Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using REST API and Web Scraping
  - Data Wrangling
  - Exploratory data analysis with SQL and Data Visualization
  - Creation of interactive visual analytics with Folium and Plotly DASH
  - Creation of predictive analytics with multiple machine learning models
- Summary of all results
  - Results of exploratory data analysis
  - Interactive visual analytics (screenshots)
  - Results of predictive analytics

Introduction

- Project background and context
- Problems you want to find answers

Section 1

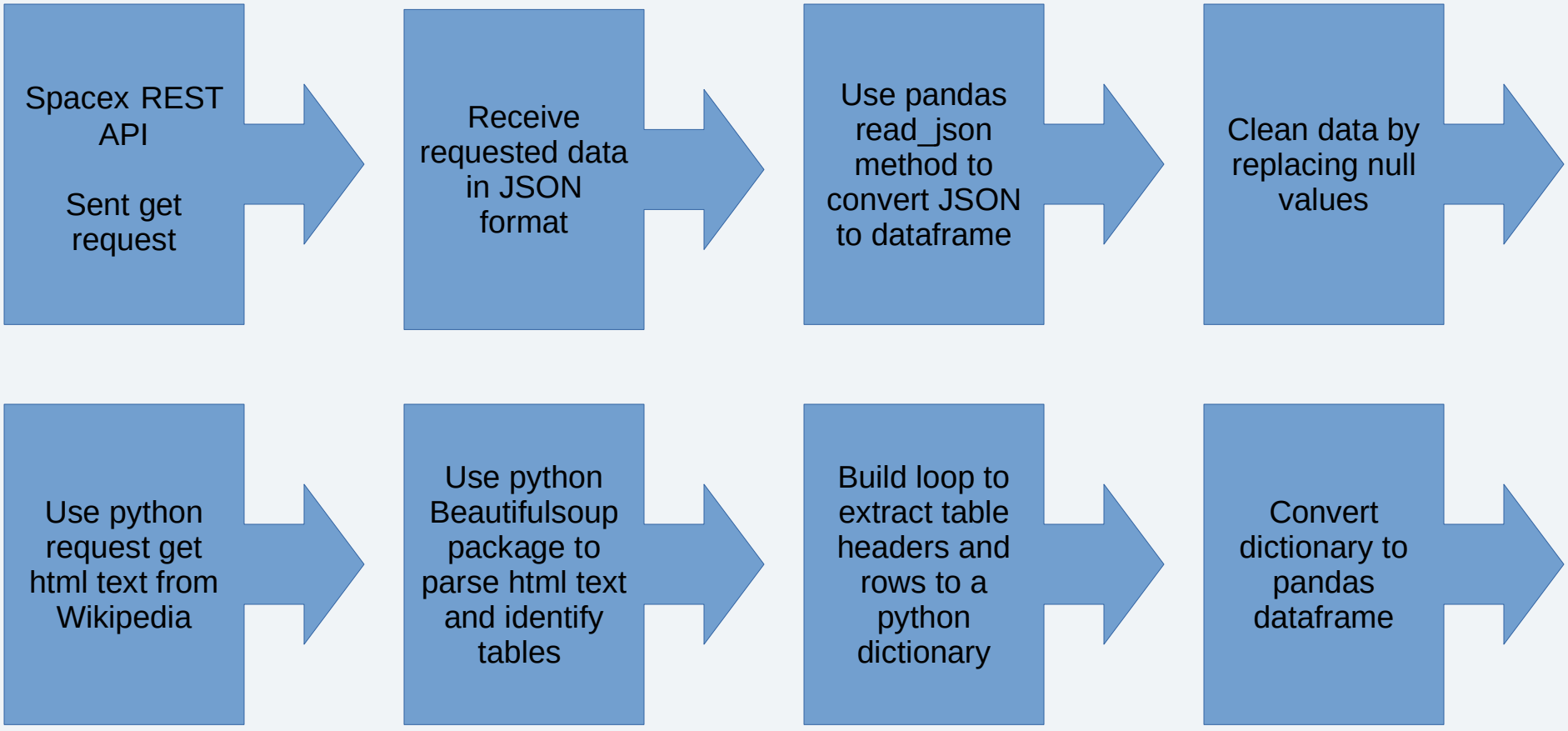
Methodology

Methodology

Executive Summary

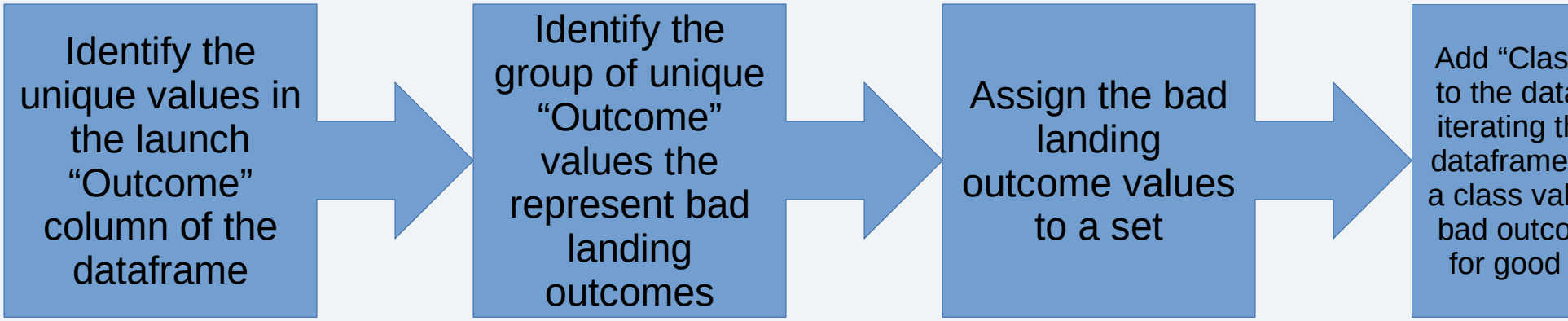
- Data collection methodology:
  - Data was gathered using SpaceX REST API and web scraping from Wikip
- Perform data wrangling
  - Describe was processed by replacing null values and utilizing one-hot encoding to add categorical features to the data set
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Das
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection



Data Wrangling

Describe how data were processed



Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

Link:
https://github.com/e-m-vanderveen/IBM-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data wrangling-v2.ipynb

EDA with Data Visualization

Summarize what charts were plotted and why you used those charts

- Scatter plots showing the relationship between
  - Payload and Flight Number
  - Flight Number and Launch Site
  - Payload and Launch Site
  - Orbit type and Flight Number
  - Payload and Orbit Type
- Bar Chart showing relationship between launch success rate and orbit type
- Line chart showing percent success rate and year

Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

EDA with SQL

Using bullet point format, summarize the SQL queries you perform

- 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 succesful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass gr 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass usin subquery
- List the records which will display the month names, failure landing_outcomes in drone ship versions, launch_site for the months in year 2015.
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) l date 2010-06-04 and 2017-03-20, in descending order.

Add the GitHub URL of your completed EDA with SQL notebook, as an ext reference and peer-review purpose

Build an Interactive Map with Folium

Summarize what map objects such as markers, circles, lines, etc. you crea
added to a folium map

- Created interactive maps displaying the different locations where Spac
  launches have occurred
- Added circle markers for launch locations and cluster markers for the la
  outcomes for each launch that occurred at each location
- Added polylines and distance annotations to measure distance from la
  to the nearest coastline and nearest city

Add the GitHub URL of your completed interactive map with Folium map, a
external reference and peer-review purpose

Build a Dashboard with Plotly Dash

Summarize what plots/graphs and interactions you have added to a dashboard

- Displayed a pie chart to show the number of launches per launch site a percentage of the total Spacex launches
- Displayed a scatter plot of landing outcome and payload mass to display the relationship between outcome and payload for the differen booster versions
- Both plots adjust with the dropdown filter and slider range so that use can interact with the plots by selecting landing site and payload mass

Add the GitHub URL of your completed Plotly Dash lab, as an external reference a peer-review purpose

Predictive Analysis (Classification)

Summarize how you built, evaluated, improved, and found the best performing classification model

You need present your model development process using key phrases and flowchart

Add the GitHub URL of your completed predictive analysis lab, as an exter reference and peer-review purpose

Results

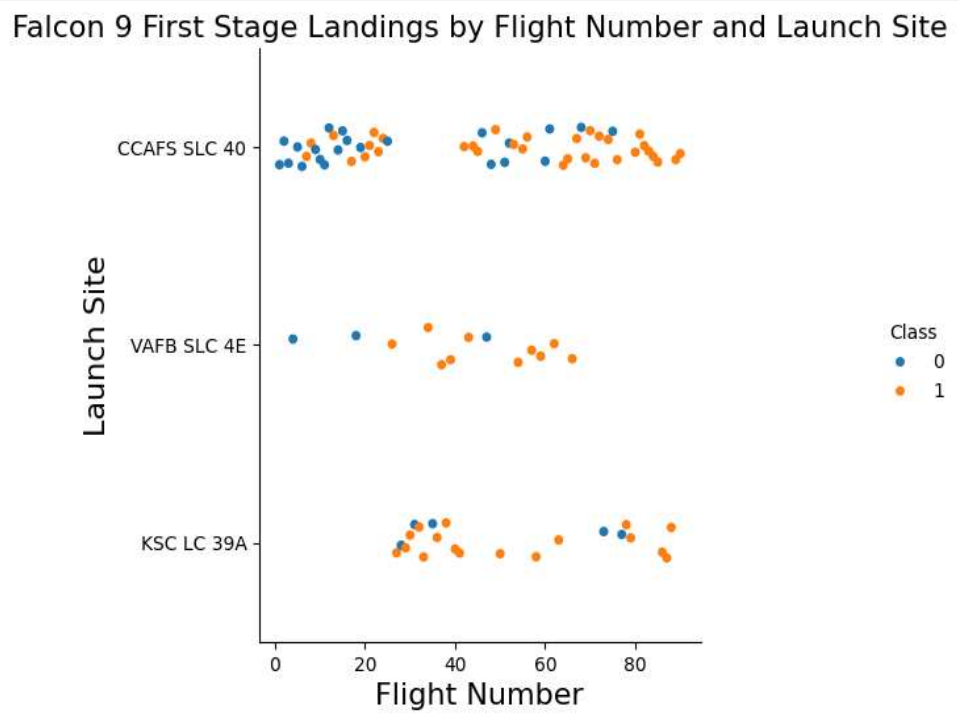
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

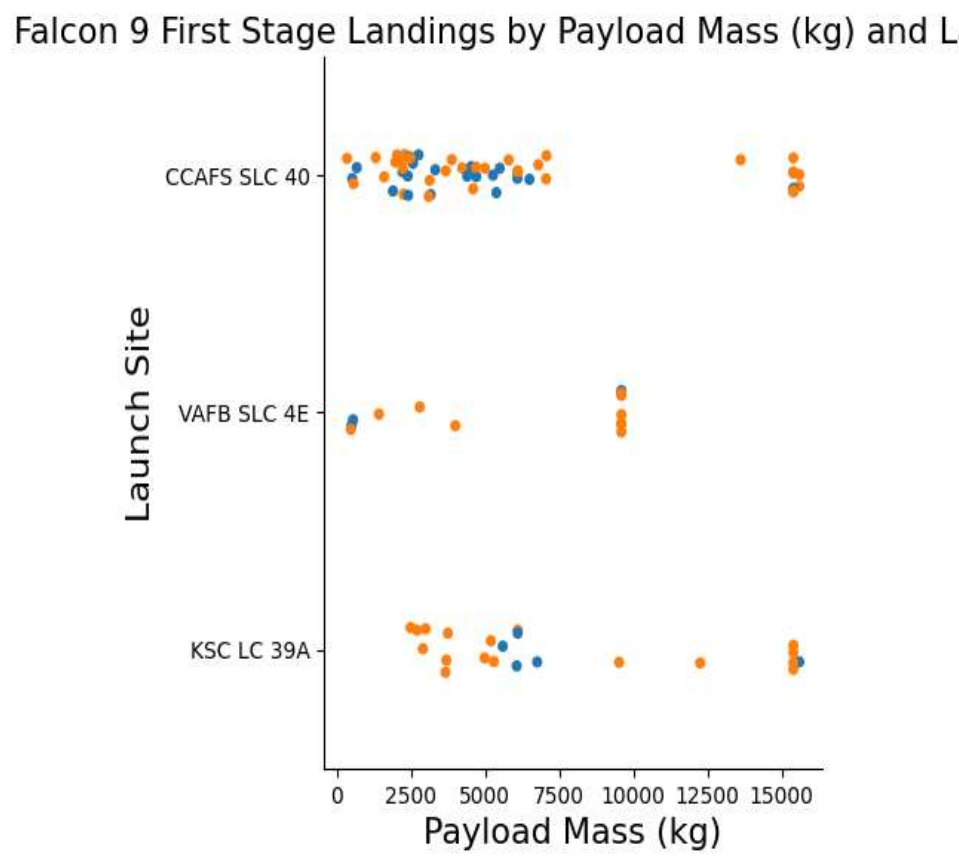
- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations



Link:
https://github.com/e-m-vanderveen/IBM-Data-Science-Capstone/blob/main/jupyter-labs-eda-da

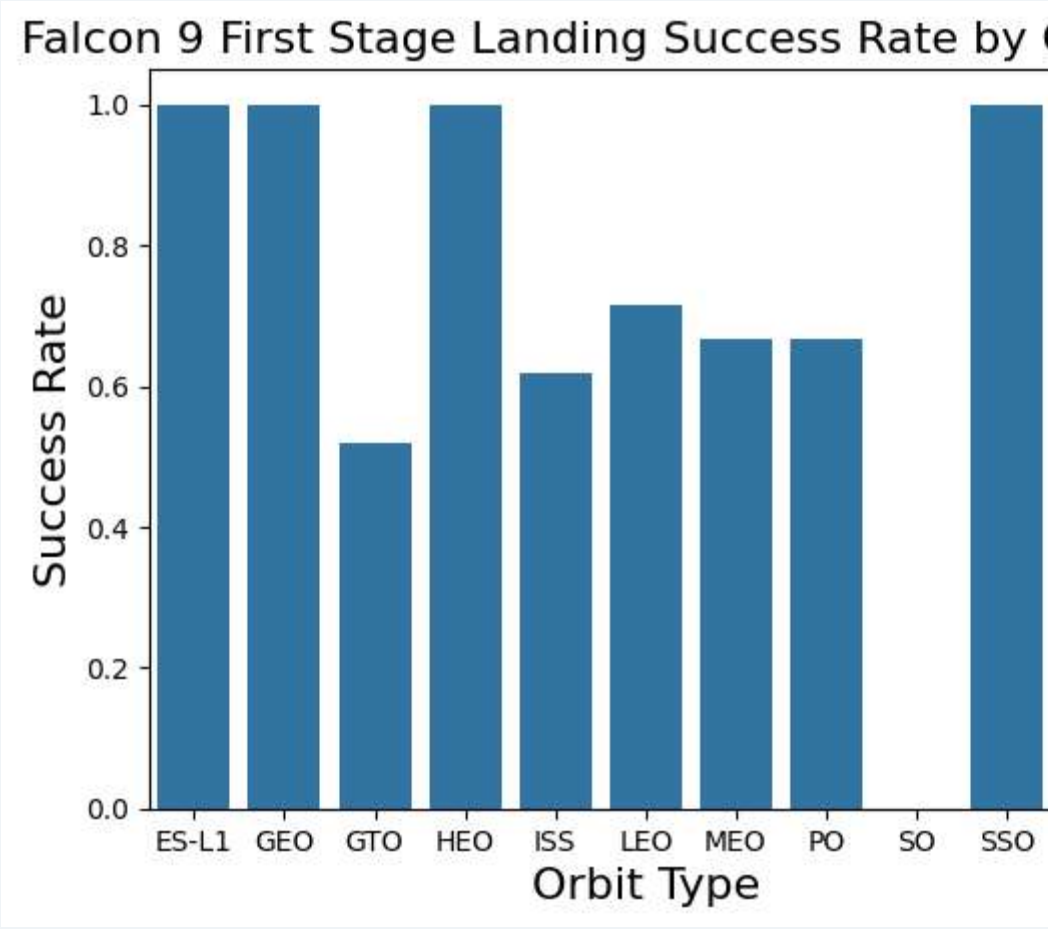
Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- Show the screenshot of the scatter plot with explanations



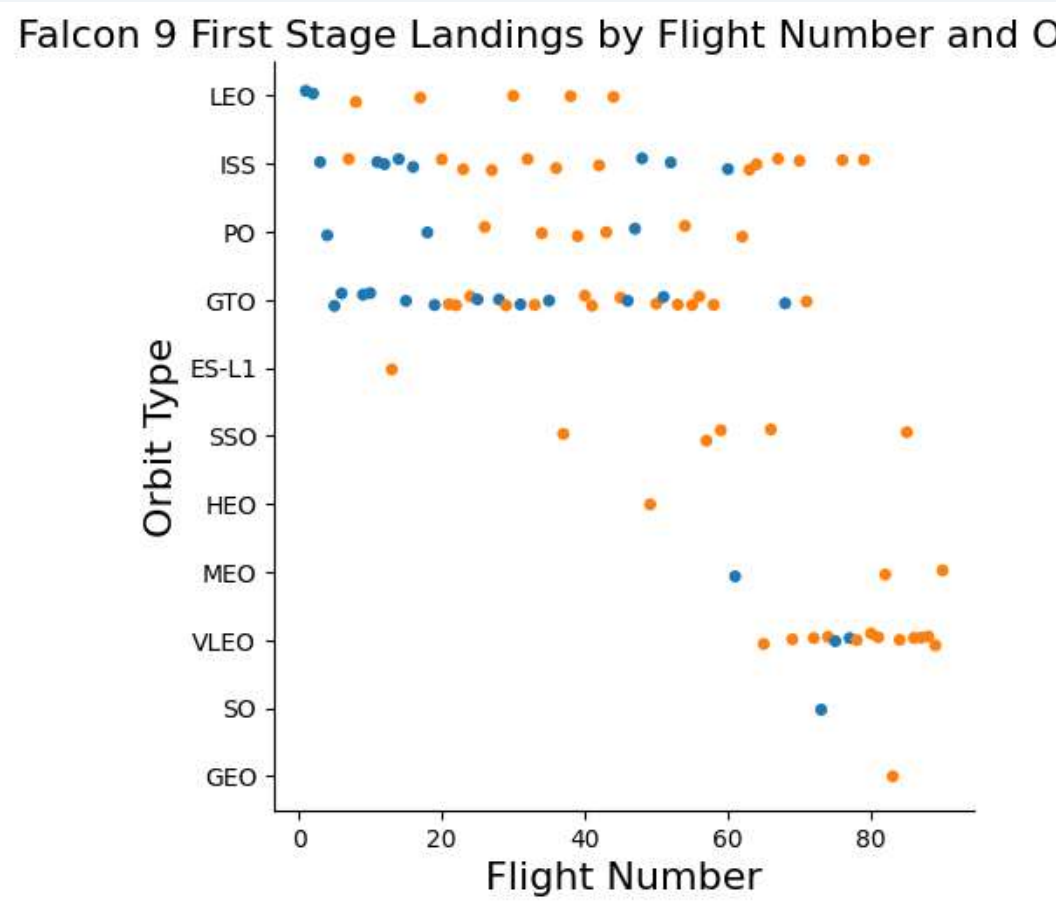
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations



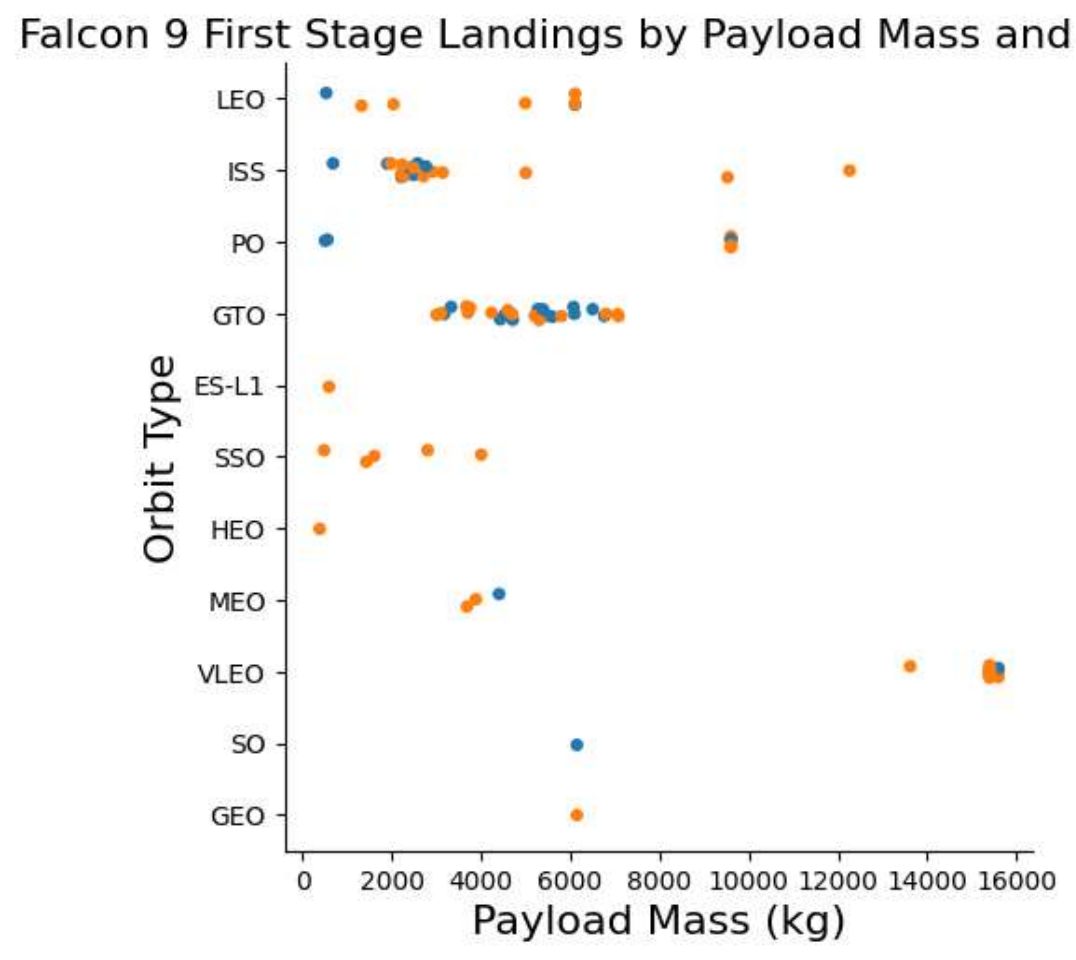
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations



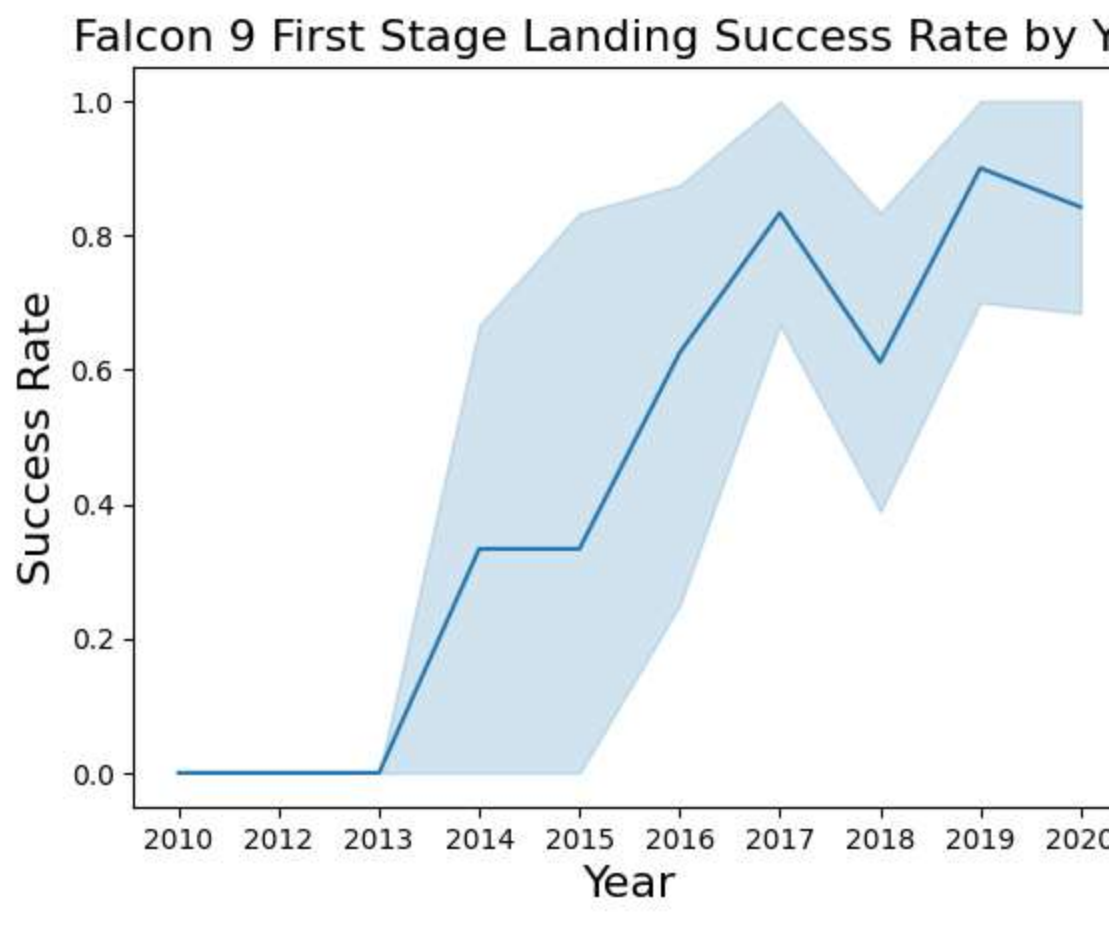
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations



All Launch Site Names

Find the names of the unique launch sites

Present your query result with a short explanation here

```
1  %sql select distinct "Launch_Site" from SPACEXTABLE;
2
```

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

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

Link:
https://github.com/e-m-vanderveen/IBM-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

Present your query result with a short explanation here

```
1  %sql select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	L
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	
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	
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	

Total Payload Mass

Calculate the total payload carried by boosters from NASA

Present your query result with a short explanation here

```
1  %sql select Customer, sum("PAYLOAD_MASS__KG_") from SPACEXTABLE where Customer like 'NASA (CRS)' group by Customer
```

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

Customer	sum(PAYLOAD_MASS__KG_)
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1

Present your query result with a short explanation here

```
1  %sql select "Booster_Version", avg("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Booster_Version" like 'F9 v1.1' group by "Bo
```

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

Booster_Version	avg(PAYLOAD_MASS__KG_)
F9 v1.1	2928.4

First Successful Ground Landing Date

Find the dates of the first successful landing outcome on ground pad

Present your query result with a short explanation here

```
1  %sql select "Landing_Outcome", min("Date") from SPACEXTABLE where "Landing_Outcome" like 'Success (ground%'
```

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

Landing_Outcome	min(Date)
Success (ground pad)	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of boosters which have successfully landed on dro
ship and had payload mass greater than 4000 but less than 6000

Present your query result with a short explanation here

```
1  %sql select "Landing_Outcome", "PAYLOAD_MASS__KG_" from SPACEXTABLE where "Landing_Outcome" like 'Success (drone%' and ("PAYLOAD_MASS__KG_" between 4000
```

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

Landing_Outcome	PAYLOAD_MASS__KG_
Success (drone ship)	4696
Success (drone ship)	4600
Success (drone ship)	5300
Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcor

Calculate the total number of successful and failure mission outcomes

Present your query result with a short explanation here

```
1  %sql select distinct "Mission_Outcome", count("Mission_Outcome") from SPACEXTABLE group by "Mission_Outcome"
```

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

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

Boosters Carried Maximum Payload

List the names of the booster which have carried the maximum payload ma

Present your query result with a short explanation here

```
1  %sql select "Booster_Version", "PAYLOAD_MASS__KG_"  from SPACEXTABLE where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_") from SPACEXTABLE)
```

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

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

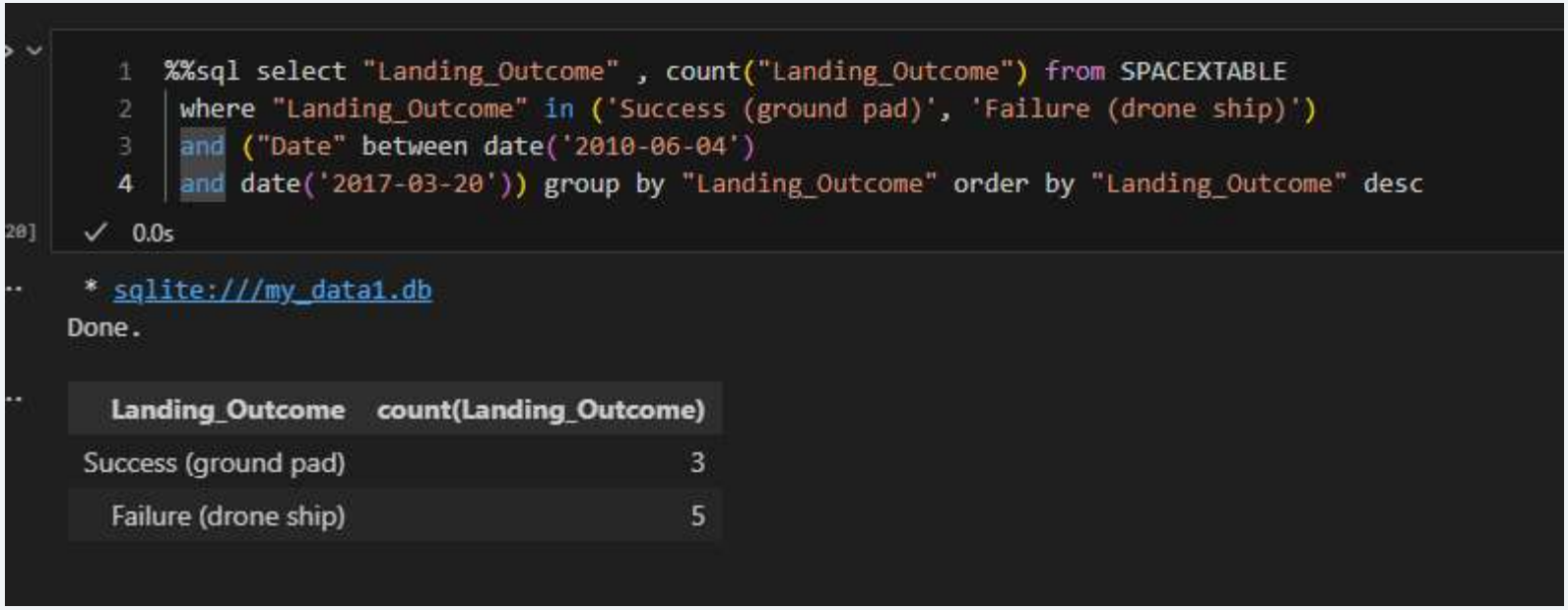
```
%%sql select
case substr("Date",6,2)
when '01' then 'January'
when '02' then 'Febuary'
when '03' then 'March'
when '04' then 'April'
when '05' then 'May'
when '06' then 'June'
when '07' then 'July'
when '08' then 'August'
when '09' then 'September'
when '10' then 'October'
when '11' then 'November'
when '12' then 'December'
else '' end as  'Month',
substr("Date",0,5) as 'Year', "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTABLE where "Landing_Outcome" like 'Failure (
```

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

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
January	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
January	2016	Failure (drone ship)	F9 v1.1 B1017	VAFB SLC-4E
March	2016	Failure (drone ship)	F9 FT B1020	CCAFS LC-40
June	2016	Failure (drone ship)	F9 FT B1024	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017

Rank the count of landing outcomes (such as Failure (drone ship) Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order



```
%%sql select "Landing_Outcome" , count("Landing_Outcome") from SPACEXTABLE
 where "Landing_Outcome" in ('Success (ground pad)', 'Failure (drone ship)')
 and ("Date" between date('2010-06-04')
 and date('2017-03-20')) group by "Landing_Outcome" order by "Landing_Outcome" desc
```

✓ 0.0s

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

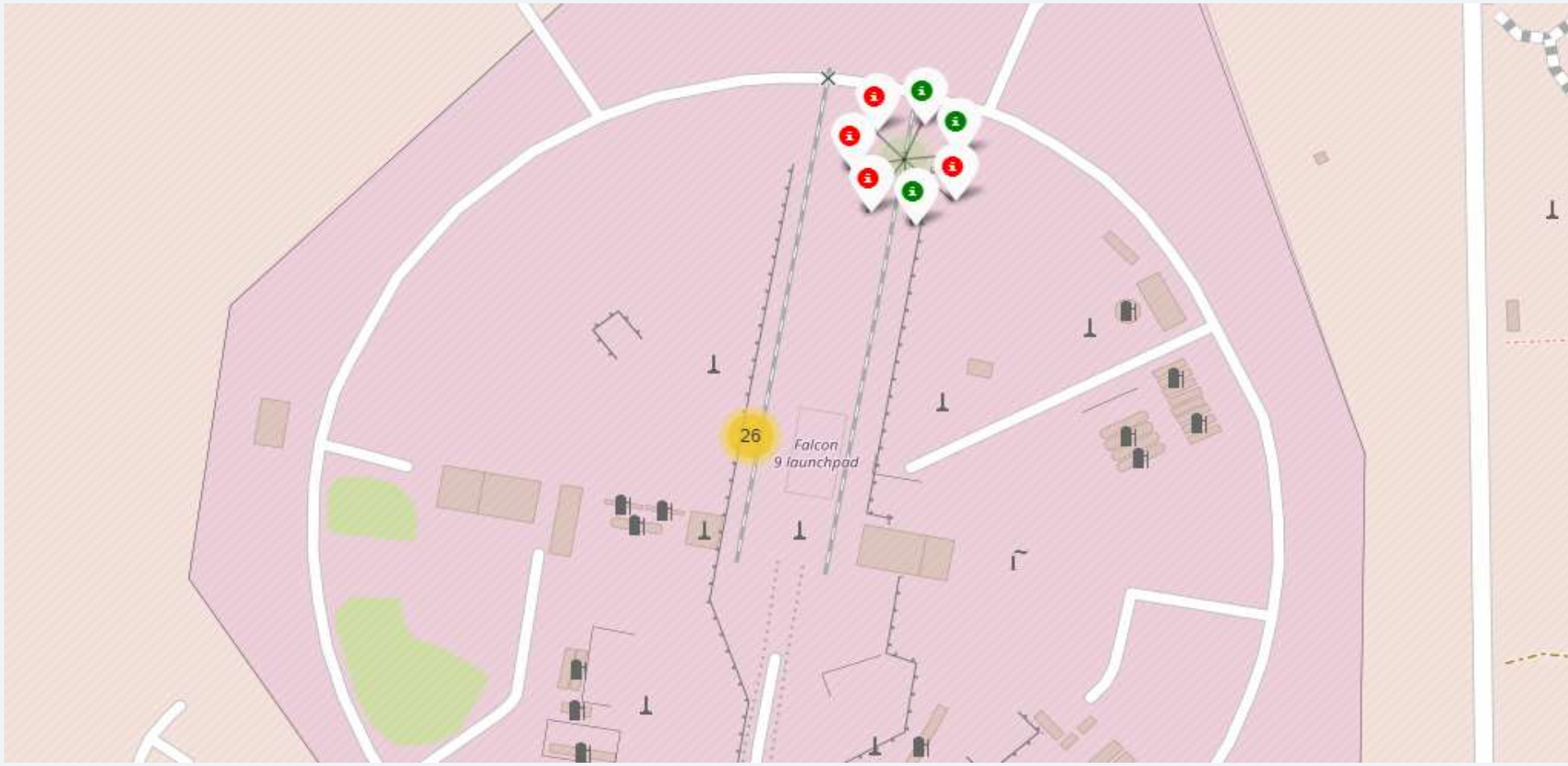
Landing_Outcome	count(Landing_Outcome)
Success (ground pad)	3
Failure (drone ship)	5


Section 3
Launch Sites
Proximities Analysis

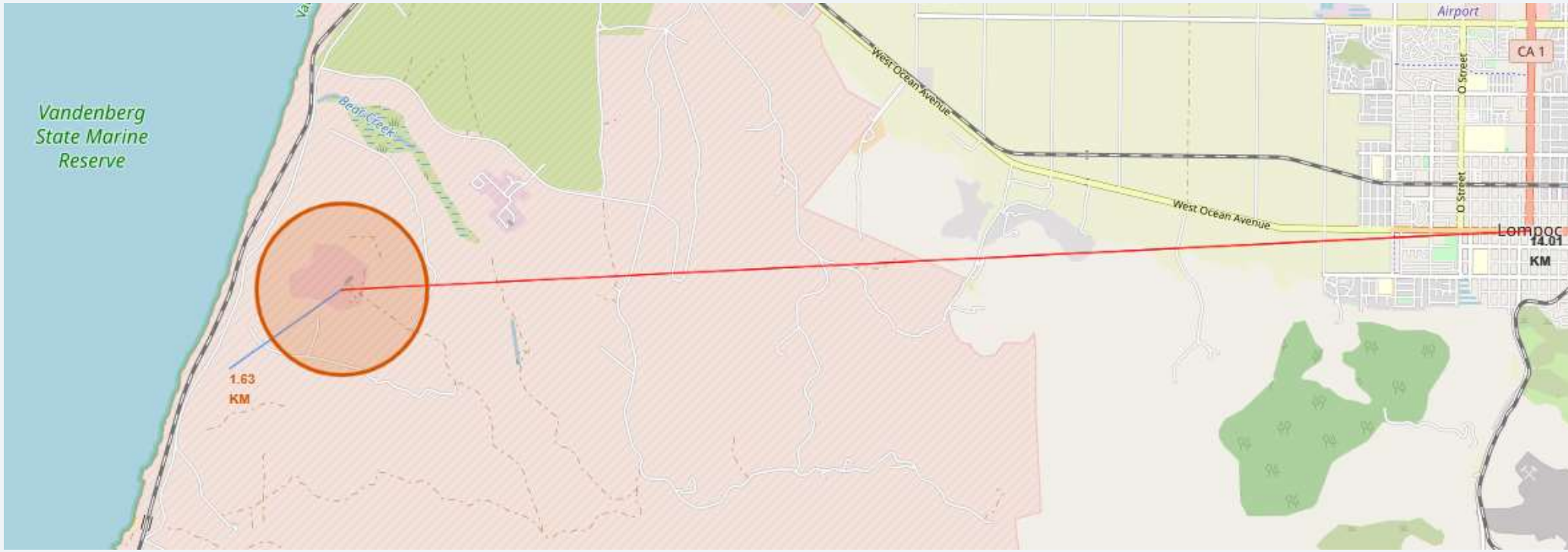
Launch Locations Map

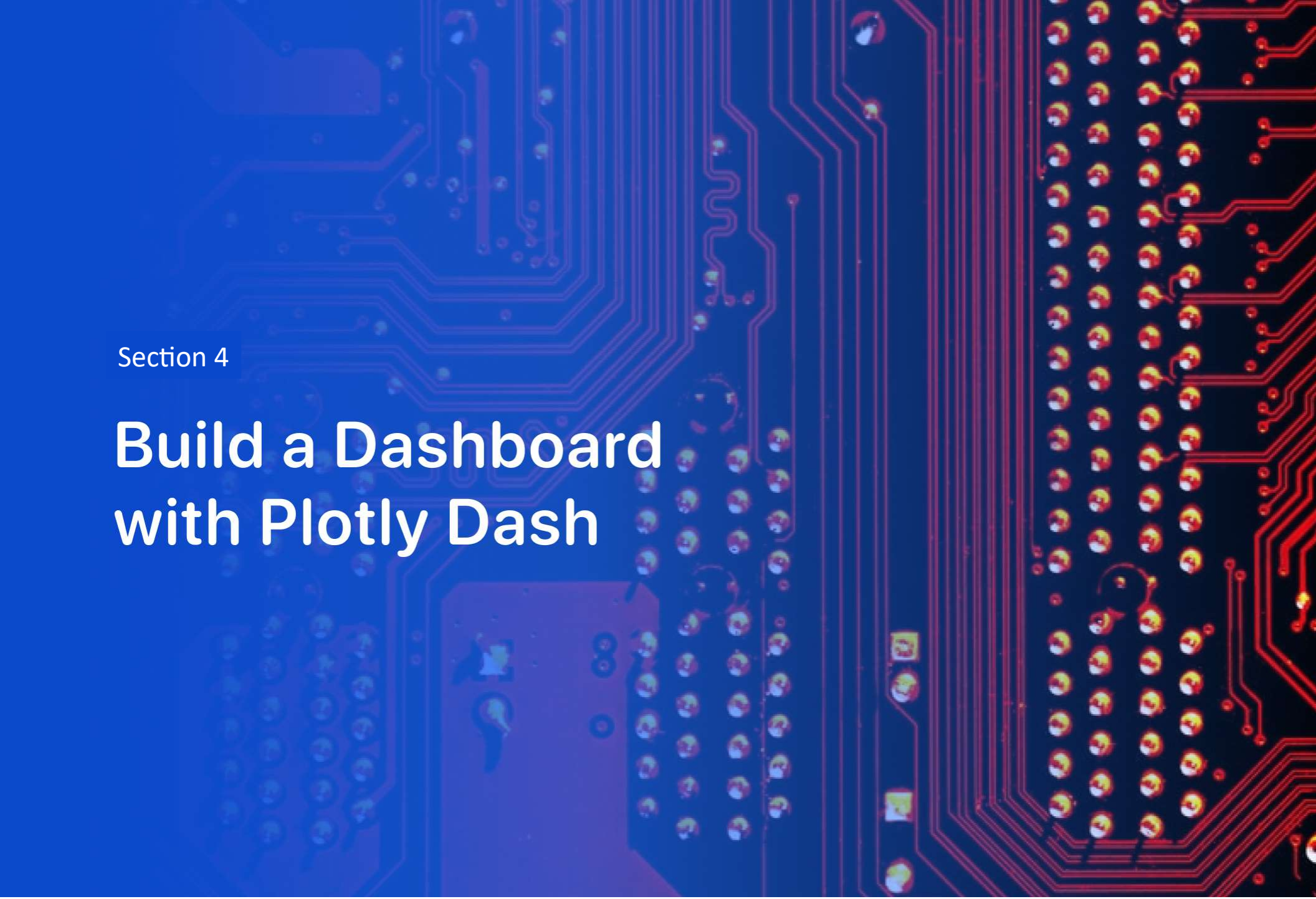
Link:
https://github.com/e-m-vanderveen/IBM-Data-Science-Capstone/blob/main/lab-jupyter-launch-site-location-v2.ipynb

Launch Outcome Marker Cluster



<Folium Map Screenshot 3>





Section 4

Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard

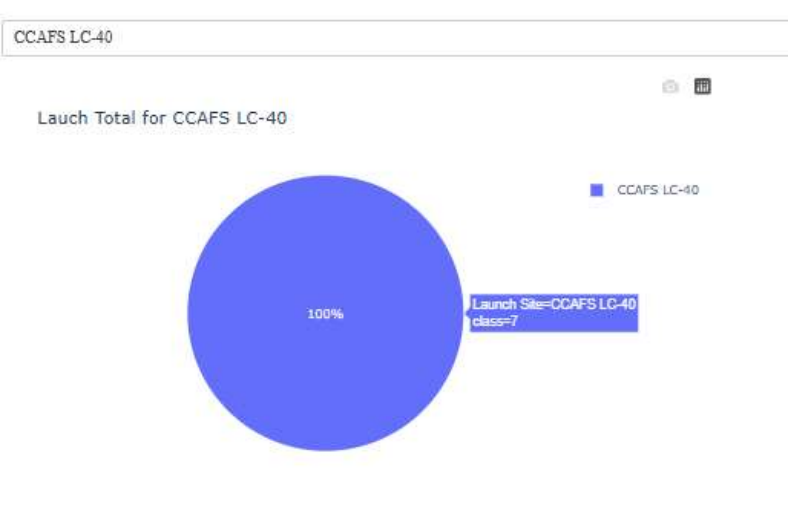
Show the screenshot of launch success count for all sites, in a piechart

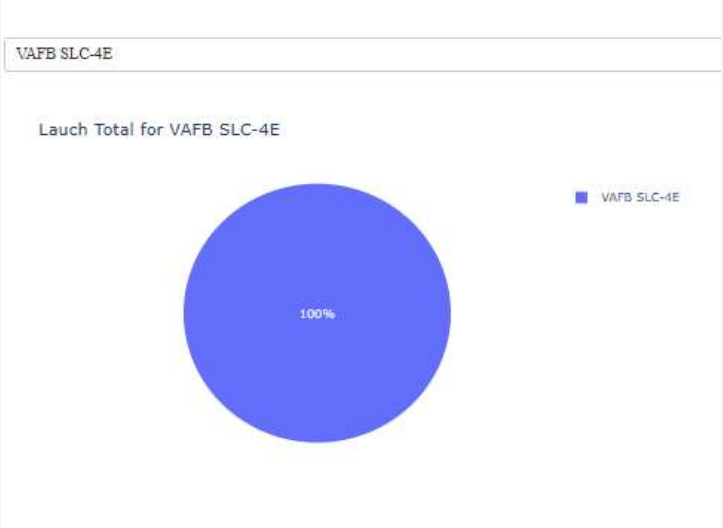
Explain the important elements and findings on the screenshot

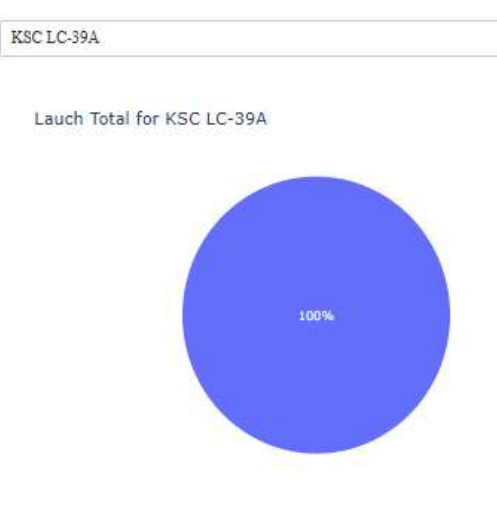
Link:
https://github.com/e-m-vanderveen/IBM-Data-Science-Capstone/blob/main/spacex_dash_app.py

<Dashboard Screenshot 2>

Interesting that the course assignment creates the piechart as a percentage of all launches that occurred at a particular site. Not t percentage of successful launches per site. Accordingly, all the sit specific pie charts show 100%, which is not very useful. However, the result that one gets when following the course instructions.

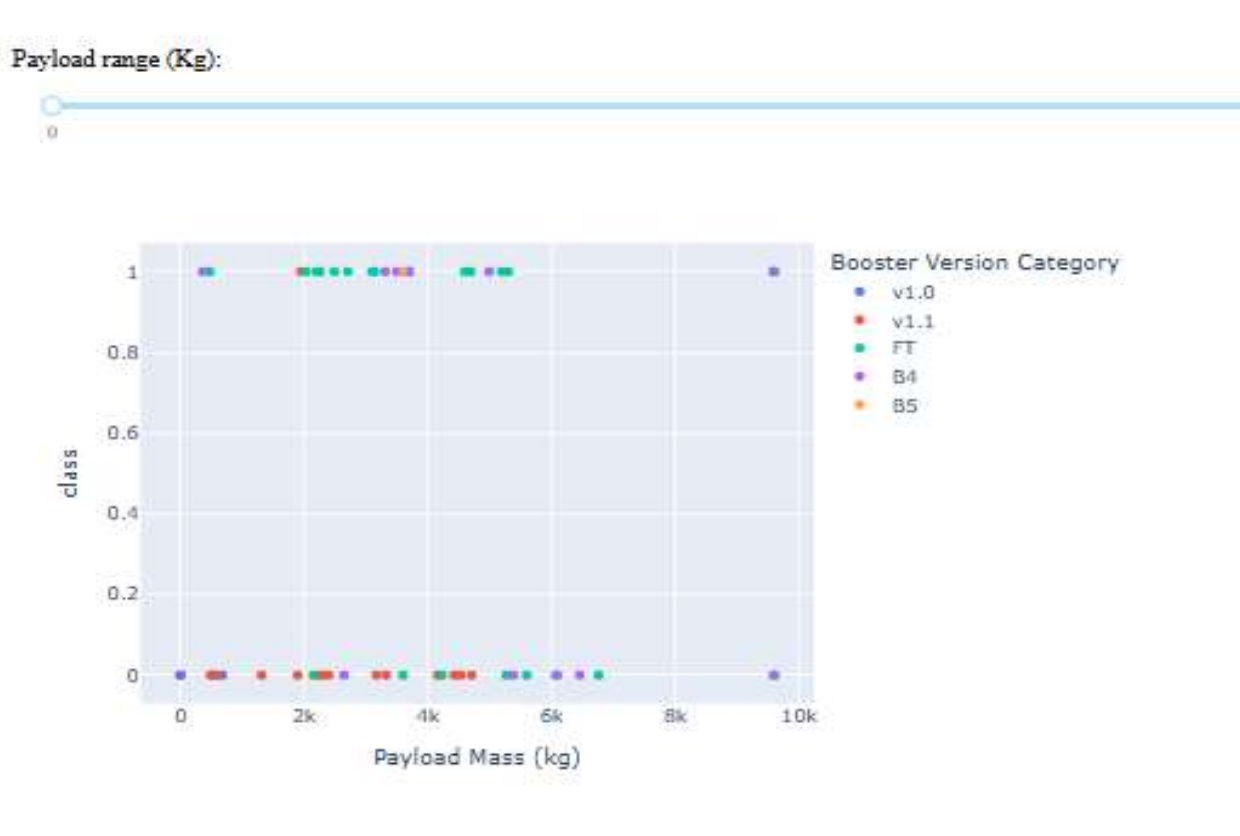






<Dashboard Screenshot 3>

Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, different payload selected in the range slider



Section 5

Predictive Analysis (Classification)

Accuracy

```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

0.0s

```
tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

```
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

0.0s

```
tuned hpyerparameters :(best parameters)  {'C': np.float64(1.0), 'gamma': np.float64(0.03162277660168379), 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

0.0s

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'spli
accuracy : 0.8767857142857143
```

```
print("tuned hpyerparameters :(best parameters) ",knn_cv.best_params_)
print("accuracy :",knn_cv.best_score_)
```

0.0s

```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858
```

Link:
https:/
-vando
-Scier
b/mair
ne-Le
-Part-5

Classification Accuracy

Find the method performs best:

```
method = [{'name': 'knn', 'score': knn_score}, {'name': 'svm', 'score': svm_score}, {'name': 'logreg', 'score': logreg_score}, {'name': '

for i in method:

    print(f"this is the score of the {i['name']} method is {i['score']}")
```

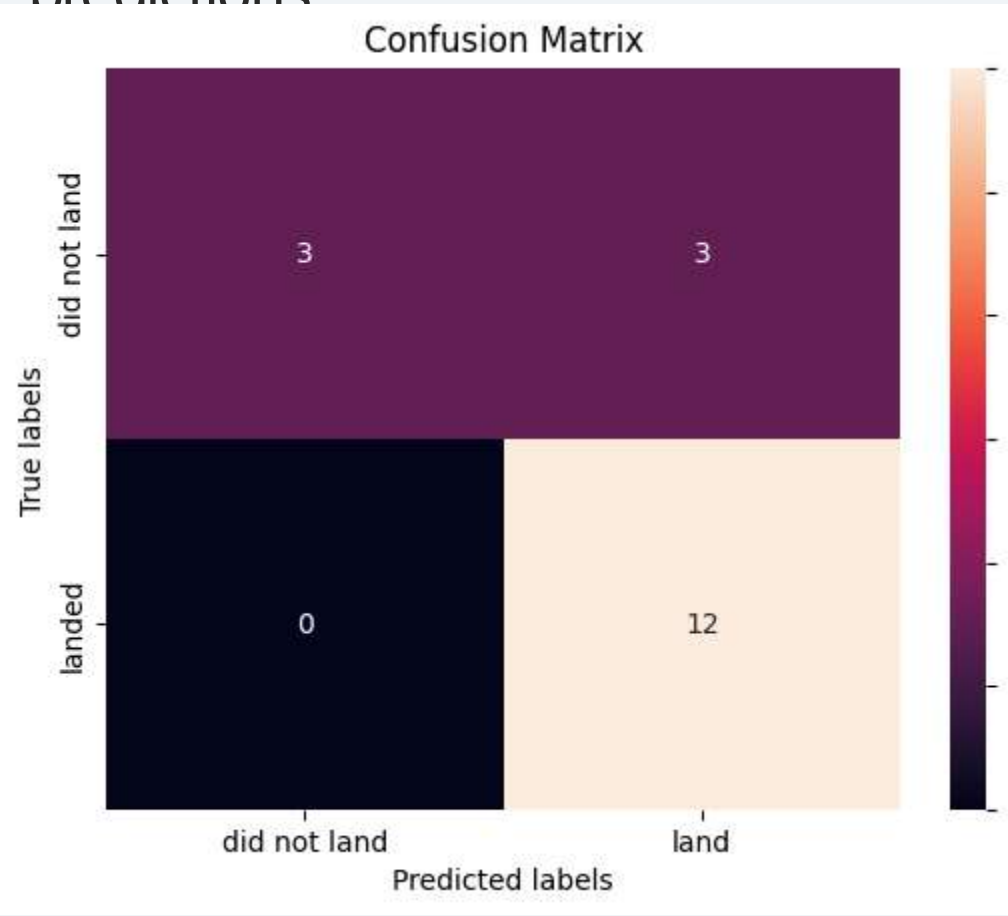
1] ✓ 0.0s

```
this is the score of the knn method is 0.8333333333333334
this is the score of the svm method is 0.8333333333333334
this is the score of the logreg method is 0.8333333333333334
this is the score of the tree method is 0.8333333333333334
```

Confusion Matrix

- All models produced an identical confusion matrix with no false negatives, and three false positive predictions

Link:
https://github.com/e-m-vanderveen/IBM-Data-Science-Capstone/blob/main/SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb



Thank you!