IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

John A. Coleman
November 2022

Outline

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

Executive Summary

- In this capstone project, we predict if the SpaceX Falcon 9 first stage will land successfully using several machine learning classification algorithms and prediction as well as various interactive visualization approaches.

- 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

  Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Common problems that needed to be studied and solved:
  - What factors and influencers determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing
  - What operating conditions needs to be in place to ensure a successful landing program

Section 1

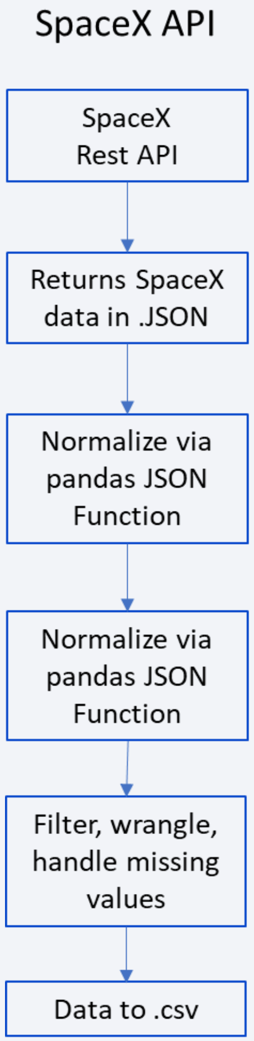
Methodology

Methodology

- The overall methodology includes:
  1. Data collection, wrangling, and formatting, using:
     - SpaceX API
     - Web scraping
  2. Exploratory data analysis (EDA), using:
     - Pandas and NumPy
     - SQL
  3. Data visualization, using:
     - Matplotlib and Seaborn
     - Folium
     - Dash
  4. Machine learning prediction, using
     - Logistic regression
     - Support vector machine (SVM)
     - Decision tree
     - K-nearest neighbors (KNN)
     - DT models evaluated for the best classifier

Data Collection via the SpaceX API

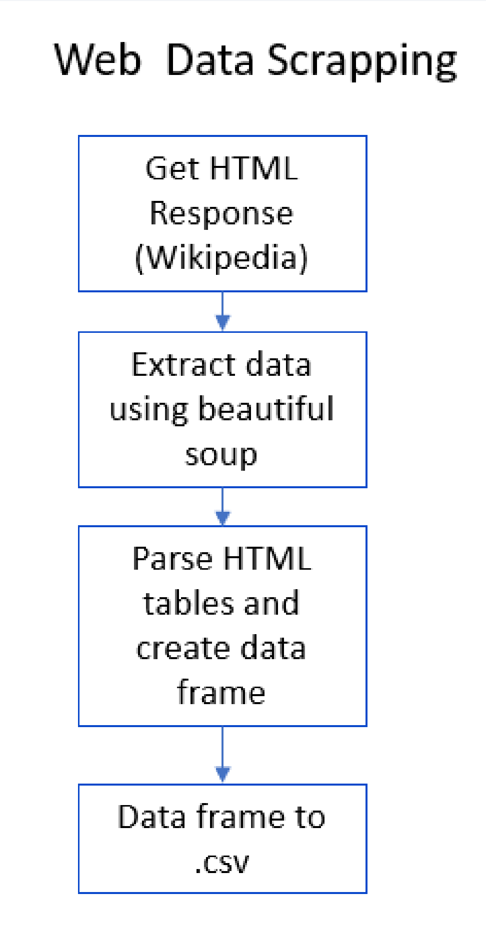
- SpaceX Launch data which was acquired by using the Spacex REST API https://api.spacexdata.com/v4/rockets/.
- Falcon 9 launch data using Beautiful soup to parse html Wikipedia data
- The API provides data about many types of rocket launches done by SpaceX, the data is therefore filtered to include only Falcon 9 launches.
- Every missing value in the data is replaced the mean the column that the missing value belongs to.
- We end up with 90 rows or instances and 17 columns or features.



git DataScienceCapstone/1_Data Collection API Lab.ipynb at main · jac-devop/DataScienceCapstone (github.com)

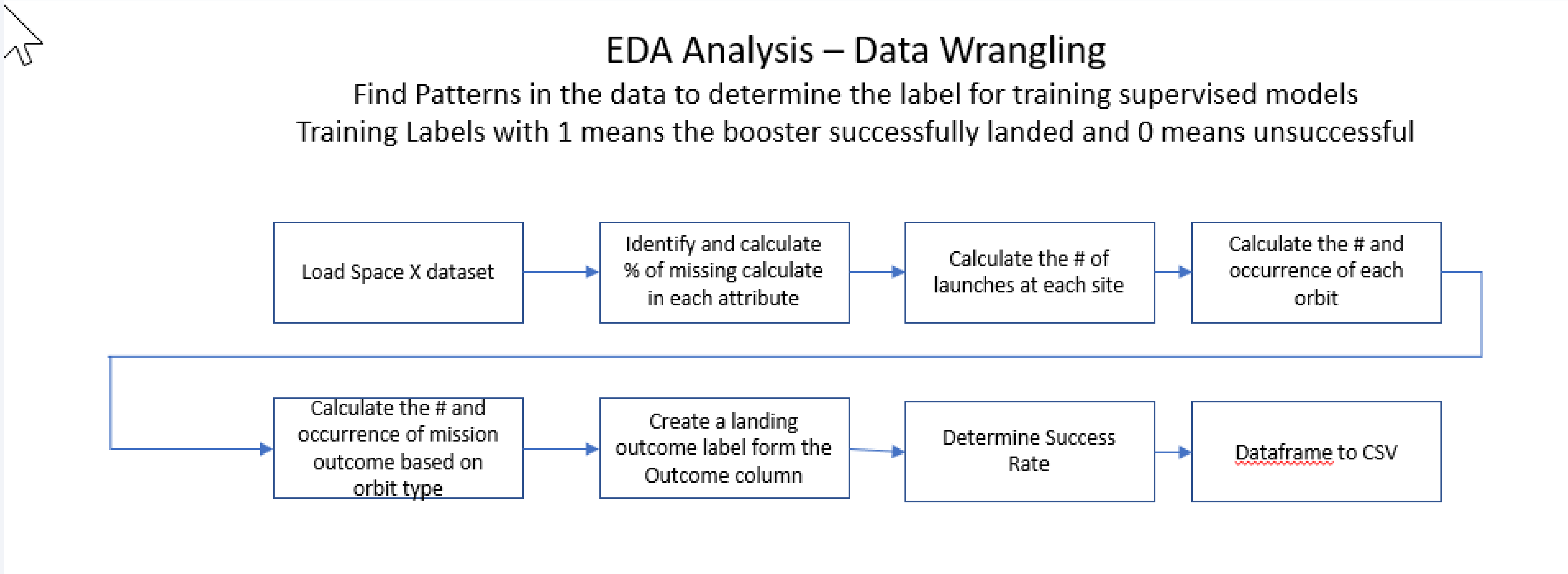
Data Collection – Web Scraping

- Web scraping was used to collect Falcon 9 historical launch records from the Wikipedia page titled “List of Falcon 9 and Falcon Heavy launches”



git [DataScienceCapstone/2_Data Collection Web Scrapingwikipedia.ipynb at main · jac-devop/DataScienceCapstone (github.com)]

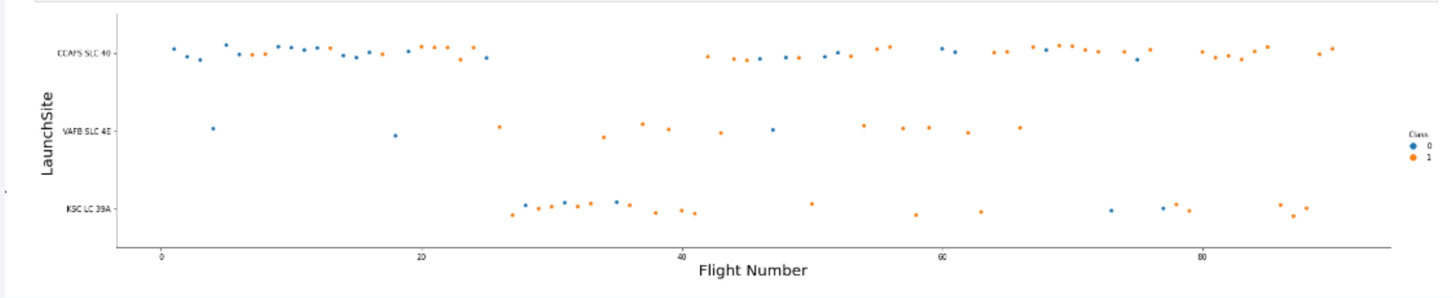
Data Wrangling

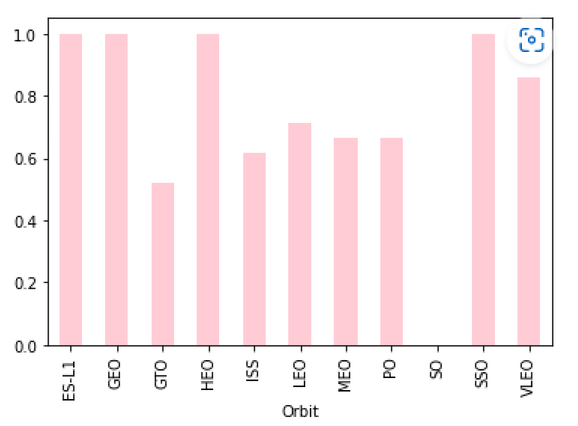


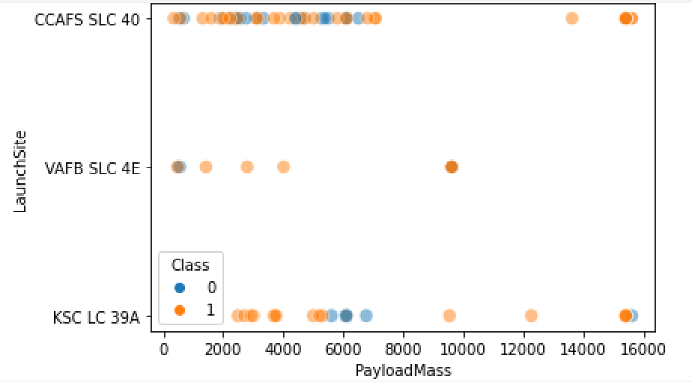
- We performed exploratory data analysis and determine the training labels
- We calculated the number of launches at each site, and the number and occurrence of each orbit
- We created landing outcome labels from the outcome column and exported the results to a .CSV

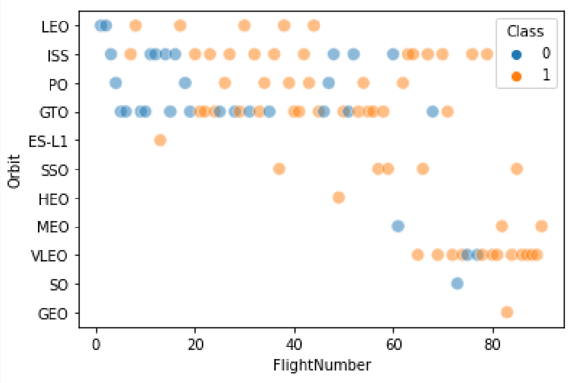
git [DataScienceCapstone/3_Data wrangling Lab.ipynb at main · jac-devop/DataScienceCapstone (github.com)]

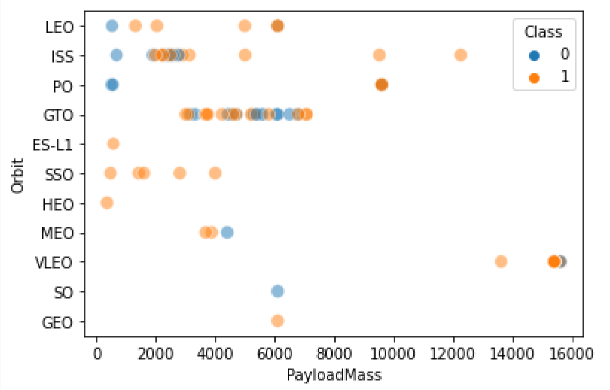
EDA with Data Visualization

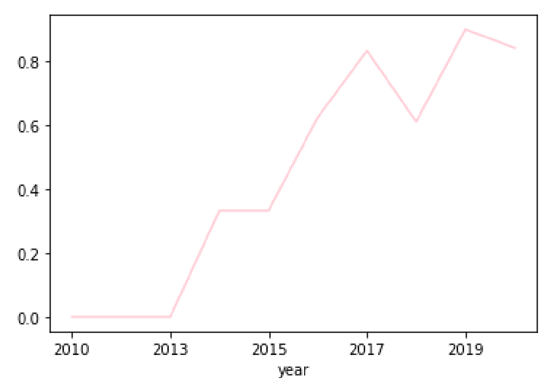












- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type and the launch success yearly trend

[EDA with Data Visualization - IBM Cloud Pak for Data]

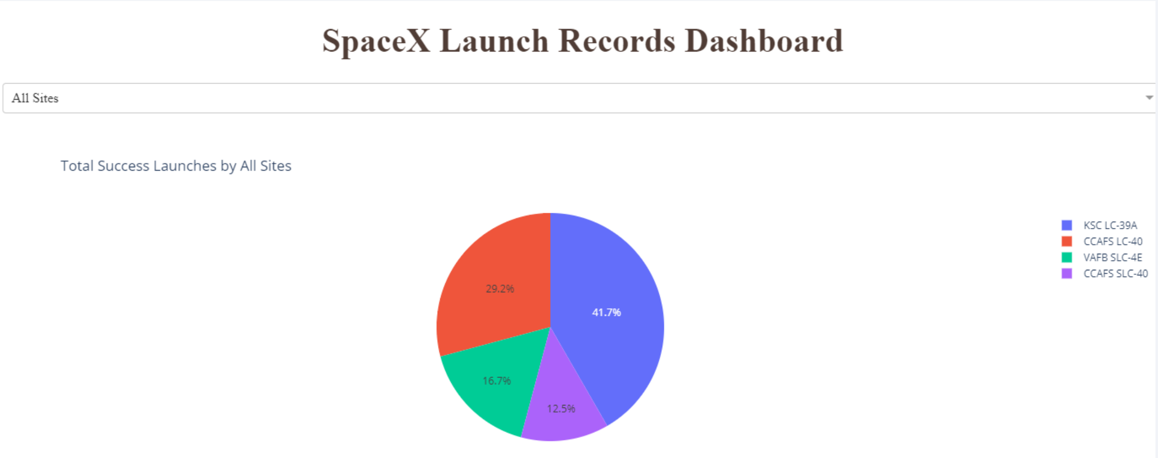
Build an Interactive Map with Folium

- Launch sites were marked [folium.Marker()] and circled with a text label [folium.Circle()] on a map [folium.Map] using the sites latitude and longitude coordinates. The reason for the marks was to visualize the geographical location rather then just referencing the Lat and Long coordinates which are not intuitive.
- Each site on the map was then marked to indicate success or failure of the launches which used the class column that indicated success or failure. Marker clusters [markerCluster()] where also used to simplify the map as many markers can have the same coordinates
- Distance between launch sites to its proximities where calculated. Folium.Maker() was used to show the distance and folium.PolyLine was used between the locations.

DataScienceCapstone/6_Interactive Visual Analytics with Folium lab.ipynb at main · jac-devop/DataScienceCapstone (github.com)

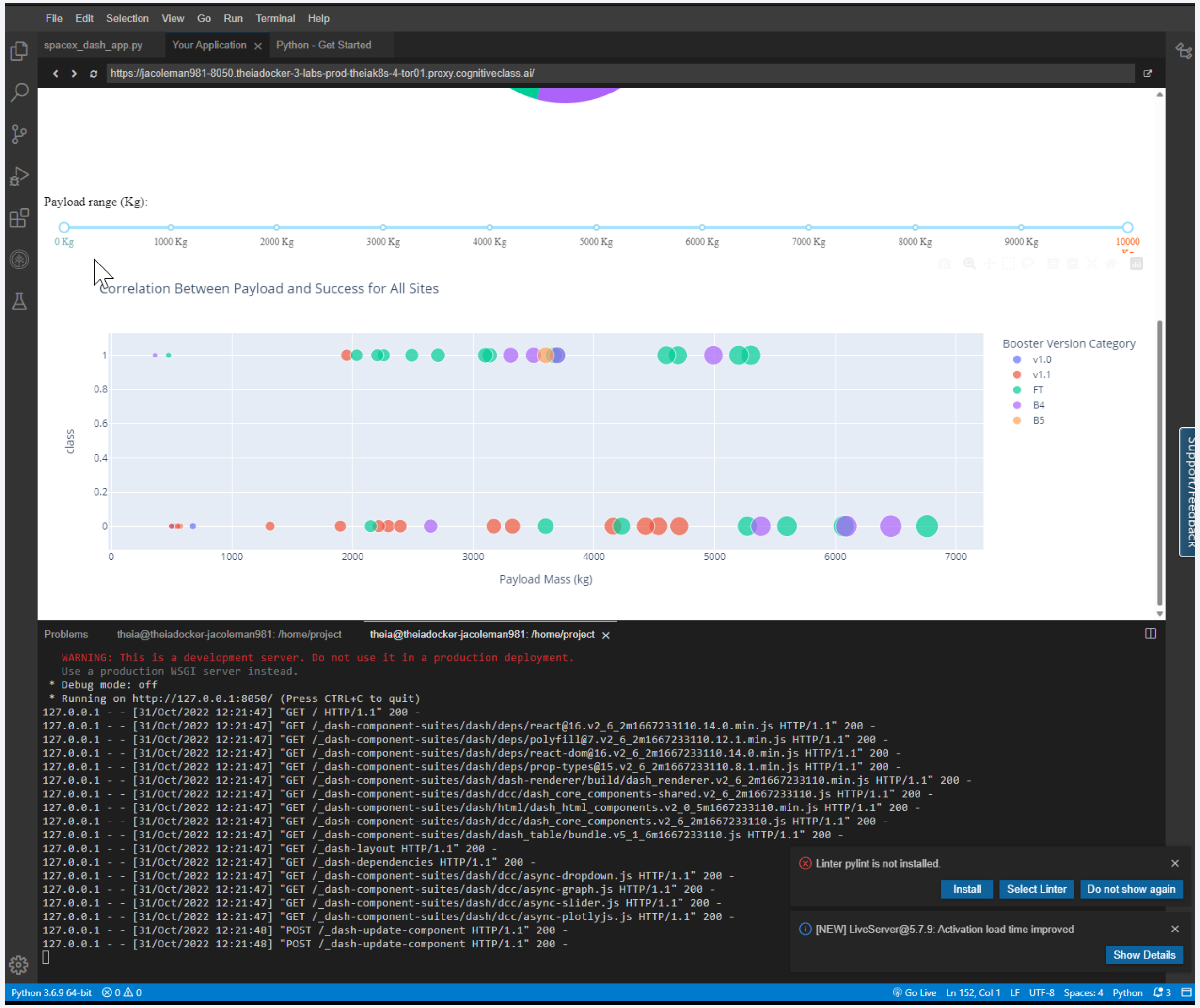
Build a Dashboard with Plotly Dash

- Dash components and ploty components were imported to build out the specified dashboard.
- HTML components were used to add a dropdown list to enable launch site selection
- An HTML graph (dcc.Graph) was used to create a ‘Success Pie Chart”. Chart shows the total of successful launches count for all sites unless a specific launch was selected which then showed the Success vs. Failed counts for the site
- Callback functions were used for the ‘site-dropdown’ as input and ‘success-pie-chart’ as output.
- Additional callback functions were added for ‘site-dropdown’ and ‘payload-slider as inputs and ‘success-payload-scatter-chart’ as output.



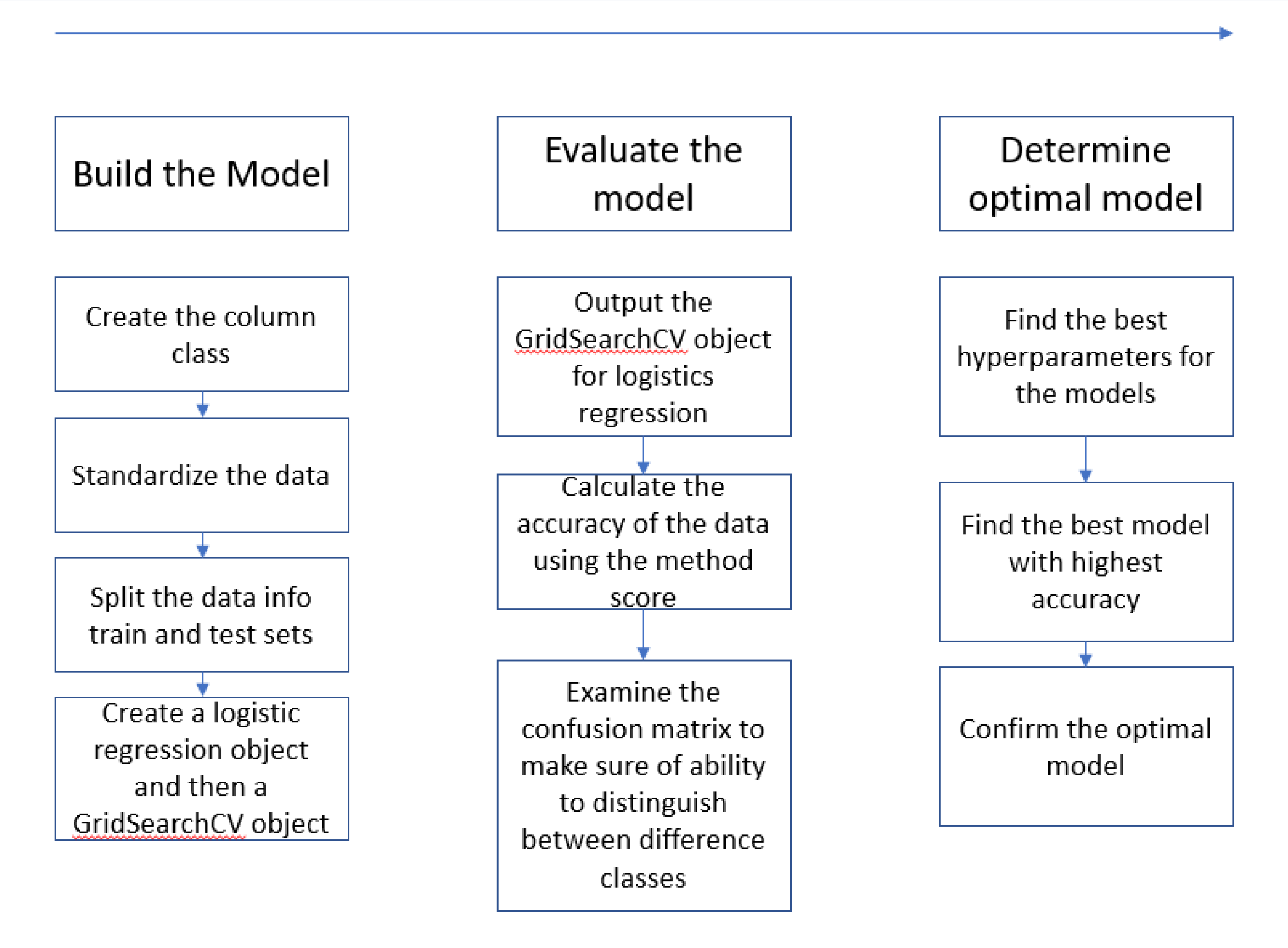
[DataScienceCapstone/7_spacex_dash_app.py at main · jac-devop/DataScienceCapstone (github.com)](github.com)

Build a Dashboard with Plotly Dash



DataScienceCapstone/7_spacex_dash_app.py at main · jac-devop/DataScienceCapstone (github.com)

Predictive Analysis (Classification)

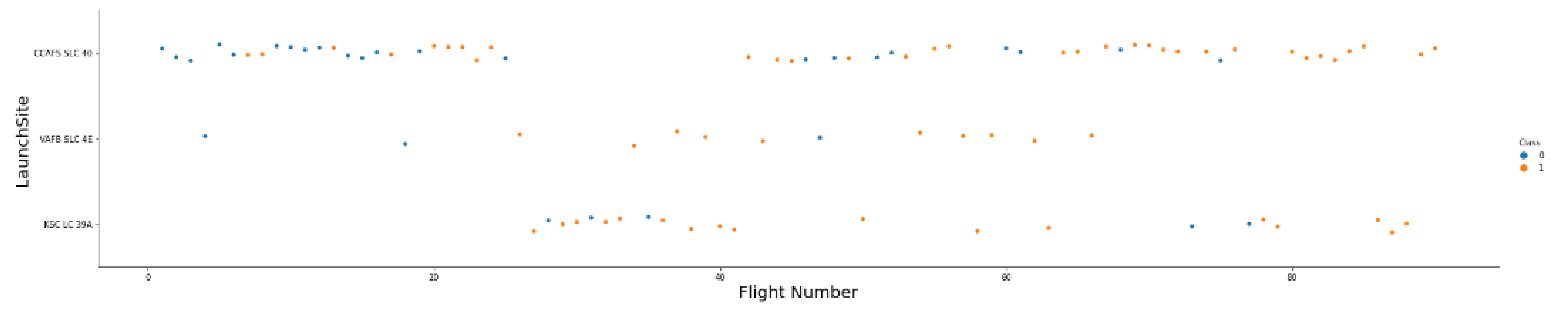


DataScienceCapstone/8_Machine Learning Prediction lab.ipynb at main · jac-devop/DataScienceCapstone (github.com)

Section 2

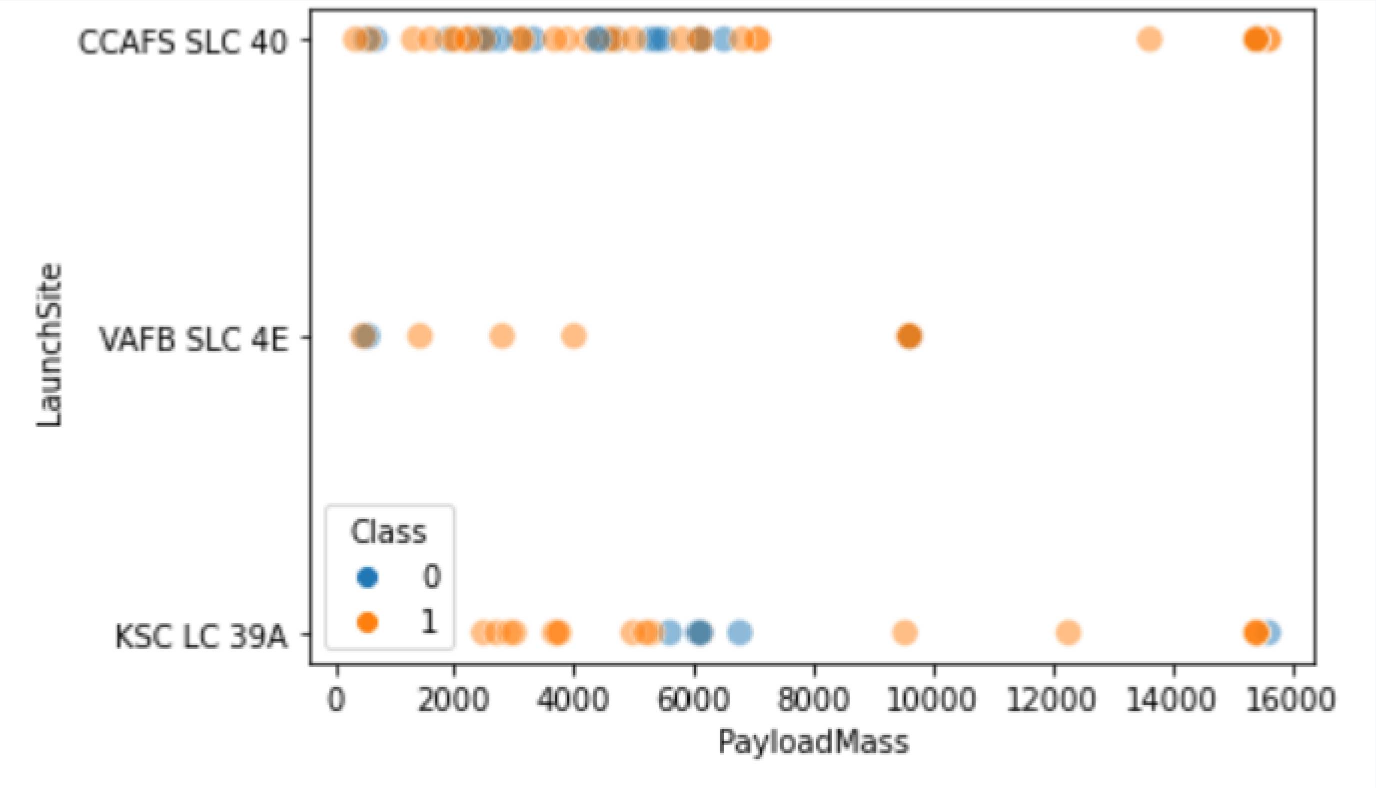
Insights drawn from EDA

Flight Number vs. Launch Site



Launches from the site of CCAFS SLC 40 are more numerous than other launch sites

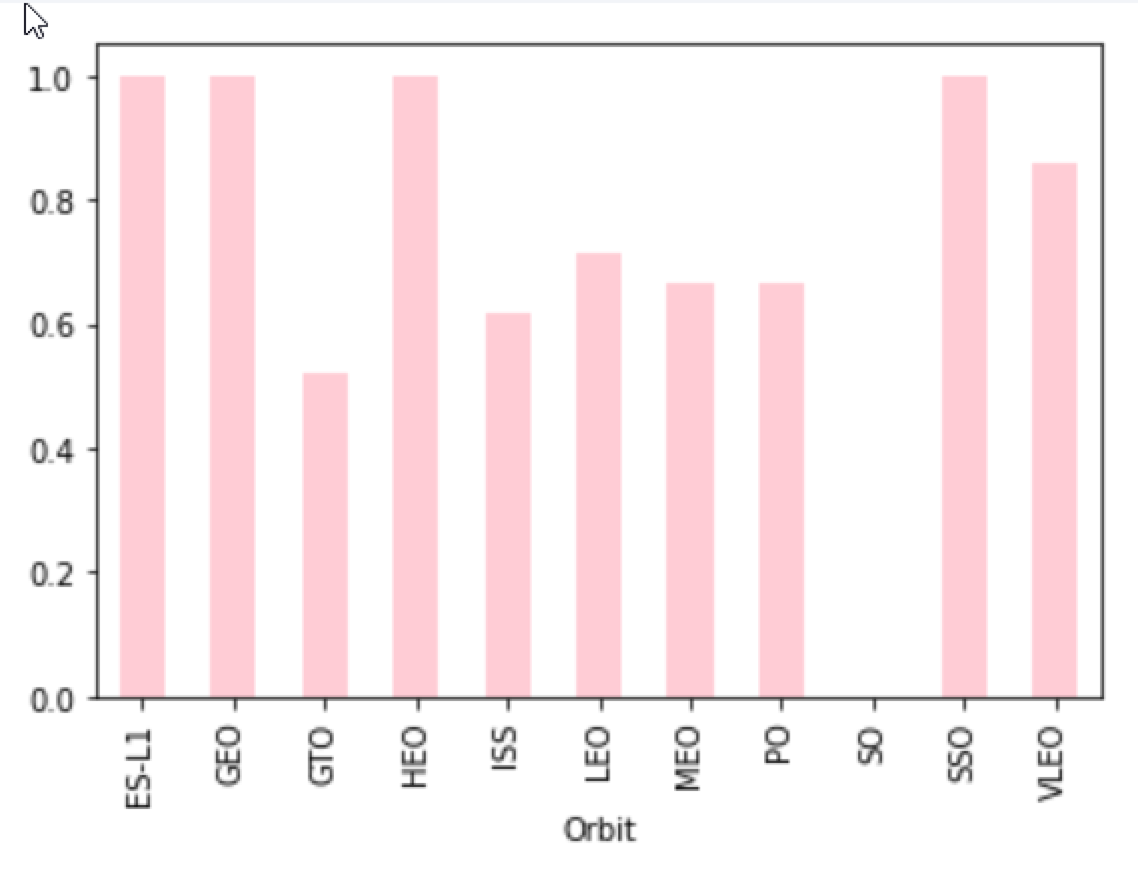
Payload vs. Launch Site



- Payload Vs. Launch Site scatter point chart shows for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

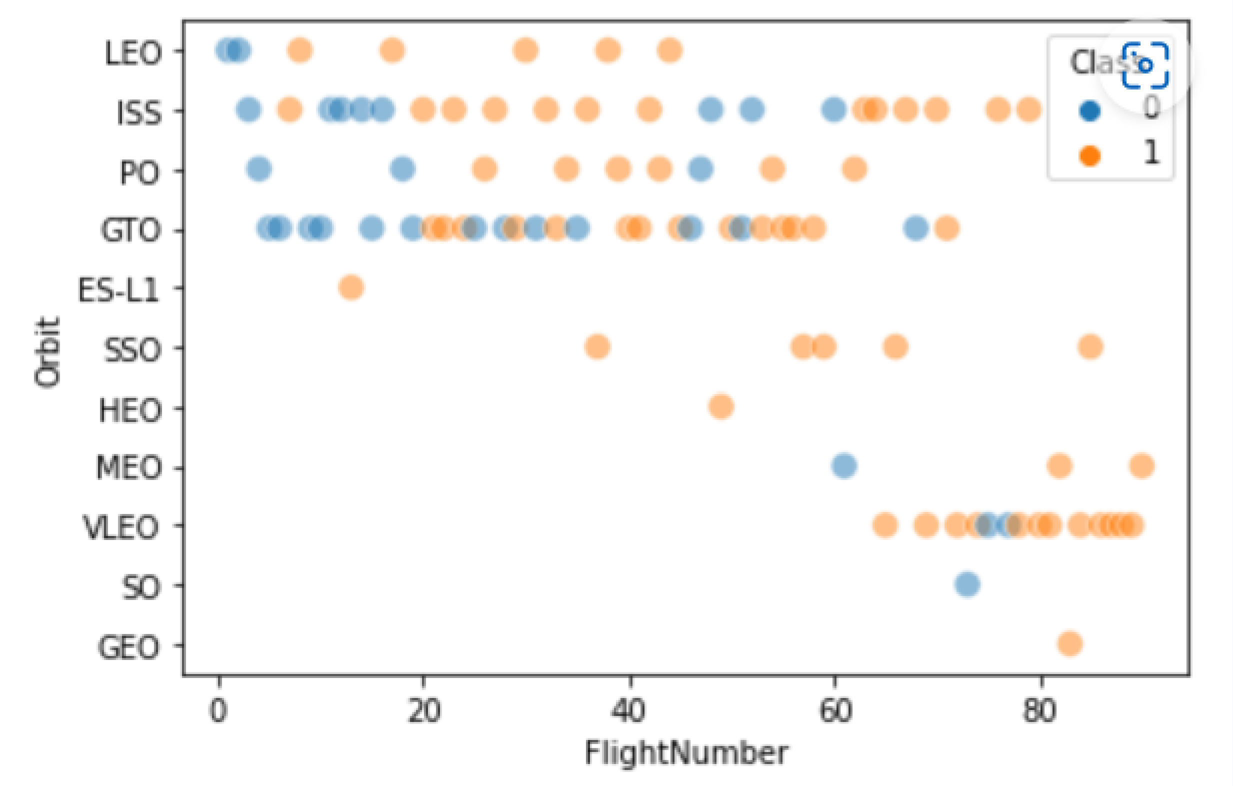
- Payloads with lower mass have been launched from CCAFS SLS40

Success Rate vs. Orbit Type



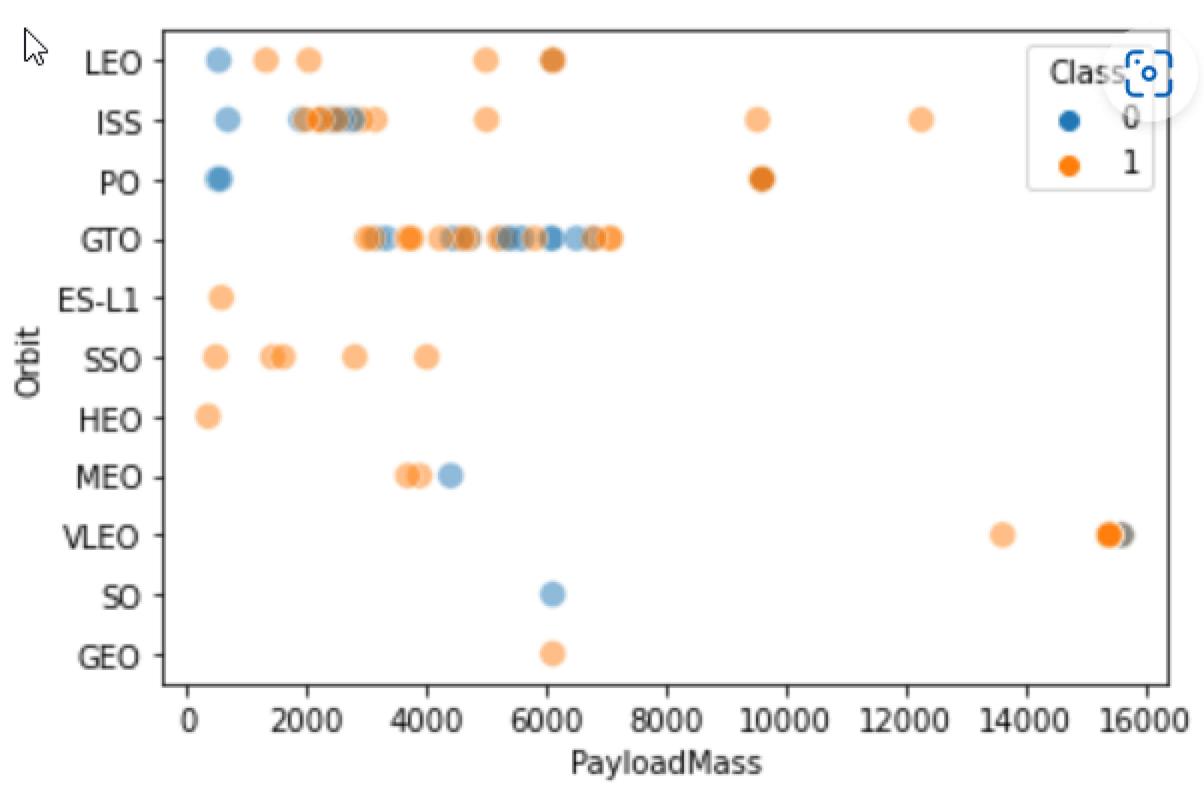
- The orbit types of ES-L1, GEO, HEO, SSO are among the highest success rate

Flight Number vs. Orbit Type



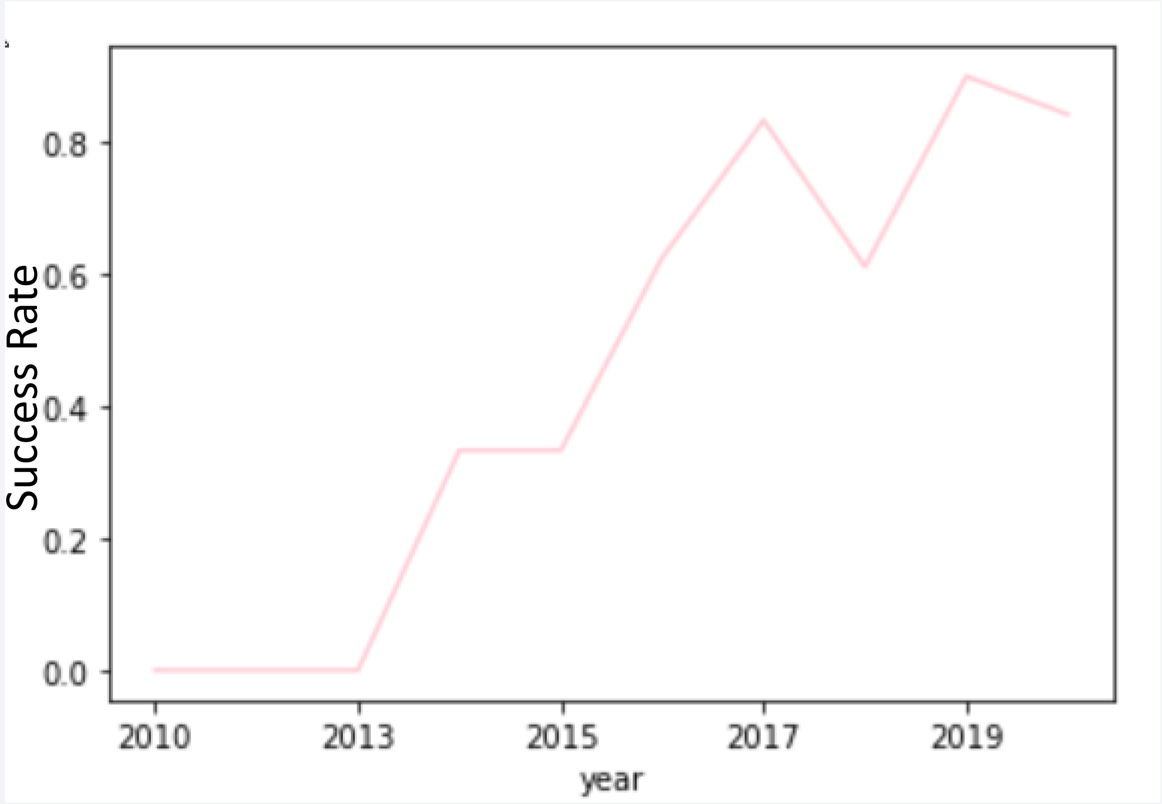
- The LEO orbit success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit
- An observed trend is there is a shift to VLEO launches in recent years

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Pola, LEO and ISS
- GTO is hard to distinguish as both positive and negative landing rates are both indicated

Launch Success Yearly Trend



- The success rate since 2013 kept increasing until 2020. Likely due to learning from failures and technology innovation

All Launch Site Names

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

In [35]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM QGX36240.SPACEX;
```

```
 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[35]:

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

- A selection of all the Distinct launch sites

git [DataScienceCapstone/4_EDA-sql-coursera (1).ipynb at main · jac-devop/DataScienceCapstone (github.com)]

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

In [36]:
```
%sql SELECT * FROM QGX36240.SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[36]:

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

- A filter on launch sites that have a name begining with ‘CCA’

git [DataScienceCapstone/4_EDA-sql-coursera (1).ipynb at main · jac-devop/DataScienceCapstone (github.com)]

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

In [37]:
```
%sql SELECT SUM(payload_mass__kg_) FROM QGX36240.SPACEX WHERE customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[37]:

1
45596

- Customer = ‘NASA (CRS) clause returns the sum of payload_mass_kg_ from the database.table (QGX36240.SPACEX)
- Result of the SQL query is shown above

[DataScienceCapstone/4_EDA-sql-coursera (1).ipynb at main · jac-devop/DataScienceCapstone (github.com)]

Average Payload Mass by F9 v1.1

```
In [14]: %sql SELECT AVG(payload_mass__kg_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM QGX36240.SPACEX WHERE BOOSTER_VERSION = 'F9 v1.1';

 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.

Out[14]:
```

Average Payload Mass by Booster Version F9 v1.1
2928

- An average can be obtained for any numeric table value using AVG

First Successful Ground Landing Date

```
In [26]: %sql SELECT MIN(DATE) AS "First Succesful Landing Outcome Ground Pad" FROM QGX36240.SPACEX WHERE landing__outcome = 'Success (ground pad)';

 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[26]:

First Succesful Landing Outcome Ground Pad
2015-12-22

- MIN function used for this query to determine First Successful Outcome with a filter of Success (ground pad)

git [DataScienceCapstone/4_EDA-sql-coursera (1).ipynb at main · jac-devop/DataScienceCapstone (github.com)]

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [40]:

```
%sql select booster_version from QGX36240.SPACEX where landing__outcome = 'Success (drone ship)'\
and payload_mass__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[40]:

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

git [DataScienceCapstone/4_EDA-sql-coursera (1).ipynb at main · jac-devop/DataScienceCapstone (github.com)]

Total Number of Successful and Failure Mission Outcomes

```
In [27]: %sql SELECT COUNT(mission_outcome) AS "Count of All Missions" FROM QGX36240.SPACEX ;

 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[27]:

Count of All Missions
101

```
In [28]: %sql SELECT COUNT(mission_outcome) As "Count of Successful Missions" FROM QGX36240.SPACEX where mission_outcome LIKE 'Success%';

 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[28]:

Count of Successful Missions
100

```
In [29]: %sql SELECT COUNT(mission_outcome) As "Count of Failed Missions" FROM QGX36240.SPACEX where mission_outcome LIKE 'Failure%';

 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[29]:

Count of Failed Missions
1

git [DataScienceCapstone/4_EDA-sql-coursera (1).ipynb at main · jac-devop/DataScienceCapstone (github.com)]

Boosters Carried Maximum Payload

```
In [34]: %sql SELECT DISTINCT booster_version as "Booster Versions which carried the Maximum Payload Mass" FROM QGX36240.SPACEX WHERE payload_mass__kg_  =(SELECT MAX(payload_mass__kg_) FROM QGX36240.SPACEX);

 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.

Out[34]:
```

Booster Versions which carried the Maximum Payload Mass
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

git [DataScienceCapstone/4_EDA-sql-coursera (1).ipynb at main · jac-devop/DataScienceCapstone (github.com)]

2015 Launch Records

[DataScienceCapstone/4_EDA-sql-coursera (1).ipynb at main · jac-devop/DataScienceCapstone (github.com)](https://github.com)

```
[35]: %sql select month(DATE) as Month, booster_version, launch_site from QGX36240.SPACEX where landing__outcome = 'Failure (drone ship)' and year(DATE) = '2015' ;
 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[35]:

MONTH	booster_version	launch_site
1	F9 v1.1 B1012	CCAFS LC-40
4	F9 v1.1 B1015	CCAFS LC-40

We can get the months by using month(DATE) and in the WHERE function we assigned the year value to '2015'. This query shows 'Failure (drone ship)

```
In [40]: %sql select * from  QGX36240.SPACEX where year(DATE) = '2015';
 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[40]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2015-01-10	09:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-02-11	23:03:00	F9 v1.1 B1013	CCAFS LC-40	DSCOVR	570	HEO	U.S. Air Force NASA NOAA	Success	Controlled (ocean)
2015-03-02	03:50:00	F9 v1.1 B1014	CCAFS LC-40	ABS-3A Eutelsat 115 West B	4159	GTO	ABS Eutelsat	Success	No attempt
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-04-27	23:03:00	F9 v1.1 B1016	CCAFS LC-40	Turkmen 52 / MonacoSAT	4707	GTO	Turkmenistan National Space Agency	Success	No attempt
2015-06-28	14:21:00	F9 v1.1 B1018	CCAFS LC-40	SpaceX CRS-7	1952	LEO (ISS)	NASA (CRS)	Failure (in flight)	Precluded (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

We can get the months by using month(DATE) and in the WHERE function we assigned the year value to '2015'. This query shows all 2015 records

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

```
In [43]: %sql select landing__outcome as "Landing Outcome", count(landing__outcome) as "Total Count"  from QGX36240.SPACEX \
where DATE between '2010-06-04' and '2017-03-20' group by landing__outcome\
order by count(landing__outcome) desc

 * ibm_db_sa://qgx36240:***@19af6446-6171-4641-8aba-9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30699/bludb
Done.
```

Out[43]:

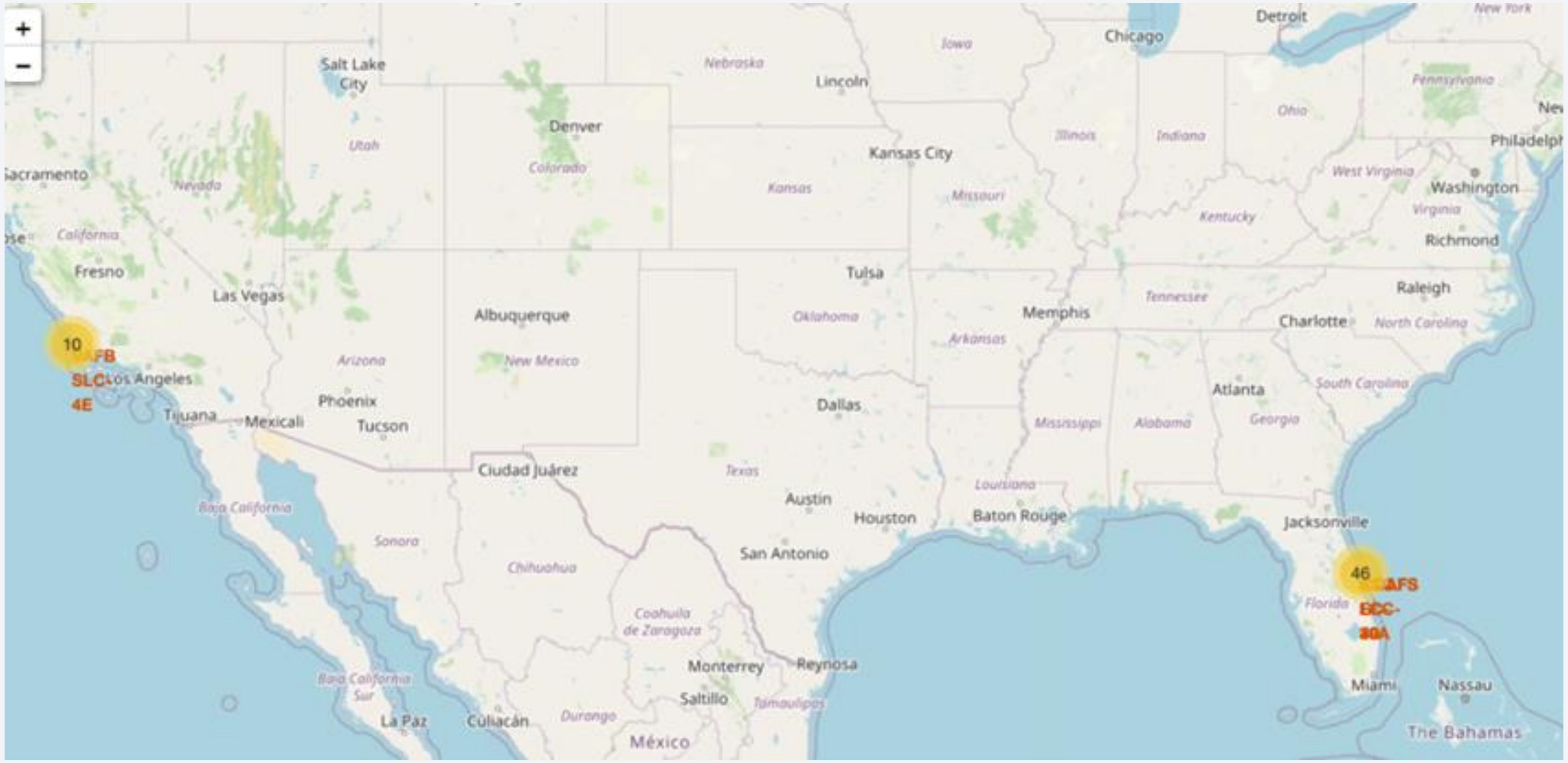
Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

By using “ORDER” we can order the values in descending order, and with “COUNT” we can count all records meeting the criteria

Section 3

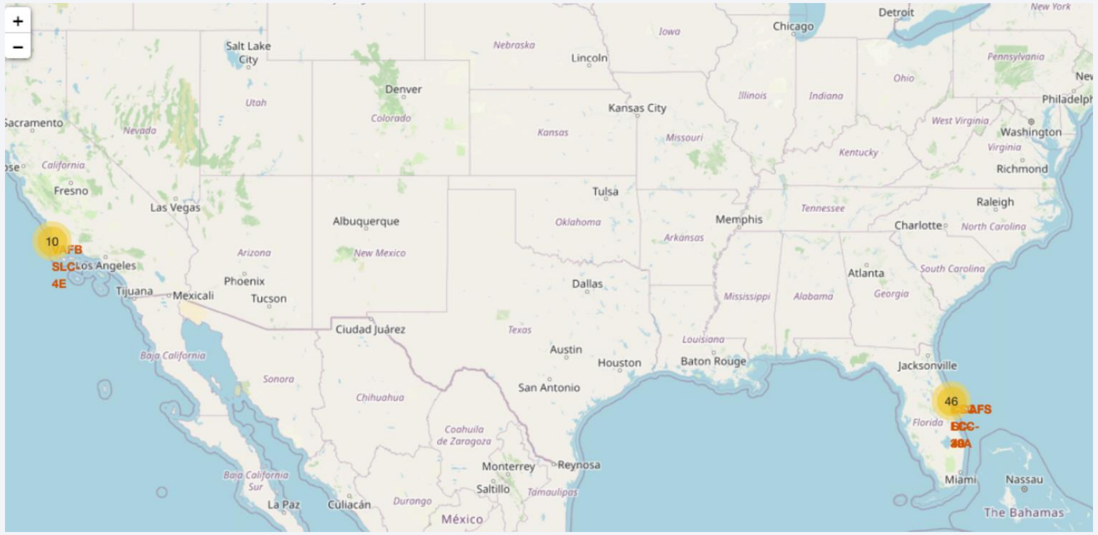
Launch Sites Proximities Analysis

All Launch Site Location Markers



All launches are in the USA and within Florida or California

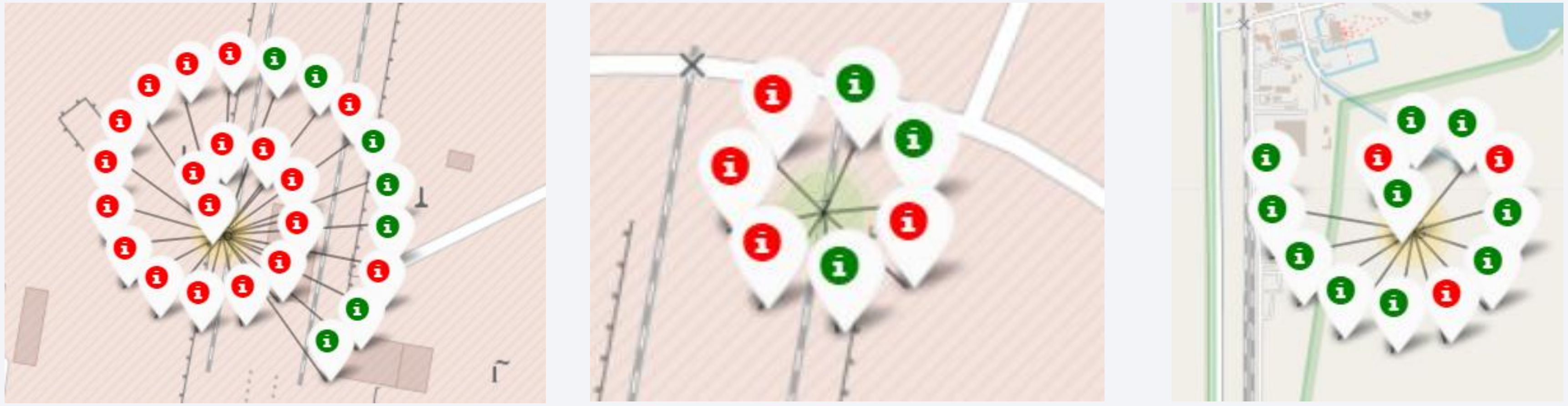
Color-labeled Markers in clusters - Launch Outcomes



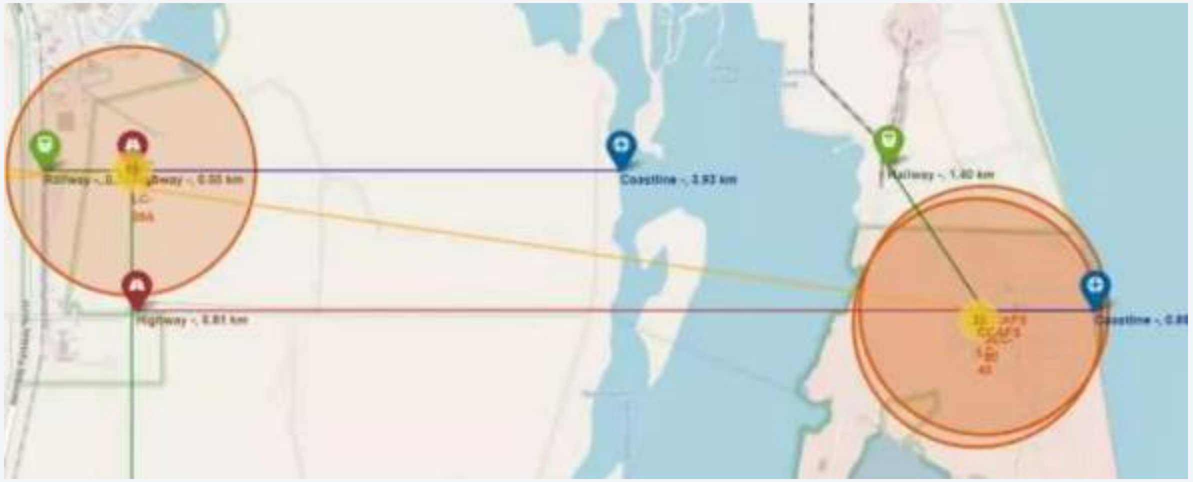
VAFB SLS-4E

Identifying the launch sites with easily identifiable success or failure rates using color-label markers in marker clusters

CCAFS LC-40



Launch Sites to its Proximities

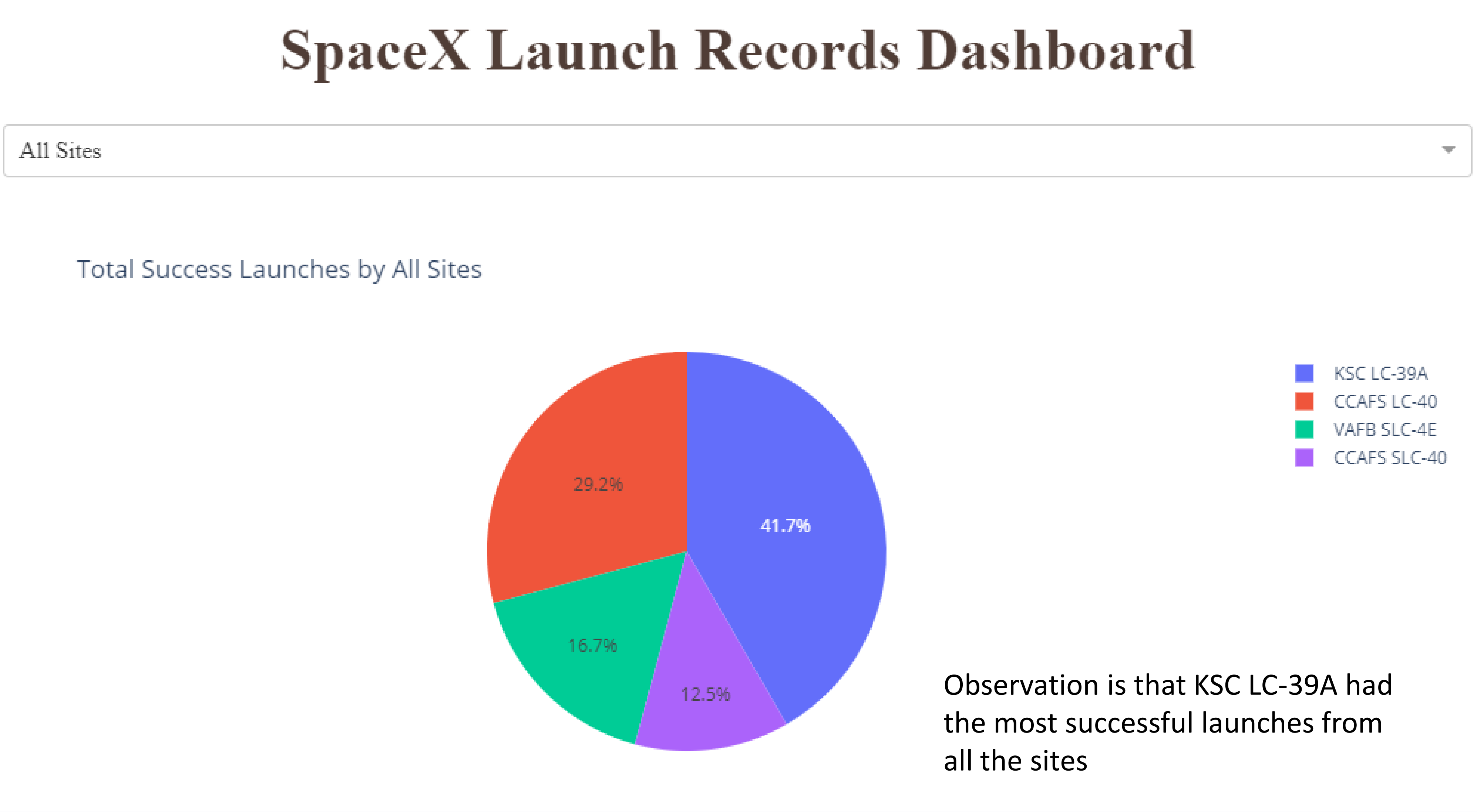


- Are launch sites near railways ? No except for space center railways to move equipment
- Are launch sites near highways ? No
- Are launch sites near coastlines ? Yes
- Are launch sites aways from populated cities and areas ? Yes

Section 4

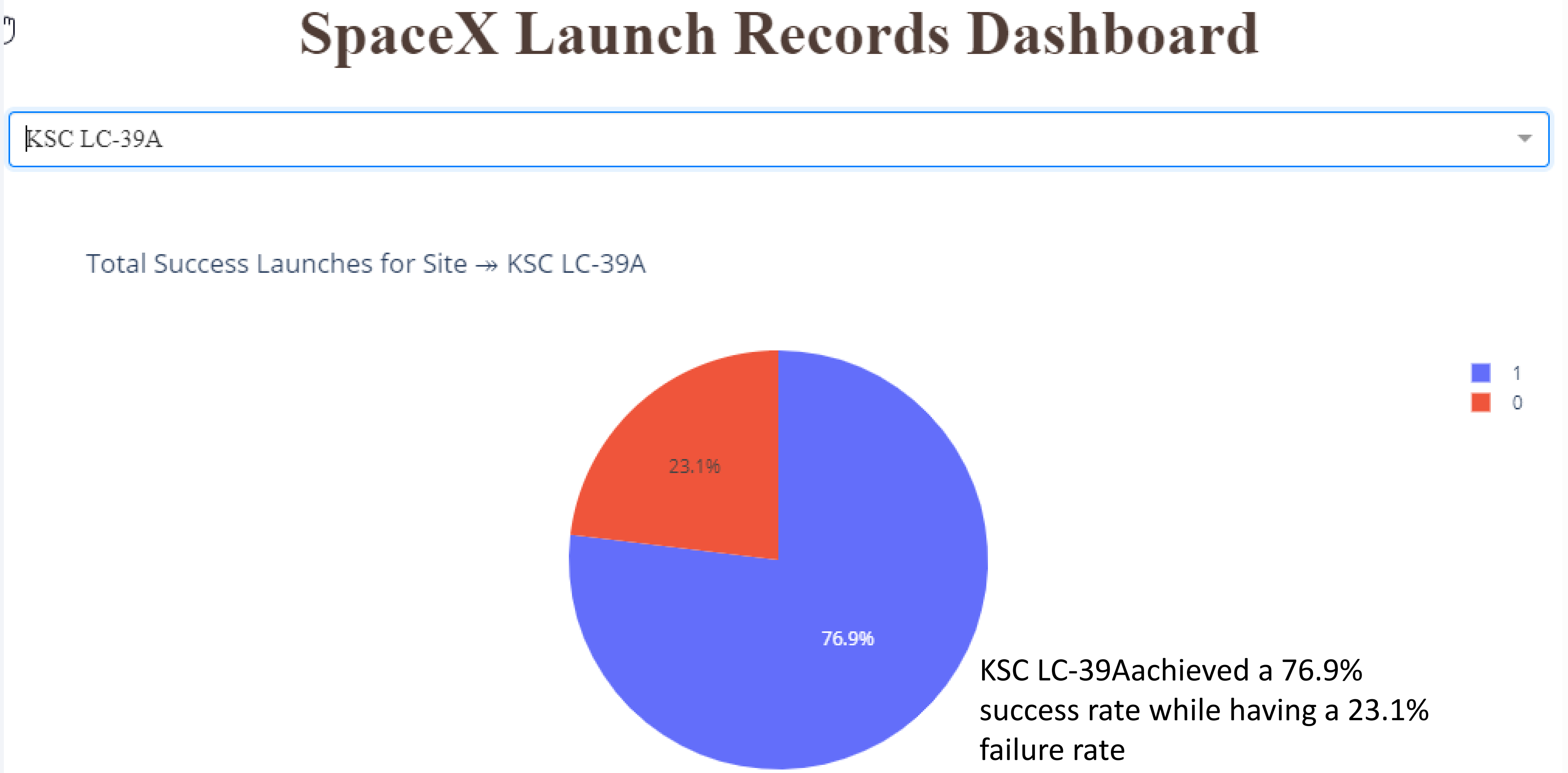
Build a Dashboard with Plotly Dash

Total Success launches by all sites



Observation is that KSC LC-39A had the most successful launches from all the sites

Success rate by site



KSC LC-39Aachieved a 76.9% success rate while having a 23.1% failure rate

Payload vs launch outcome

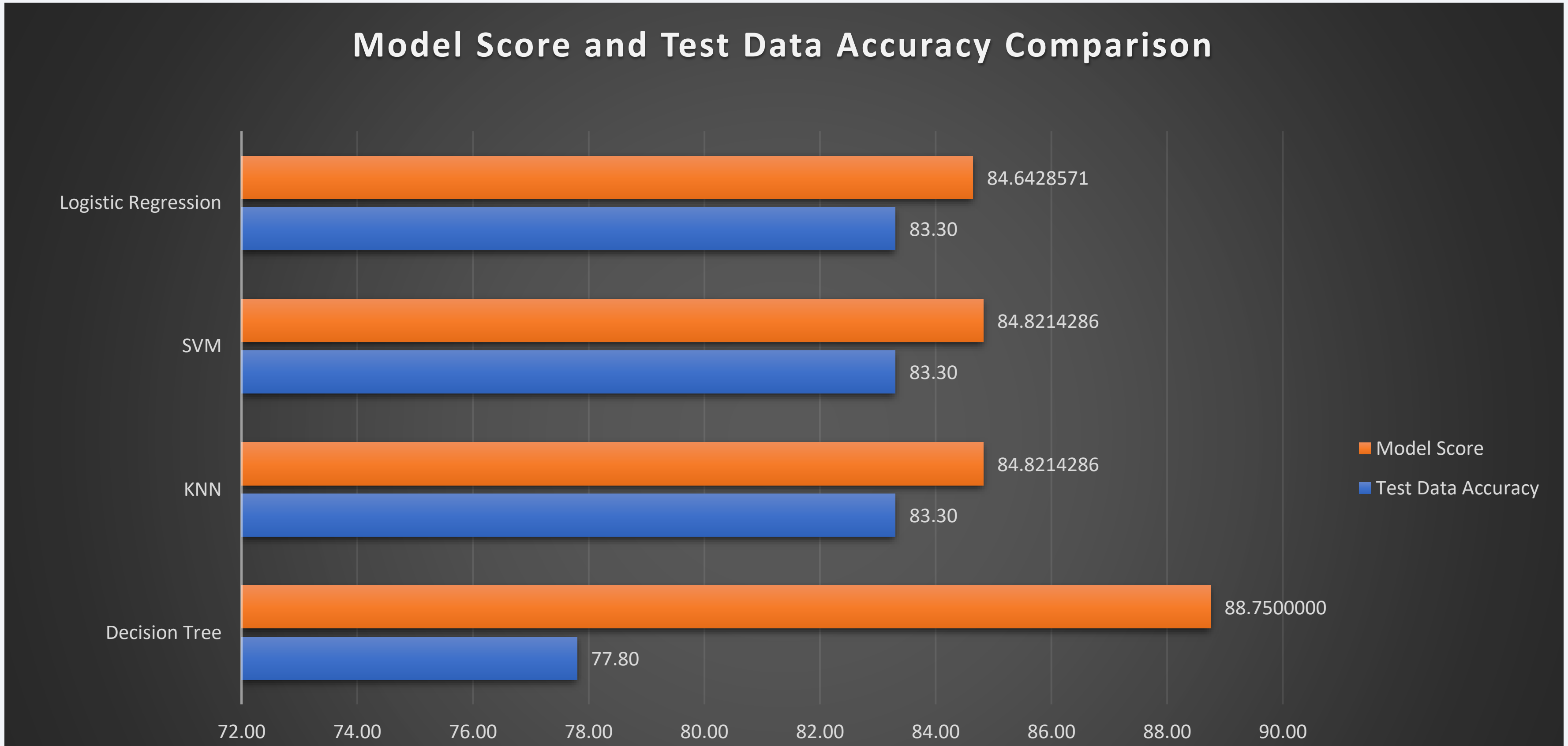
Payload 0 kg – 5000 kg

Payload 6000 kg – 10000 kg

Section 5

Predictive Analysis (Classification)

Classification Score and Accuracy



	Decision Tree	KNN	SVM	Logistic Regression
Model Score	88.7500000	84.8214286	84.8214286	84.6428571
Test Data Accuracy	77.80	83.30	83.30	83.30

Results

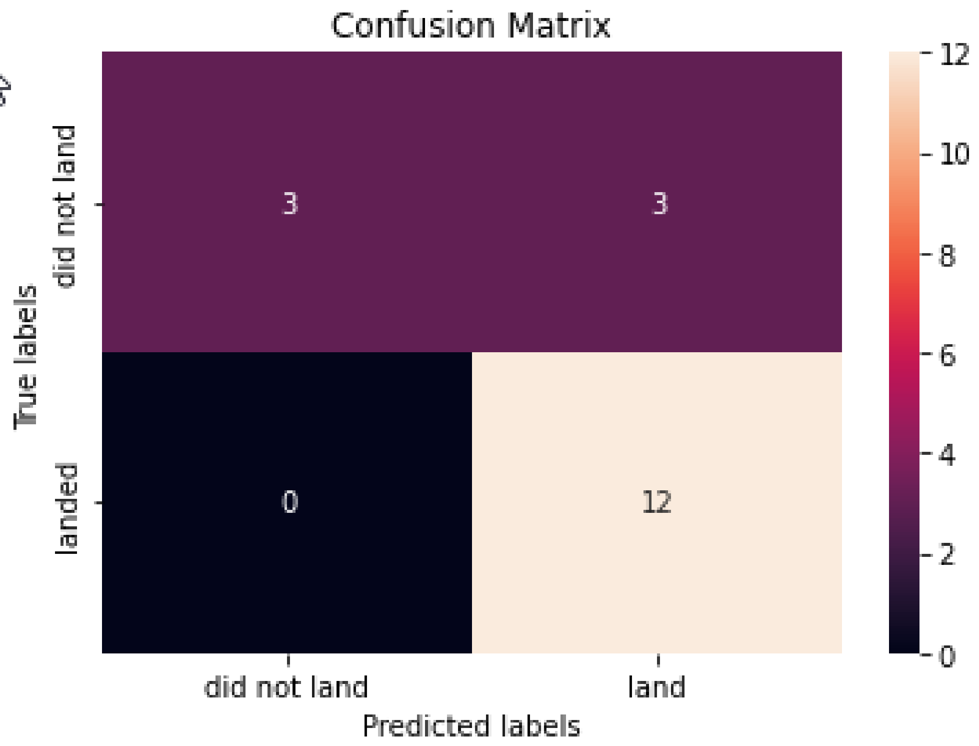
- Based on the GridSearchCV best scores the following results are shown in ranked order with Decision Tree classifier having the highest classification accuracy

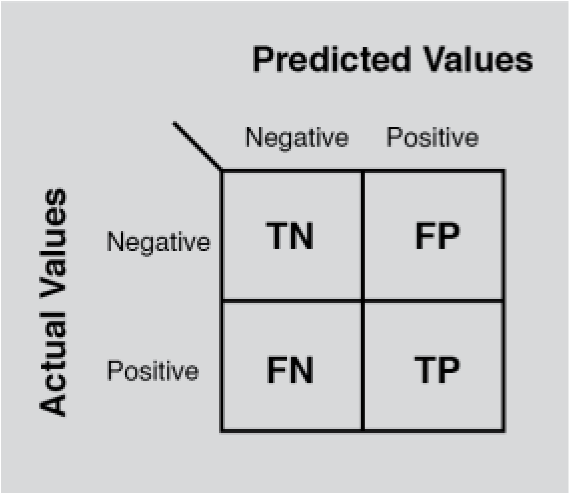
1. **Decision tree - GridSearchCV best score 0.8875 (Test data accuracy 0.7777777777777778)**
2. K nearest neighbors (KNN) – GridSearchCV best score: 0.8482142857142858 (Test data accuracy 0.8333333333333334)
3. Support vector machine (SVM) – GridSearchCV best score: 0.8482142857142856 (Test data accuracy 0.8333333333333334)
4. Logistic regression – GridSearchCV best score: 0.8464285714285713 (Test data accuracy 0.8333333333333334)

Results

Logistic Regression

- GridSearchCV best score: 0.8464285714285713
- Accuracy score on test set: 0.8333333333333334
- Confusion Matrix:

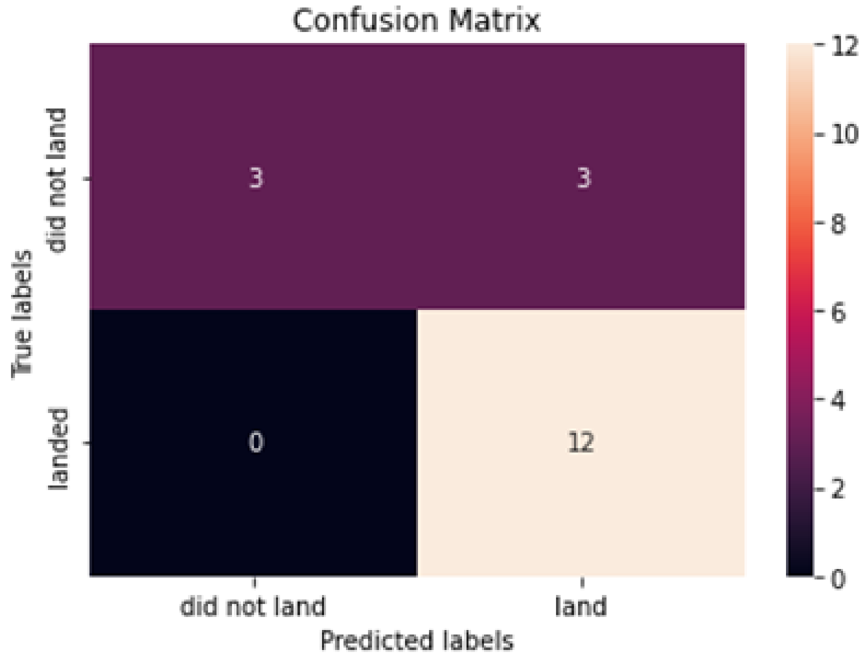


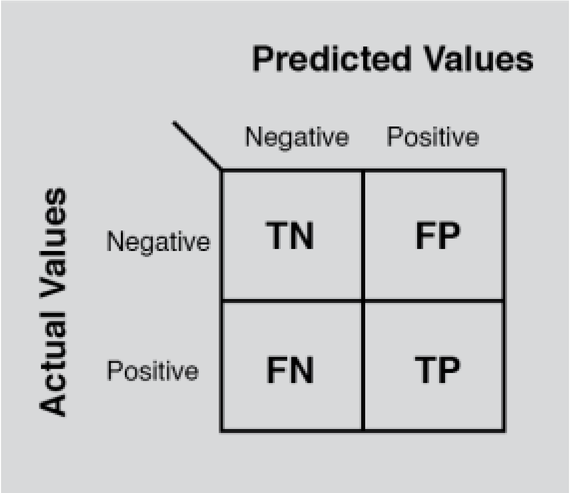


Results

Support vector machine (SVM)

- GridSearchCV best score: 0.8482142857142856
- Accuracy score on test set: 0.8333333333333334
- Confusion Matrix:

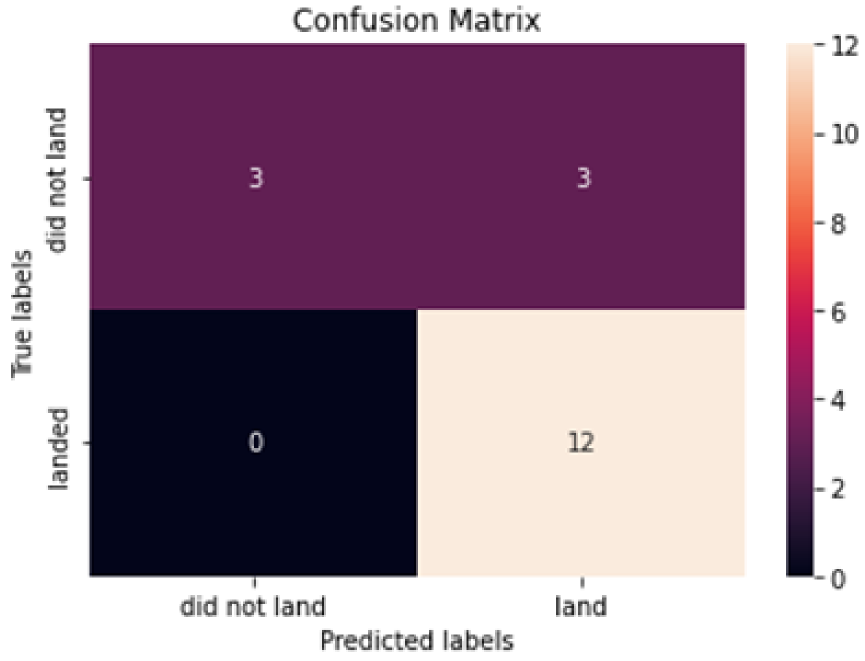


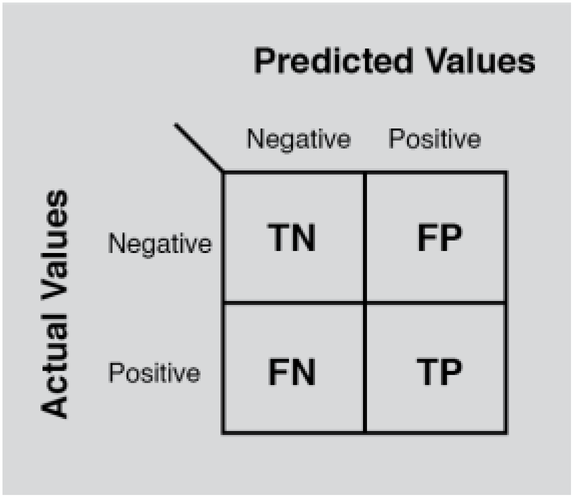


Results

Decision tree

- GridSearchCV best score: 0.8767857142857143
- Accuracy score on test set: 0.7777777777777778
- Confusion Matrix:

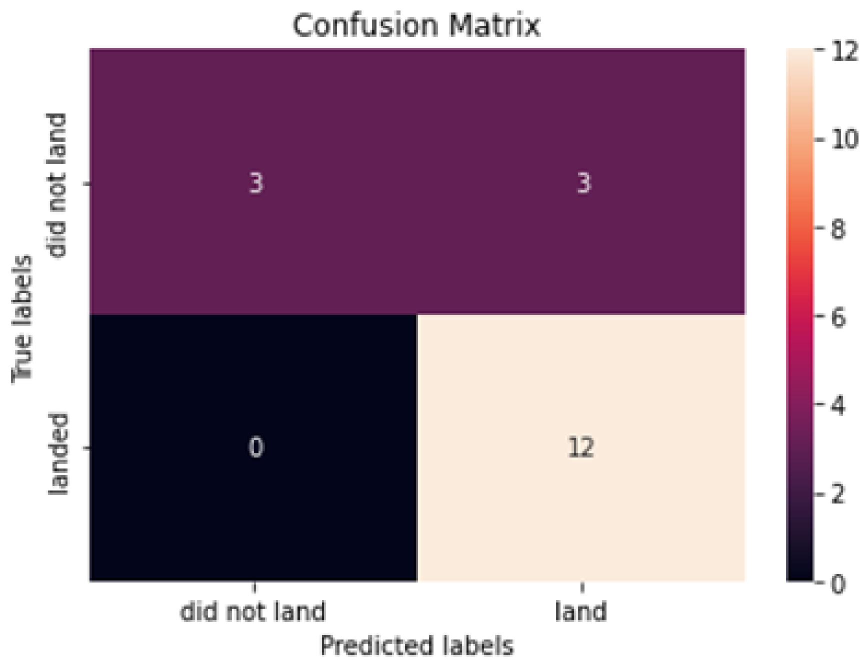


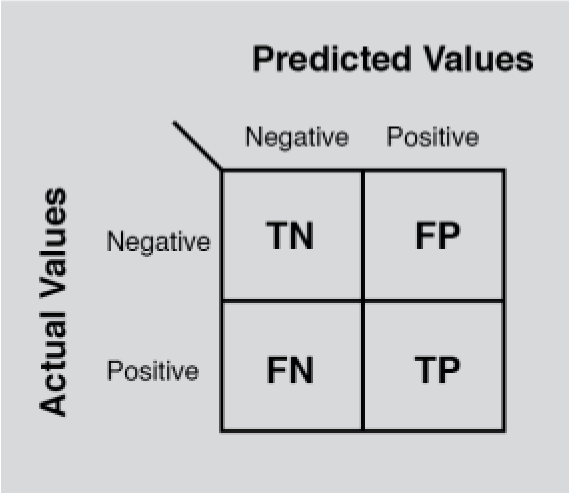


Results

Decision tree

- K nearest neighbors: 0.8482142857142858
- Accuracy score on test set: 0.8333333333333334
- Confusion Matrix:





Conclusions

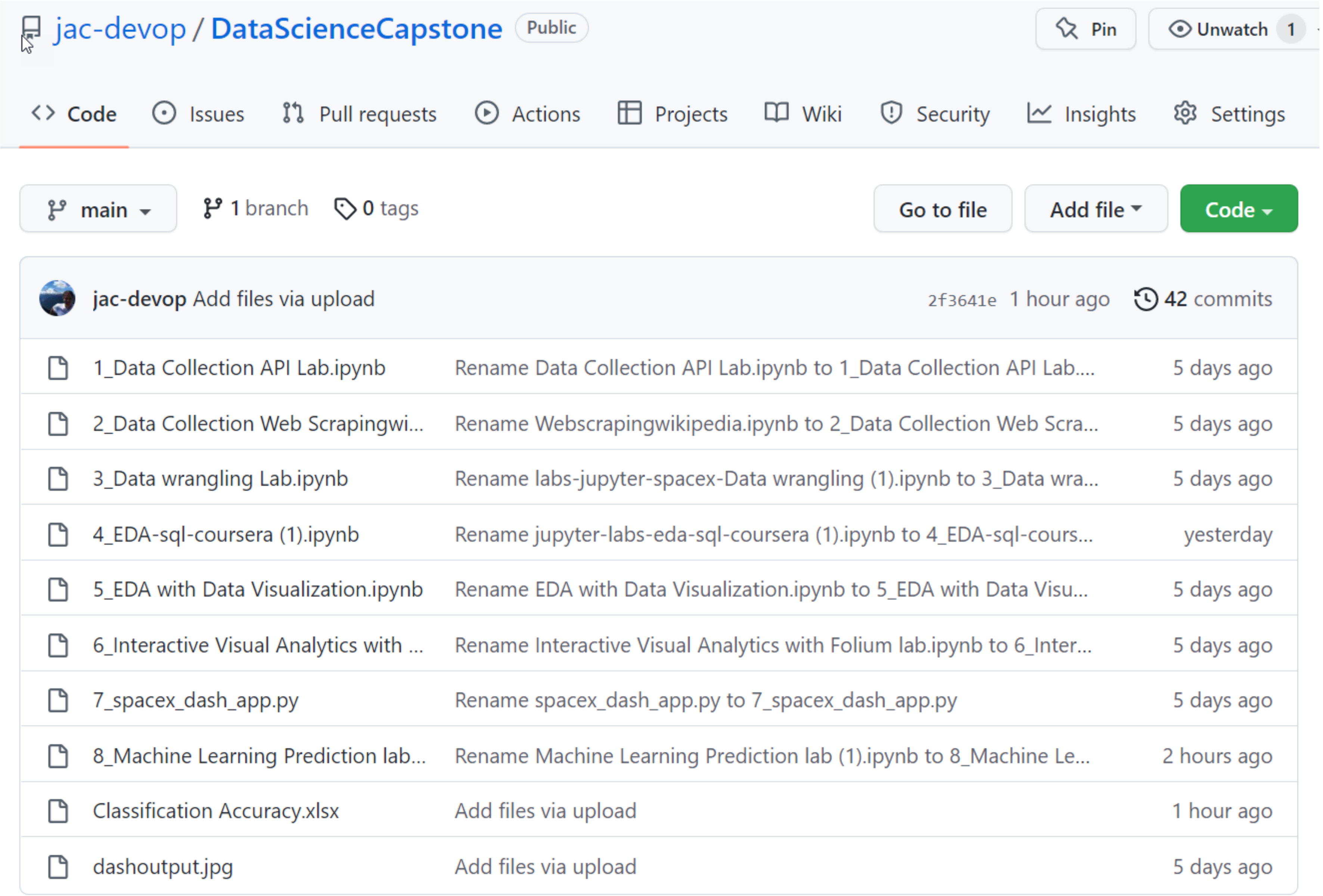
Following observations and conclusions where made:

- The Decision tree classifier is the best machine learning algorithm for this scenario and dataset
- Launch success rate started to increase in 2013 until 2020 likely due to gained learning and technical advances
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate
- KSC LC-39 Had the most successful launched of any site
- Low weighted payloads perform better than the heavier payloads

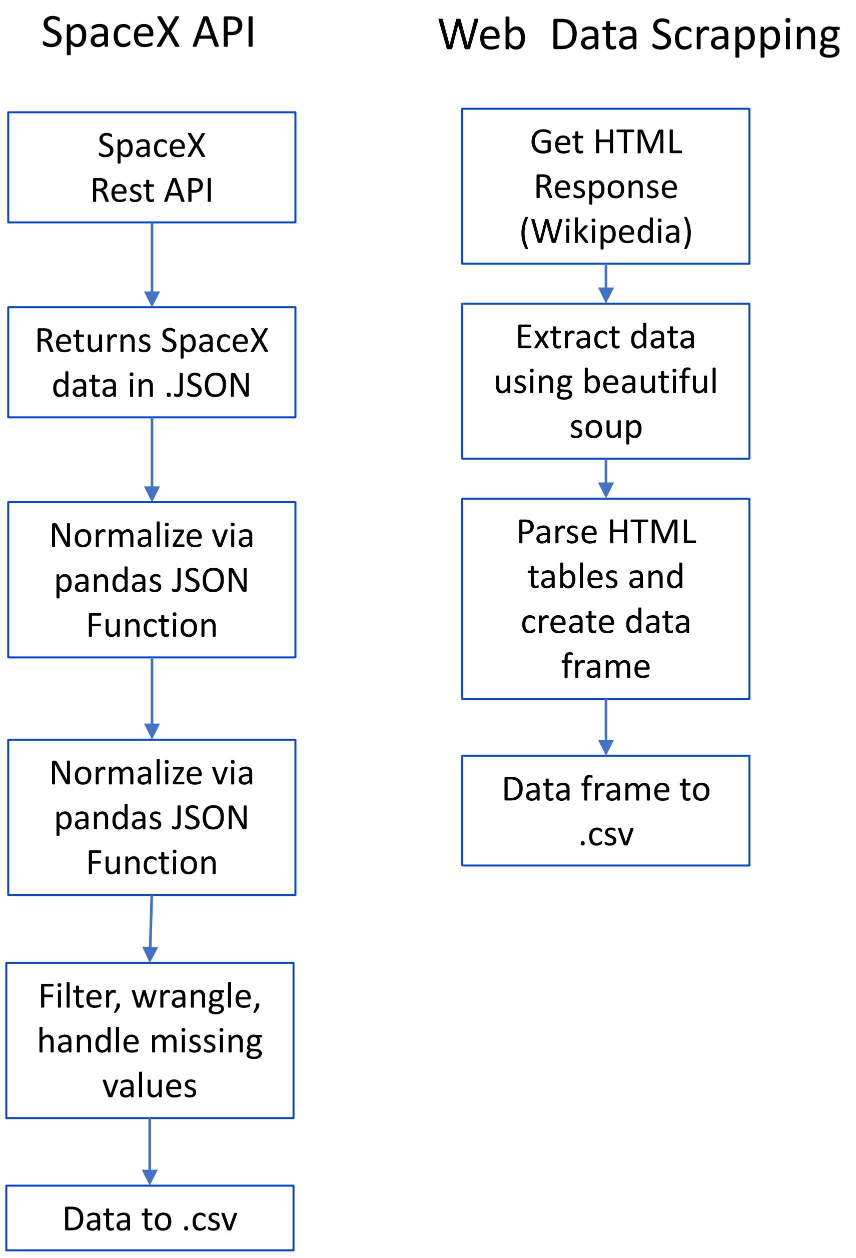
Thank you!

Appendix

- Github content enumeration
- Various flow charts used in the presentation



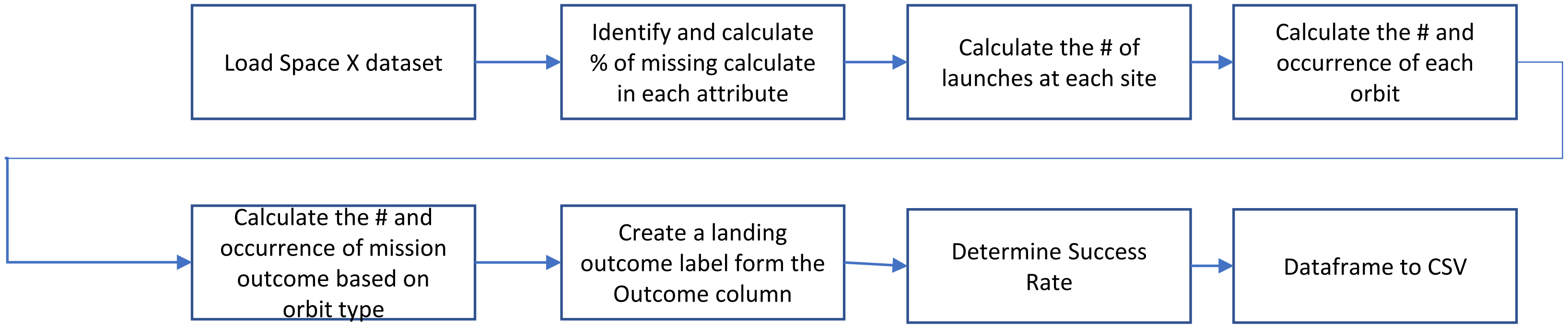
git [jac-devop/DataScienceCapstone: Applied Data Science Capstone (github.com)]

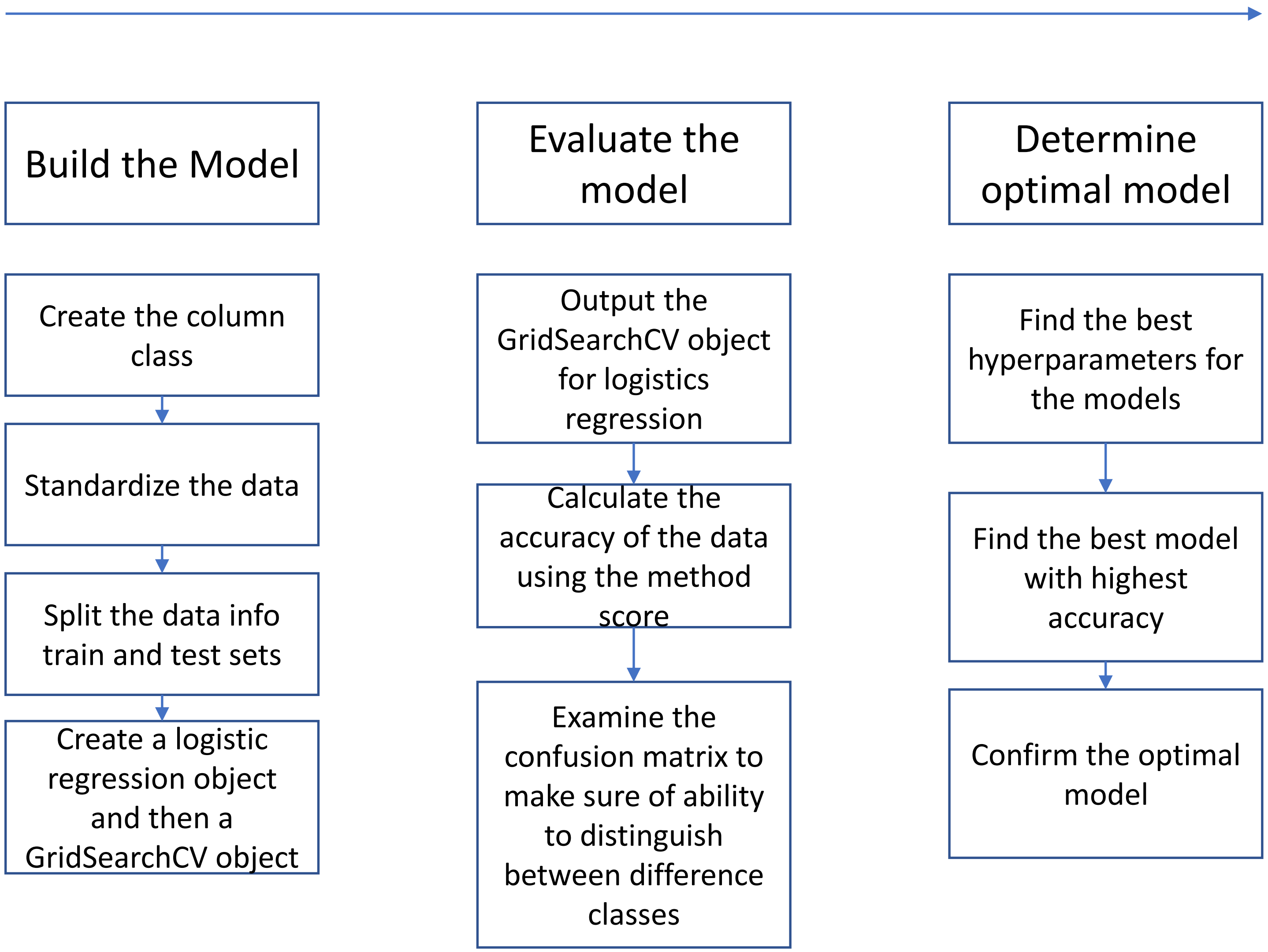

SpaceX API
Web Data Scrapping
SpaceX Rest API
Returns SpaceX data in .JSON
Normalize via pandas JSON Function
Normalize via pandas JSON Function
Filter, wrangle, handle missing values
Data to .csv
Get HTML Response (Wikipedia)
Extract data using beautiful soup
Parse HTML tables and create data frame
Data frame to .csv

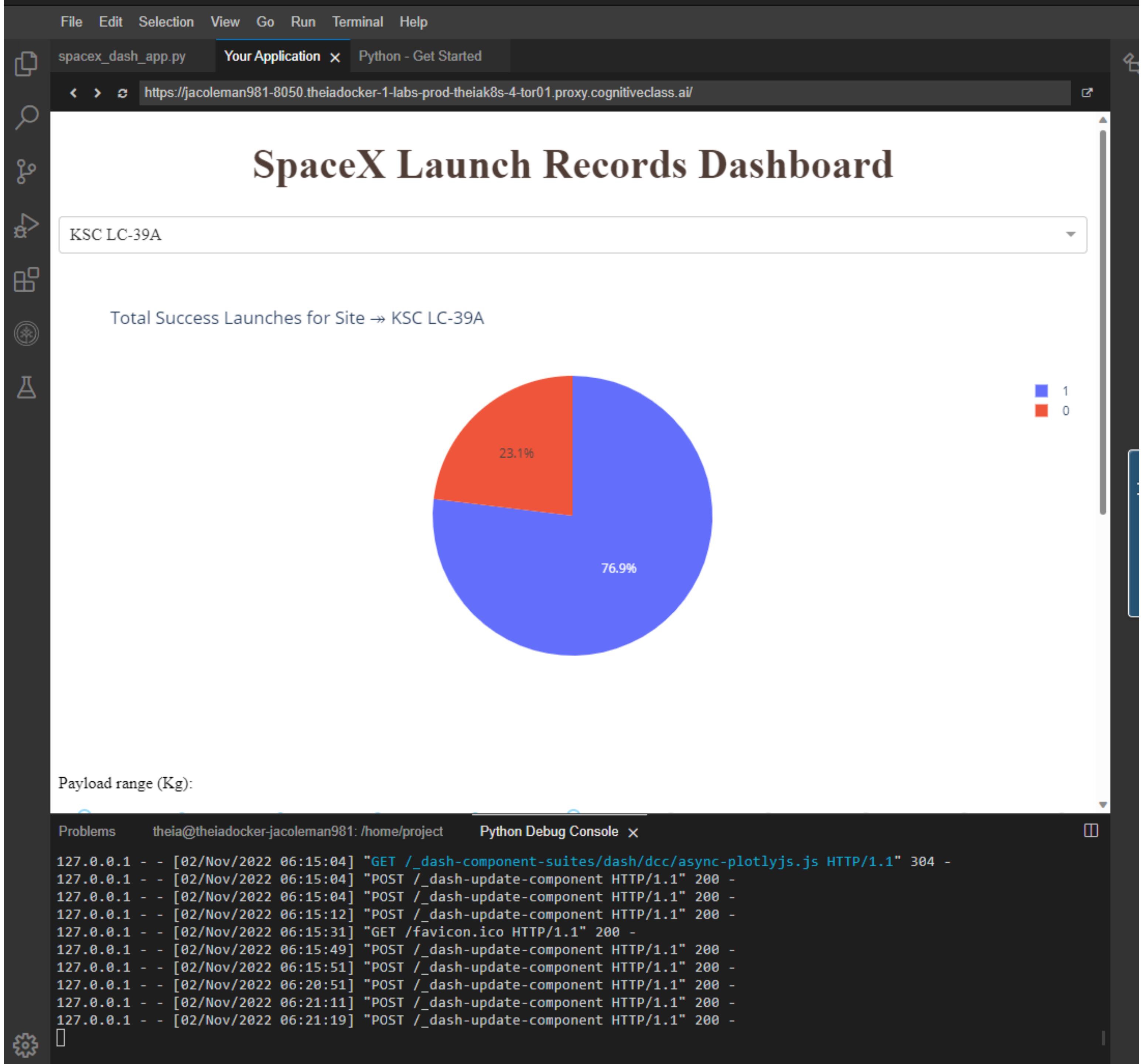
EDA Analysis – Data Wrangling

Find Patterns in the data to determine the label for training supervised models

Training Labels with 1 means the booster successfully landed and 0 means unsuccessful




Build the Model
Create the column class
Standardize the data
Split the data info train and test sets
Create a logistic regression object and then a GridSearchCV object
Evaluate the model
Output the GridSearchCV object for logistics regression
Calculate the accuracy of the data using the method score
Examine the confusion matrix to make sure of ability to distinguish between difference classes
Determine optimal model
Find the best hyperparameters for the models
Find the best model with highest accuracy
Confirm the optimal model


File Edit Selection View Go Run Terminal Help
spacex_dash_app.py
Your Application
Python - Get Started
https://jacoleman981-8050.theiadocker-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/
SpaceX Launch Records Dashboard
KSC LC-39A
Total Success Launches for Site → KSC LC-39A
23.1%
76.9%
1
0
Payload range (Kg):
Problems
theia@theiadocker-jacoleman981: /home/project
Python Debug Console
127.0.0.1 - - [02/Nov/2022 06:15:04] "GET /_dash-component-suites/dash/dcc/async-plotlyjs.js HTTP/1.1" 304 -
127.0.0.1 - - [02/Nov/2022 06:15:04] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2022 06:15:04] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2022 06:15:12] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2022 06:15:31] "GET /favicon.ico HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2022 06:15:49] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2022 06:15:51] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2022 06:20:51] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2022 06:21:11] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2022 06:21:19] "POST /_dash-update-component HTTP/1.1" 200 -