IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Joseph Grippi
6-8-2022

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
    - Interactive analytics in screenshots
    - Predictive Analytics result

Introduction

- Project background and context

  At ***Space Y*** we would like to directly compete with Space X. Space X boasts that their Falcon 9 rocket launches at a cost of $62 Million. This is due to Space X being able to reuse the first stage of their launch. In order to compete with Space X, we will need to replicate this ability.

- Problems you want to find answers to
  - Success rate of launches
  - Success rate of landings
  - Models of rockets that were launched
  - Payload of rocket launches
  - Location of rocket launches

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected using SpaceX API and Web Scraping a Wikipedia Source.
- Perform data wrangling
    - One hot encoding was applied to categorical features.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - We used several Machine Learning Models. They are SVM, KNN, Decision Tree, and Logistic Regression to find the one that predicts success with the best accuracy.

Data Collection

- Data sets were collected from the following sources

    SpaceX API

    Web Scraping a Wikipedia Source

- We then performed Data Wrangling techniques to ensure data was properly remediated, cleaned and formatted.

- During our research the following Data was taken into consideration –

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

Data Collection – SpaceX API

1. Get Data from SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

2. Convert to a JSON File and normalize

```
# Use json_normalize method to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

3. Filter Data

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a sing
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

4. Create a new dictionary

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Export to CSV

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

https://github.com/joegrippi/IBM_Data_Science_Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- 1. Get Data from Wikipedia

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

```
# use requests.get() method with the provided static_url
response = requests.get(static_url)
# assign the response to a object
print(response.content)
```

- 2. Extract Columns

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
for x in first_launch_table.find_all('th'):
    name = extract_column_from_header(x)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

3. Create a Data Frame

```
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']=[]
```

2. Extract Columns

```
df1.to_csv('spacex_web_scraped.csv', index=False)
```

https://github.com/joegrippi/IBM_Data_Science_Capstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- Data was Wrangled in the following steps
    1. Load Data that was saved during our data collection step.

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")
df.head(10)
```

    2. Examine our data

```
# Apply value_counts() on column LaunchSite
df.LaunchSite.value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

- 3. Add columns where needed

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

```
df['Class']=landing_class
df[['Class']].head(8)
```

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

- 4. Export to CSV

```
df.to_csv("dataset_part_2.csv", index=False)
```

https://github.com/joegrippi/IBM_Data_Science_Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

The Following was visualized using our formatted data:

- **Scatter charts showing how correlation between variables:**

  Flight Number vs. Launch Site
  Payload vs. Launch Site
  Flight Number vs. Orbit Type
  Payload VS. Orbit Type

- **Bar Chart to show the visual relationship between a category and a discrete value:**

  Success Rate of each Orbit Type

- **Line Charts to show in which direction a trend is pointing in:**

  Year Vs. Success Rate

https://github.com/joegrippi/IBM_Data_Science_Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- **Dataset was loaded into a table into an IBM Cloud Db2 database, and we executed the following SQL Queries in Python:**
- 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_version's which have carried the maximum payload mass
- List the failed landing_outcome's in drone ship, their booster versions, and launch site names for in year 2015
- 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

https://github.com/joegrippi/IBM_Data_Science_Capstone/blob/main/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- The following was marked in an interactive Folium Map
  - Markers that show all launch sites on a map
  - Markers that show the success/failed launches for each site on the map
  - Lines that show the distances between a launch site to its proximities
- The following questions were asked
  - Success rates at each location was examined
  - Distance from coastline was examined
  - Distance from railways was examined
  - Distance from highways was examined
  - Distance from cities was examined

https://github.com/joegrippi/IBM_Data_Science_Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- The Plotly dashboard application displays an interactive pie chart and a scatter point chart

Pie chart

Shows total success launches by all sites combined or each singular site

Scatter chart

Display the relationship between Outcomes and Payload mass (Kg) by different boosters

2 inputs are available:

All sites/individual site & Payload mass on a slider between 0 and 10000 kg

This chart helps the end user determine how success depends on the launch point, payload mass, and booster version categories

https://github.com/joegrippi/IBM_Data_Science_Capstone/blob/main/IBM_Data_Science_Dash_Plotly.ipynb

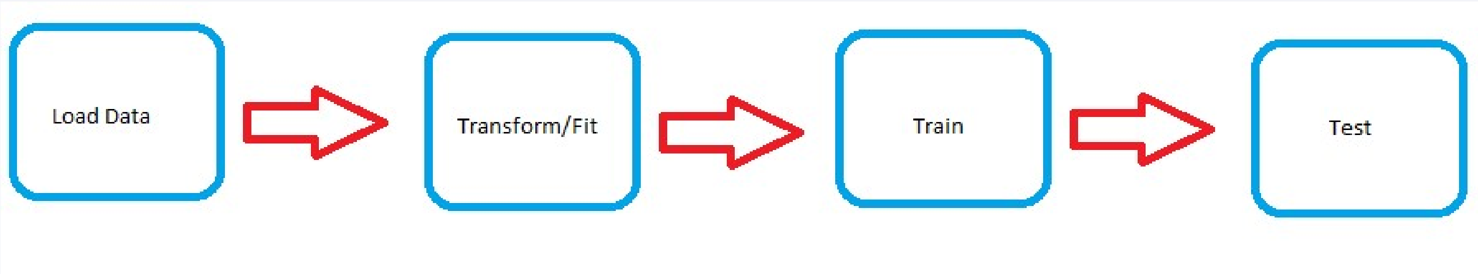
Predictive Analysis (Classification)

- 1. Load our data

```
data = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv")

X = pd.read_csv('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_3.csv')
```

- 2. Then transform/fit, train and test our data against different methods to see which is the most accurate.



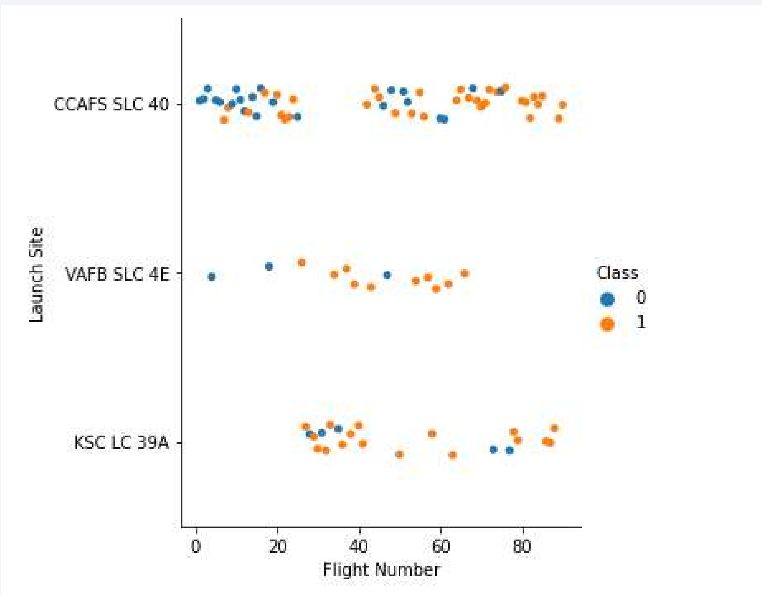
- 3. For our study we applied the following methods, SVM, KNN, Decision Tree, and Logistic Regression.
- 4. Compare accuracy of each Model and choose the best one for future use.

https://github.com/joegrippi/IBM_Data_Science_Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Section 2

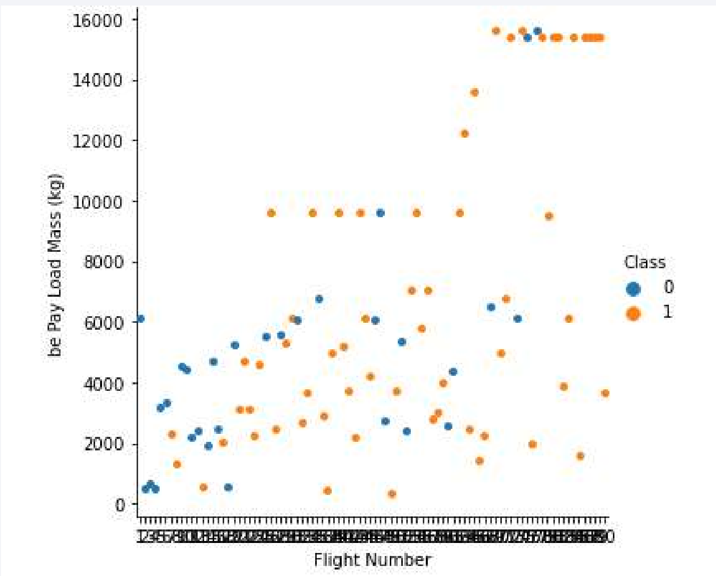
Insights drawn from EDA

Flight Number vs. Launch Site



The blue dots *(Class 0)* represent unsuccessful launches, the orange dots *(Class 1)* represent successful launches at Flight Number Vs. Launch Site.
The chart shows that the success rate increased as the flight number increased.

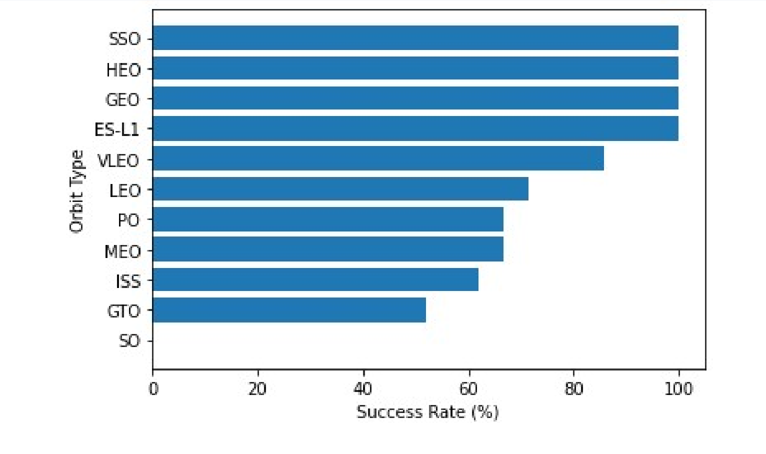
Payload vs. Launch Site



The blue dots *(Class 0)* represent unsuccessful launches, the orange dots *(Class 1)* represent successful launches at Payload Vs Launch Site.

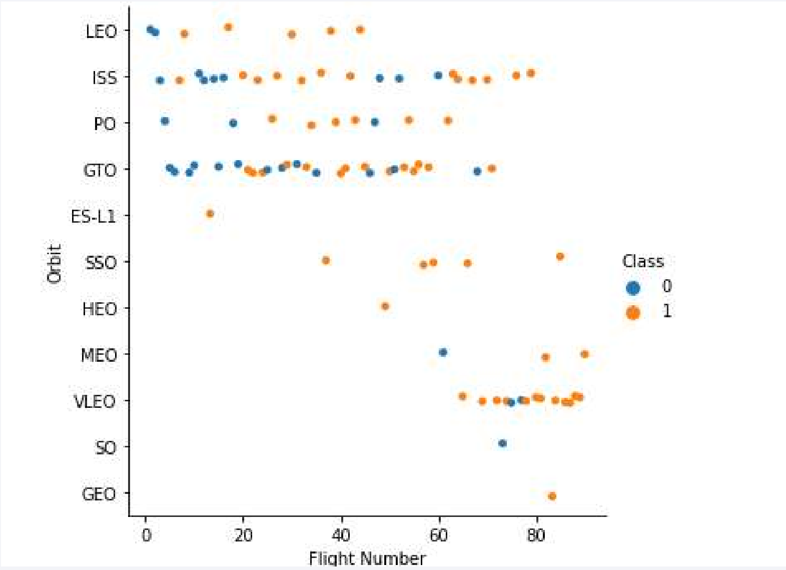
The chart shows that earlier flights with low payload were highly unsuccessful.

Success Rate vs. Orbit Type



The horizontal bar chart shows the success rate for each Orbit. We can see that the four most successful Orbits are SSO, HEO, GEO, and ES-L1.

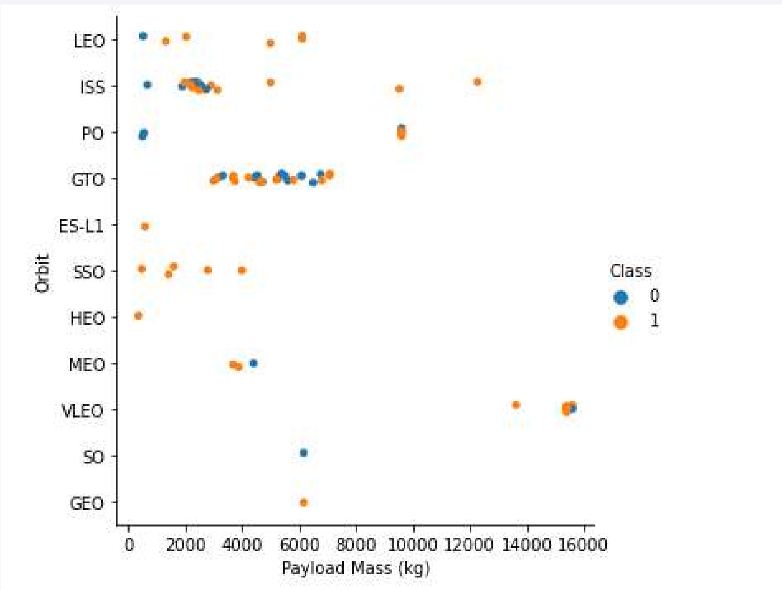
Flight Number vs. Orbit Type



The blue dots *(Class 0)* represent unsuccessful launches, the orange dots *(Class 1)* represent successful launches at Flight Number Vs. Orbit Type.

We see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

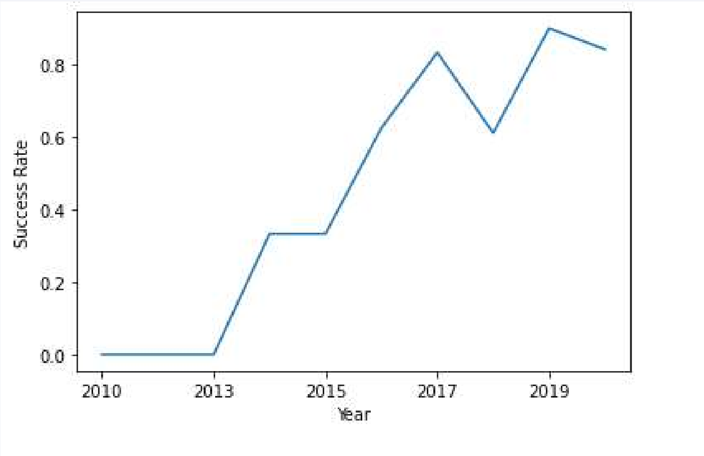
Payload vs. Orbit Type



The blue dots *(Class 0)* represent unsuccessful launches, the orange dots *(Class 1)* represent successful launches at Payload Vs. Orbit Type.

With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

Launch Success Yearly Trend



This chart shows success rate over time. We can see that time and success rate is positively correlated. This is most likely due to research and development.

All Launch Site Names

```
%sql select DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

```
 * ibm_db_sa://rqy96889:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
Done.
```

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

- This query is performed to list all the distinct launch sites in the SPACEXTBL DB2 Database Table.

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTBL where LAUNCH_SITE LIKE '%CCA%' Limit 5;
```

```
 * ibm_db_sa://rqy96889:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/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

- This query shows 5 launches that begin with CCA.

Total Payload Mass

```
%sql select SUM(PAYLOAD_MASS__KG_) AS NASA_CRS_Payload_Mass from SPACEXTBL where Customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://rqy96889:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
Done.
```

nasa_crs_payload_mass
45596

- This query shows the total Payload that has been dissipated over all flights.

Average Payload Mass by F9 v1.1

```
%sql select AVG(PAYLOAD_MASS__KG_) AS F9_V1_1_Payload_Mass from SPACEXTBL where Booster_Version = 'F9 v1.1';
```

```
 * ibm_db_sa://rqy96889:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
Done.
```

f9_v1_1_payload_mass
2928

- This SQL query shows the total Payload Mass by the F9 v1.1.

First Successful Ground Landing Date

```
%sql select MIN(DATE) AS_First_Success_Ground_Pad from SPACEXTBL where landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://rqy96889:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
Done.
```

as_first_success_ground_pad
2015-12-22

- This query shows us the first successful landing date. We can see the first successful landing date was performed on December 22nd, 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select BOOSTER_VERSION, landing__outcome, PAYLOAD_MASS__KG_ from SPACEXTBL where landing__outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_
```

```
 * ibm_db_sa://rqy96889:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
Done.
```

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

- This SQL Query shows the Booster Version of all successful Drone ships with a payload between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

```
%sql select Mission_Outcome, count(*) As Total_Number from SPACEXTBL Group by Mission_Outcome;
```

```
 * ibm_db_sa://rqy96889:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
Done.
```

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

- This SQL query shows the total number of successful launches to failures. As you can see SpaceX has a high success rate.

Boosters Carried Maximum Payload

```
%sql select BOOSTER_VERSION, PAYLOAD_MASS__KG_ from SPACEXTBL \
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

```
 * ibm_db_sa://rqy96889:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
Done.
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

- This SQL query shows the different Booster Version’s that have carried maximum Payload.

2015 Launch Records

```
%sql select Landing__Outcome, Booster_Version, Launch_Site from SPACEXTBL where Landing__Outcome = 'Failure (drone ship)' and Date like '%2015%';
```

```
 * ibm_db_sa://rqy96889:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/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 SQL query shows the failed Drone ship launches and their Booster version for 2015.

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

```
%sql select Landing__Outcome, count(Landing__Outcome) as Total_Number from SPACEXTBL where Date Between '2010-06-04' and '2017-03-20' Group By Landing
```

```
 * ibm_db_sa://rqy96889:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/BLUDB
Done.
```

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

- This SQL query shows all the types of attempts between June 2011 and March 2017.
- We can see that there were three successful ground pad landings.

Section 3

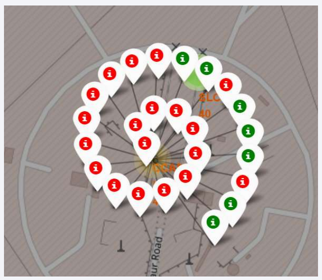
Launch Sites Proximities Analysis

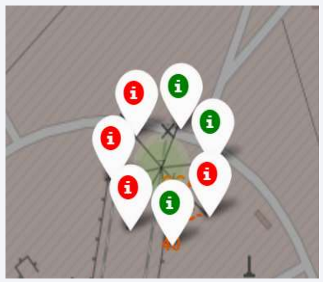
All Launch Sites

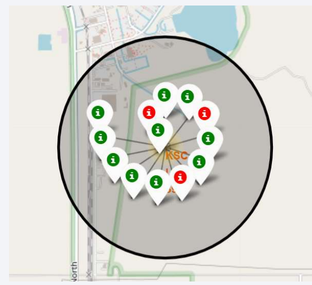
- This map shows all of the SpaceX Launch Sites. We can see that they are all located on the coast.

Launch Successful and Unsucessful Launches

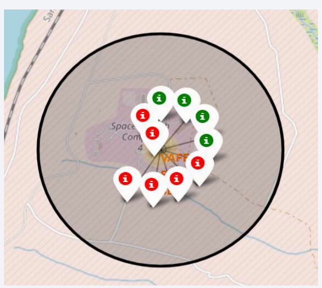
Florida Launch Sites







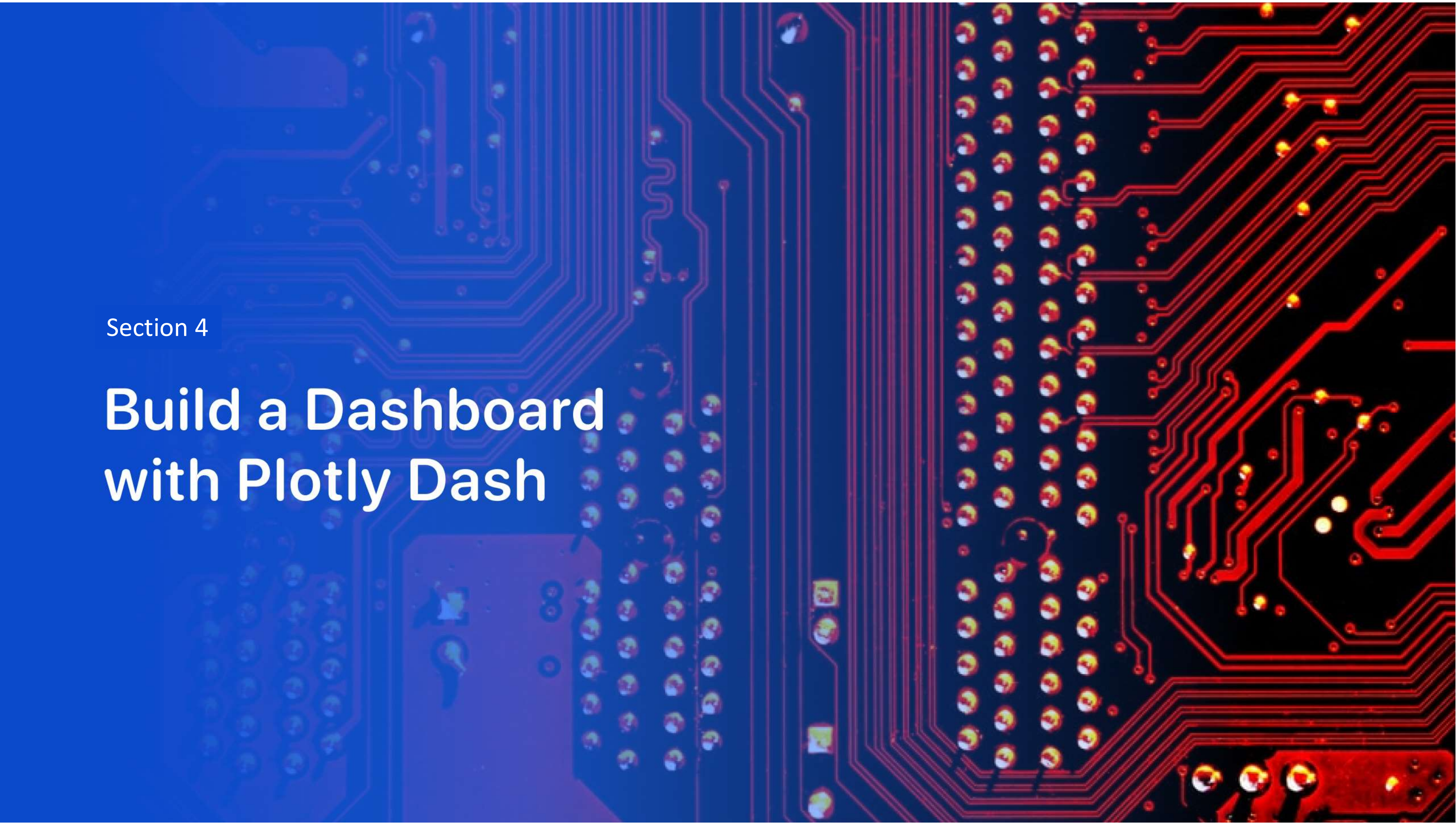
California Launch Site



The green marker represents a successful launch while red marker represents an unsuccessful launch.

Launch Site Surroundings.

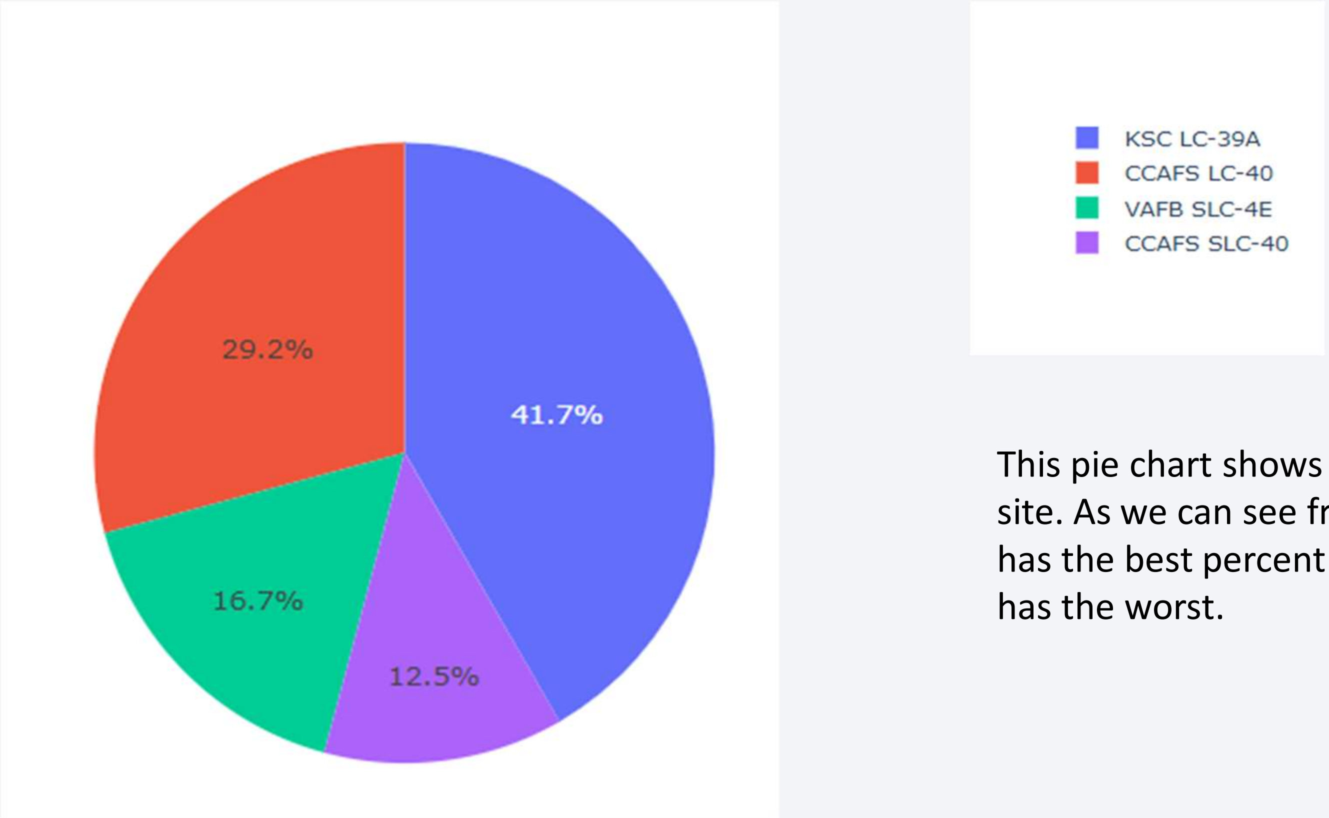
- These maps show the proximity of railway, highway, and coastline.
- We find that the Launch Site is close to the coast and transportation while being for from cities.



Section 4

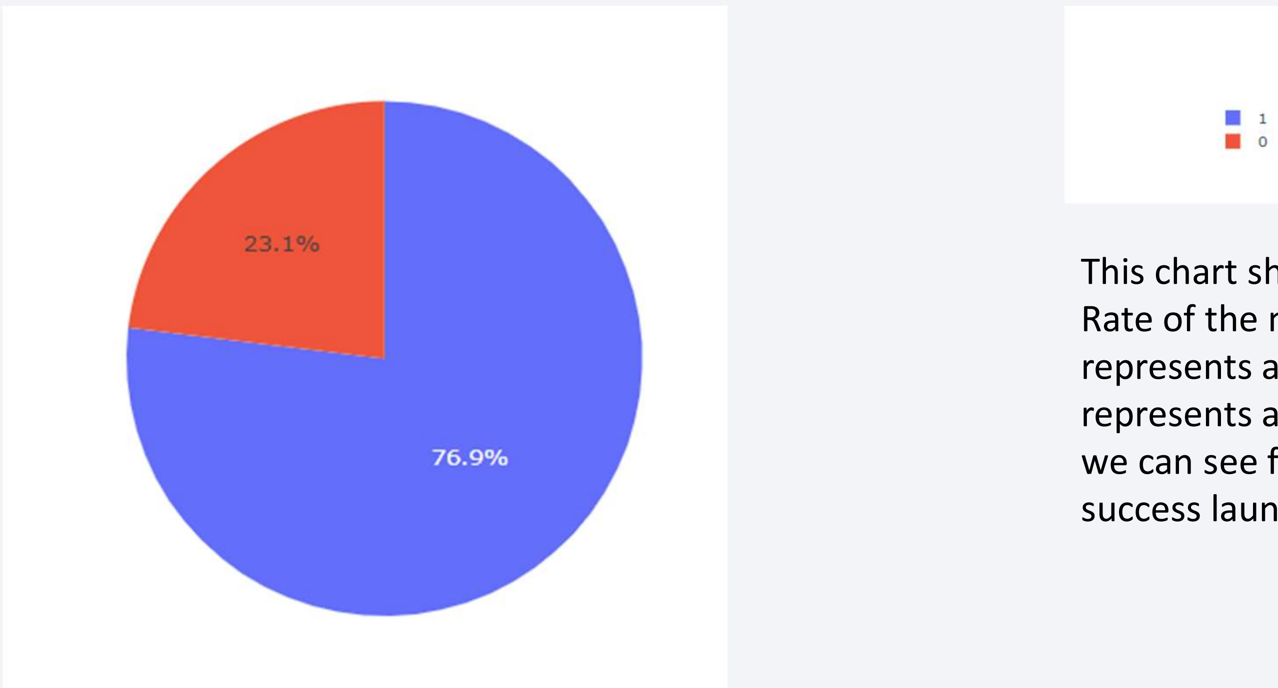
Build a Dashboard with Plotly Dash

Total Percentage Successful Launches Per-Site



This pie chart shows the successful launches by site. As we can see from this chart KSC LC-39A has the best percent rate while VAFB SLC-4E has the worst.

Launch Success Rate of Most Successful Site

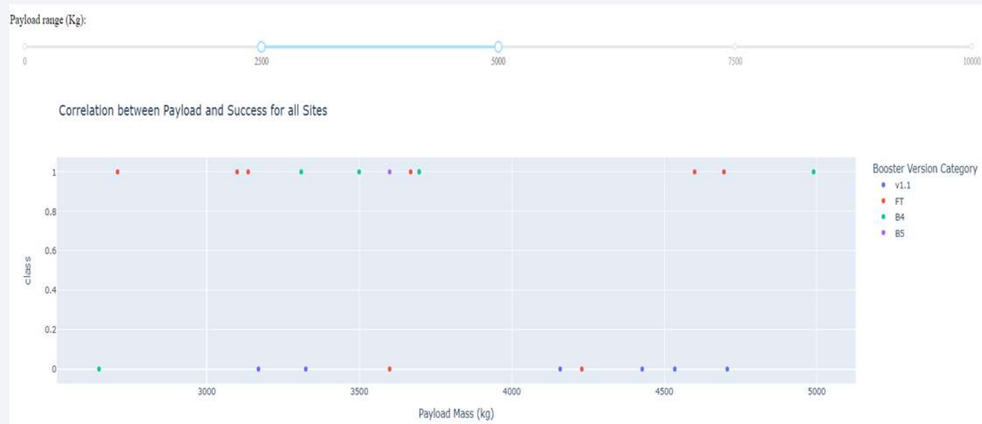


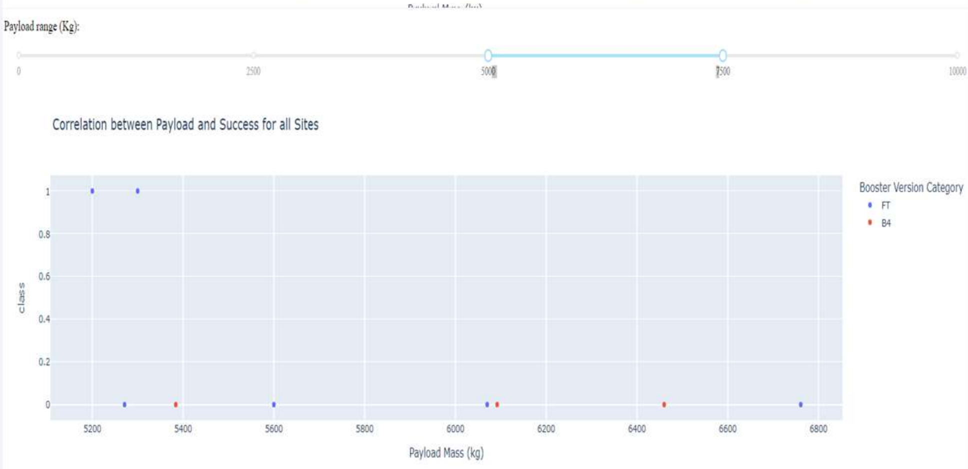
This chart shows the Launch Success Rate of the most successful site. Red represents a fail while Blue represents a successful launch. As we can see from the chart shows a success launch rate of 76.9%

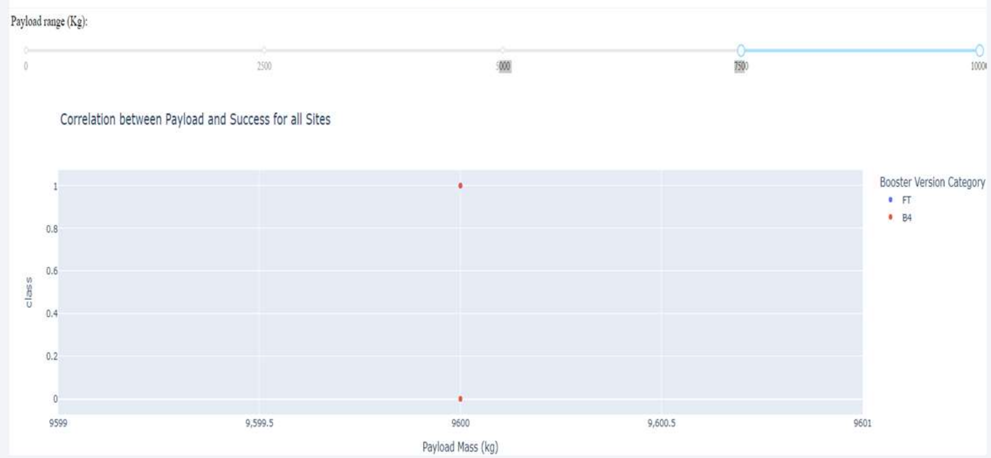
Correlation Between Payload and Success Rate For All Sites

This shows correlation for all Payloads across all sites.

Payload Success Rate For each Payload Range



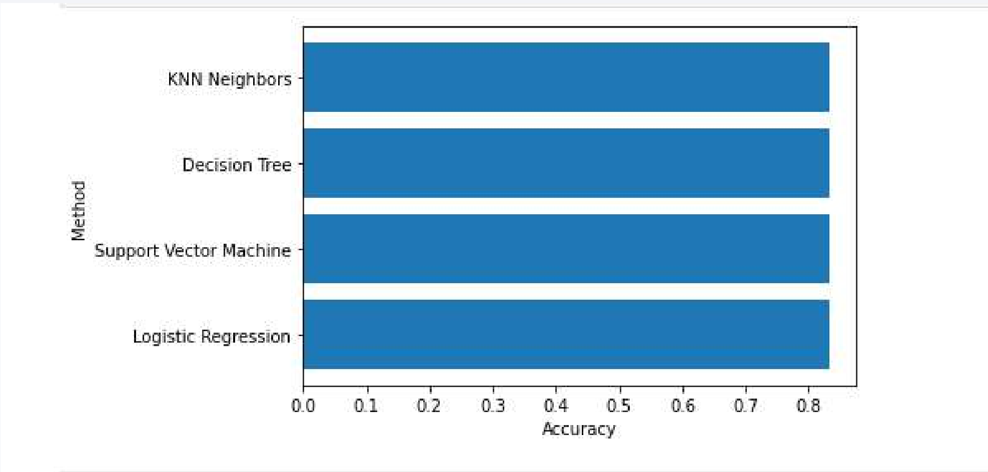




Section 5

Predictive Analysis (Classification)

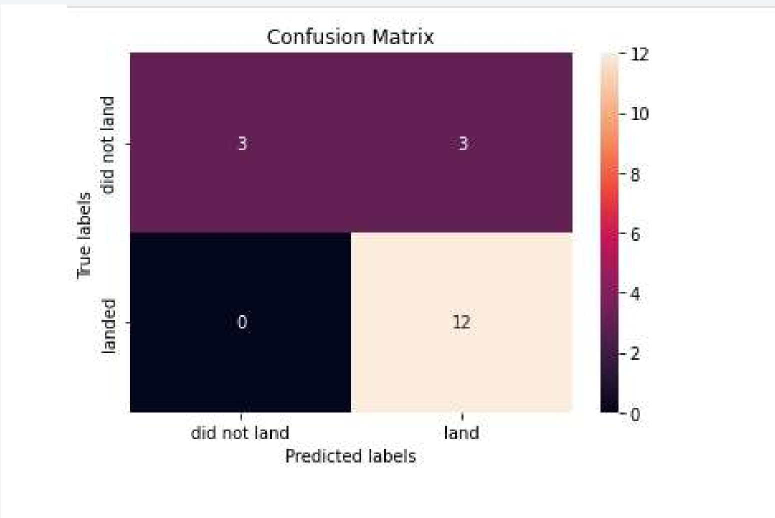
Classification Accuracy



	Accuracy
Logistic Regression	0.833333
Support Vector Machine	0.833333
Decision Tree	0.833333
KNN Neighbors	0.833333

- We can see that all models performed exactly the same. This is most likely due to a small sample space. (Training and Test Set)

Confusion Matrix



- All Confusion Matrixes performed the same due to the same accuracy rate.

Conclusions

- Launch Success Rate has increased over time.
- Orbit Types SSO, HEO, GEO, and ES-L 1 have the highest success rate.
- Launches are convenient to travel to and close to the coast line.
- Due to a small sample set we currently do not know what could be the best predictive model to use. All models had a success rate of 83.3%.

Appendix

- https://github.com/joegrippi/IBM_Data_Science_Capstone


Thank you!