DATA SCIENCE PROJECT

SPACEX PROJECT

- PARTH PATIL

30th June,2024

IBM Developer

SKILLS NETWORK

OUTLINE FOR THE PROJECT

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- 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 (Classification)
- Results

INTRODUCTION FOR THE PROJECT

- SpaceX Overview: SpaceX, the leading company in the commercial space age, aims to make space travel more affordable.
- Falcon 9 Cost: SpaceX advertises Falcon 9 rocket launches for $62 million, compared to over $165 million from other providers.
- Reusability Savings : Much of the cost savings comes from reusing the first stage of the rocket.
- Predicting Reusability : By predicting if the first stage will land successfully, we can estimate the launch cost.
- Project Goal : Using public information and machine learning models, we aim to predict the reusability of SpaceX's first stage.

METHODOLOGY FOR THE PROJECT

- **Data Collection Methodology**
  - **SpaceX REST API**: Collecting data directly from SpaceX's API.
  - **Web Scraping**: Gathering additional data from Wikipedia.
- **Data Wrangling**
  - **Filtering**: Cleaning and organizing the data.
  - **Handling Missing Values**: Addressing incomplete data entries.
  - **One Hot Encoding**: Preparing data for binary classification.
- **Exploratory Data Analysis (EDA)**
  - **Visualization and SQL**: Analyzing data patterns and insights.
- **Interactive Visual Analytics**
  - **Tools**: Using Folium and Plotly Dash for interactive visualizations.
- **Predictive Analysis**
  - **Classification Models**: Building, tuning, and evaluating models for best results.

DATA COLLECTION

- **API Requests**: Collected data from SpaceX's public API.
- **Web Scraping**: Extracted data from SpaceX's Wikipedia table.

Collecting the data

- The process involves SpaceX API requests, JSON normalization, DataFrame conversion, filtering for Falcon 9, and imputing missing PayloadMass values.
- GitHub: DATA COLLECTION

Websracpping

- The process involves requesting Wikipedia HTML, parsing with BeautifulSoup, finding the launch info table, extracting data to a dictionary, and casting to a DataFrame.
- GitHub: WEBSRACPPING

DATA WRANGLING

- Generate a new 'class' column to categorize landing outcomes: assign 1 for successful missions based on the 'Mission Outcome' being true, indicating success, and 0 for failures. This label distinguishes between successful and unsuccessful landings, providing a clear binary classification for training and analysis purposes.
- True ASDS, True RTLS, & True Ocean are set as 1
- None None, False ASDS, None ASDS, False Ocean, False RTLS are set as 0

- GitHub: DATA WRANGLING

EDA WITH DATA VISUALIZATION

- Scatter plots show the relationship between variables. If a relationship exists, they could be used in machine learning model.
- Bar charts show comparisons among discrete categories. The goal is to show the relationship between the specific categories being compared and a measured value
- Line charts show trends in data over time (time series)
- Charts plotted: Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit Type vs. Success Rate, Flight Number vs. Orbit Type, Payload Mass vs Orbit Type and Success Rate Yearly Trend

- GitHub: EDA with DATA VISUALIZATION

EDA WITH SQL

- The analysis of space mission data involved several queries: identifying unique launch sites, examining records of 'CCA'-prefixed sites, totaling payload mass from NASA's CRS missions, averaging payload mass for F9 v1.1 boosters, pinpointing the first successful ground pad landing date, identifying boosters with successful drone ship landings and specific payload criteria, tallying total mission outcomes, listing booster versions with maximum payload achievements, and detailing failed drone ship landings in 2015 with associated booster versions and launch site names for comprehensive analysis.

- GitHub: EDA with SQL

ITERACTIVE MAP WITH FOLIUM

- NASA Johnson Space Center Marker: Added a circle marker with popup and text labels at NASA Johnson Space Center's latitude and longitude to denote its location.
- Launch Sites Markers: Added circle markers with popup and text labels for all launch sites, displaying their geographical positions relative to the Equator and coastlines.
- Colored Markers for Launch Outcomes: Used marker clusters to display launch outcomes (success in green, failure in red) for each launch site, highlighting sites with high success rates.
- Distances Visualization: Added colored lines to show distances from a launch site (e.g., KSC LC-39A) to nearby features like railways, highways, coastlines, and closest cities, providing spatial context.
- Marker Interactivity: Enabled popup labels to show additional details such as launch success rates or specific site information when markers are clicked.
- Visualization Clarity: Ensured markers and lines are clear and visually distinct, aiding in understanding spatial relationships and launch performance metrics at each site.
- GitHub: MAP with FOLIUM

DASHBOARD WITH PLOTLY DASH

- Launch Sites Dropdown List: Added a dropdown list for selecting a specific launch site.
- Pie Chart for Success Launches: Added a pie chart to display the total successful launches for all sites, and success vs. failed counts when a specific launch site is selected.
- Payload Mass Range Slider: Added a slider to allow users to select a specific payload mass range.
- Scatter Chart of Payload Mass vs. Success Rate: Added a scatter chart to show the correlation between payload mass and launch success rates for different booster versions.
- Interactive Elements: Ensured dropdown, pie chart, slider, and scatter chart are interactive, allowing users to dynamically filter and analyze data.
- Visualization Integration: Integrated all visual elements to provide a comprehensive, user-friendly dashboard for analyzing launch data.
- GitHub: DASHBOARD CODE

PREDICTIVE ANALYSIS

- Create a NumPy array from the "Class" column and standardize the data using `StandardScaler`.
- Split the data into training and testing sets using the `train_test_split` function.
- Create a `GridSearchCV` object with `cv = 10` to find the best parameters for Logistic Regression, SVM, Decision Tree, and KNN models.
- Apply `GridSearchCV` on each model and calculate the accuracy on the test data using the `.score()` method, then examine the confusion matrix for each model.
- Determine the best-performing method

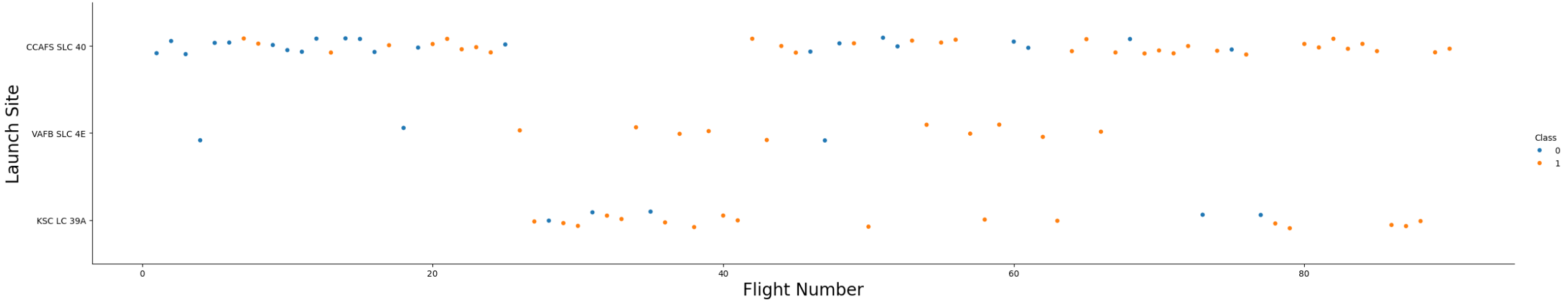
- GitHub: Machine learning

RESULTS

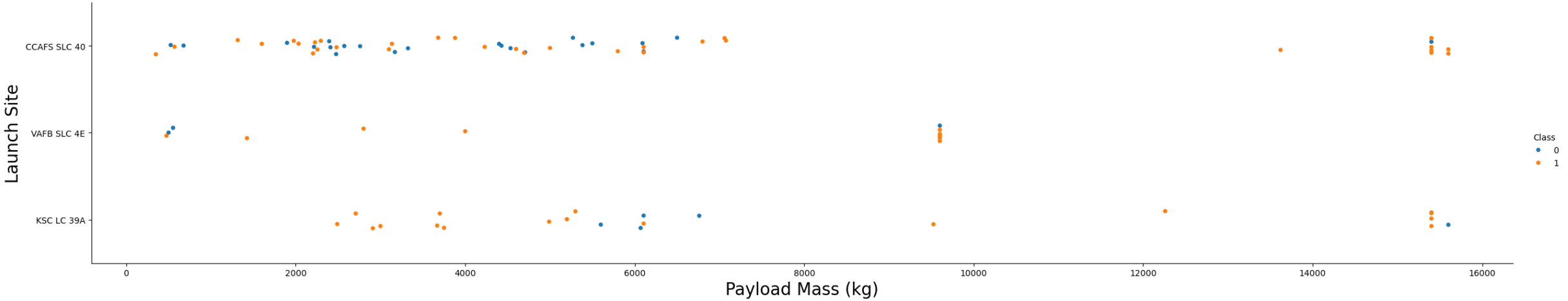
- EDA with visualization
- EDA with SQL
- Interactive map with Folium
- Plotly Dash dashboard
- Predictive Analysis

EDA WITH DATA VISUALIZATION

Flight No. VS Launch site graph shows - The earliest flights all failed while the latest flights all succeeded. It can be assumed that each new launch has a higher rate of success



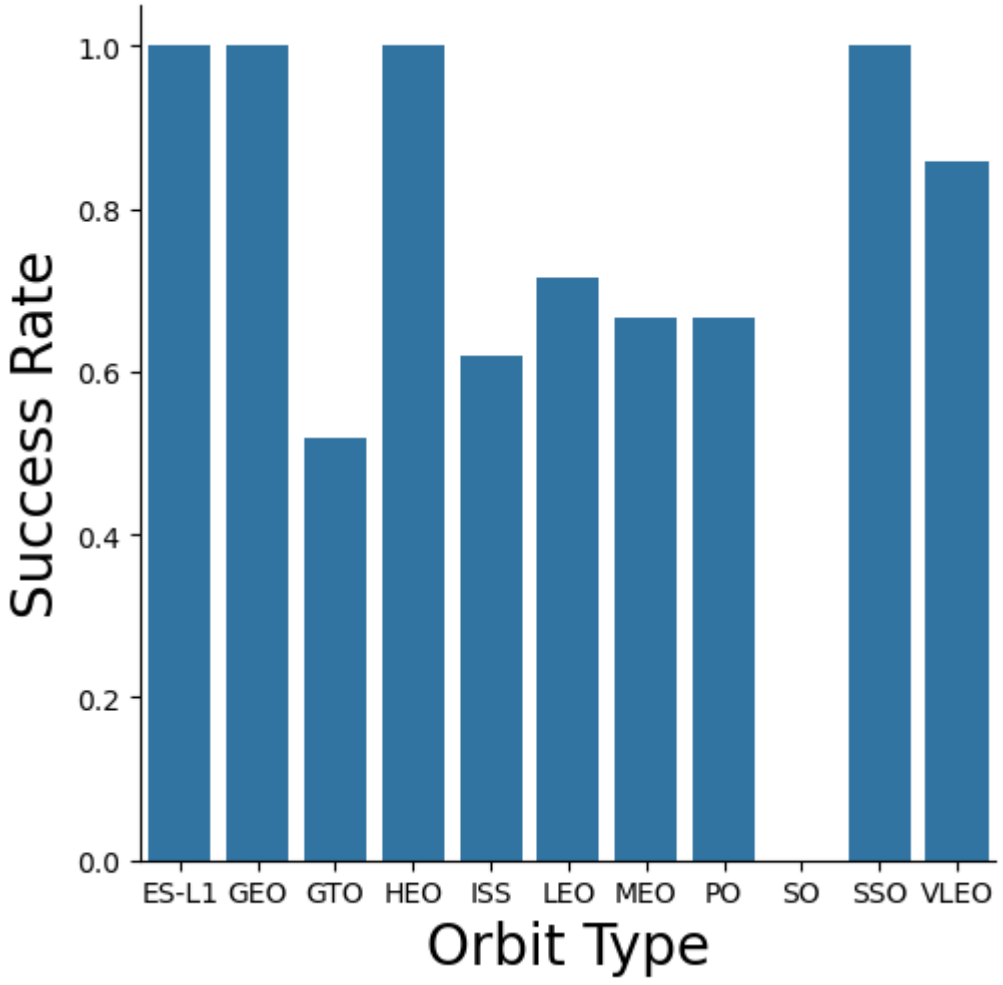
Payload Mass VS Launch Site graph show - For every launch site the higher the payload mass, the higher the success rate. Most of the launches with payload mass over 7000 kg were successful



IBM Developer

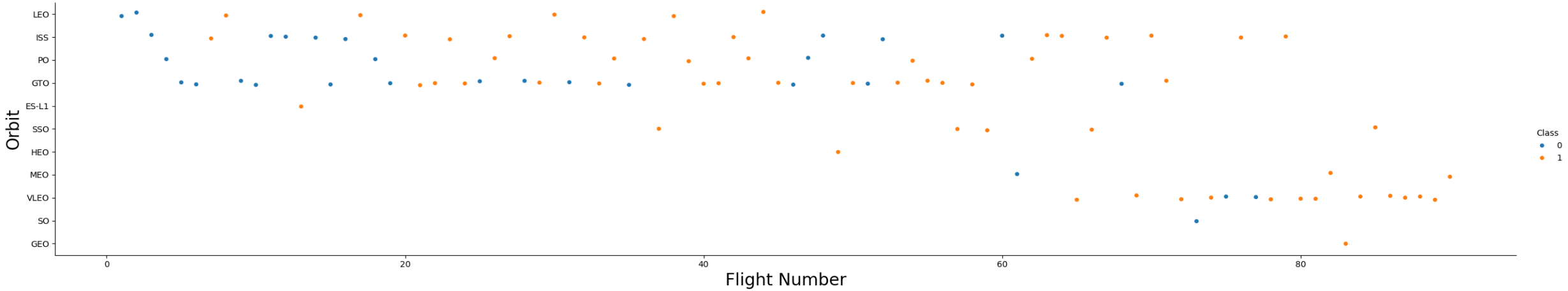
SKILLS NETWORK

Success rate vs. Orbit type



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

Flight No. VS Orbit graph shows - 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.



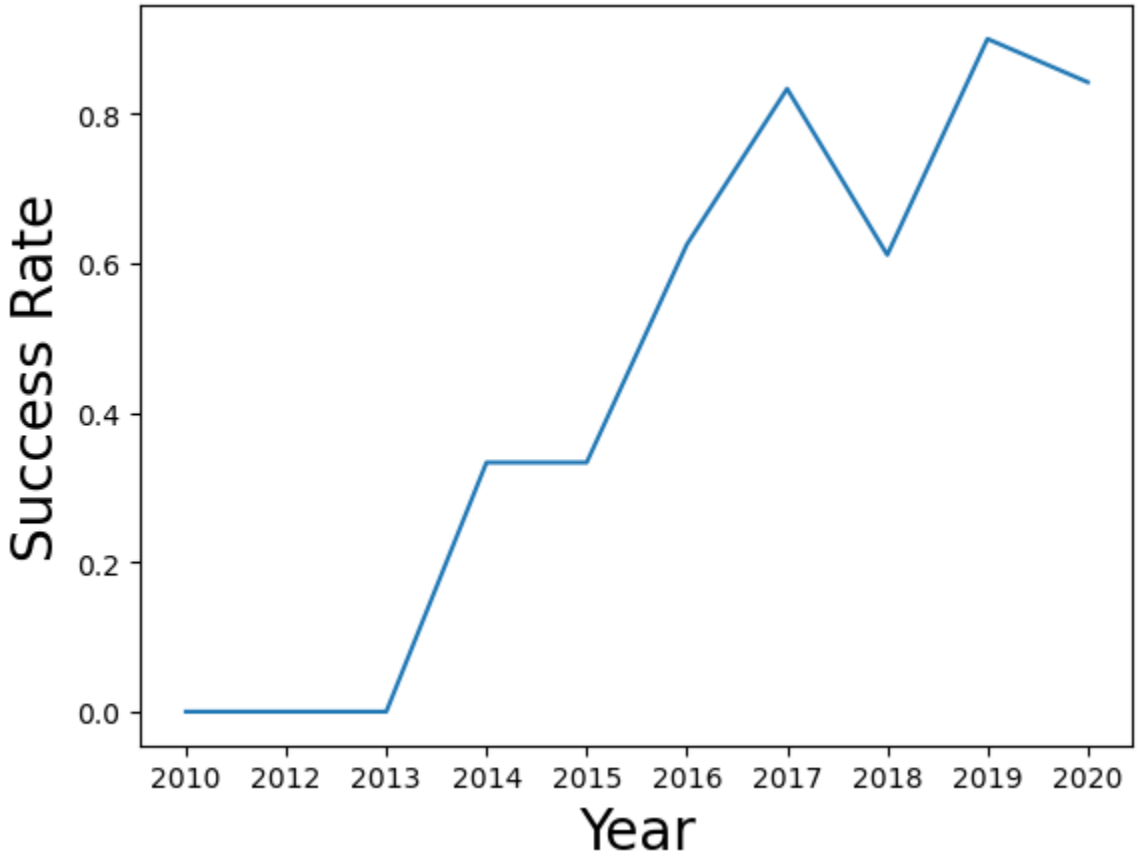
Payload Mass VS Orbit graph shows - Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

IBM Developer

SKILLS NETWORK

Launch success yearly trend

The success rate since 2013 kept **increasing** till 2020.



EDA WITH SQL

All launch site names

```
In [9]: %sql select distinct LAUNCH_SITE from SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

Out[9]:

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

Displaying the names of the unique launch sites in the space mission

Launch site names begin with `CCA`

```
In [10]: %sql select * from SPACEXTABLE where Launch_site like 'CCA%' limit 5
 * sqlite:///my_data1.db
Done.
```

Out[10]:

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

```
In [11]: %sql select sum(PAYLOAD_MASS__KG_) as sum from SPACEXTABLE where customer like 'NASA (CRS)'
 * sqlite:///my_data1.db
Done.
Out[11]:
```

sum
45596

Total payload mass

```
In [13]: %sql select min(Date) as Date from SPACEXTABLE where Mission_outcome like 'success%'
 * sqlite:///my_data1.db
Done.
Out[13]:
```

Date
2010-06-04

First successful ground landing date

```
In [12]: %sql select avg(payload_mass__kg_) as Average from SPACEXTABLE where booster_version like 'F9 v1.1%'
 * sqlite:///my_data1.db
Done.
Out[12]:
```

Average
2534.6666666666665

Average payload mass by F9 v1.1

In [14]:
```
%sql select Booster_Version from SPACEXTABLE where (mission_outcome like 'success%') and (payload_mass__kg_ between 4000 and
```

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

Out[14]:

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

Successful drone ship landing with payload between 4000 and 6000

In [16]:
```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXTABLE GROUP by mission_outcome ORDER BY mission_outcome
```

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

Out[16]:

Mission_Outcome	Count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Total number of successful and failure mission outcomes

In [17]:
```
%sql select booster_version from SPACEXTABLE where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTABLE);
```

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

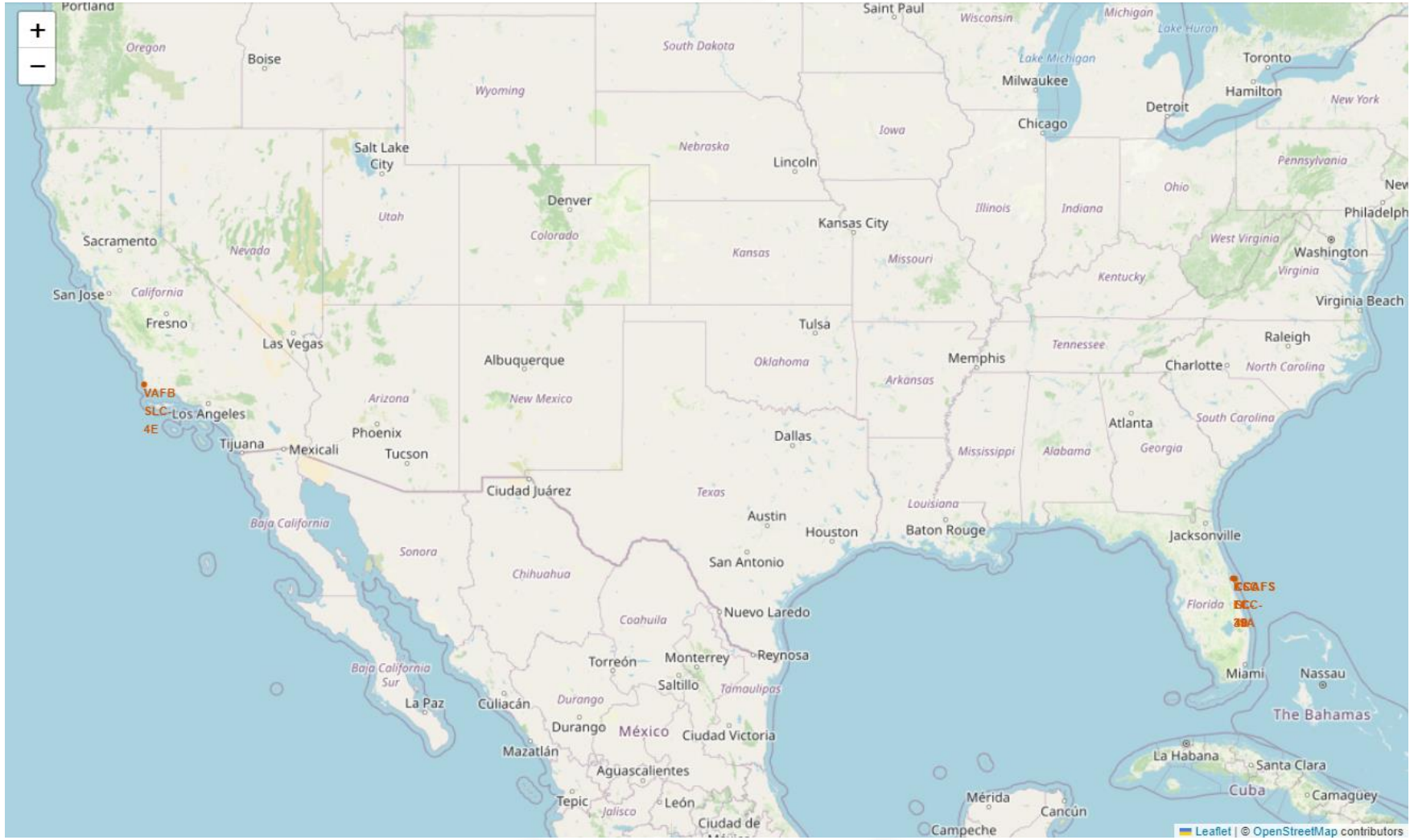
Out[17]:

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

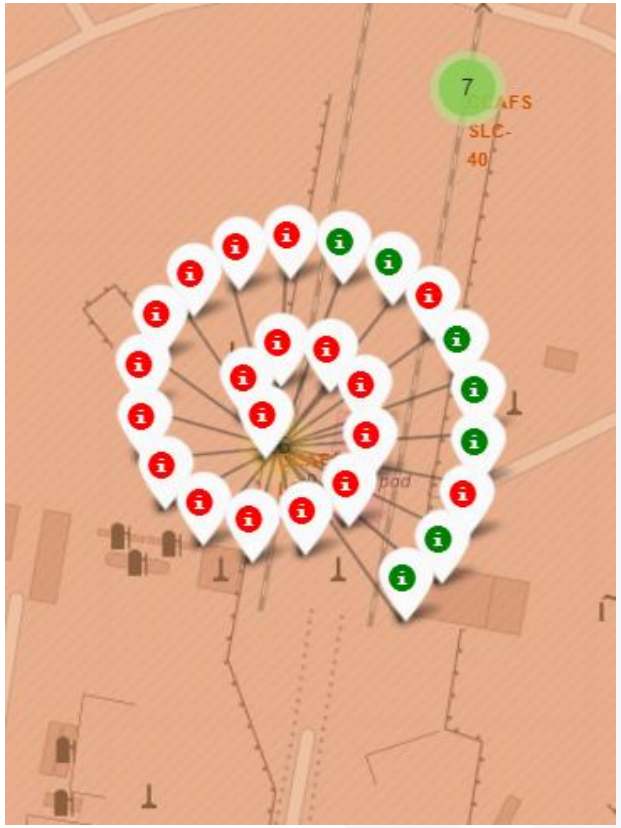
Boosters carried maximum payload

ITERACTIVE MAP WITH FOLIUM

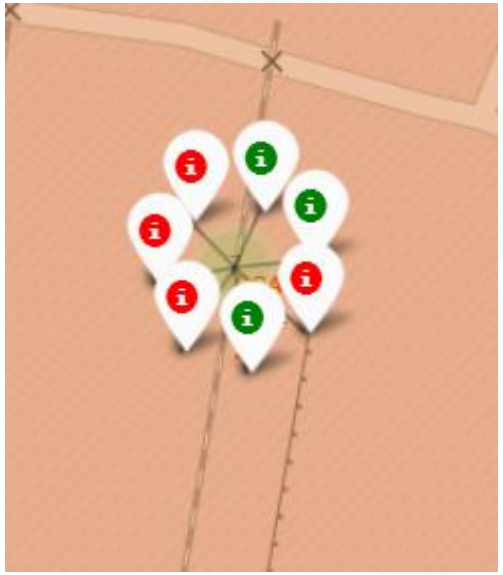
All launch sites' location markers on a global map



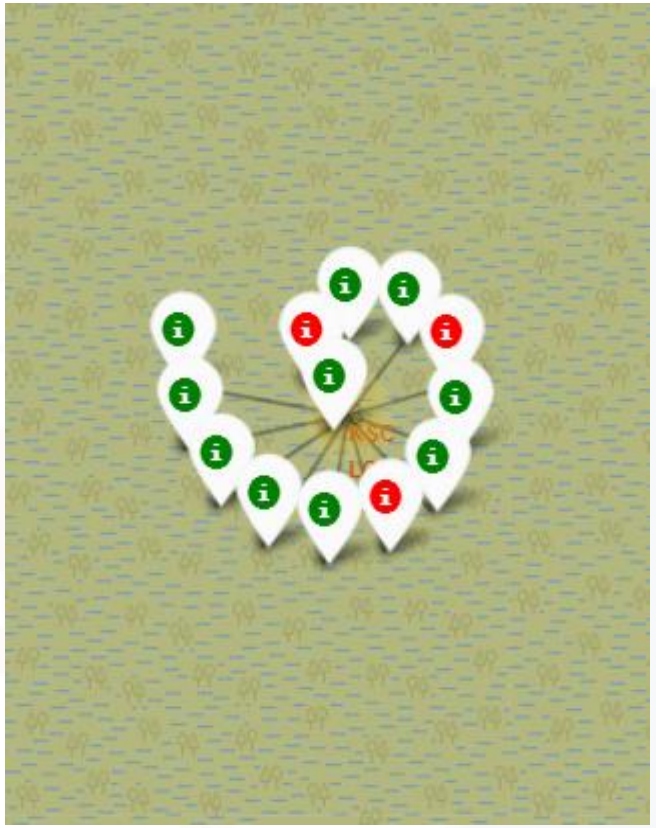
Colour-labeled launch records on the map



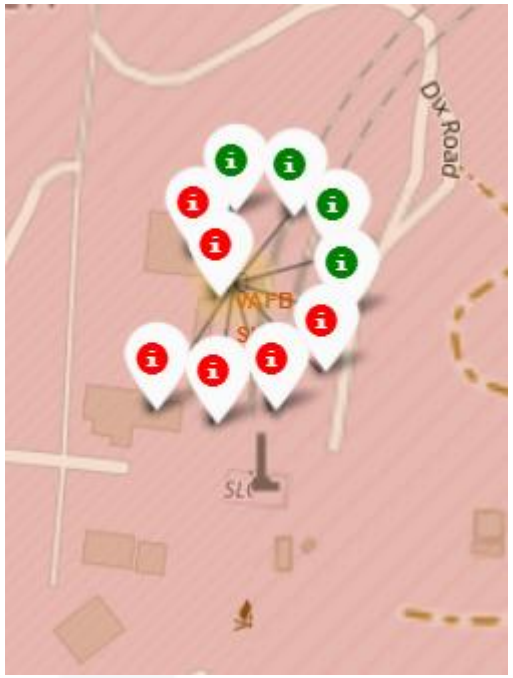
CCAFS LC-40



CCAFS SLC-40



KSC LC-39A

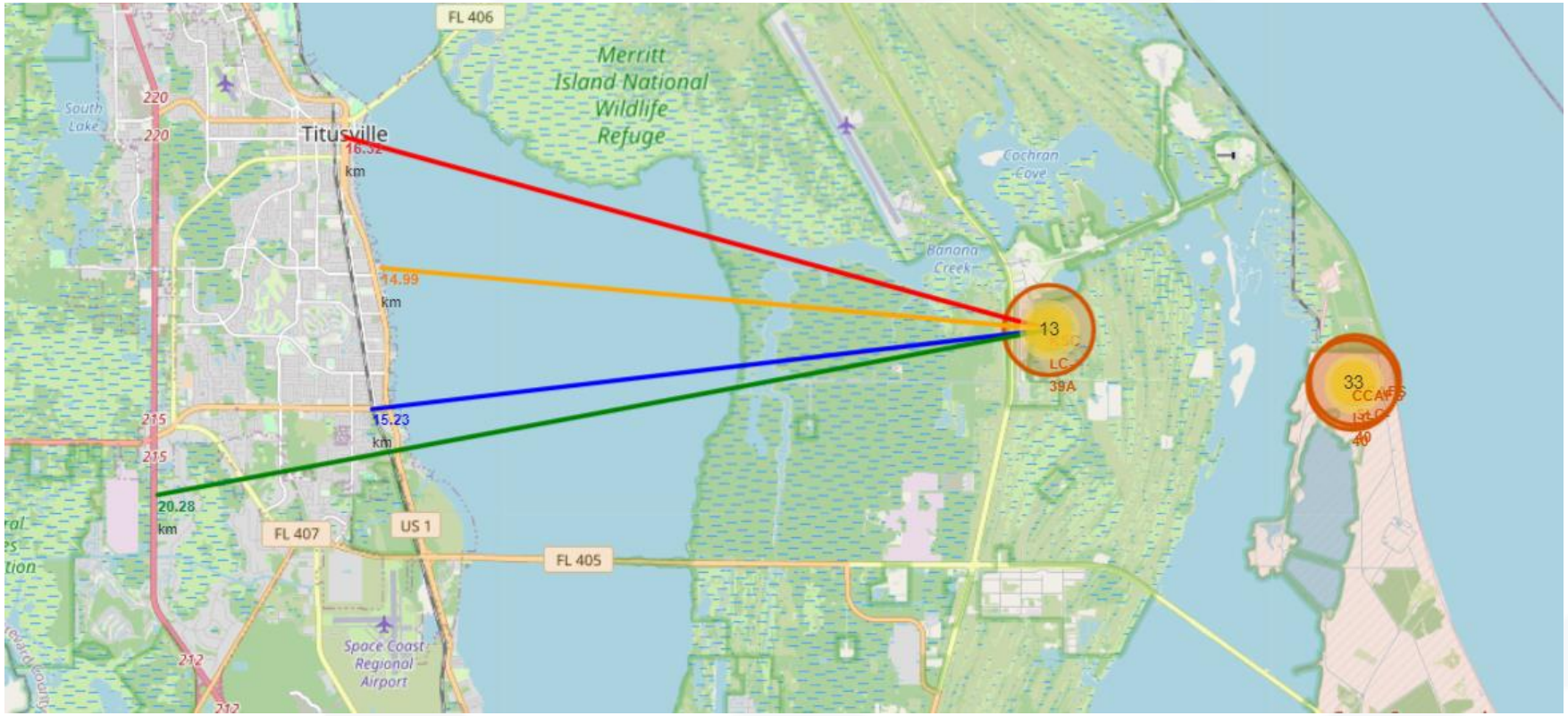


VAFB SLC-4E

Green Marker = Successful Launch

Red Marker = Failed Launch

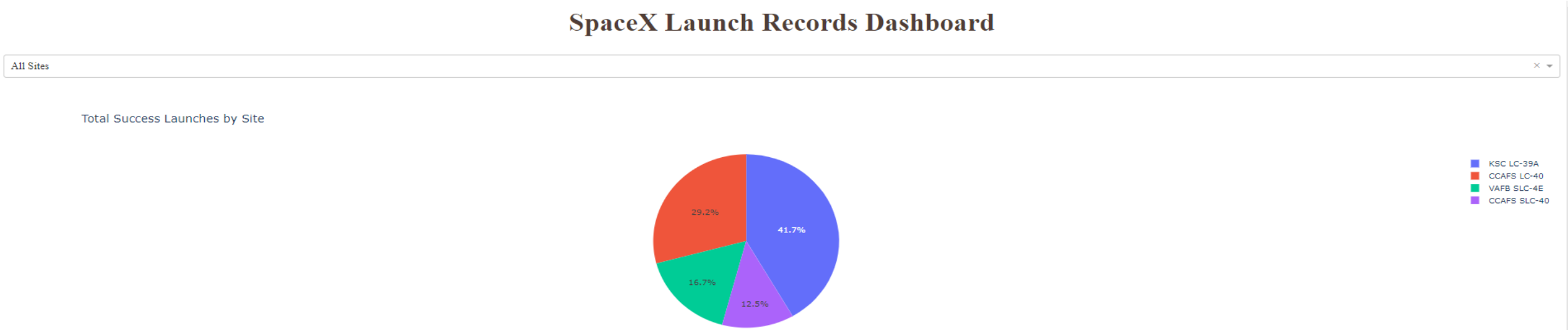
Distance from the launch site KSC LC-39A to its proximities



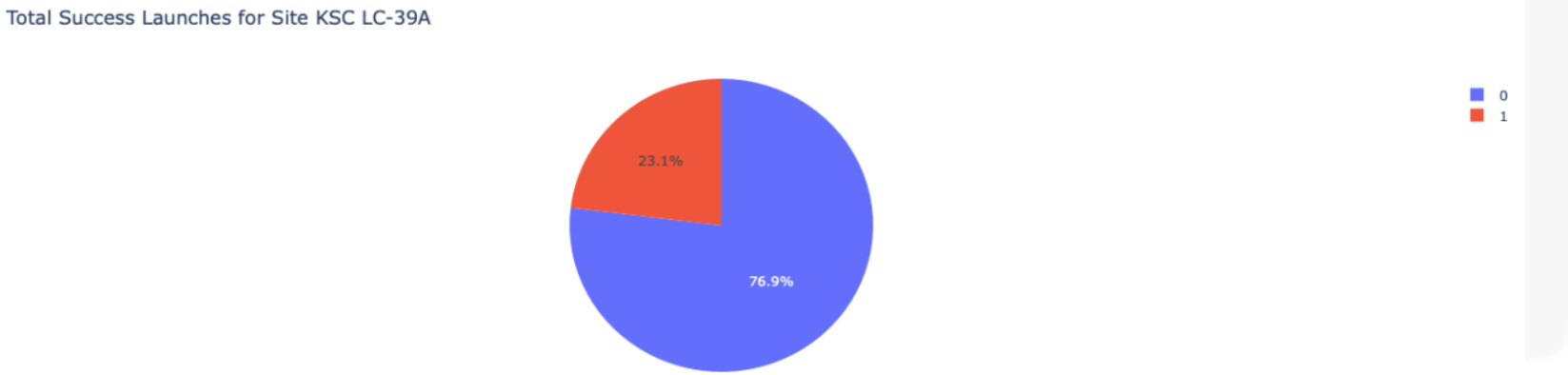
- Coastline (14.99 km)
- Railway (15.23 km)
- Closest city Titusville (16.32 km)
- Highway (20.28 km)

DASHBOARD WITH PLOTLY DASH

Launch success count for all sites



Launch site with highest launch success ratio



Payload Mass vs. Launch Outcome for all sites

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

PREDICTIVE ANALYSIS

Classification Accuracy

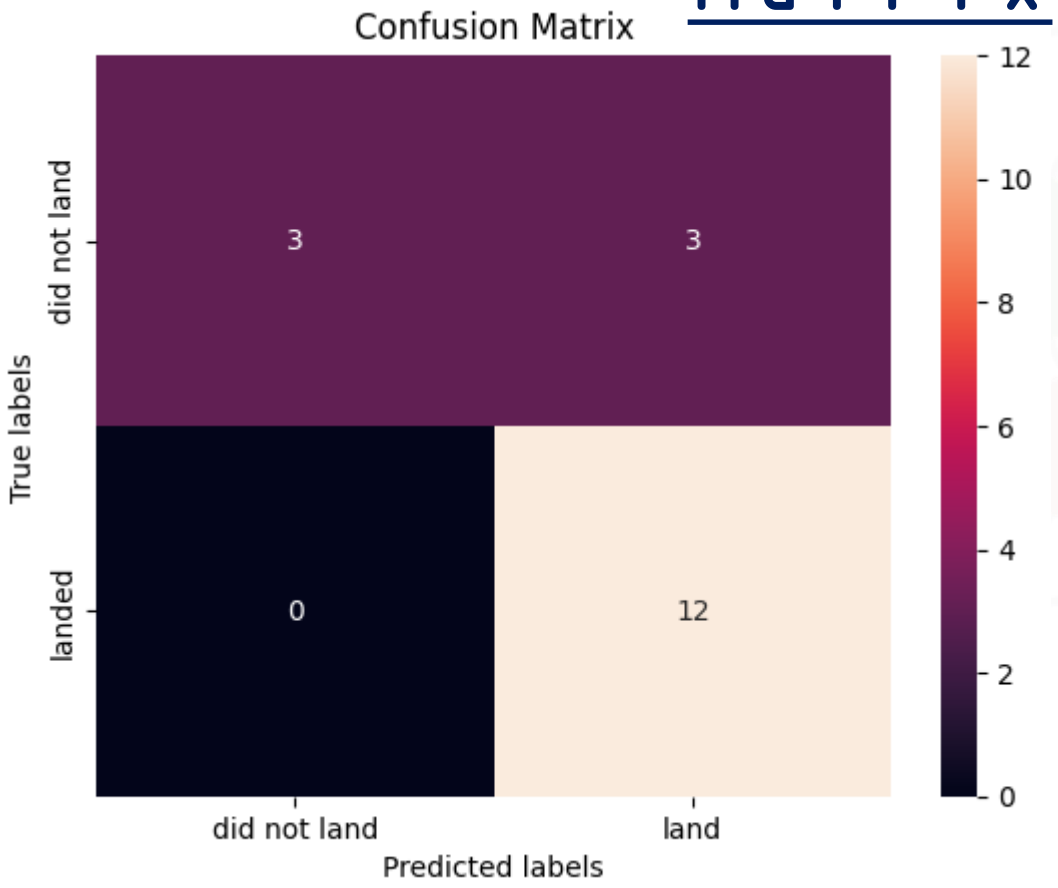
Find the method performs best:

In [32]:
```
print(methods)
print(accu)
```

```
['logistic regression', 'support vector machine', 'decision tree classifier', 'k nearest neighbors']
[0.8333333333333334, 0.8333333333333334, 0.8333333333333334, 0.8333333333333334]
```

- All models had virtually the same accuracy on the test set at 83.33% accuracy.

Confusion Matrix

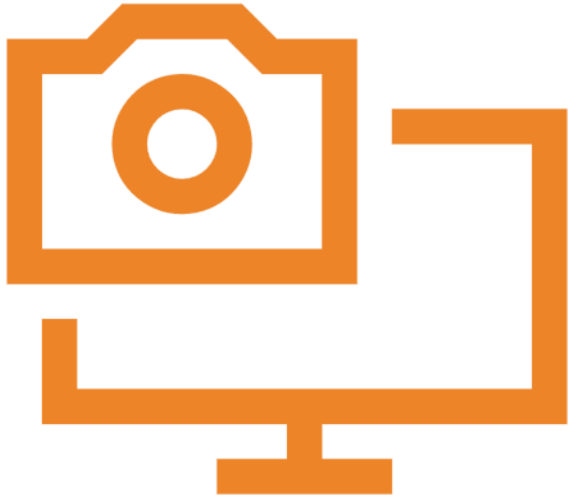


- All models performed the same on the test set.
- The confusion matrix is identical for all models.
- Models predicted 12 successful landings correctly.
- Models predicted 3 unsuccessful landings correctly.
- Models incorrectly predicted 3 successful landings (false positives), showing an over-prediction of successful landings.

CONCLUSION

- Develop a machine learning model for Space Y to compete with SpaceX.
- Predict successful Stage 1 landings to save ~$100 million USD.
- Use data from a public SpaceX API and web scraping the SpaceX Wikipedia page.
- Create data labels and store data in a DB2 SQL database.
- Develop a dashboard for data visualization.
- Achieve an accuracy of 83% with the machine learning model.
- Allon Mask of SpaceY can use this model to predict the success of Stage 1 landings before launch, aiding in launch decision-making.
- Collect more data to determine the best machine learning model and improve accuracy further.

THANK YOU



TO

- IBM
- COURSERA
- IBM INSTRUCTORS