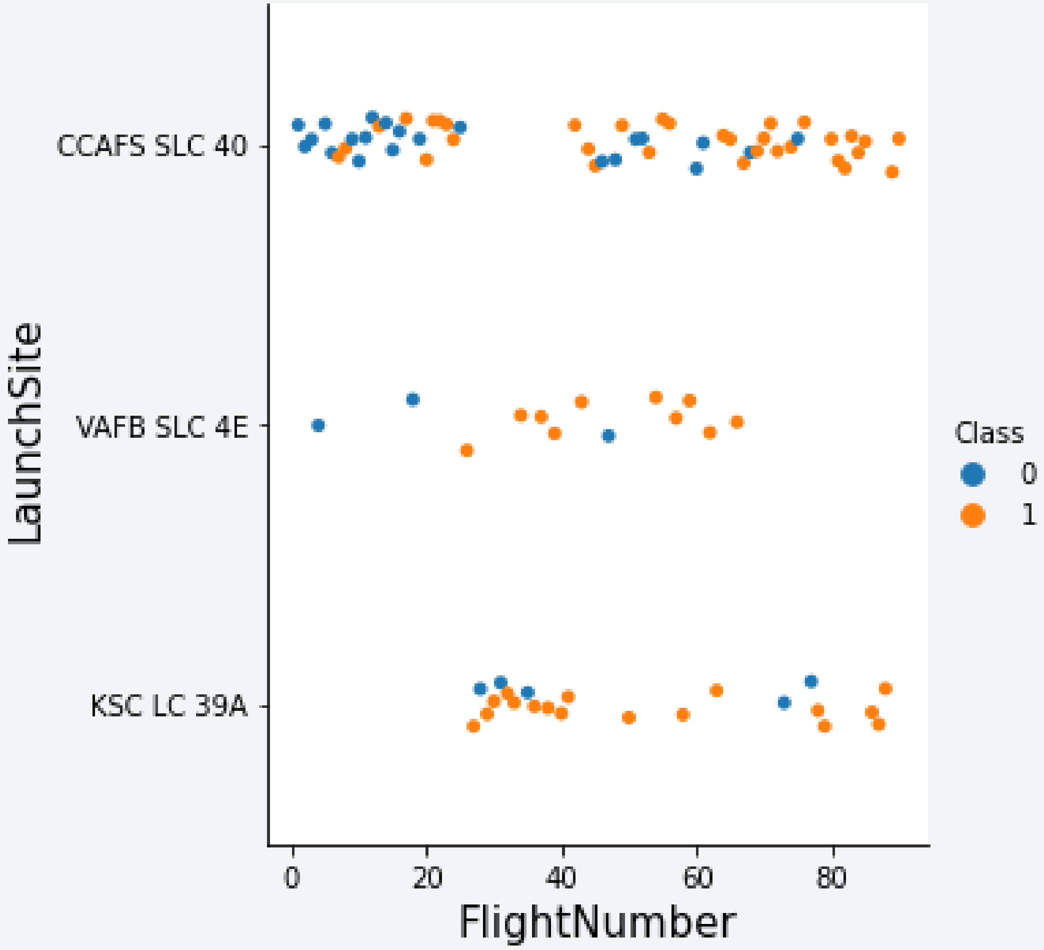
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

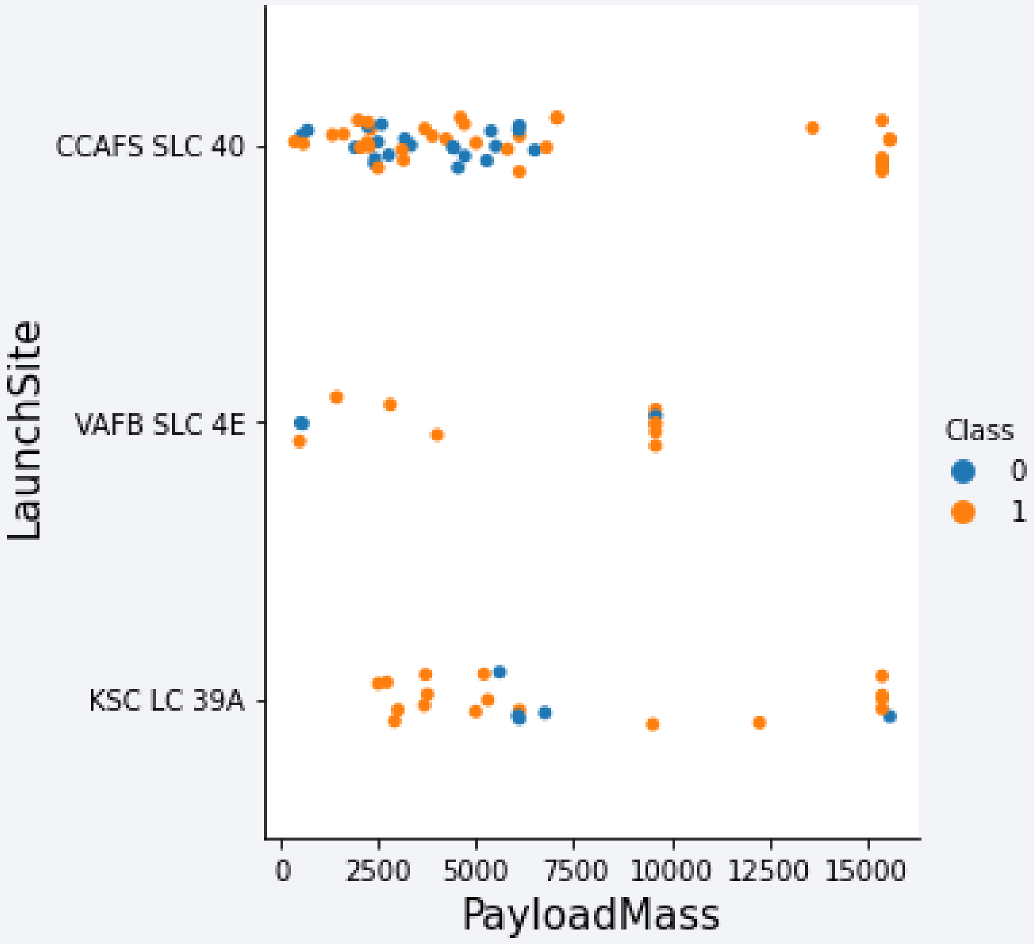
Flight Number vs. Launch Site

- For larger number of flights, the larger probability of success for each site



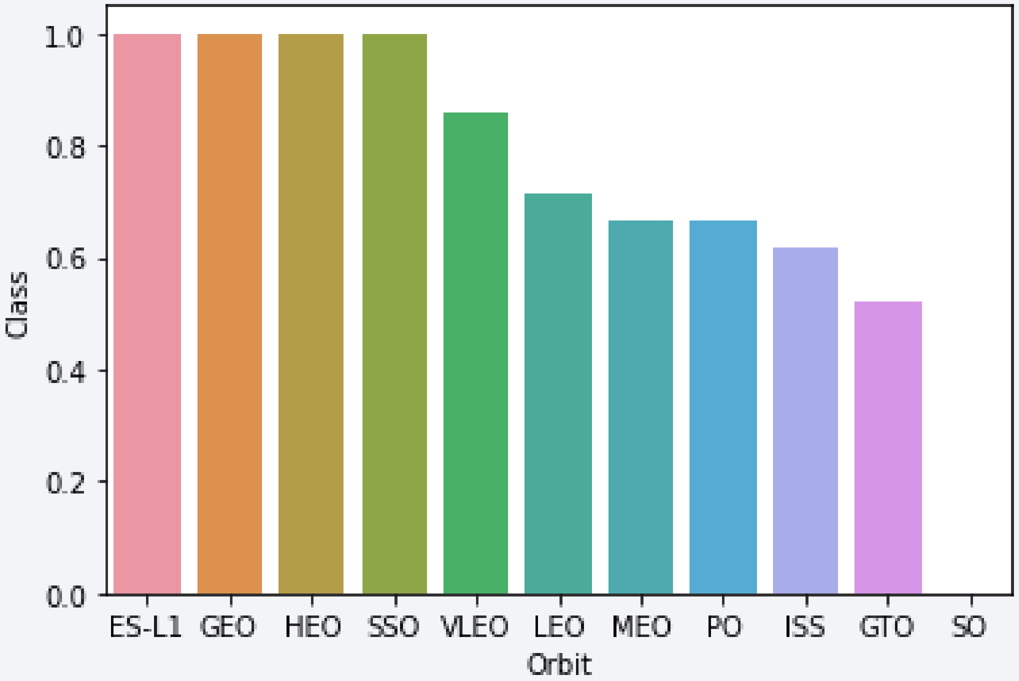
Payload vs. Launch Site

- For Launch Site CCAFS SLC 40 there is a slight pattern showing more probability of success rate for larger payloads (as per around 15,000kg).
- Similar analysis for VAFB SLC 4E as per around 10,000kg range payloads.
- For KSC LC 39A there is no clear pattern to remark on.



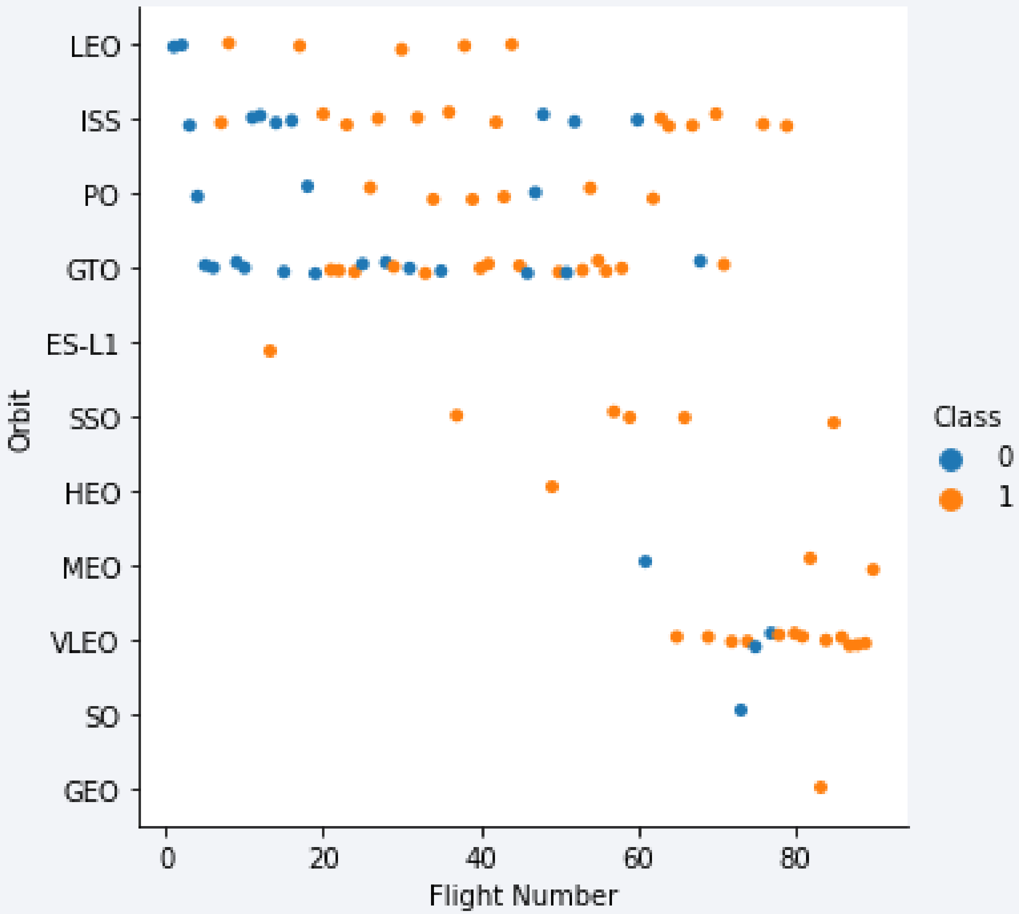
Success Rate vs. Orbit Type

- Bar chart shows that ES-L1, GEO, HEO and SSO are the orbits with highest success rate.



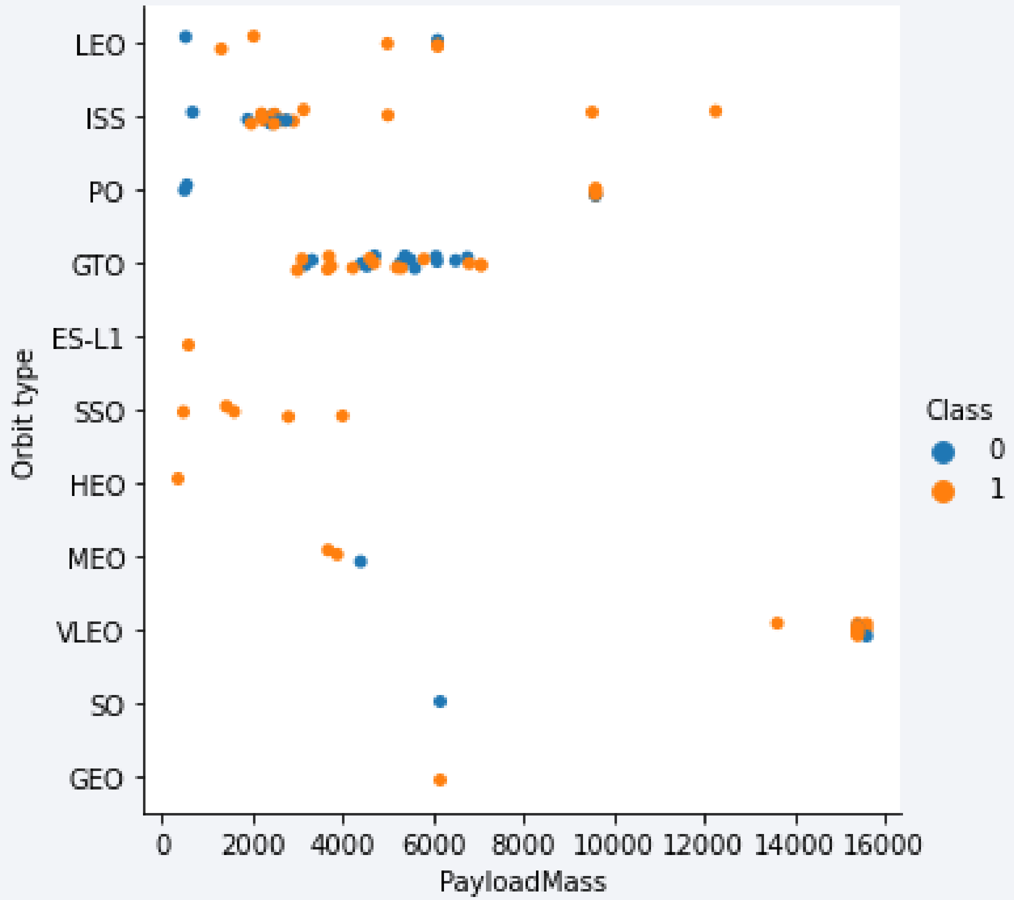
Flight Number vs. Orbit Type

- Scatter point plot showing Flight number vs. Orbit type. Most clear pattern seems to be for LEO orbit, where the higher flight number, the higher success rate. On the other hand, for GTO there is no clear correlation.



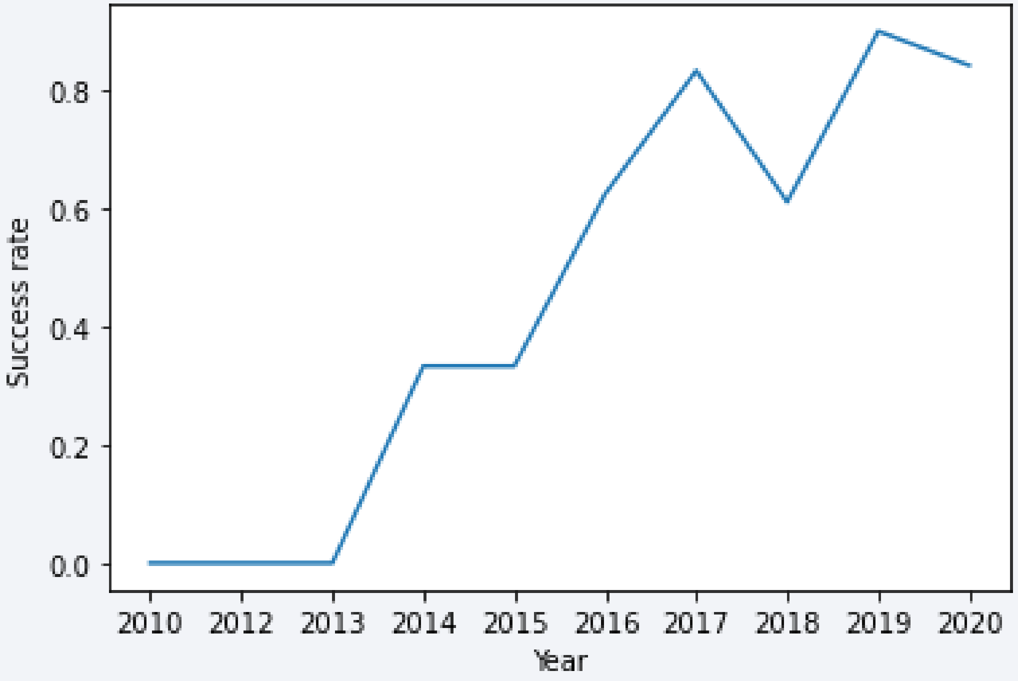
Payload vs. Orbit Type

- This Scatter plot shows slight positive correlation for SSO, LEO, ISS and PO, as per when Payload increases, success rate landing does too.



Launch Success Yearly Trend

- Line chart for yearly average success rate, it shows remarkable increasing trend from 2013 until 2020.



All Launch Site Names

- Unique launch sites list generated by SQL Distinct query from SpaceX data, as follows:

```
%sql select distinct (Launch_Site) from (SPACEXTBL)

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

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

Launch Site Names Begin with 'CCA'

- SQL Query: to find 5 records where launch sites begin with `CCA`.

Task performed with “Like” and “Limit” SQL commands, as shown below.

```
In [8]: %sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;

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

Out[8]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- SQL Query: to calculate the total payload carried by boosters from NASA.

Query performed through function Sum, as shown below.

```
%sql select sum(PAYLOAD_MASS__KG_) as Total_Payload_Mass_Kg from SPACEXTBL where Customer like "NASA (CRS)";
```

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

Total_Payload_Mass_Kg
45596

Average Payload Mass by F9 v1.1

- SQL Query: to calculate the average payload mass carried by booster version F9 v1.1

Query performed through Avg function, as shown below.

```
[12]: %sql select avg(PAYLOAD_MASS__KG_) as Avg_Payloadmass from SPACEXTBL where Booster_Version = 'F9 v1.1';

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

[12]:

Avg_Payloadmass
2928.4

First Successful Ground Landing Date

- SQL Query: to find the dates of the first successful landing outcome on ground pad.

Query performed through function Min, as shown below.

```
%sql select min(DATE) as first_succesful_landing from SPACEXTBL where "Landing _Outcome"='Success (drone ship)';
```

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

first_succesful_landing
06-05-2016

Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL Query: to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

Where and And conditions were applied to set this query, as shown below.

```
[12]: %sql select Booster_Version from SPACEXTBL where "Landing _Outcome"='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;

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

[12]:

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

Total Number of Successful and Failure Mission Outcomes

- SQL Query: to calculate the total number of successful and failure mission outcomes.

Subqueries and Like features were applied in this query, as shown below.

```
[22]: %sql select (select count(MISSION_OUTCOME) from SPACEXTBL where Mission_Outcome LIKE '%Success%') as Successful_Mission_Outcomes,\
(select count(MISSION_OUTCOME) from SPACEXTBL where Mission_Outcome LIKE '%Failure%') as Failure_Mission_Outcomes;

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

[22]:

Successful_Mission_Outcomes	Failure_Mission_Outcomes
100	1

Boosters Carried Maximum Payload

- SQL Query: to list the names of the boosters which have carried the maximum payload mass.

Query performed through Subqueries and Max function, among others, as shown below.

```
%sql select distinct BOOSTER_VERSION, max(PAYLOAD_MASS__KG_) from SPACEXTBL where PAYLOAD_MASS__KG_= \
(SELECT MAX(PAYLOAD_MASS__KG_) from SPACEXTBL) GROUP BY Booster_Version ORDER BY max(PAYLOAD_MASS__KG_);
```

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

Booster_Version	max(PAYLOAD_MASS__KG_)
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- SQL Query: to list the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.

Substring and subqueries features where performed to obtain this query, as shown below.

```
[40]: %sql SELECT SUBSTRING (DATE,4,2) as Month,\
(select ("Landing _Outcome") from SPACEXTBL where "Landing _Outcome" LIKE '%drone%') as Landing_Outcome,\
BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where SUBSTRING (DATE, 7, 4)='2015';

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

[40]:

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
02	Failure (drone ship)	F9 v1.1 B1013	CCAFS LC-40
03	Failure (drone ship)	F9 v1.1 B1014	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1016	CCAFS LC-40
06	Failure (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	Failure (drone ship)	F9 FT B1019	CCAFS LC-40

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

- SQL Query: to 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. Query made by using Where, Group by and Order by features, as

```
[54]: %sql SELECT "Landing _Outcome", count ("Landing _Outcome") as "Count" FROM SPACEXTBL WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017'\
Group by "Landing _Outcome" ORDER BY Count ("Landing _Outcome") DESC;

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

[54]:

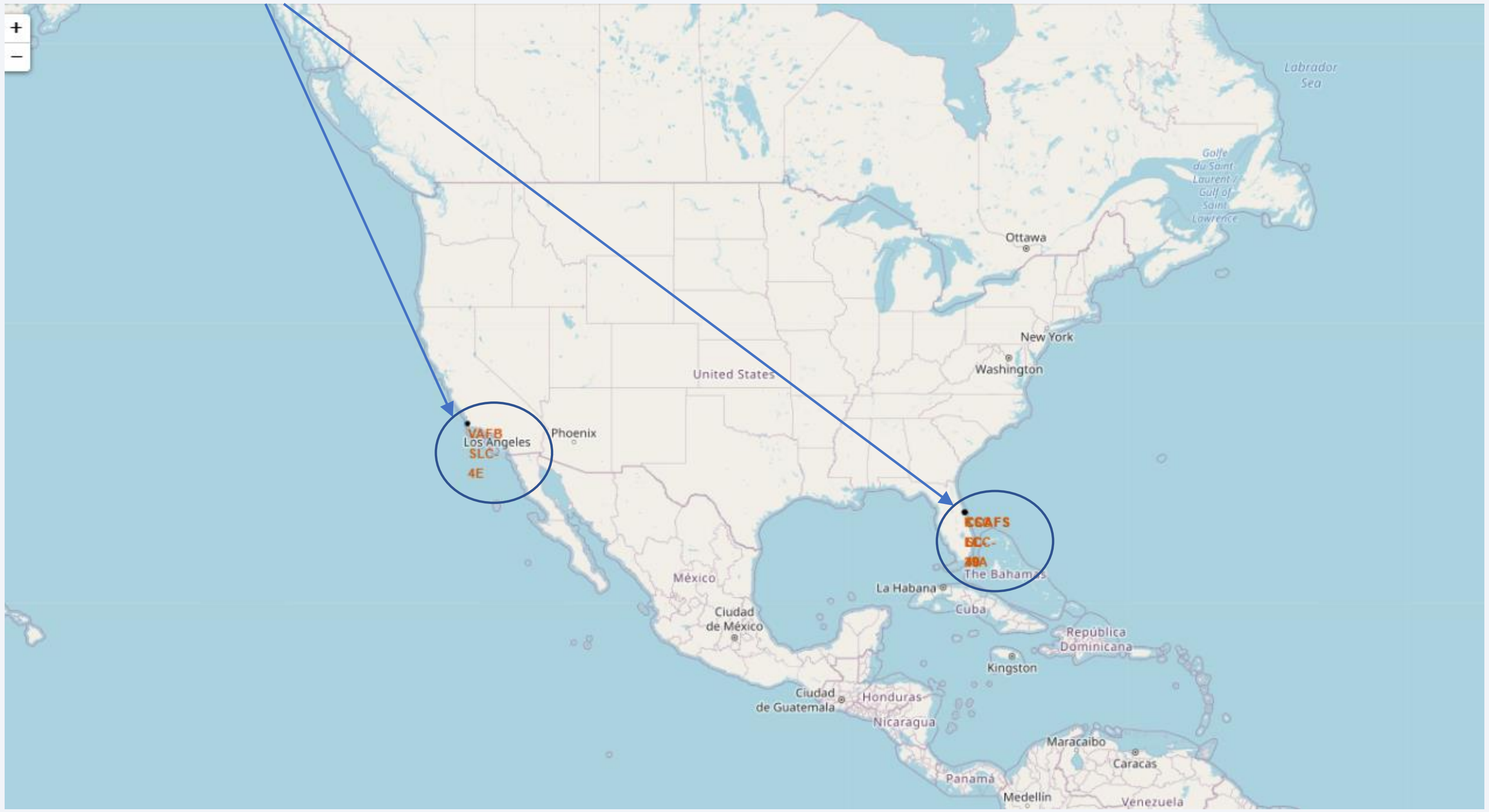
Landing _Outcome	Count
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

Section 3

Launch Sites Proximities Analysis

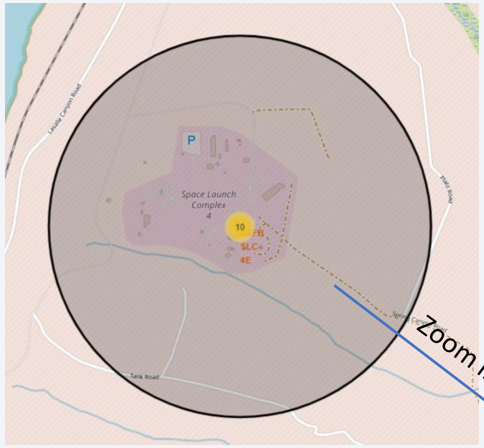
All Launch Sites Map

- As shown on map screenshot, launch sites are located on USA coasts, near Florida and California.

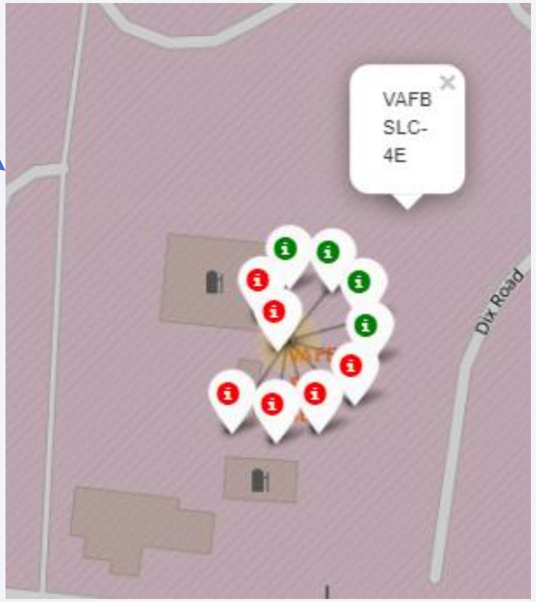


Color Labeled Markers for Launch Sites

California Launch Sites



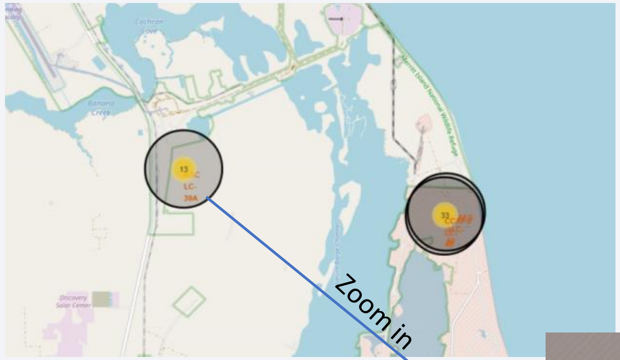
Zoom in



Green markers: successful Launch Sites

Red markers: failure Launch Sites

Florida Launch Sites

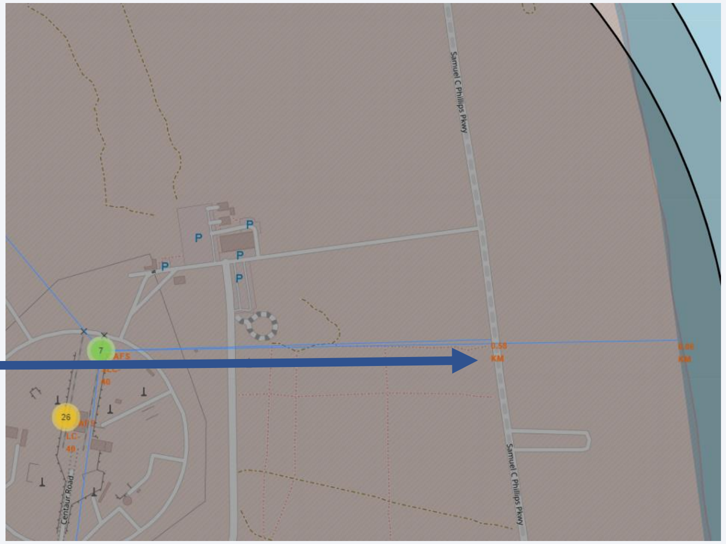


Zoom in

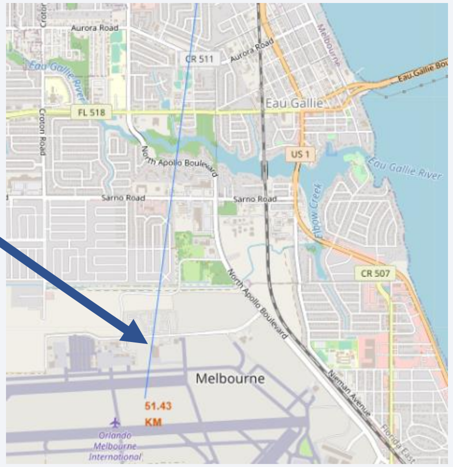
Launch Site Distance to POIs

Distance to Railroad

Distance to Highway and Coastline



Distance to City



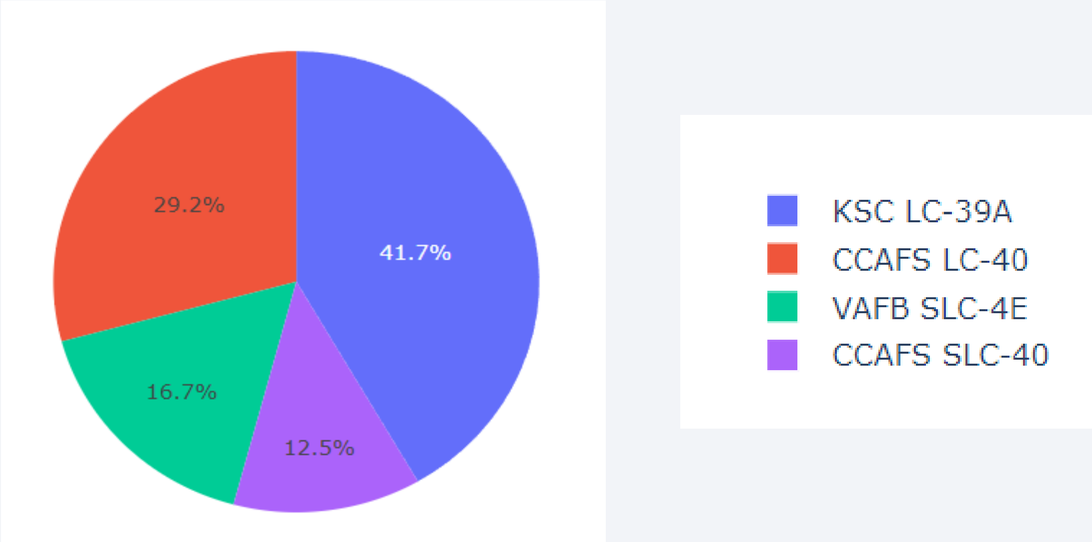
CCAFS SLC40 Launch Site findings:
Are launch sites in close proximity to railways? Yes, for logistic reasons.
Are launch sites in close proximity to highways? Yes, for logistic reasons.
Are launch sites in close proximity to coastline? Yes, for safety reasons.
Do launch sites keep certain distance away from cities? Yes, for safety reasons.

Section 4

Build a Dashboard with Plotly Dash

Dashboard: pie chart for all sites

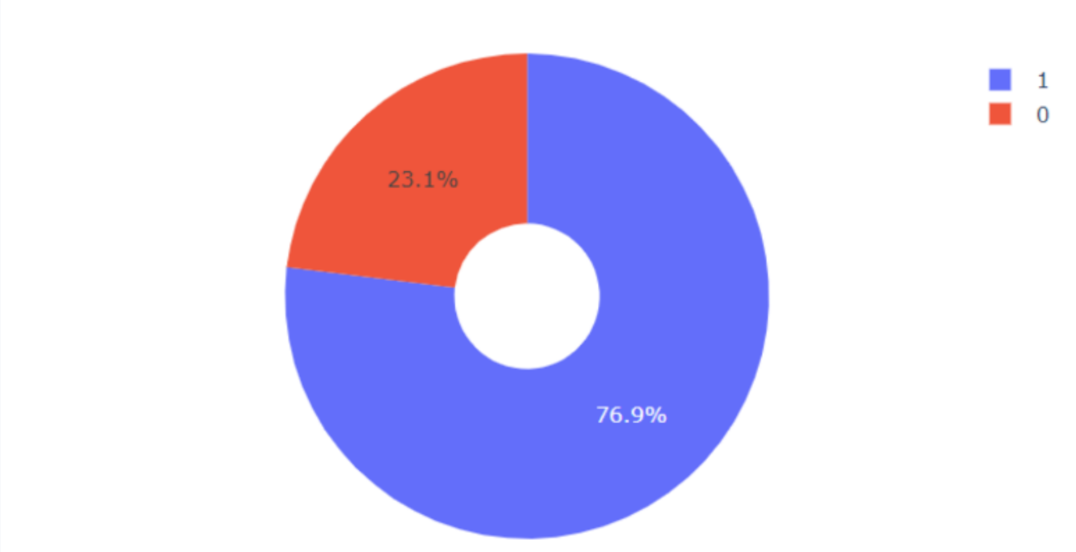
Pie Chart for Total Success Launches for All Sites



KSC LC-39A with 41.7% has the most successful launch rate as seen on above chart

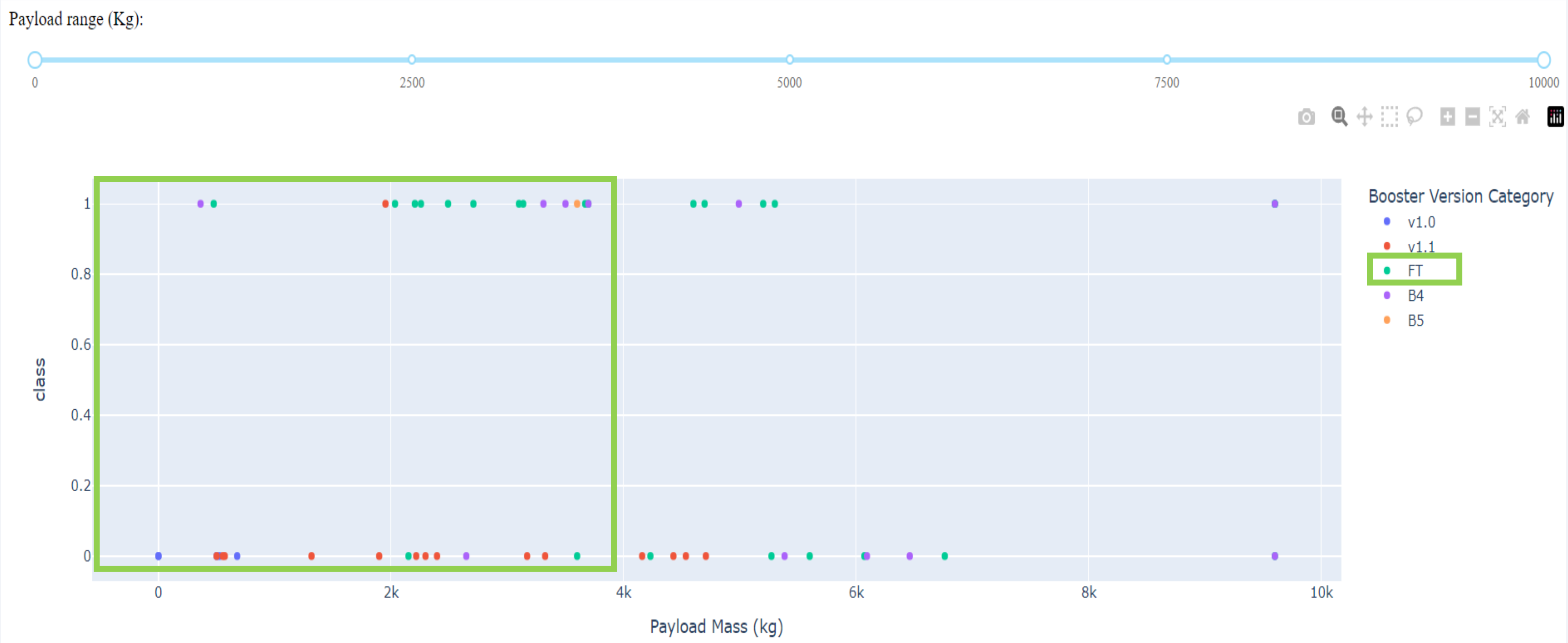
Dashboard: pie chart for KSC LC-39A site

Pie Chart for KSC LC-39A site: Success and Failure Launches rates



KSC LC-39A shows a Success Launch rate of 76.9%, while Failure Launch rate is 23.1%.

Dashboard: scatter plot for Payload vs. Launch Outcome

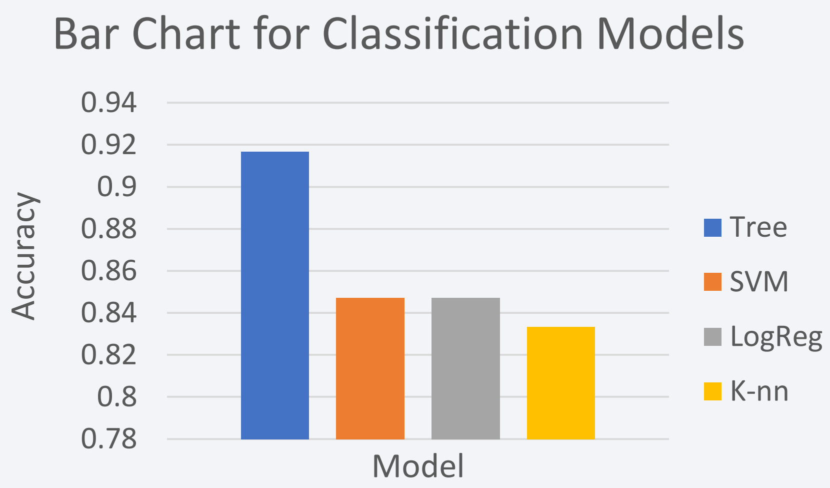


Scatter plot shows that payload range 0-4k kg has higher success rate than payload range 4k-10k. Also, the graph shows that FT booster version has the most successful launch rate.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



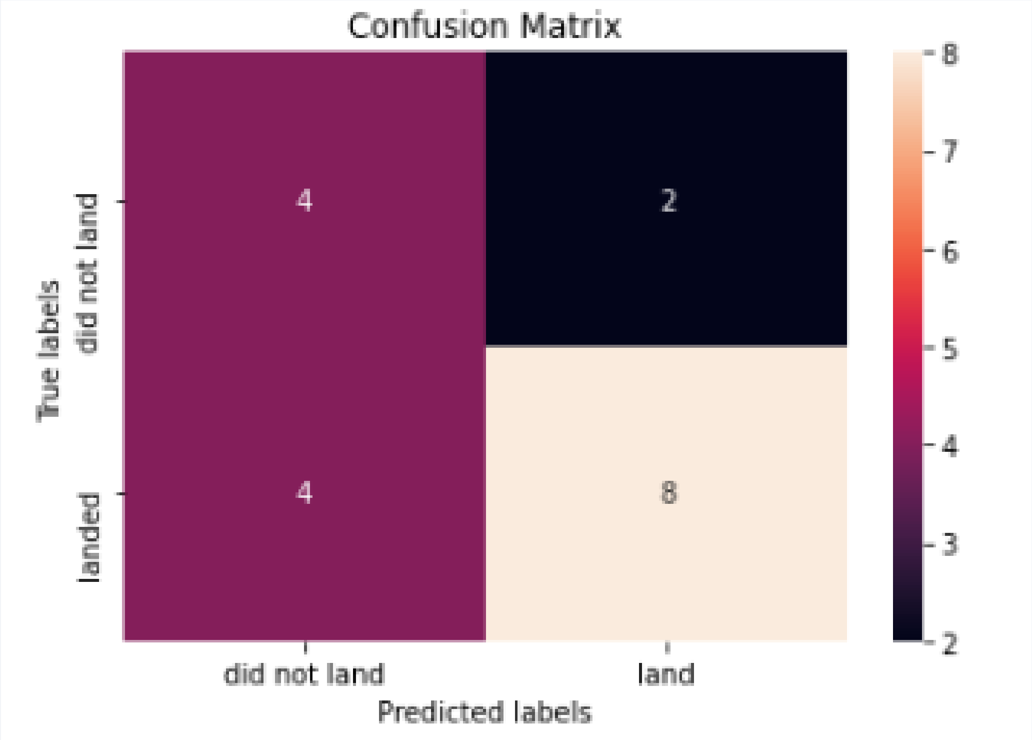
Finding best model with Python:

```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.9166666666666666
```

Confusion Matrix

Confusion Matrix for Decision Tree Model (best model calculated)



The above chart shows that Decision Tree model can distinguish among different classes. Major problem identified is for False Negatives, when successful landings are marked as failure landings by the classifier.

Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site
- Low weighted payloads perform better than heavier payloads.
- Success average rate for SpaceX launches has an increasing trend since 2013 to 2020.
- KSC LC 39A has the most successful launches rate from all the sites.
- Orbits GEO,HEO,SSO and ES L1 have the best Success Rate.
- Tree Classifier Algorithm is the best classifier model for this dataset.

Thank you!