IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Malik Hadi Ahmed
28/1/25

Outline

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

Executive Summary

- Summary of methodologies:
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis with Data Visualization
  - Exploratory Data Analysis with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive Analysis
- Summary of all results:
  - Exploratory Data Analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- Project background and context:
  - SpaceX is a company that specializes in making space travel affordable.
  - It advertises the 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.
  - Using machine learning and data, we can determine if the first stage will land, and therefore the cost of a launch.
- Problems you want to find answers
  - How do different variables affect the success of the first stage landing
  - Does the rate of successful findings increase over the years?
  - What is the best algorithm that can be used for binary classification in this case?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - Filtering the data, while dealing with missing values.
  - Using One Hot Encoding to prepare the data to a binary classification
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune, and evaluation of classification models to ensure the best results.

Data Collection

- Data collection involved a combo of API requests from SpaceX's REST API and Web Scraping from Space X's Wikipedia entry.
- Data Columns from REST API:
  - FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights…
- Data Columns from Web Scraping
  - Flight No., Launch Site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster…

Data Collection – SpaceX API

1. Requesting rocket launch data from SpaceX API
2. Decoding the response content using .json() and turning it into a data frame
3. Requesting needed info about the launches by applying functions
4. Constructing data we have obtained into a dictionary
5. Creating a data frame from the dictionary
6. Filtering the data frame to only include Falcon 9 launches
7. Replacing missing values of Payload Mass column with the mean
8. Exporting the data to CSV

Github: [DataScience-Falcon9LandingPrediction/jupyter-labs-spacex-data-collection-api.ipynb at main · ahmedmalikhadi/DataScience-Falcon9LandingPrediction (github.com)]

Data Collection - Scraping

1. Requesting Falcon 9 launch data from Wikipedia
2. Creating a BeautifulSoup object from HTML response
3. Extracting all column names from the HTML table header
4. Collecting the data by parsing HTML tables
5. Constructing data we have obtained into a dictionary
6. Creating a dataframe from the dictionary
7. Exporting the data to CSV

Github: DataScience-Falcon9LandingPrediction/jupyter-labs-webscraping.ipynb at main · ahmedmalikhadi/DataScience-Falcon9LandingPrediction (github.com)

Data Wrangling

- In the data set, there are several different cases where the booster did not land successfully.
- Those outcomes are converted into binary values for training, with 1 meaning the booster was successful and 0 meaning it was not.

1. Perform exploratory Data Analysis and determine Training Labels
2. Calculate the number of launches on each site
3. Calculate the number and occuurence of each orbit
4. Calculate the number and occurrence of mission outcome per orbit type
5. Create a landing outcome label from Outcome column
6. Exporting the data to CSV

Github: [DataScience-Falcon9LandingPrediction/labs-jupyter-spacex-Data wrangling.ipynb at main · ahmedmalikhadi/DataScience-Falcon9LandingPrediction (github.com)]

EDA with Data Visualization

- Flight No. vs Payload Mass, Flight No. vs Launch Site, Payload Mass vs. Launch Site, Orbit Type vs Success Rate, Flight No. vs Orbit Type, Payload Mass vs Orbit Type, and Success Rate Yearly Trend were plotted.
- Scatter plots show the relationships between variables which can be used for training Machine Learning models
- Bar charts show comparisons between discrete categories.
- Line charts show the trends in data over time.
- Github: DataScience-Falcon9LandingPrediction/edadataviz.ipynb at main · ahmedmalikhadi/DataScience-Falcon9LandingPrediction (github.com)

EDA with SQL

- Displaying the names of unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string ‘CCA’
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was achieved
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass
- Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in the year 2015
- Ranking the count of landing outcomes in drone ship, their booster versions and launch site names for the months in the year 2015
- Ranking 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
- Github: DataScience-Falcon9LandingPrediction/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · ahmedmalikhadi/DataScience-Falcon9LandingPrediction (github.com)

Build an Interactive Map with Folium

- Markers of all Launch Sites:
    - Added Marker with Circle, Popup Label and Text Label of NASA Johnson Space Center using its latitude and longitude coordinates as a start location
    - Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to Equator and coasts
- Colored Markers of launchoutcomes for each Launch Site
    - Added colored Markers of success and failure for launches using Marker Cluster to identify which launch sites have relatively high success rates
- Distances between a Launch Site to its proximities
    - Added colored lines to show distances between Launch Sites and its proximities such as Railways, Highways, etc.
- Github: DataScience-Falcon9LandingPrediction/lab_jupyter_launch_site_location.ipynb at main · ahmedmalikhadi/DataScience-Falcon9LandingPrediction (github.com)

Build a Dashboard with Plotly Dash

- Launch Sites Dropdown List
- Pie Chart showing Success Launches
- Slider of Payload Mass Range
- Scatter Chart of Payload Mass vs Success Rate
- GitHub: DataScience-Falcon9LandingPrediction/spacex_dash_app.py at main · ahmedmalikhadi/DataScience-Falcon9LandingPrediction (github.com)

Predictive Analysis (Classification)

1. Creating a NumPy array from the column “Class” in data
2. Standardizing the data with StandardScaler, then fitting and transforming it
3. Splitting the data into training and testing sets with train_test_split function
4. Creating a GridSearchCV object with cv = 10 to find the best parameters
5. Applying GridSearchCV on LogReg, SVM, Decision Tree, and KNN models
6. Calculating the accuracy on the test data using the method .score() for all models
7. Examining the confusion matrix for all models
8. Finding the method performs best by examining the Jaccard Score and F1 Score metrics

- GitHub: DataScience-Falcon9LandingPrediction/SpaceX_Machine Learning Prediction_Part_5.ipynb at main · ahmedmalikhadi/DataScience-Falcon9LandingPrediction (github.com)

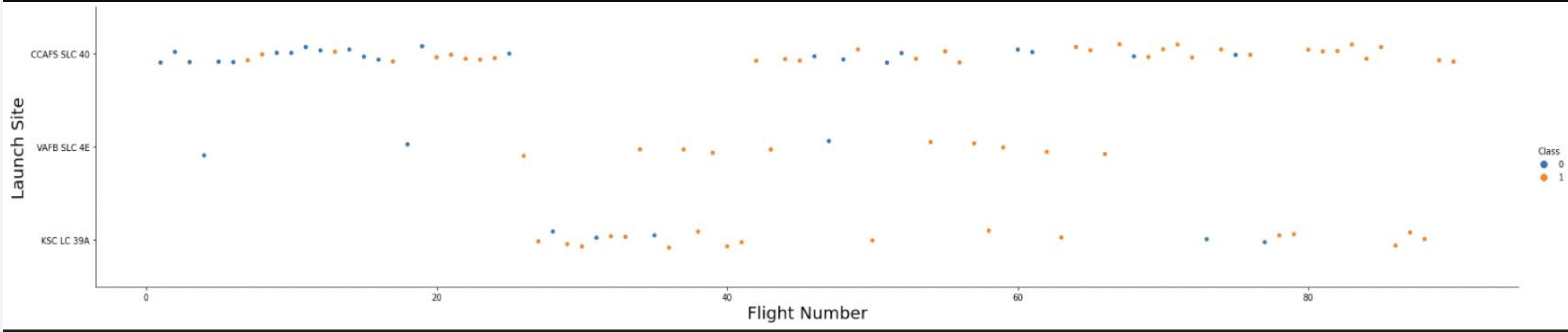
Results

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

Section 2

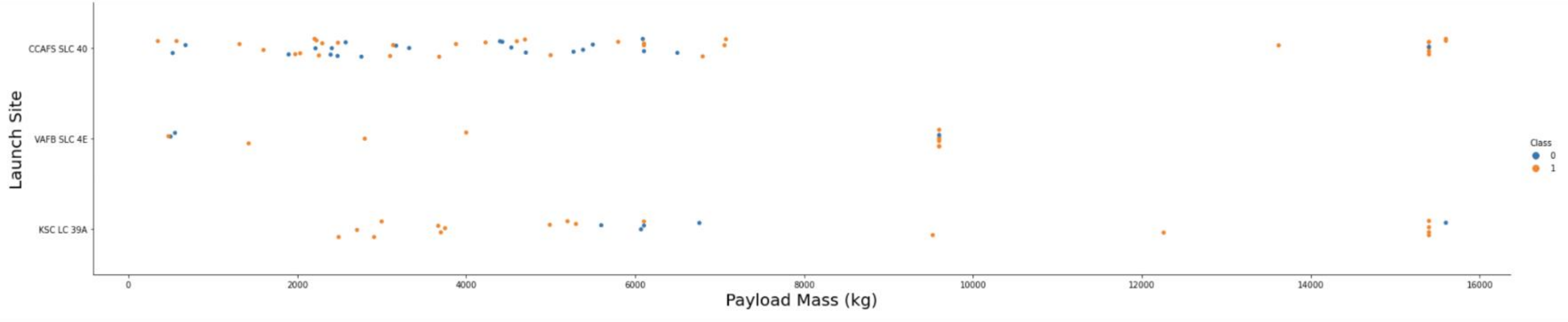
Insights drawn from EDA

Flight Number vs. Launch Site



- The earliest flights all failed while the latest flights all succeeded
- The CCAFS SLC 40 launch site has about a half of all launches
- VAFB SLC 4E and KSC LC 39A have higher success rates
- It can be assumed that each new launch has a higher rate of success

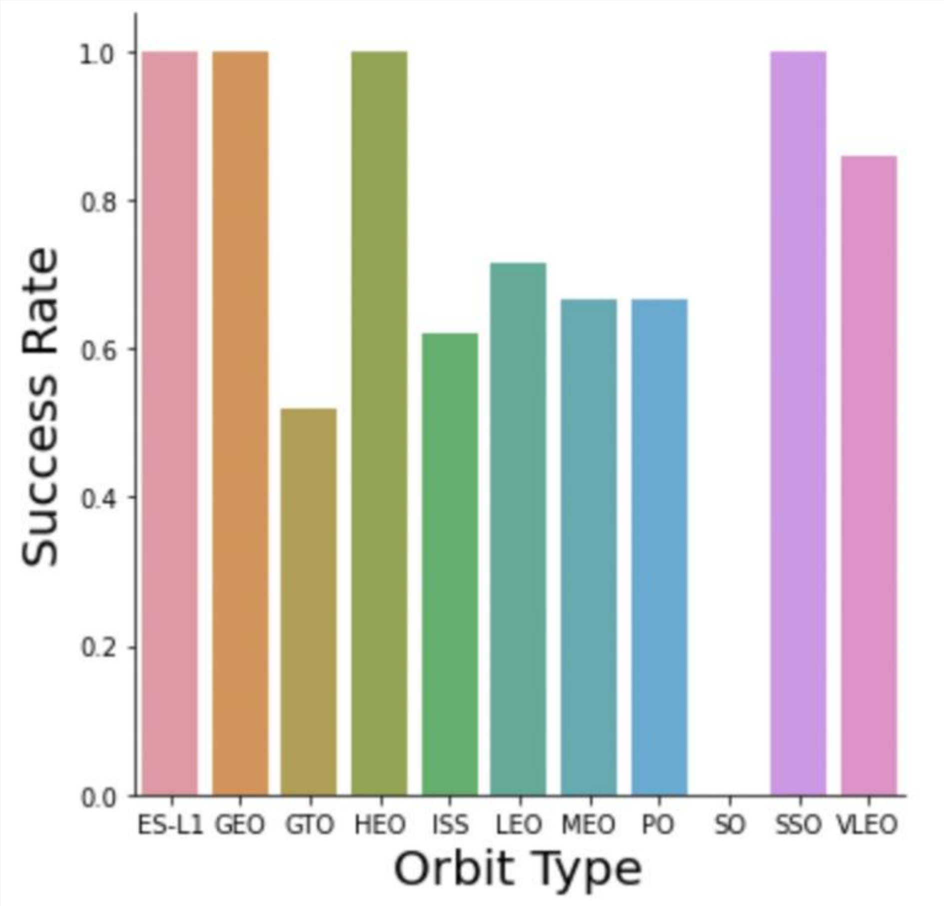
Payload vs. Launch Site



- For every launch site the higher the payload mass, the higher the success rate.
- Most of the launches with payload mass over 7000 kg were successful.
- KSC LC 39A has a 100% success rate for payload mass under 5500 kg too.

Success Rate vs. Orbit Type

- Orbits with 100% success rate:
  - ES-L1, GEO, HSO, SSO
- Orbits with 0% success rate:
  - SO
- Orbits with success rate between 50% and 85%
  - GTO, ISS, LEO, MEO, PO

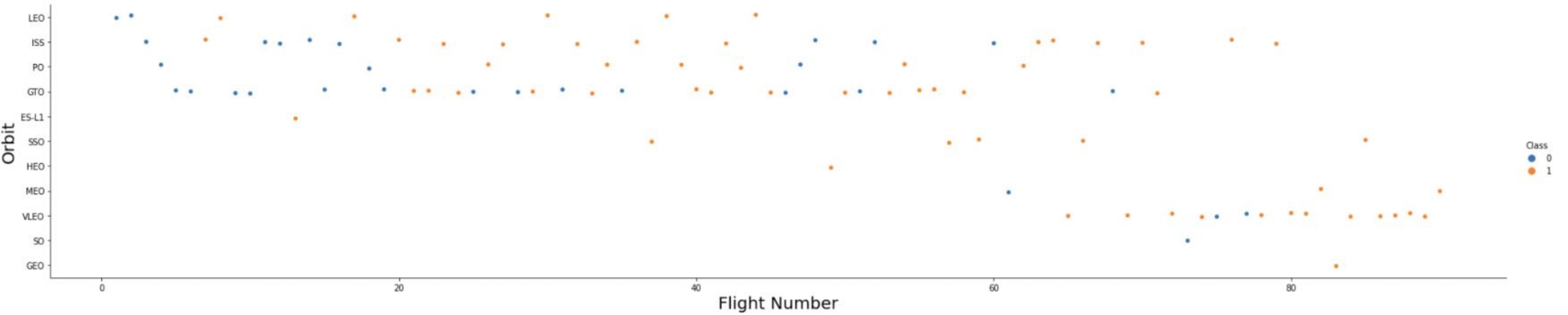


Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

- Show the screenshot of the scatter plot with explanations

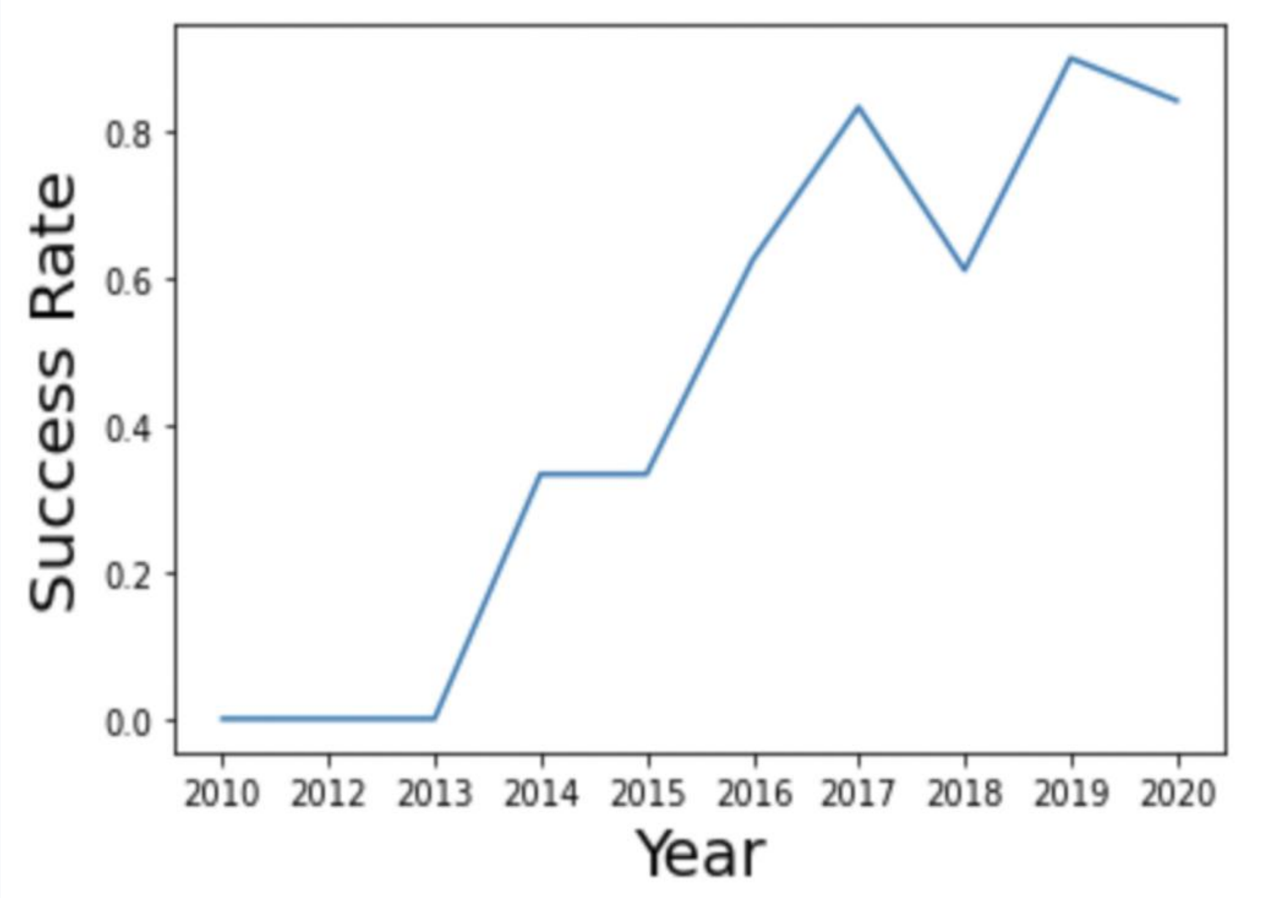
Payload vs. Orbit Type



- 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

Launch Success Yearly Trend

- The success rate since 2013 kept increasing until 2020 overall



All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

Launch Site Names Begin with 'CCA'

In [5]:
```
%sql select * from SPACEXDATASET where launch_site like 'CCA%' limit 5;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[5]:

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

```
In [6]: %sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXDATASET where customer = 'NASA (CRS)';

 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
Out[6]:
```

total_payload_mass
45596

Average Payload Mass by F9 v1.1

```
In [7]: %sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXDATASET where booster_version like '%F9 v1.1%';
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[7]:

average_payload_mass
2534

First Successful Ground Landing Date

```
In [8]: %sql select min(date) as first_successful_landing from SPACEXDATASET where landing__outcome = 'Success (ground pad)';

 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.

Out[8]:
```

first_successful_landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [9]: %sql select booster_version from SPACEXDATASET where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ between 4
        000 and 6000;

 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.

Out[9]:
```

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

Total Number of Successful and Failure Mission Outcomes

```
In [10]: %sql select mission_outcome, count(*) as total_number from SPACEXDATASET group by mission_outcome;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[10]:

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

Boosters Carried Maximum Payload

```
In [11]: %sql select booster_version from SPACEXDATASET where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXDATASET);

 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
Out[11]:
```

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

2015 Launch Records

```
In [12]: %%sql select monthname(date) as month, date, booster_version, launch_site, landing__outcome from SPACEXDATASET
                where landing__outcome = 'Failure (drone ship)' and year(date)=2015;

 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[12]:

MONTH	DATE	booster_version	launch_site	landing__outcome
January	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

In [13]:
```
%%sql select landing__outcome, count(*) as count_outcomes from SPACEXDATASET
      where date between '2010-06-04' and '2017-03-20'
      group by landing__outcome
      order by count_outcomes desc;
```

```
 * ibm_db_sa://wzf08322:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[13]:

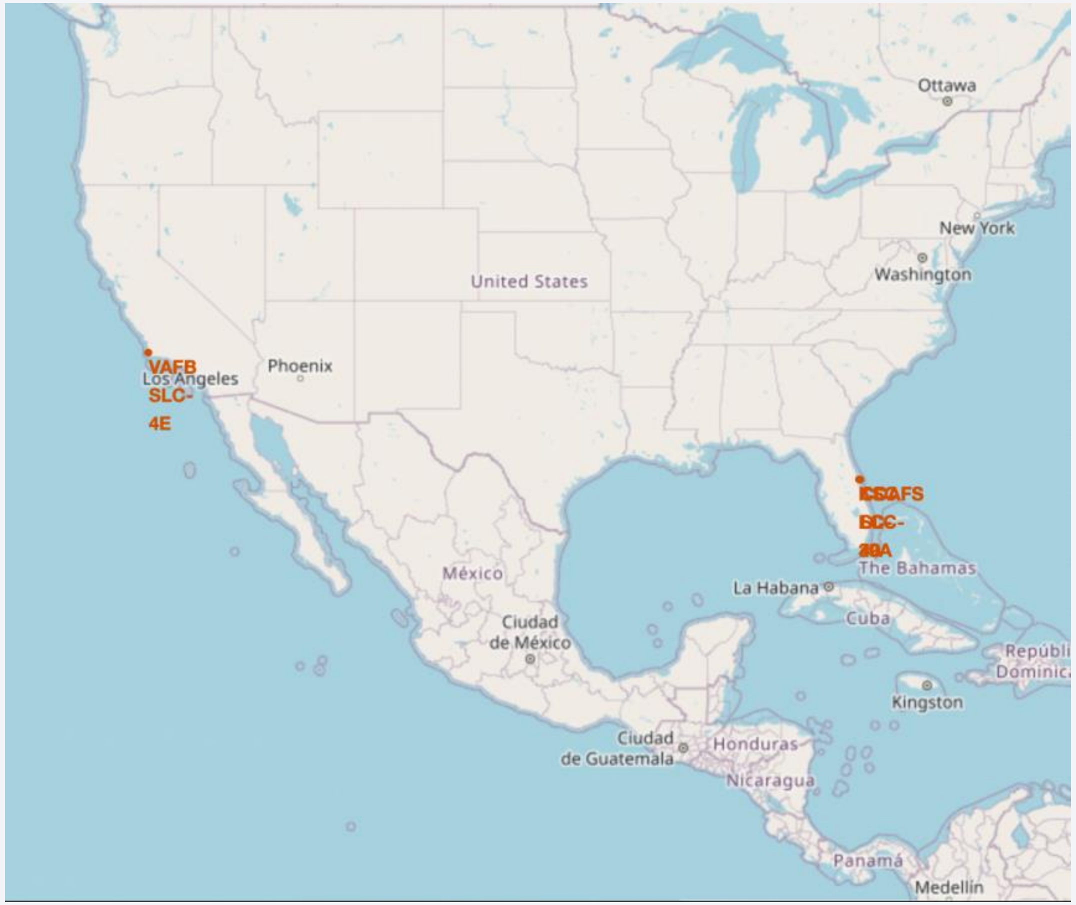
landing__outcome	count_outcomes
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

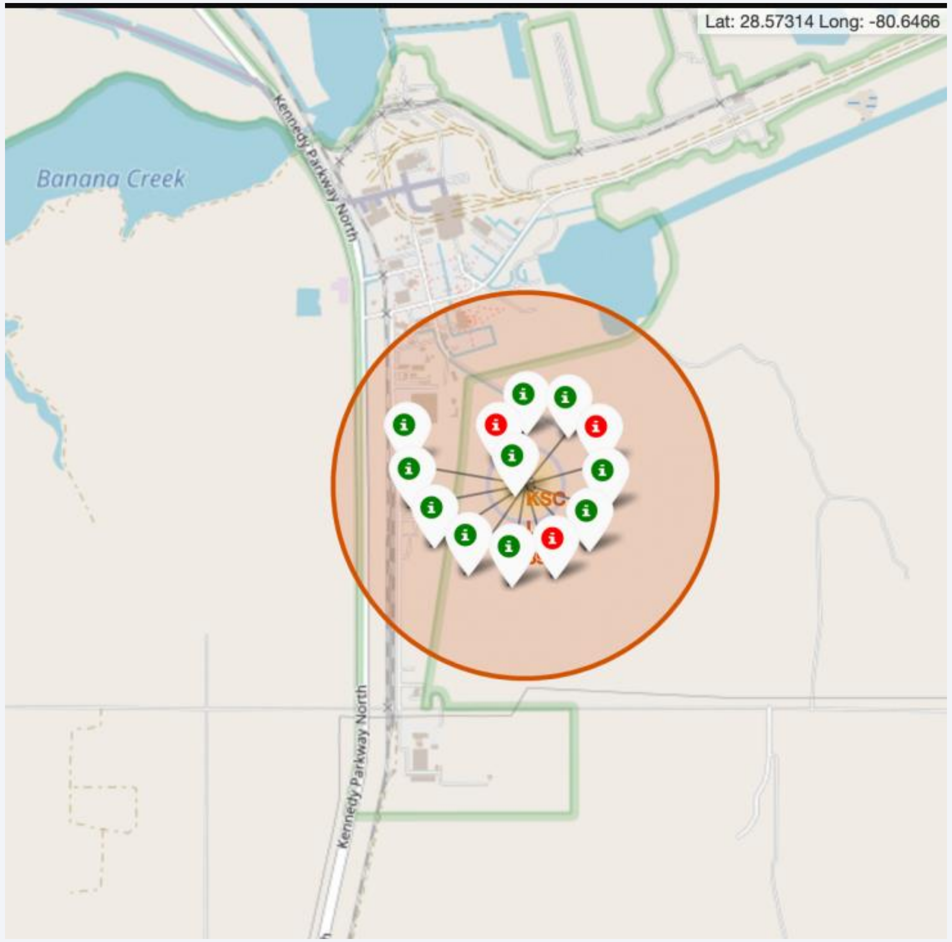
All launch sites' location markers on a global map

- Most Launch sites are in proximity to the Equator Line and Coast
- Land is moving faster at the equator than any other place on the surface of the Earth
- If a ship is launched from the equator it goes up into space
- Launching the rockets towards the ocean minimizes the risk of debris damage



Colour labeled launch records

- Green Marker = Success
- Red Marker = Failure
- Launch Site KSC LC-39A has a very high success rate



<Folium Map Screenshot 3>

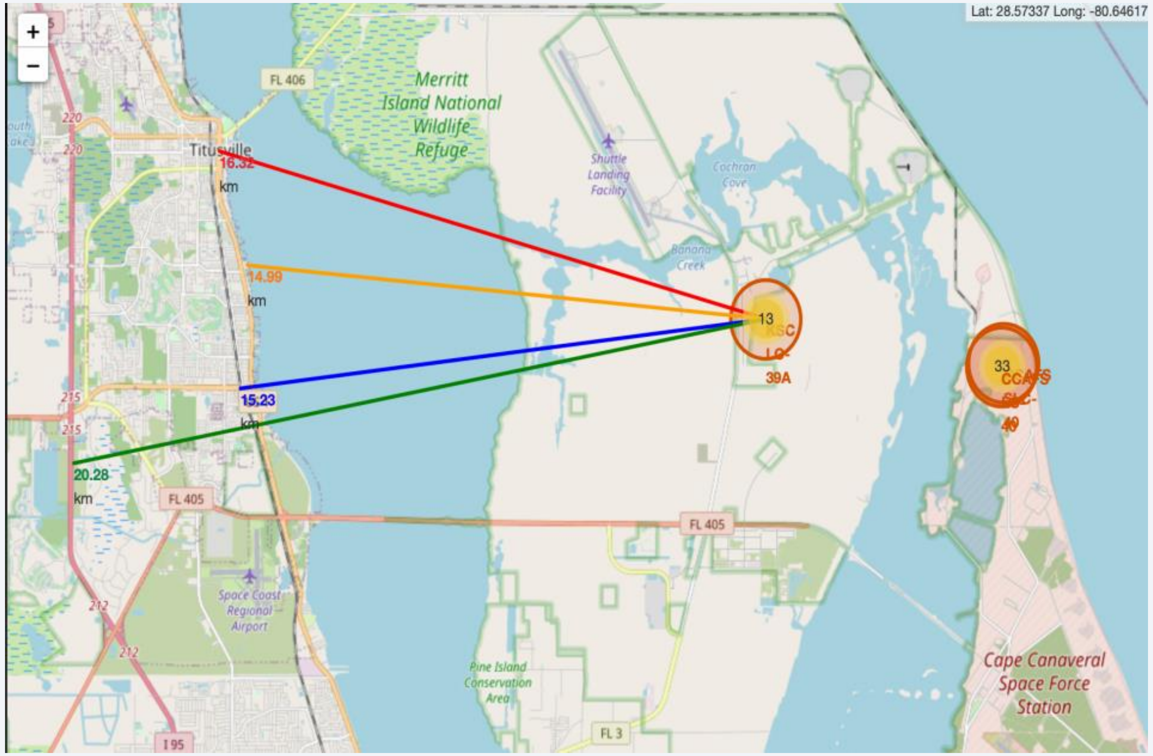
- Replace <Folium map screenshot 3> title with an appropriate title

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot

Distance from the launch site KSC LC-39A to its proximities

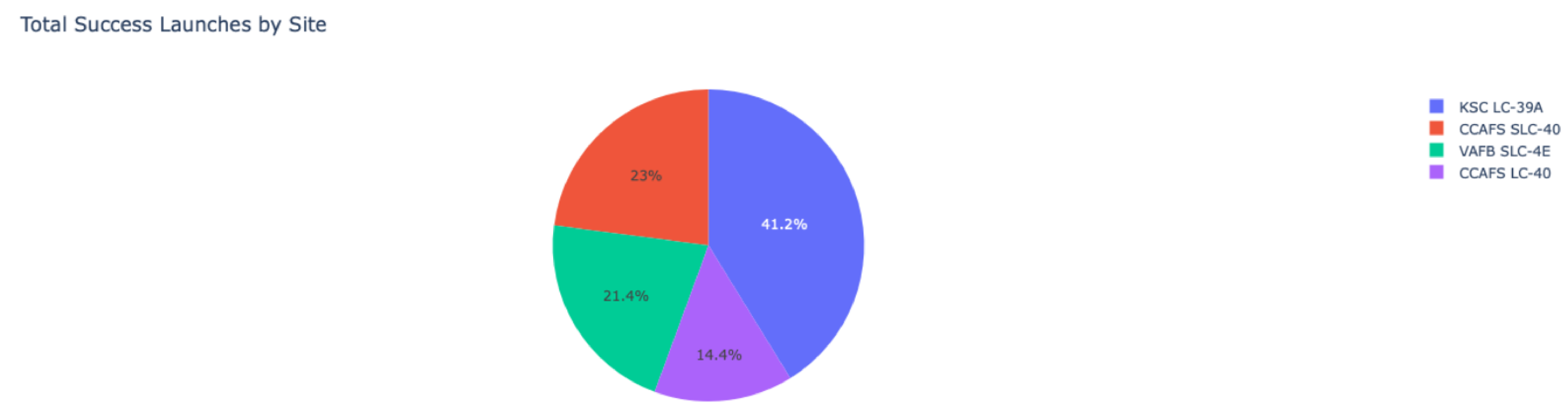
- KSC LC-39A is:
  - Close to the railway ~15km
  - Close to the highway ~20km
  - Close to the coastline ~15km
- Also the launch site is close to its closest city Titusville by 16.32 km
- Failed rockets with their high speed can cover distances ~15-20km in a few seconds, making it potentially dangerous to populated areas.



Section 4

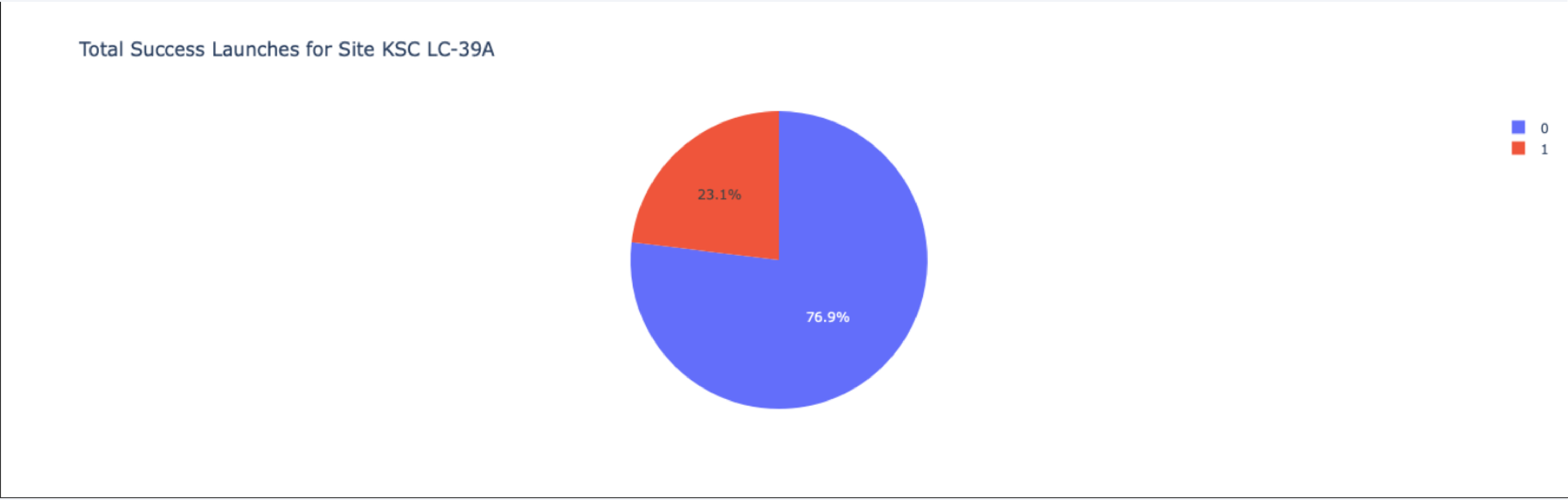
Build a Dashboard with Plotly Dash

Launch success count for all sites



- From all the sites, KSC LC-39A has the most successful launches

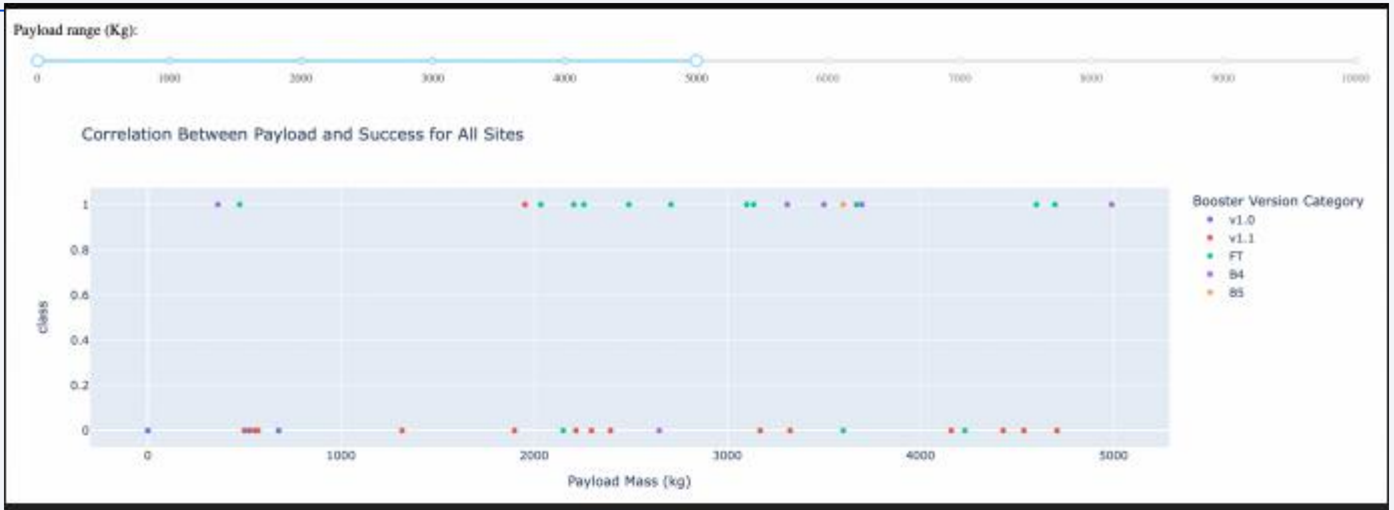
Launch site with highest launch success ratio

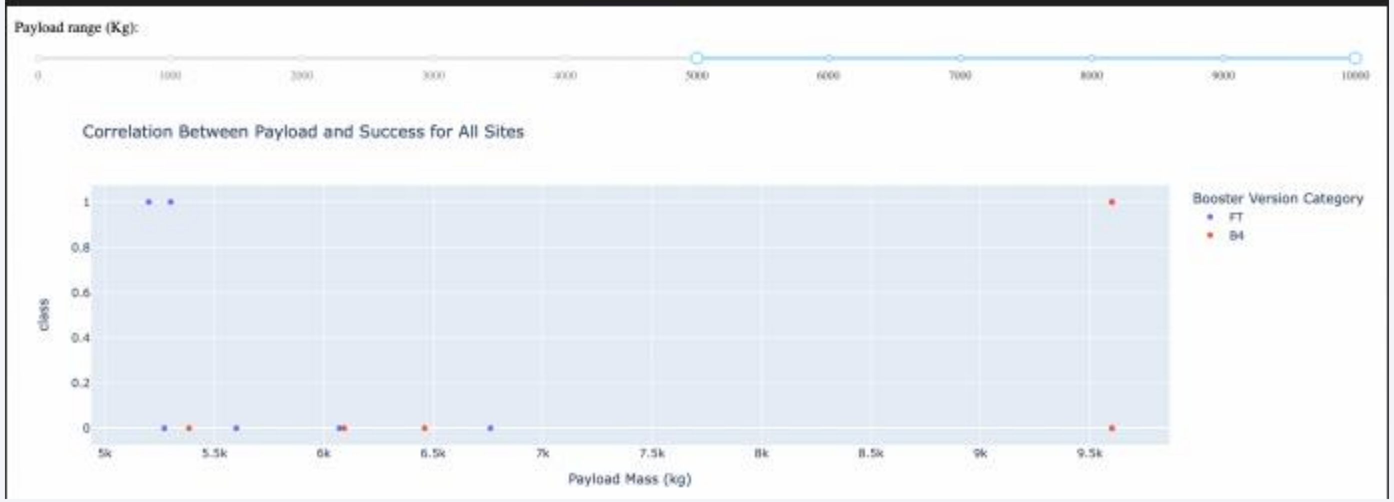


- KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings

Payload Mass vs Launch Outcome for all sites

- The charts show that payloads between 2000 and 5500 kg have the highest success rate.





Section 5

Predictive Analysis (Classification)

Classification Accuracy

- We cannot confirm which method is optimal based on the scores of the Test Set
- This may be due to the small sample size.
- The scores of the whole dataset confirm that the most optimal model is the decision tree model.

Scores and Accuracy of the Test Set

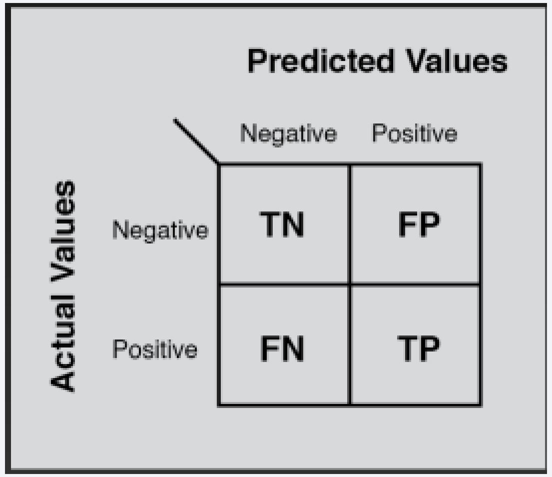
	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

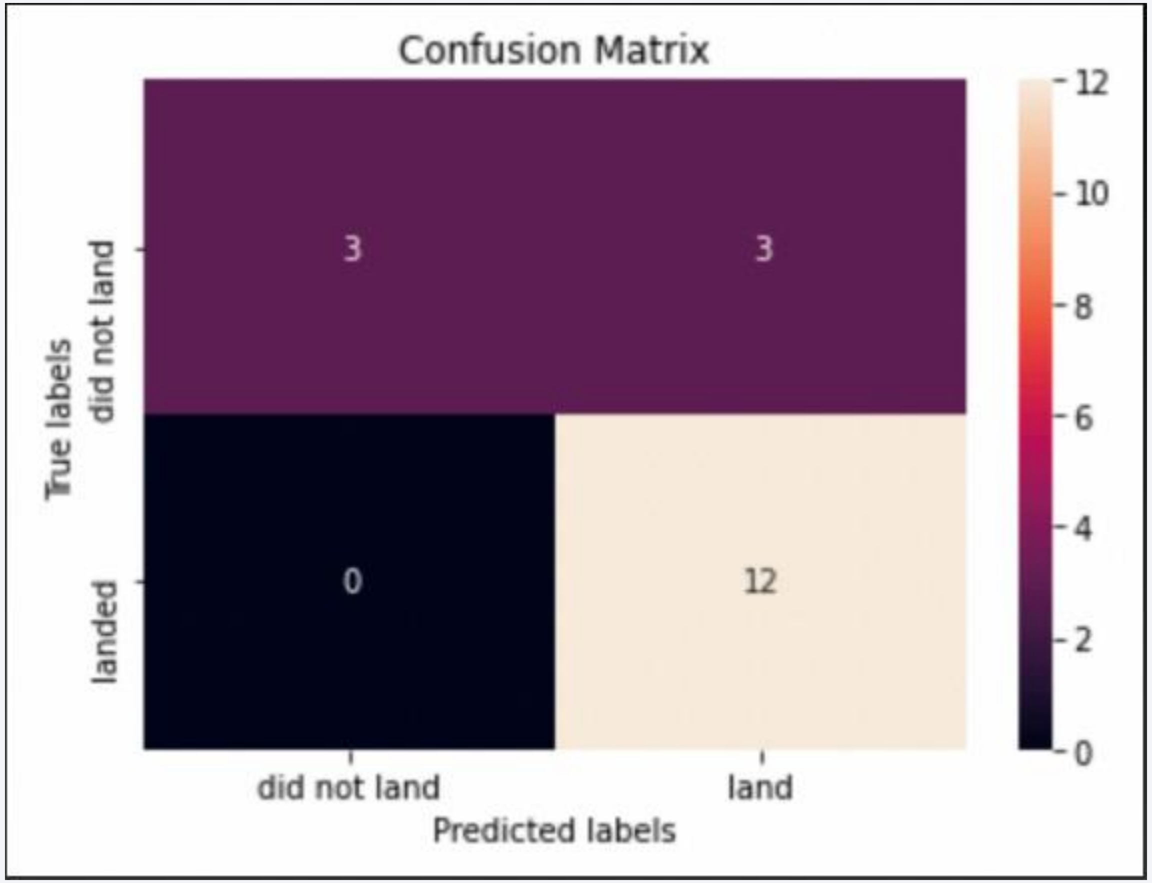
Scores and Accuracy of the Entire Data Set

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.882353	0.819444
F1_Score	0.909091	0.916031	0.937500	0.900763
Accuracy	0.866667	0.877778	0.911111	0.855556

Confusion Matrix

- Through the confusion matrix, we can identify that logistic regression can distinguish between the different classes, we can understand that the major problem is false positives.





Conclusions

- The decision tree model was the most optimal choice for this dataset
- Launches with a low payload mass show better results than launches with a larger payload mass
- Most of launch sites are in proximity to the equator line and all the sites are in very close proximity to the coast
- The success rate has increased over the years
- KSC LC-39A is the launch site with the highest success rate
- Orbits ES-L1, GEO, HEO, and SSO have 100% success rate

Thank you!