IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Caleb Pillay
16/01/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection with web scrapping
  - Data wrangling with pandas library
  - Data exploration with SQL IBM DB2 database and pandas library
  - Data visualization Matplotlib library
  - Data plotting with Folium library
  - Data mining with Plotly and Dash library
  - Machine learning with Scikit-learn library
- Summary of all results
  - SQL table outputs
  - Graphs and an interactive dashboard
  - Predicative machine algorithms

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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.

- Problems you want to find answers
  - Determine if the first stage of Falcon 9 will land.
  - Determine if there are existing variables that affect landing success rate.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Web scraped data from the Wikipedia page: List of Falcon 9 and Falcon Heavy launches using Requests and Beautiful Soup library
- Perform data wrangling
  - Data was cleaned in a Pandas data frame using python
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Models were built with Scikit-learn using the iterative process of grid search

Data Collection

- The data was web scrapped and entered into a data frame
  - Using the requests library in python a request was made to the Wikipedia server, data was recorded and stored in static-url format, using the `‘static_url’` decorator.
  - A beautifulsoup object was created using the `‘html.parser’` decorator.
  - An empty dictionary assigned with keys from the extracted column names from the beautifulsoup object.
  - A predefined function was created to extract and assign the table contents from the beautifulsoup object into a dictionary object.
  - A predefined function was created to extract the dictionary values and assign them into a data frame.

Data Collection - Scraping

- Web scraped data from the Wikipedia page: [List of Falcon 9 and Falcon Heavy launches]() using Requests and Beautiful Soup library
- [Coursera/blob/main/Webscraping_SpaceX.ipynb]()

```
[ ] import sys

    import requests
    from bs4 import BeautifulSoup
    import re
    import unicodedata
    import pandas as pd
```

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

Create a BeautifulSoup object from the HTML response

```
[ ] # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
    soup = BeautifulSoup(response, 'html.parser')
```

```
[ ] # 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')
    print(len(html_tables))

    24
```

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

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

Data Wrangling

- Exploratory Data Analysis was applied to find patterns in the data
- Pandas data frame was used for ease of functionality

```
[ ] # 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
```

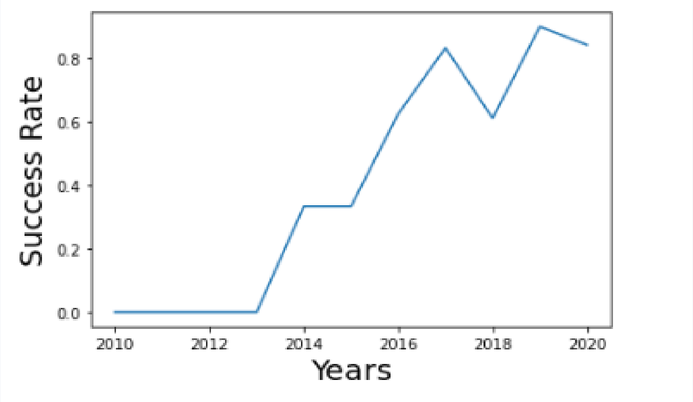
```
[ ] # landing_outcomes = values on Outcome column
    landing_outcomes.index[[1,3,5,6,7]]

    Index(['None None', 'False ASDS', 'False Ocean', 'None ASDS', 'False RTLS'], dtype='object')
```

- Launch sites were identified
- Landing outcomes were relabeled to success – ‘1’ or not – ‘0’ for model use
- Coursera/blob/main/Data_wrangling_Spacex.ipynb

EDA with Data Visualization

- Matplotlib and Seaborn plotting libraries were used
- Relationships between payload-mass, flight number, launch site, orbit were plotted to visually assess the launch success rates.



Interestingly : Launch success rate saw a sharp increase from 2013 onward

[Coursera/blob/main/EDA-DATAVIZ-SpaceX.ipynb]

EDA with SQL

- Coursera/blob/main/EDA-SQL-SpaceX.ipynb
- The data frame was inserted in the IBM's cloud faculty DB2
- Queries were performed in a Jupyter notebook using a db2 connection string
- SQL-Alchemy library allowed for sql queries to be performed
  - Identified launch sites
  - Total payload mass launched from individual sites and booster versions
  - Date of first successful landing on a ground pad.
  - Boosters which have successful landing on drone ship and payload mass between 4000 – 6000.
  - Total outcomes of launches.
  - Landing outcomes between various dates.

Build an Interactive Map with Folium

- Coursera/blob/main/Folium-SpaceX.ipynb
- Folium library was used to plot launch sites and individual launches onto a map
- A visual representation of successful lunches at each site was created
- Visually investing if there are any proximity variables that could affect a launches success rate
- The questions posed were:
  - Do proximities to cities affect a launched success?
  - Do proximities to railways, highways and coastlines affect a launched success?

Build a Dashboard with Plotly Dash

- Effectively data mining with Plotly graphs assembled into dashboard using the Dash library
- Visually investigated the proportion of successful launches per site using a pie plot
- The relationship between successful launches and payload-masses for per site using a scatter plot
- [Coursera/blob/main/Polty%20dash%20-SpaceX.ipynb]

Predictive Analysis (Classification)

- Scikit-learn library was used to predict at launch outcome
- Models: logistic regression, support vector machine, decision tree classifier and k nearest neighbors
- GridSearchCV was used to determine the optimal hyperparameters through an iterate process
- The best model was chosen by the highest number of correctly predict outcomes from the test data
- Coursera/blob/main/Machine_Learning_Prediction_SpaceX_.ipynb

Results

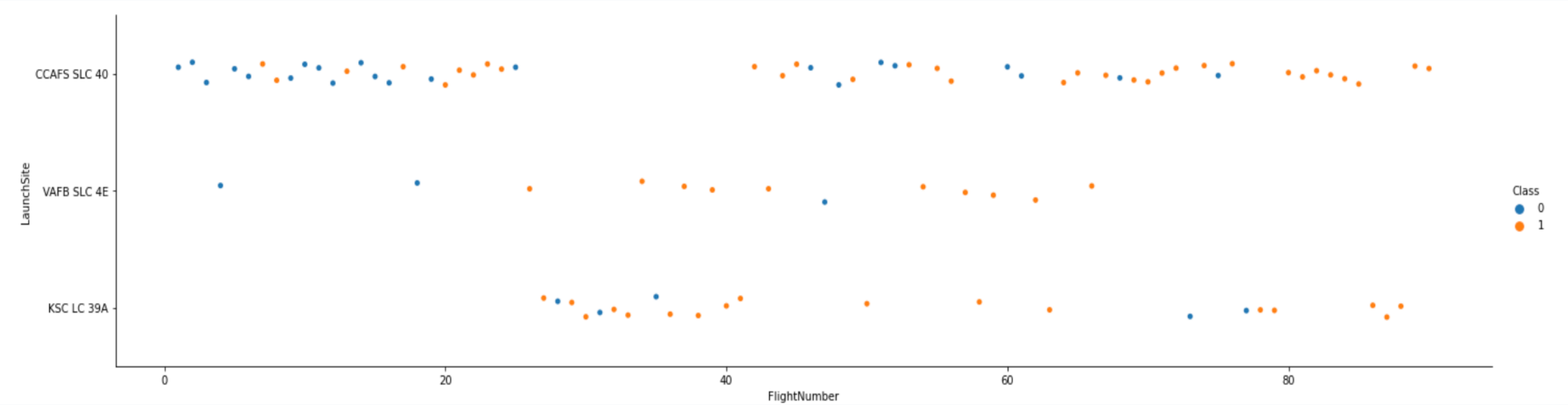
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

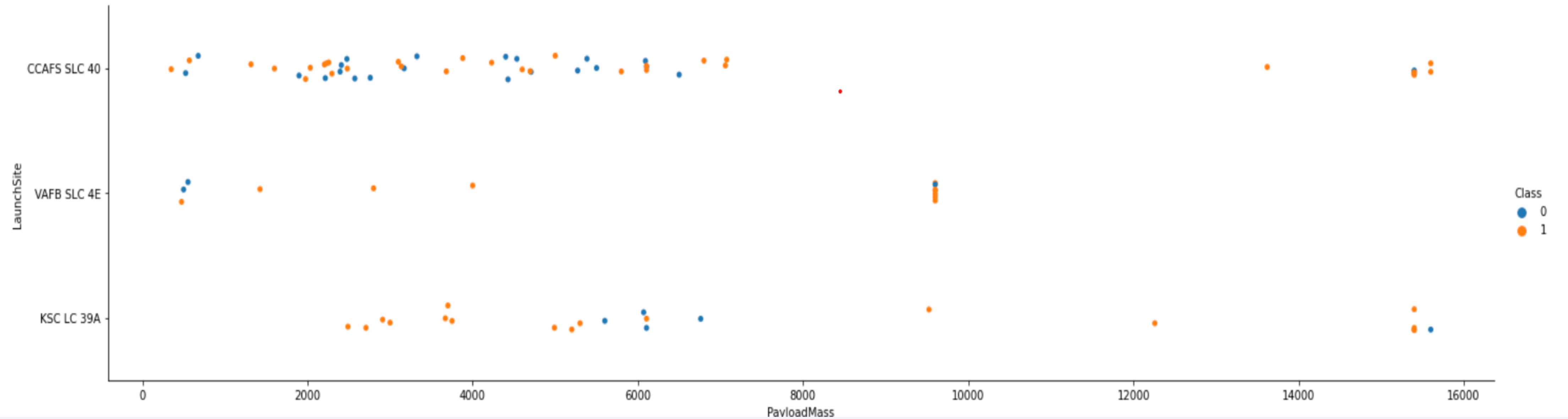
Flight Number vs. Launch Site

- CCAFS SLC has the highest number of launches and successful landings



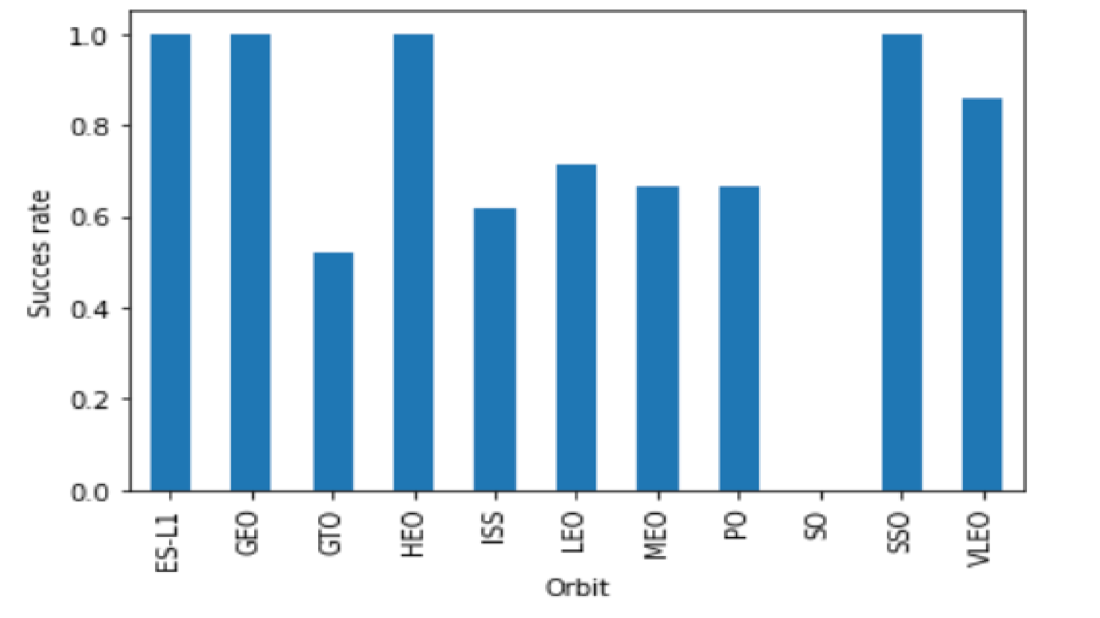
Payload vs. Launch Site

- Payloads below 2000kgs has a fewer launches across two sites
- Payloads above 8000kgs seems to be more successful at landing across all sites



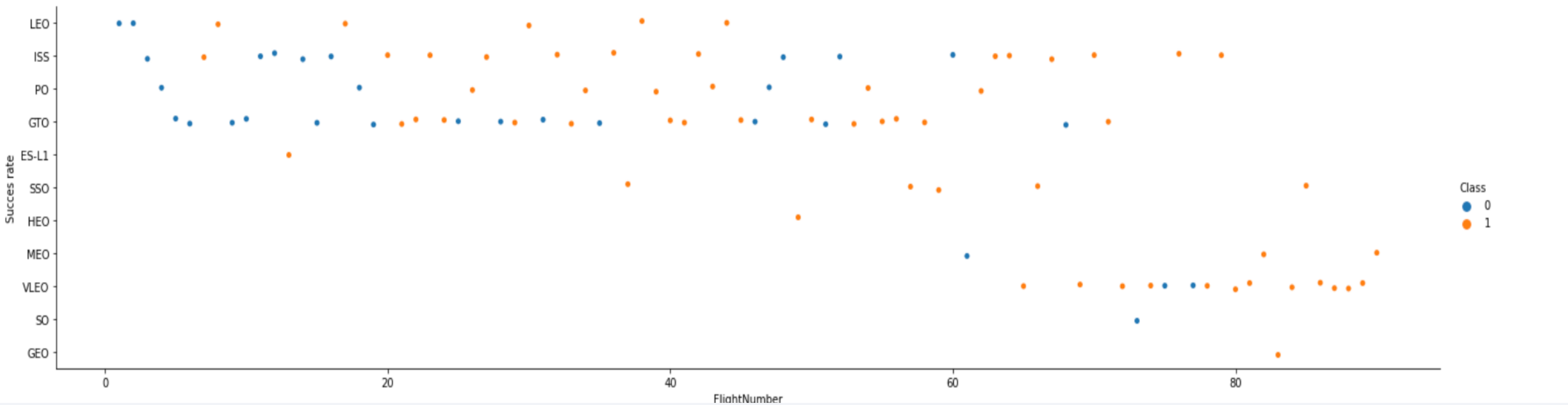
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO, VLEO had the most success rate



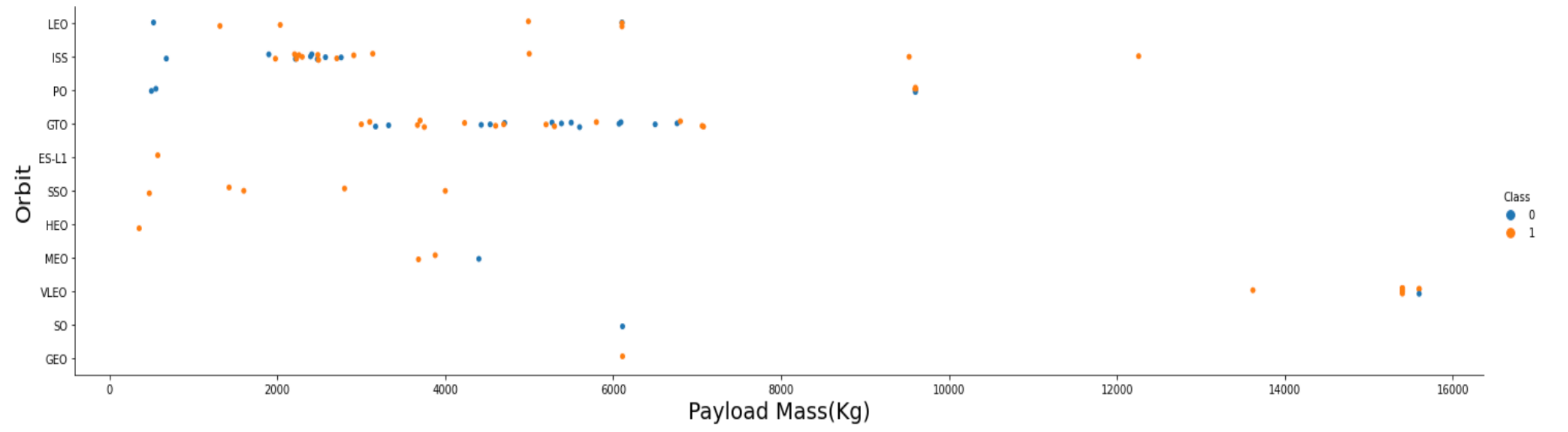
Flight Number vs. Orbit Type

- LEO, ISS, PO show an increase in success rate after 20 flights
- SSO has a 100% success rate



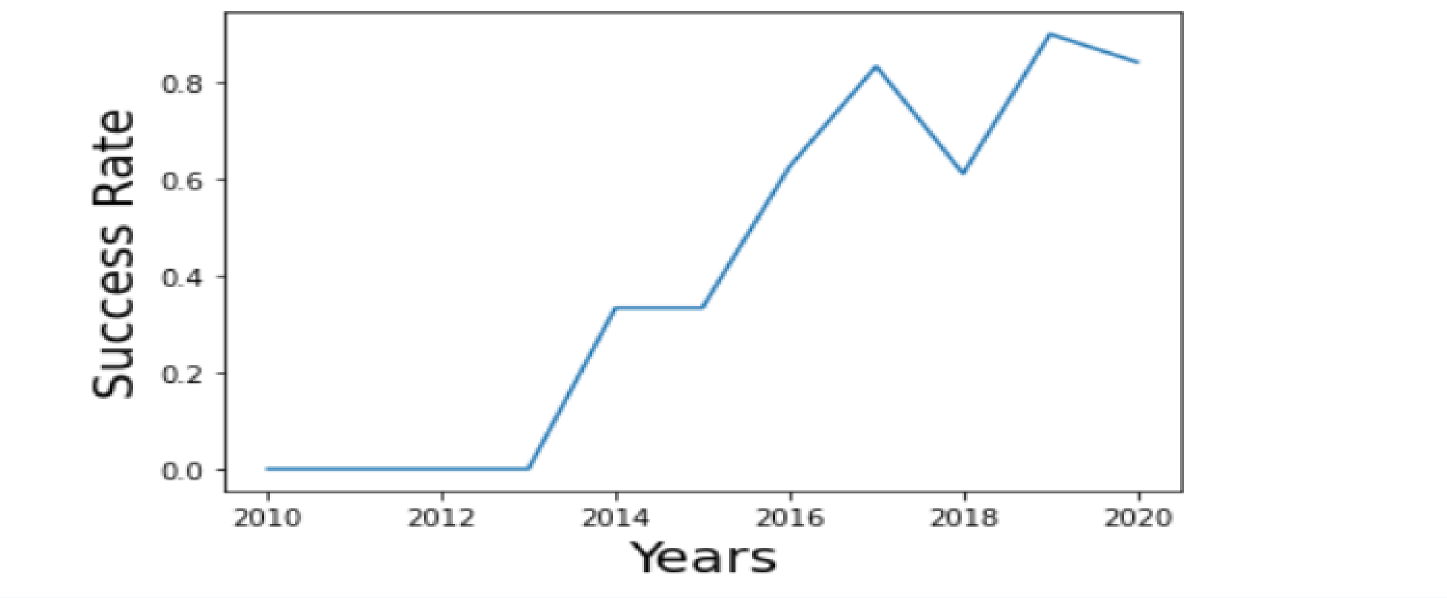
Payload vs. Orbit Type

- ISS, ISS & PO have low success rate with payloads below 3000kgs



Launch Success Yearly Trend

- 2013 onward showed a steady increase in success rate of landing



All Launch Site Names

- Distinct allowed for the unique categorical variables from launch site to be displayed

```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL;
```

```
 * ibm_db_sa://cmj12999:***@21fecfd8-47b7-4937-840d-d
Done.
```

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

Launch Site Names Begin with 'CCA'

- Like clause with wildcat expression “%”, queried all CCA launch sites

```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://cmj12999:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Sum() function calculated the total payload mass of WHERE clause customer "NASA (CRS)"

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';

 * ibm_db_sa://cmj12999:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/bludb
Done.
```

1
2928

Average Payload Mass by F9 v1.1

- AVG() function calculated the average payload mass of WHERE clause booster version is "F9 v1.1"

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
```

```
 * ibm_db_sa://cmj12999:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.datab
Done.
```

1
2928

First Successful Ground Landing Date

- Min() function was applied to date column to determine the earliest successful ground pad landing

```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME= 'Success (ground pad)';
```

```
 * ibm_db_sa://cmj12999:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.d
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Two filters were applied in the WHERE clause, specified payload mass range and landing outcome

```
%%sql SELECT BOOSTER_VERSION FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND Landing__outcome='Success (drone ship)';
```

```
 * ibm_db_sa://cmj12999:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases
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

- GROUP BY clause allowed for the aggregation of the “Mission Outcome” column

```
%%sql SELECT MISSION_OUTCOME, count(*) AS total,
sum(case when MISSION_OUTCOME = 'Failure (in flight)' then 1 else 0 end) AS Failure,
sum(case when MISSION_OUTCOME = 'Success' then 1 else 0 end) AS Success
FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://cmj12999:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg
Done.
```

mission_outcome	total	failure	success
Failure (in flight)	1	1	0
Success	99	0	99
Success (payload status unclear)	1	0	0

Boosters Carried Maximum Payload

- A sub-query was performed in the WHERE clause to filter the for the maximum payload

```
%sql SELECT DISTINCT(BOOSTER_VERSION) FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ =
(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * ibm_db_sa://cmj12999:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1o
Done.
```

booster_version
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

- Two filters were used in the WHERE clause a date range and a landing outcome, both had to be satisfied to display a result

```
%%sql
SELECT Landing__Outcome AS LANDING_OUTCOME, Booster_Version AS BOOSTER_VERSION, Launch_Site AS LAUNCH_SITE
from  SPACEXTBL  where Landing__Outcome like 'Failure %' and (DATE between '2014-12-31' and '2016-01-01');
```

```
 * ibm_db_sa://cmj12999:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cl
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

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

```
%%sql
SELECT Landing__Outcome, COUNT(Landing__Outcome) Count
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing__Outcome
ORDER BY COUNT(Landing__Outcome) desc;
```

```
 * ibm_db_sa://cmj12999:***@21fecfd8-47b7-4937-840d-d791d0218660
Done.
```

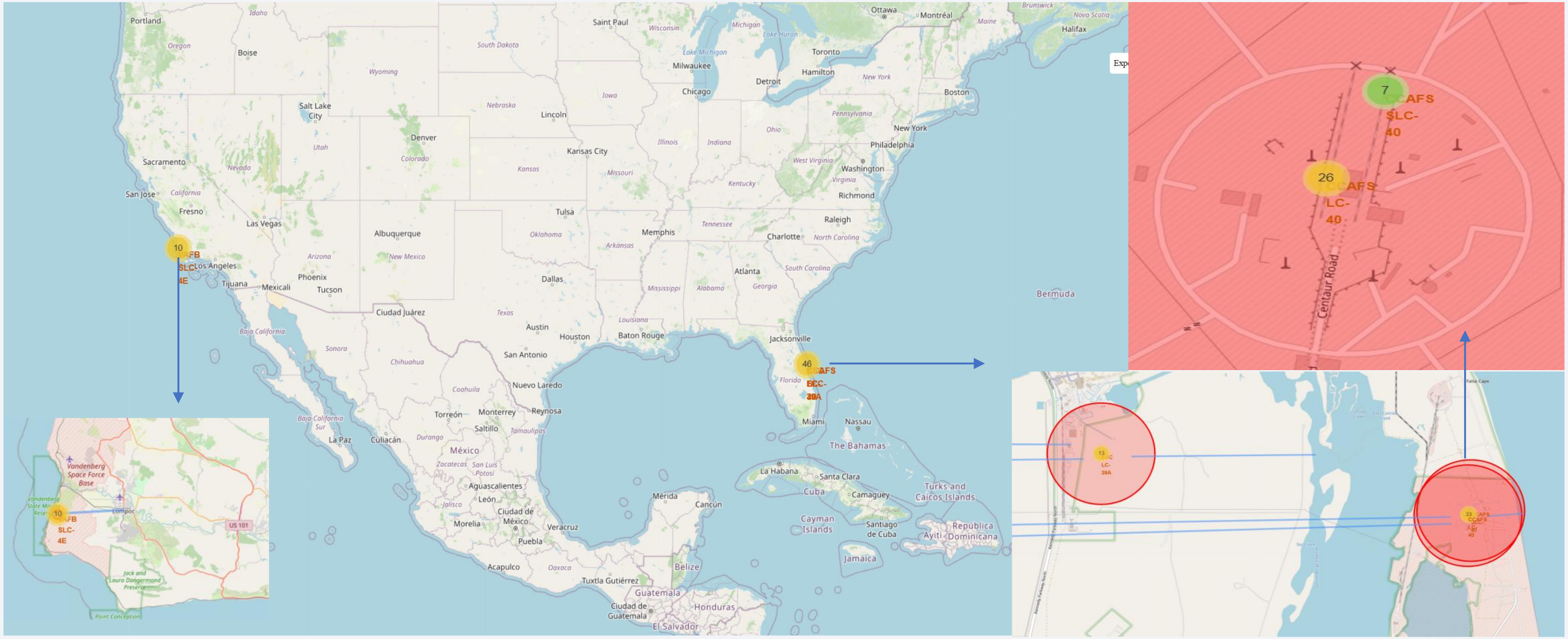
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- Count() clause numerated the categorical variables in the Landing_Outcome column
- Data was aggregated by the Group By clause
- Data was ordered by the Order By clause

Section 3

Launch Sites Proximities Analysis

Space X Launch Sites plotted with Folium



Folium Map Launch Sites

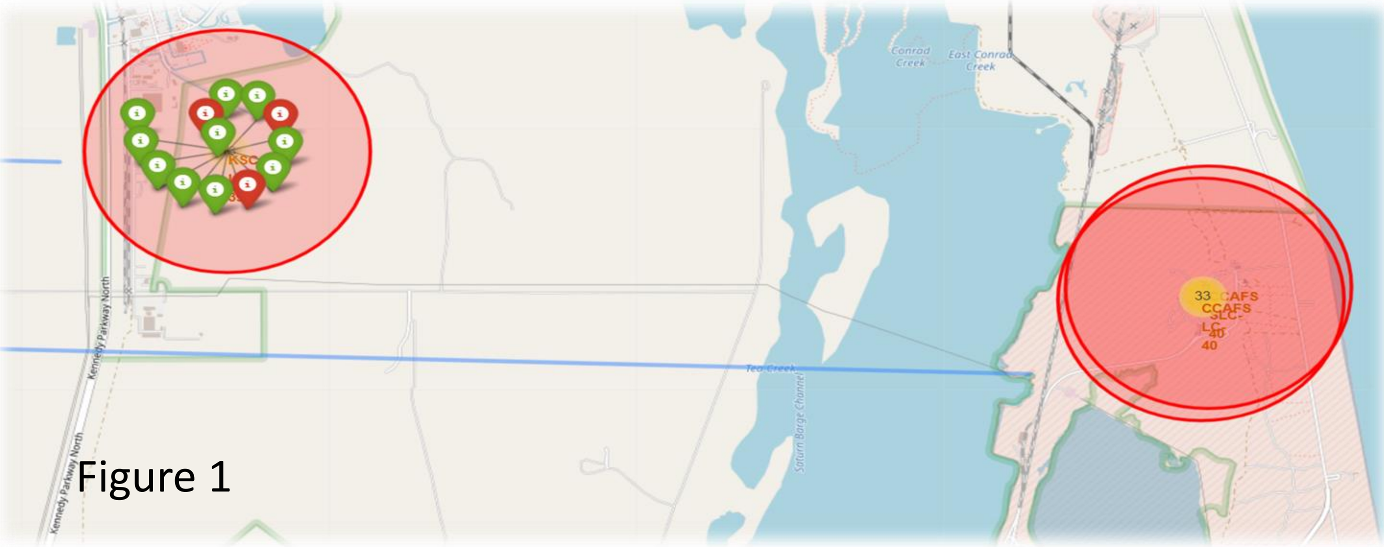


Figure 1

Figure 2

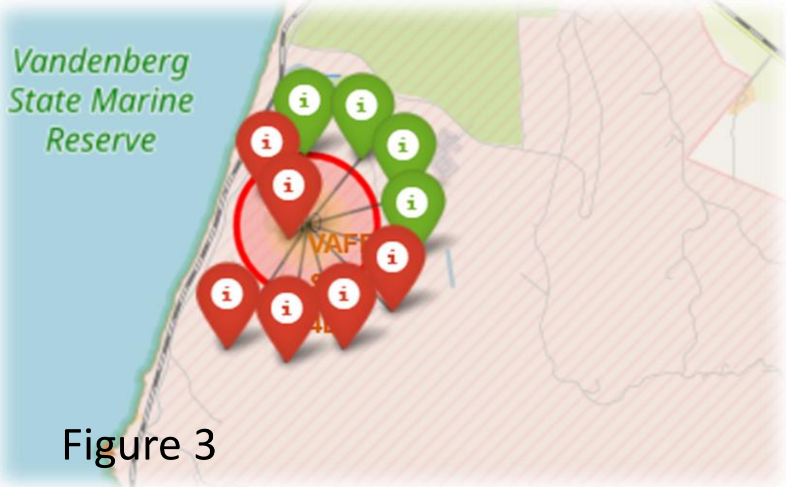


Figure 3

Figure 1: Florida launch sites KSC LC-39A, CCAFS LC-40, CCAFS SLC-40

Figure 2: CCAFS SLC-40 with 7 launches and 3 successful landings

Figure 3: California launch site VAFB SLC-4E, 10 launches and 4 successful landings

Folium Map Launch Sites Distances to Landmarks

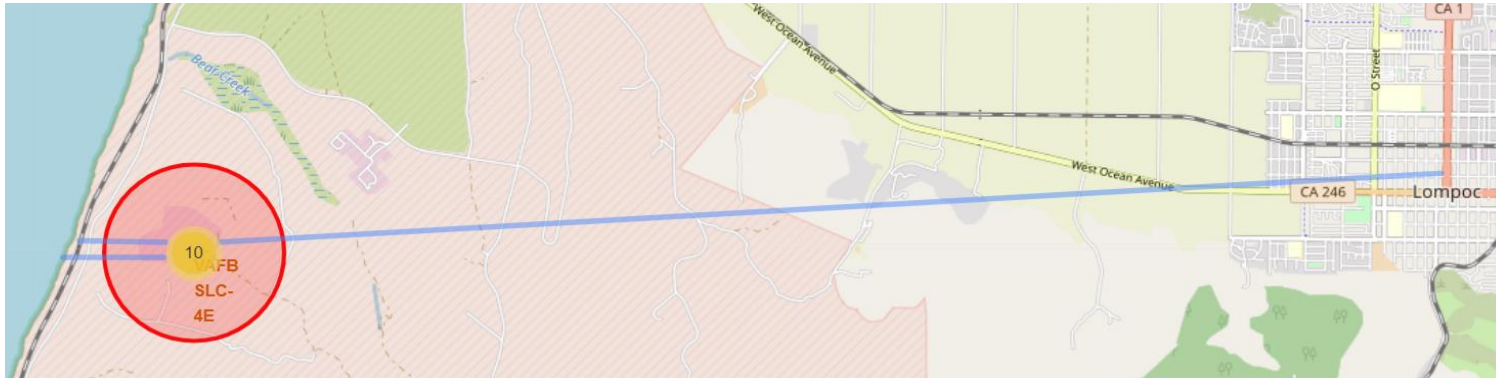


Figure 1: California launch site VAFB SLC-4E

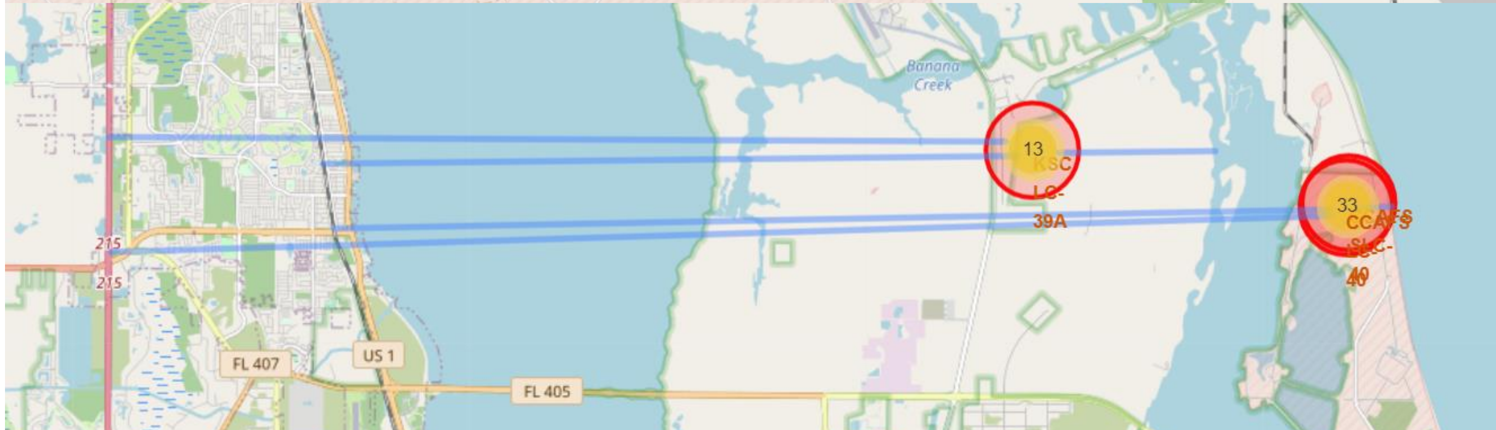


Figure 2: Florida launch sites KSC LC-39A, CCAFS LC-40, CCAFS SLC-40

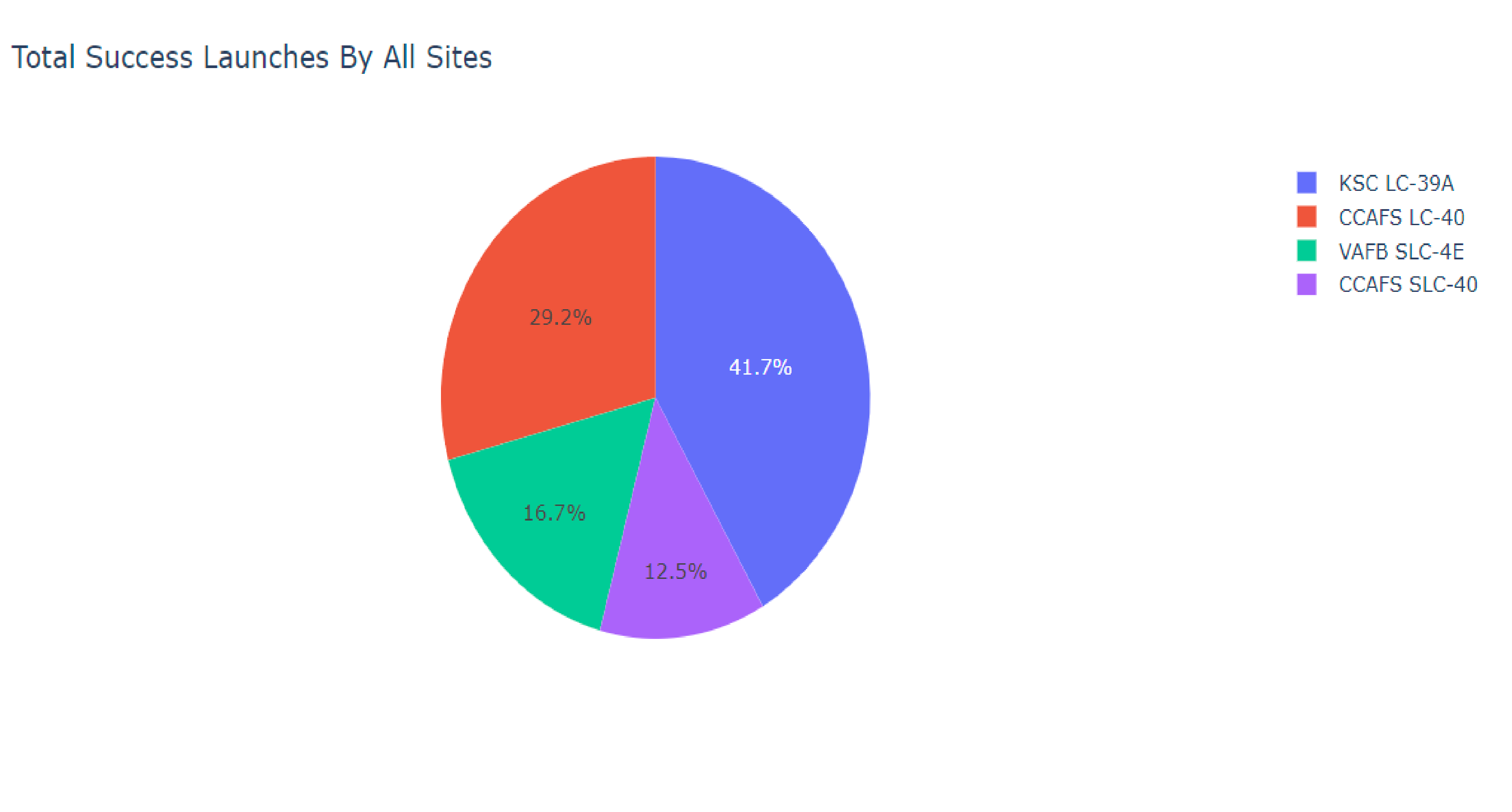
	Distance in Km's		
	VAFE SLC- 4E	KSC LC 39A	CCAFS's sites
Railway	1	15	21,6
Highway	13,67	19,41	26,5
Coast Line	1,03	3,9	0,6

- CCAFS sites where grouped into on column above due to the close prolixity to each other
- All sites are with 1km from the coast line
- Varying distances from the rail-road and greater than 10km from highways

Section 4

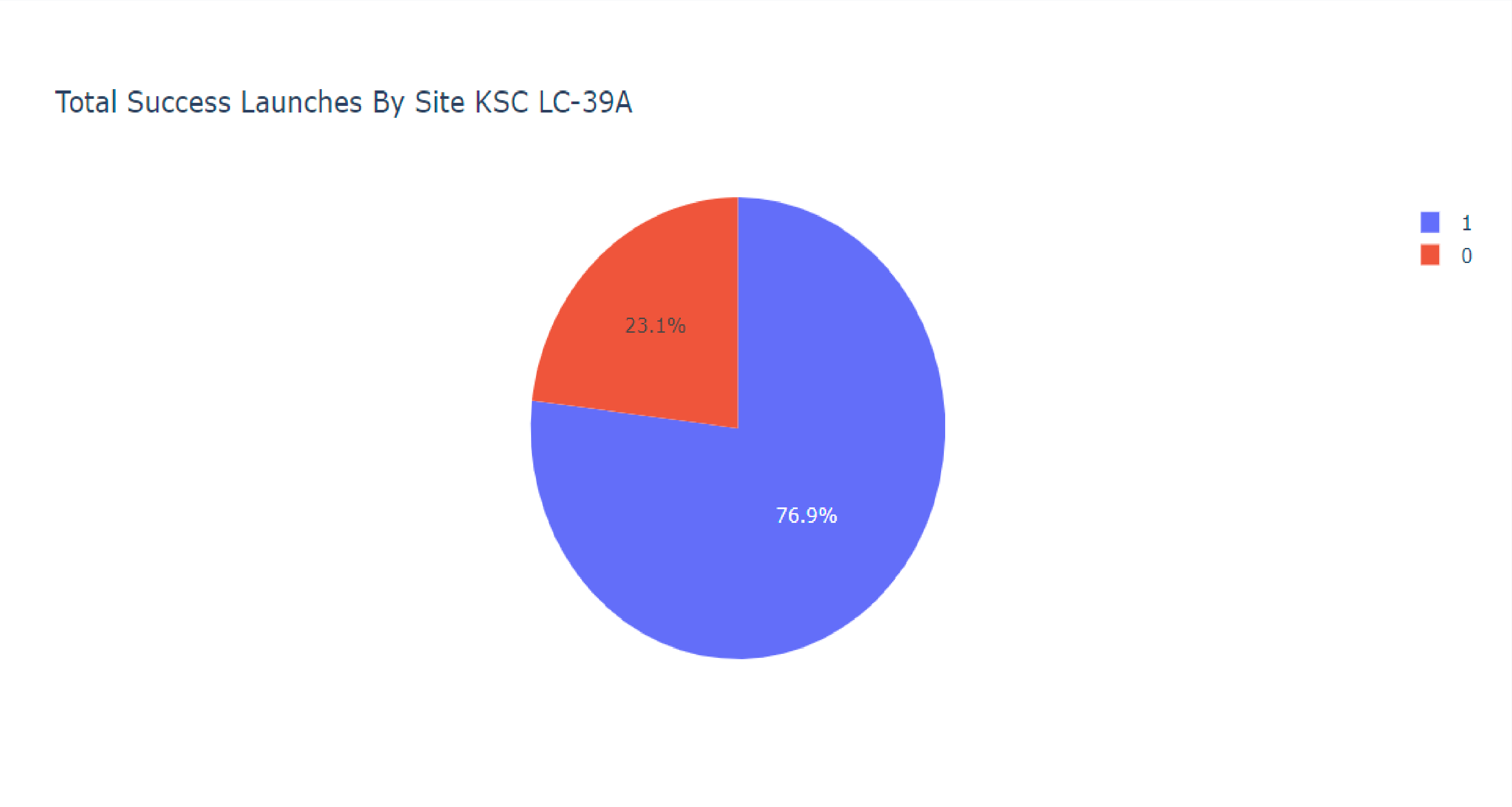
Build a Dashboard with Plotly Dash

Dashboard Pie Chart of Proportional Launch Site Success



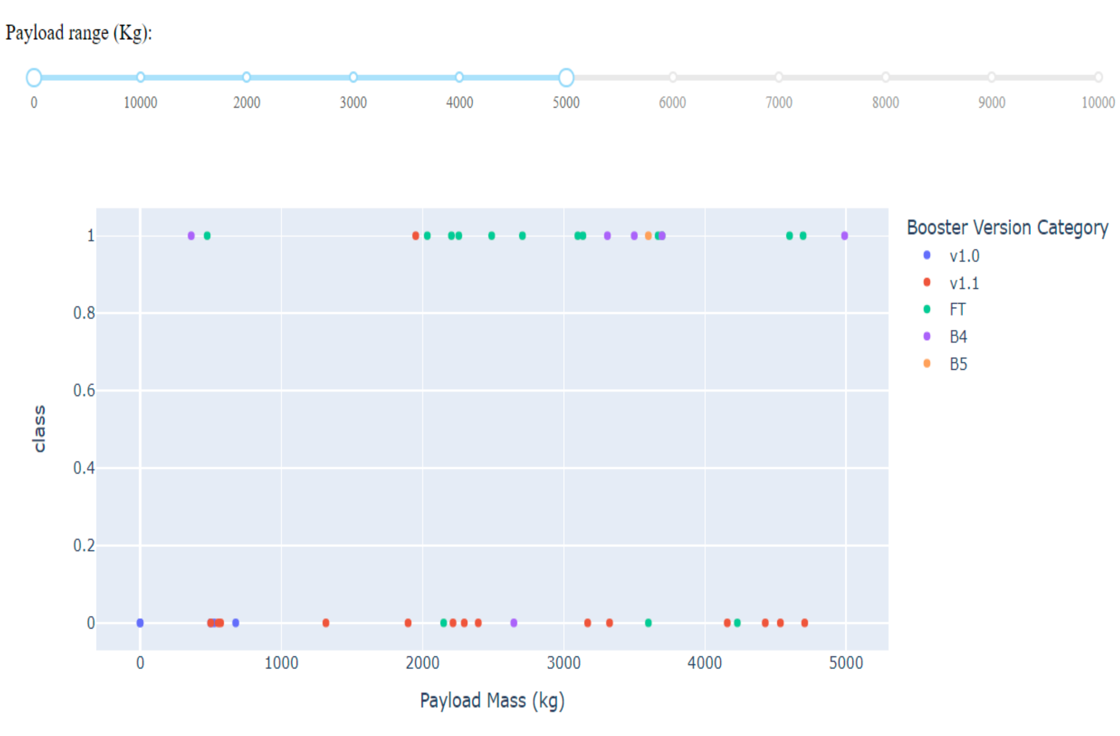
- KSC LC 39A had the highest proportion of successful launches of 41,7%
- Followed by CCAFS LC-40 with 29,2% successful launches

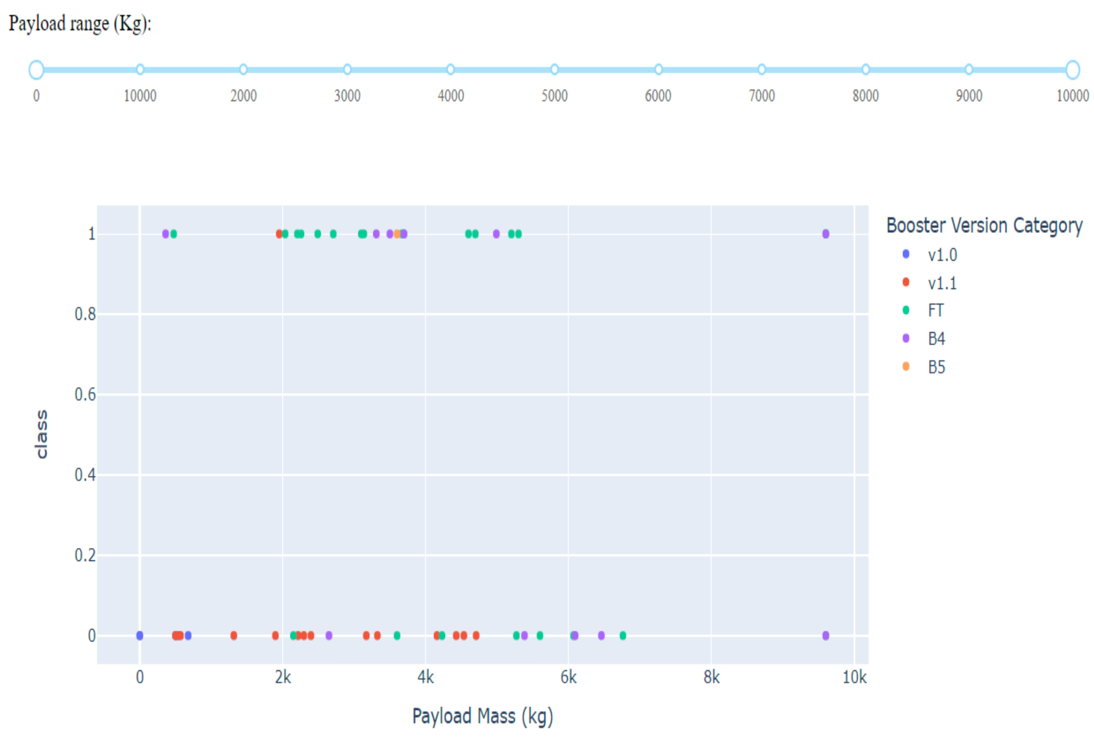
KSC LC 39A Pie Chart of Proportional Launch Site Success



- This launch site had the furthest distance (3,9km) from the coast line compared to the other sites which could have attributed to its successful landing percentage

Dashboard Scatter Plot with Payload Range Selector





0 – 5000kg Payloads

0 – 10000kg Payloads

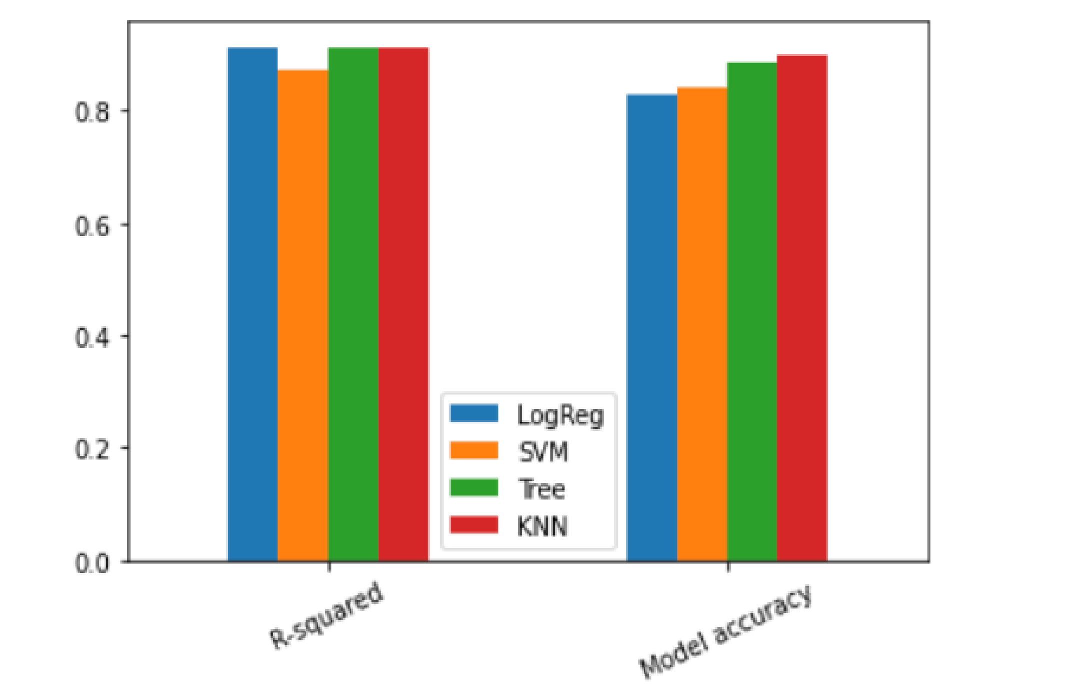
Section 5

Predictive Analysis (Classification)

Classification Accuracy

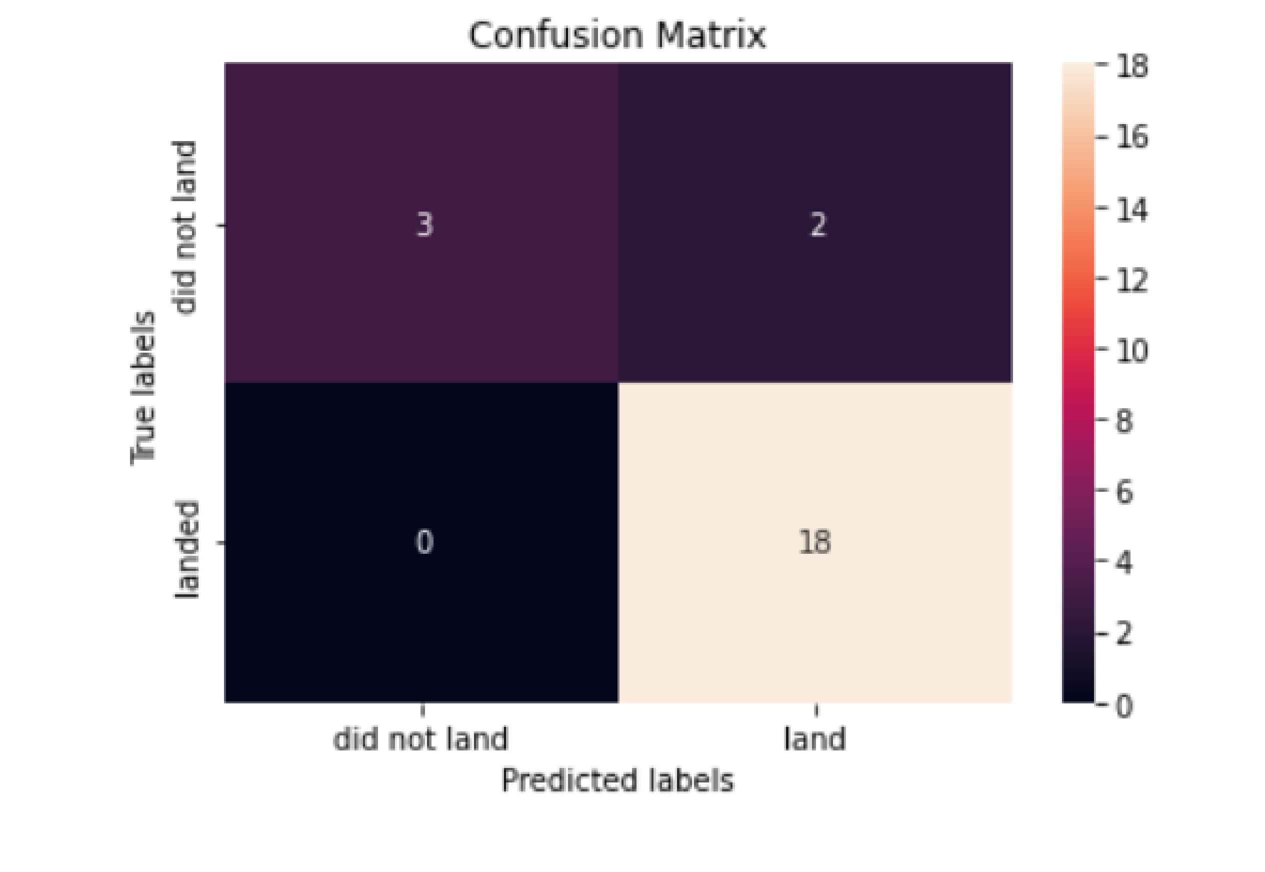
- R-squared is the mean measure of cv=10 grid search best parameter
- Model accuracy is the r-squared measure of the test data
- KNN model performed the best

	LogReg	SVM	Tree	KNN
R-squared	0.913043	0.869565	0.913043	0.913043
Model accuracy	0.826190	0.840476	0.883333	0.900000



Confusion Matrix

- KNN confusion matrix
- 0 false negatives
- 2 false positives



Conclusions

- KSC LC 39A had the highest successful landing outcomes, which also had the further coastline of 3,9km's. Coastal wind and humidity could have a role on landing success. This would require further investigation.
- Successful landings was seen with payloads between 2000 and 6000kg's.
- Orbits LEO, ISS, PO showed an increase in success rate after 20 flights, and SSO had a 100% success rate.
- Launches after 2013 had a greater probability of landing successfully.
- KNN algorithm produced the most accurate predictions.

Thank you!