IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Christopher Eberhart
12/02/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through Web API.
  - Data loaded into a relation database to utilize SQL commands to cleanup and normalize the data.
  - Utilized data visualization of static and interactive charts to approach correlations.
  - Initiated machine learning algorithms to verify hypothesis from visualizations.
- Summary of all results
  - The results were striking in that heavier payloads from the Kennedy Space Center LC-39A launch site would yield the most success for the Falcon 9 booster.

Introduction

- Project background and context
  - In order to maximize the reduction in costs compared to competitors and ensure the success of a launch, it is imperative to examine the real-world success of data. By increasing the chances of success, SpaceX can provide the best value compared to its competitors.
- Problems you want to find answers
  - What factors were more likely to reenter and land successfully.
  - What characteristics of a launch had a higher probability of success.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected through utilization of the web scraping API's in Python through publicly available data.
- Perform data wrangling
  - Missing data points were dropped from the analysis in order to preserve real-world outcomes.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Compared the outcome of various modeling to find the best predictor.

Data Collection

- Describe how data sets were collected.
  - Data collection was through utilizing Python and its extensive libraries to scrape the web through built-in API into a dataframe.

Data Collection – SpaceX API

- Utilize a web API call to SpaceX to collect the data needed to complete the analysis.

- Link for API code is https://github.com/SineOverdirve/CapstoneProject/blob/07001f00bd848873744b6372b5f7c95b1c6d9200/jupyter-labs-spacex-data-collection-api.ipynb

1. Get request for rocket launch data using API

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]: response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json result to dataframe

```
In [12]: # Use json_normalize method to convert the json result into a dataframe

         # decode response content as json
         static_json_df = res.json()
```

```
In [13]: # apply json_normalize
         data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

```
In [30]: rows = data_falcon9['PayloadMass'].values.tolist()[0]

         df_rows = pd.DataFrame(rows)
         df_rows = df_rows.replace(np.nan, PayloadMass)

         data_falcon9['PayloadMass'][0] = df_rows.values
         data_falcon9
```

Data Collection - Scraping

- Utilizing BeautifulSoup in Python, scrapped data from Wikipedia on the Falcon 9.
- https://github.com/SineOverdirve/CapstoneProject/blob/17c14a6722b62db1d53ddfd799dcce63d0703c4f/jupyter-labs-webscraping.ipynb

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]:
```
200
```

2. Create a BeautifulSoup object from the HTML response

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

In [10]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

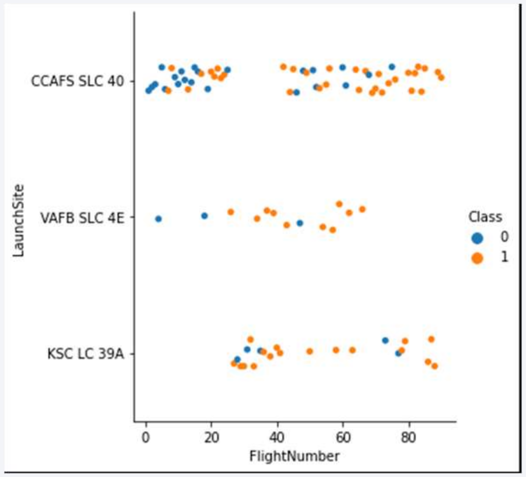
4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

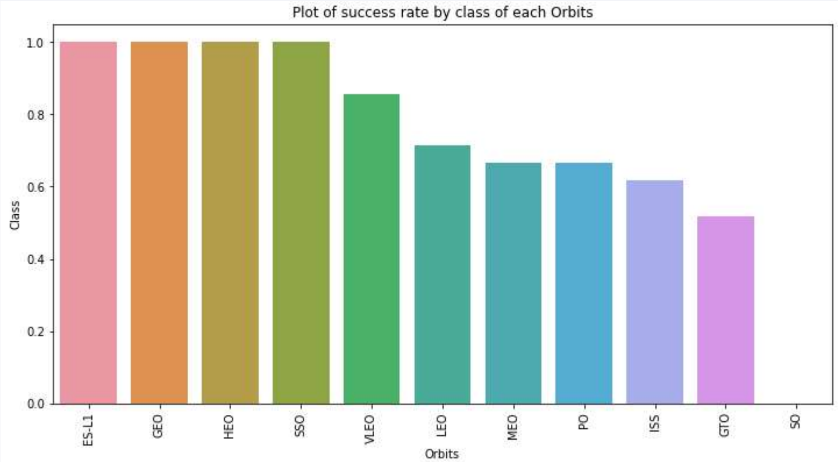
Data Wrangling

- Combined the data from the SpaceX API call data and what was scrapped from the Falcon 9 Wikipedia page.
- Calculated number of launches at each site along with destination orbital region.
- Through wrangling also created a binary outcome of the launch if success or failed.
- https://github.com/SineOverdirve/CapstoneProject/blob/17c14a6722b62db1d53ddfd799dcce63d0703c4f/Capstone%20Project%20Data%20Wrangling.ipynb

EDA with Data Visualization

- Utilized a scatter plot of payload vs launch site to find the better location along with a bar chart of the success rate by orbit.





- https://github.com/SineOverdirve/CapstoneProject/blob/3a87b5d4e73ef8cba4244ee6a5c5f4b0628fc780/EDA%20with%20Visualization%20Lab.ipynb

EDA with SQL

- Loaded the SpaceX dataset into a PostgreSQL database and confirmed data loaded correctly with jupyter notebook.
- Applied EDA utilizing SQL to gain insight through queries:
  - The unique launch site names and identifiers.
  - The payload mass carried by each Falcon 9 launch.
  - The average payload mass carried by the Falcon 9 booster.
  - The total count of successful and failure missions.
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- https://github.com/SineOverdirve/CapstoneProject/blob/3a87b5d4e73ef8cba4244ee6a5c5f4b0628fc780/Capstone%20EDA%20w_SQL.ipynb

Build an Interactive Map with Folium

- Utilizing the Folium library, all launch sites are marked with a marker, circle, and text. Additionally, each site shows the number of launches with a marker along with green for successful and red for failure.
- These objects were added to provide us with a quick visual representation of what sites were used more often along with which had more successful uses of the Falcon 9 booster.
- https://github.com/SineOverdirve/CapstoneProject/blob/3a87b5d4e73ef8cba4244ee6a5c5f4b0628fc780/Launch%20Site%20Location%20Lab.ipynb

Build a Dashboard with Plotly Dash

- Through the use of the Plotly library, interactive charts were able to be examined to look for trends.
- Through using a scatter plot of the payload vs the Falcon 9 booster version, we gain insight into what version is showing the most success.
- Utilizing a pie chart of the launch locations and number of successful missions, we can visually see which sites provide the best probability of a successful outcome.
- https://github.com/SineOverdirve/CapstoneProject/blob/3a87b5d4e73ef8cba4244ee6a5c5f4b0628fc780/spacex_dash_app.py

Predictive Analysis (Classification)

- Utilized the numby, pandas, and scikit learn libraries to be able to ETL the data and run against various ML algorithms.
- By spinning the data and pushing it through the various classification models, we were able to determine that the decision tree provided the most accurate predictor.
- https://github.com/SineOverdirve/CapstoneProject/blob/3a87b5d4e73ef8cba4244ee6a5c5f4b0628fc780/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- The decision tree predictor gave us an accuracy against the test data of 87.5%

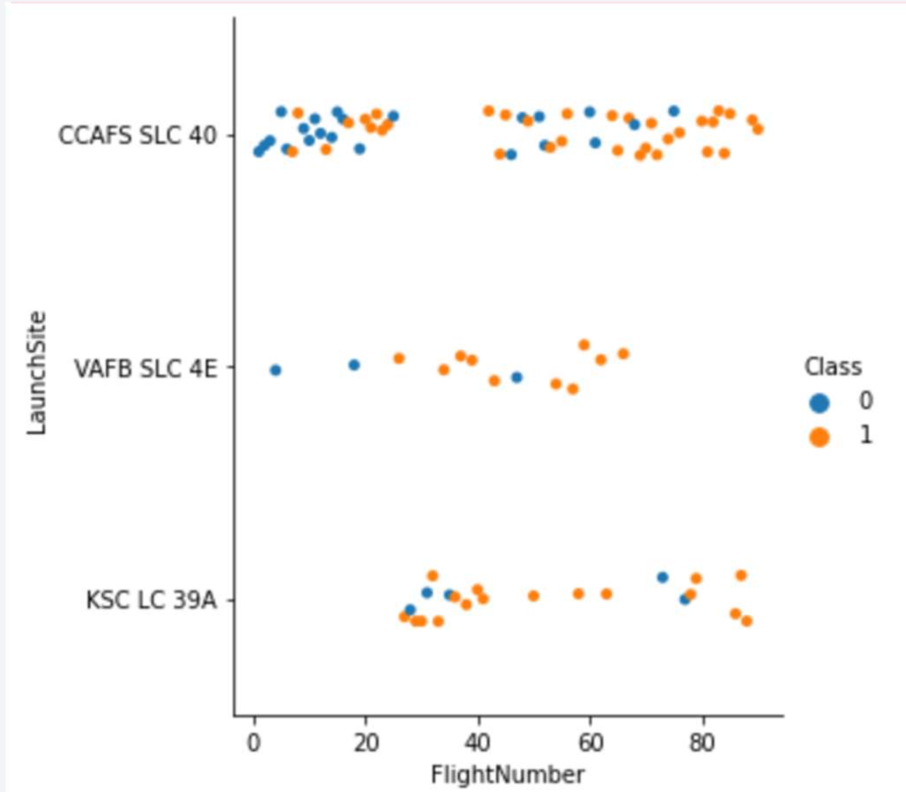
```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 14, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_spli
t': 10, 'splitter': 'random'}
accuracy : 0.875
```

Section 2

Insights drawn from EDA

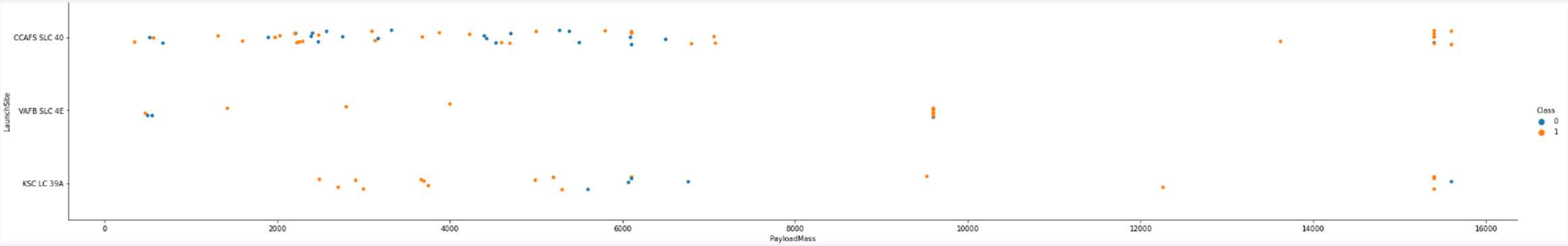
Flight Number vs. Launch Site

- Through a scatter plot visualization, we were able to determine that LC-39A at Kennedy Space Center provided the highest probability of success.



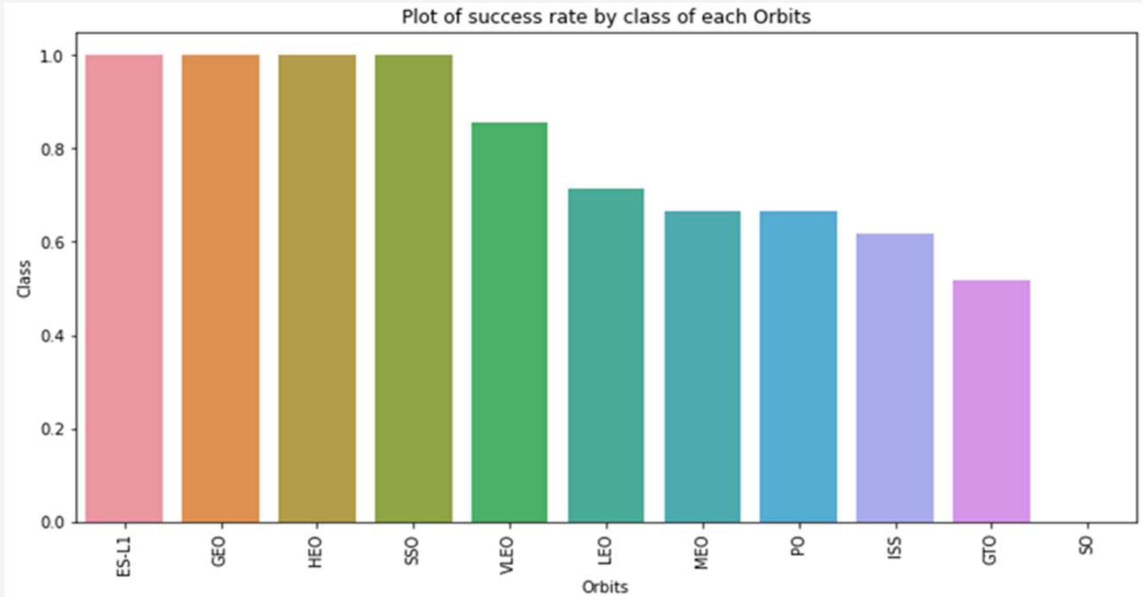
Payload vs. Launch Site

- Utilizing a scatter plot visualization, we were able to see that heavier payloads led to a high chance of success at each site.



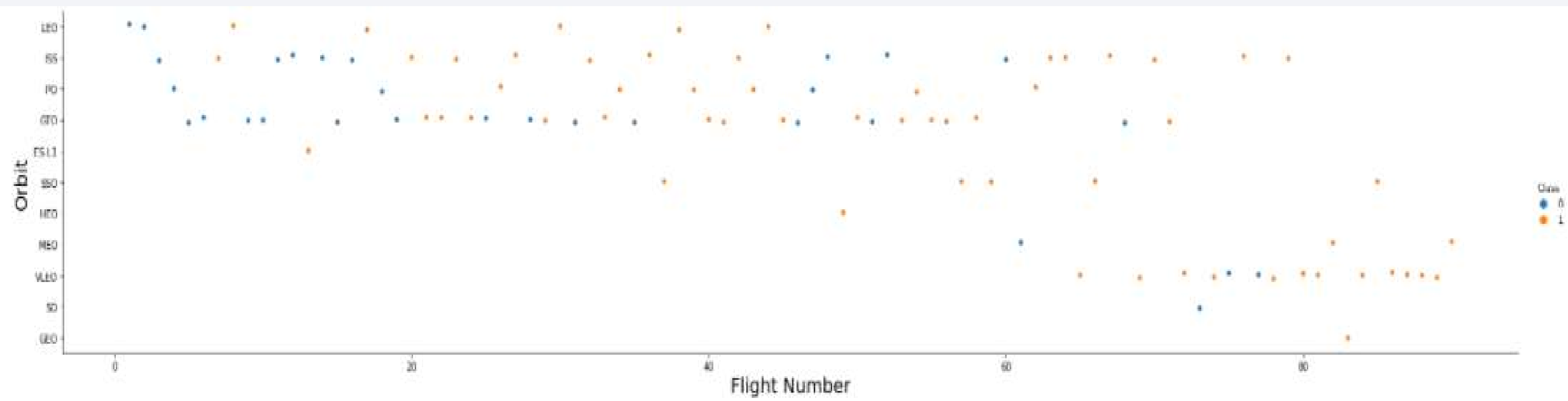
Success Rate vs. Orbit Type

- Using a bar chart visualization, we can see that ES-L1, GEO, HEO, SSO, and VLEO have the higher probability of success.



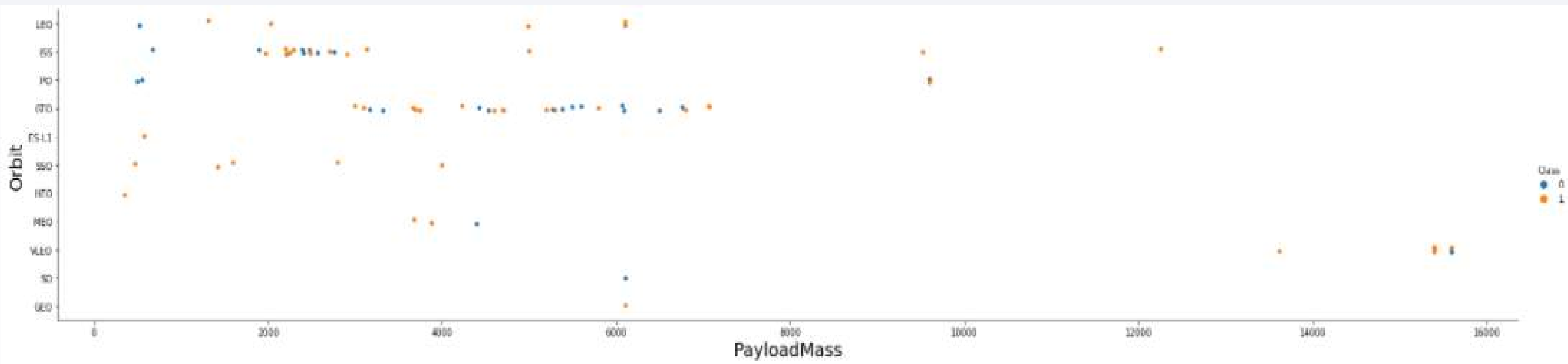
Flight Number vs. Orbit Type

- Using a scatter plot visualization of the orbit type vs the iteration of attempts, we can see a strong correlation of the GTO orbit leading to more successes.



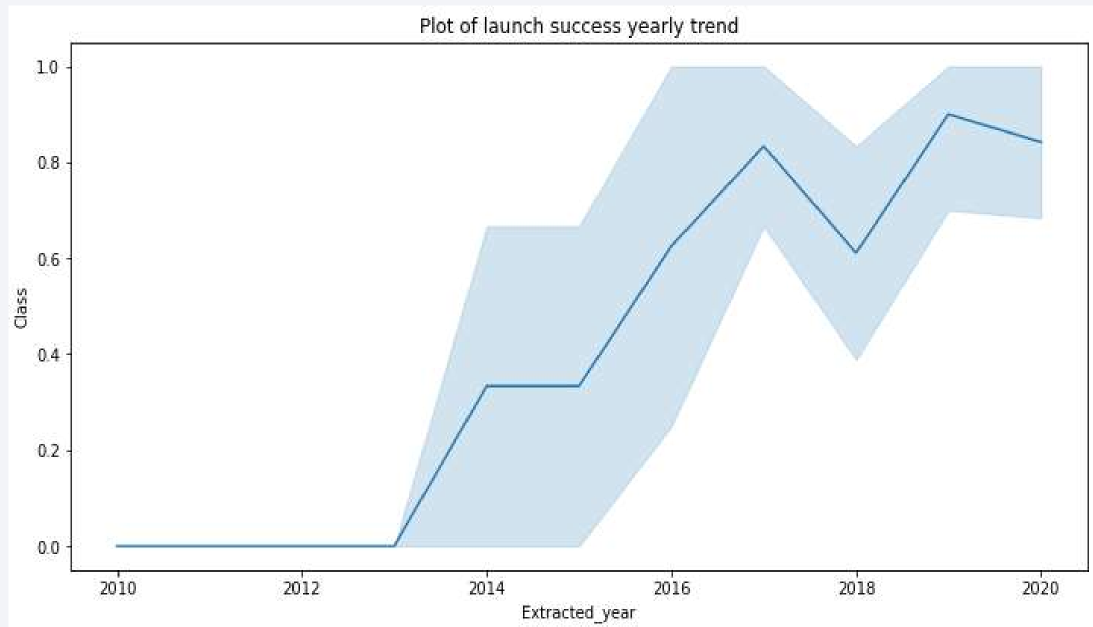
Payload vs. Orbit Type

- Through the use of a scatter plot of payloads vs orbit, there is a greater chance of success using the LEO and ISS orbits.



Launch Success Yearly Trend

- On the line plot, we can see that the success rate has increased quite rapidly from 2013 to 2017 and has held nearly steady at 80%.



All Launch Site Names

- Attached is a screenshot of the query used to compile the unique launch sites utilized by SpaceX for the Falcon 9 booster.

```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'KSC'

- Find 5 records where launch sites' names start with `KSC`

```
%sql SELECT * FROM SPACEX_SQL_EDA WHERE LAUNCH_SITE LIKE '%KSC%' LIMIT 5
```

```
 * ibm_db_sa://kdt73268:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TotalPayload, CUSTOMER FROM SPACEX_SQL_EDA WHERE CUSTOMER LIKE '%NASA (CRS)%' GROUP BY CUSTOMER
```

```
 * ibm_db_sa://kdt73268:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

totalpayload	customer
45596	NASA (CRS)
2617	NASA (CRS), Kacific 1

Average Payload Mass by F9 v1.1

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

```
%sql SELECT BOOSTER_VERSION, AVG(PAYLOAD_MASS__KG_) FROM SPACEX_SQL_EDA WHERE BOOSTER_VERSION LIKE '%F9 v1.1%' GROUP BY BOOSTER_VERSION
```

```
 * ibm_db_sa://kdt73268:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

booster_version	2
F9 v1.1	2928

First Successful Ground Landing Date

- The screenshot of the query shows that the first successful landing of a booster to the ground was on December 22nd, 2015.

```
%sql SELECT Landing__Outcome, MIN(DATE) AS MinDate FROM SPACEX_SQL_EDA GROUP BY LANDING__OUTCOME
```

```
 * ibm_db_sa://kdt73268:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

landing__outcome	mindate
Controlled (ocean)	2014-04-18
Failure	2018-12-05
Failure (drone ship)	2015-01-10
Failure (parachute)	2010-06-04
No attempt	2012-05-22
Precluded (drone ship)	2015-06-28
Success	2018-07-22
Success (drone ship)	2016-04-08
Success (ground pad)	2015-12-22
Uncontrolled (ocean)	2013-09-29

Successful Drone Ship Landing with Payload between 4000 and 6000

- Using a Select query with a between in the where clause, we are able to find all the drone ship landings that were successful that carried payloads between 4000 to 6000 kg.

```
SELECT BoosterVersion
FROM SpaceX
WHERE LandingOutcome = 'Success (drone ship)'
    AND PayloadMassKG > 4000
    AND PayloadMassKG < 6000
...
```

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Through a count with a group by, we are able to count the number of missions that resulted in a success or failure. This approach was used instead of wild card on “fail” or “success” to show the breakdown of each category.

```
%sql SELECT Landing__Outcome, CouNT(Landing__Outcome) AS Cntr FROM SPACEX_SQL_EDA WHERE Landing__Outcome LIKE '%Success%' OR Landing__Outcome LIKE '
```

```
 * ibm_db_sa://kdt73268:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

landing__outcome	cntr
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Success	38
Success (drone ship)	14
Success (ground pad)	9

Boosters Carried Maximum Payload

- The attached screenshot shows all boosters that launched at max capacity.

```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEX_SQL_EDA WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX_SQL_EDA)
```

```
 * ibm_db_sa://kdt73268:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
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

2017 Launch Records

- The attaches screenshot shows the records by month names that resulted in successful landing outcomes in the year 2017.

```
%sql SELECT MonthName(Date) AS MoName, Landing__Outcome, Booster_Version, Launch_Site FROM SPACEX_SQL_EDA WHERE Year(Date)=2017 AND UPPER(Landing__C
```

```
 * ibm_db_sa://kdt73268:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

moname	landing__outcome	booster_version	launch_site
February	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
May	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
June	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
August	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
September	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
December	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

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

- The attached screenshot shows the query and count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

```
%sql SELECT Landing__Outcome, Count(Landing__Outcome) FROM SPACEX_SQL_EDA WHERE Date >= '20100604' AND Date <='20170320'  AND Landing__Outcome LIKE
```

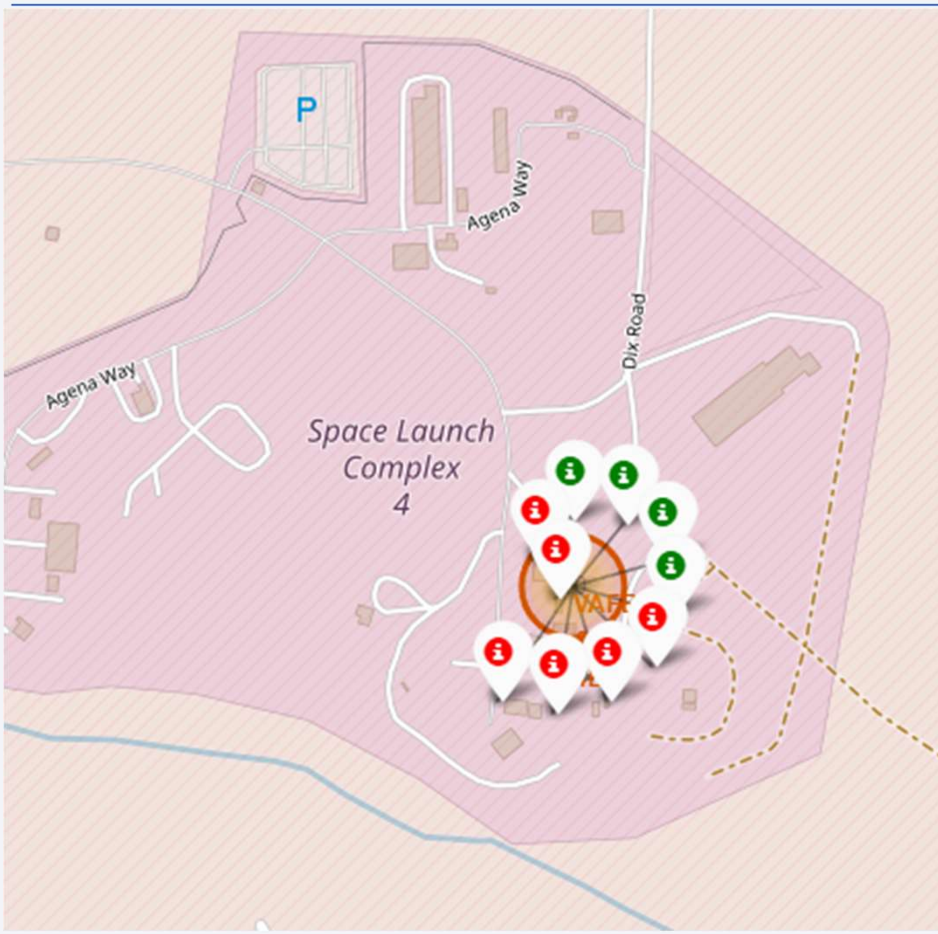
```
 * ibm_db_sa://kdt73268:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
Done.
```

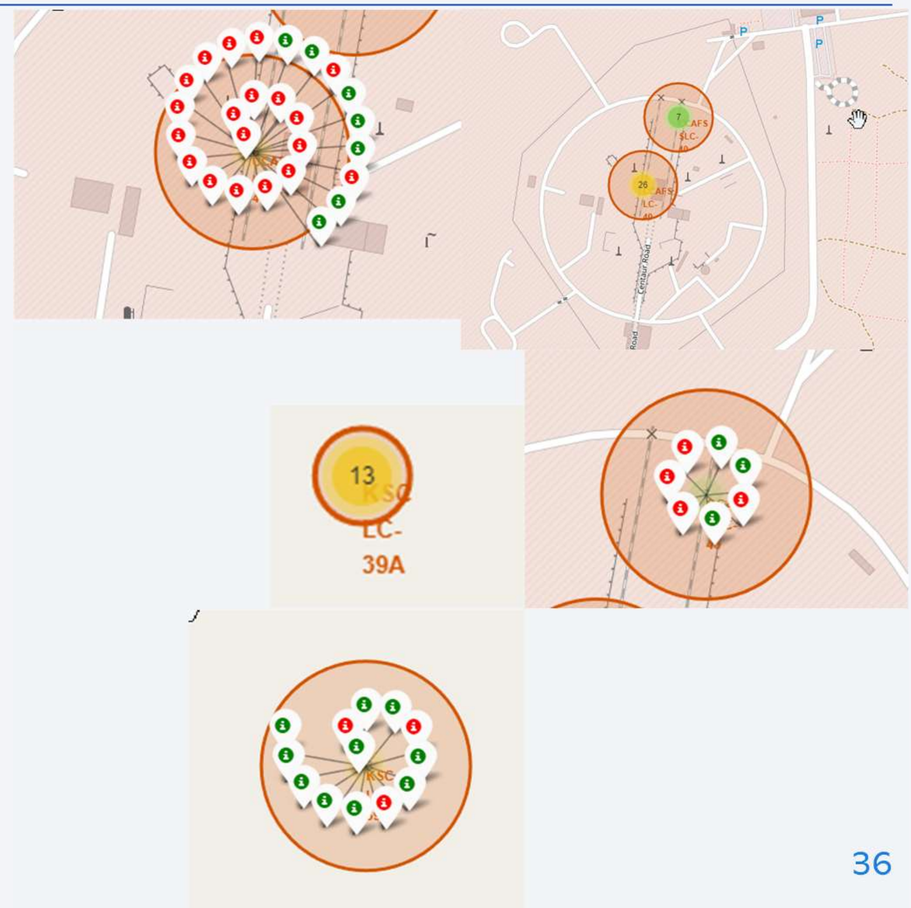
landing__outcome	2
Success (drone ship)	5
Success (ground pad)	3


Section 3
Launch Sites
Proximities Analysis

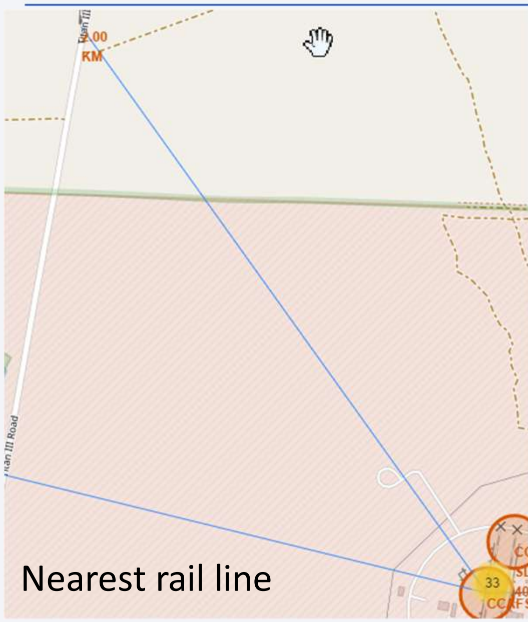
All launch sites displayed

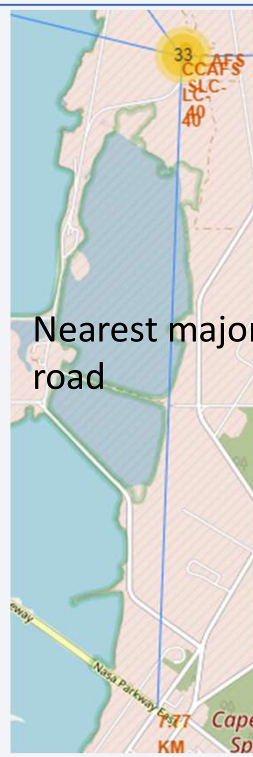
Launch Site Markers with Success Icons

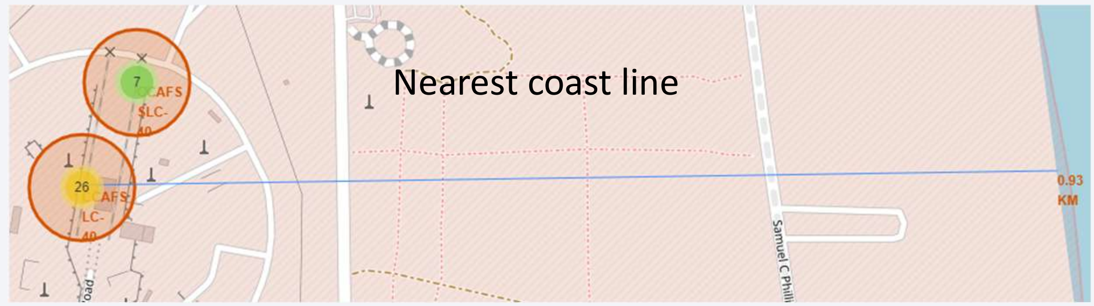


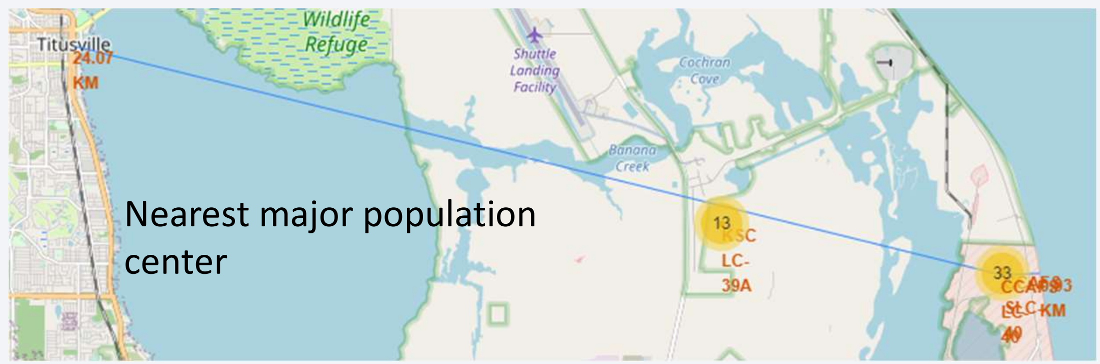


Showing Distances to Major land markers





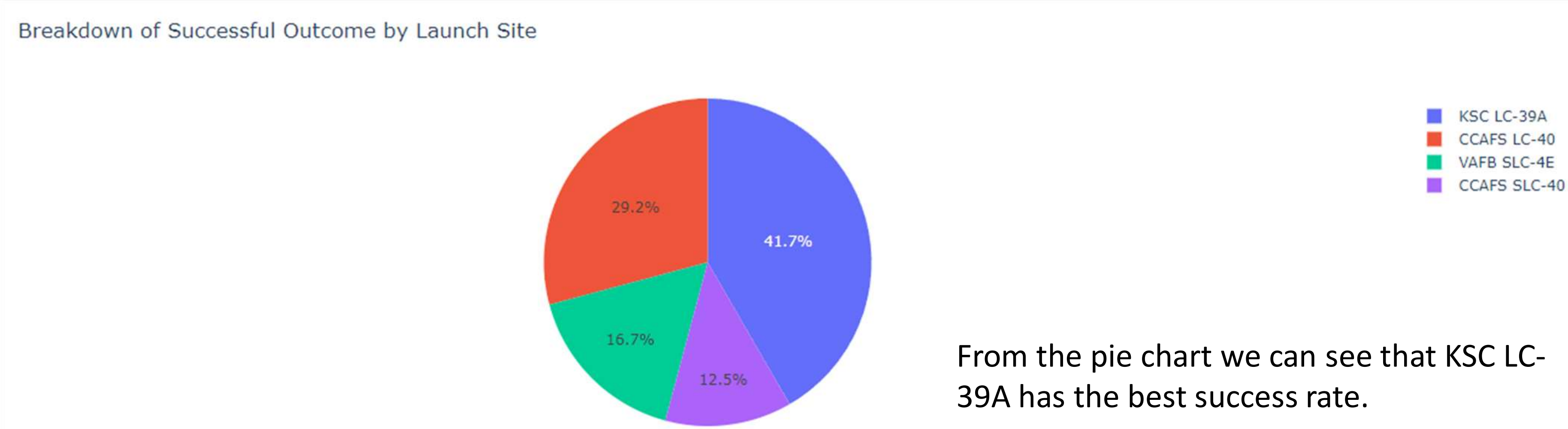




Section 4

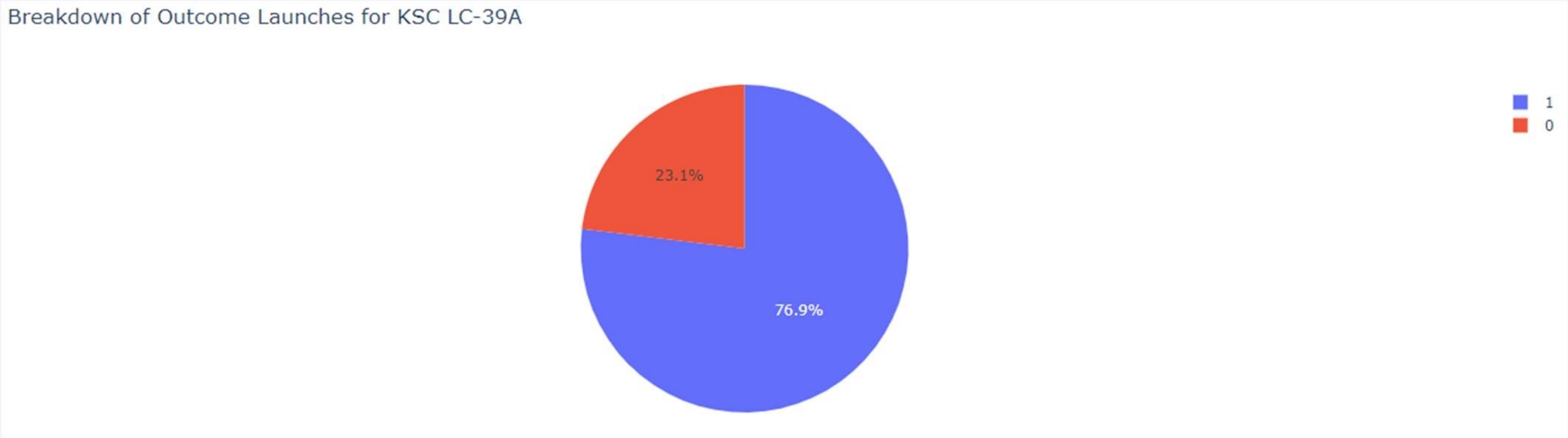
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



From the pie chart we can see that KSC LC-39A has the best success rate.

Pie chart showing the KSC LC-39A site success rate



Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

We can see that heavier payloads led to a higher success rate.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Below is a screenshot showing the various classification models along with their best score. We can see that the decision tree method led to the most accurate results.

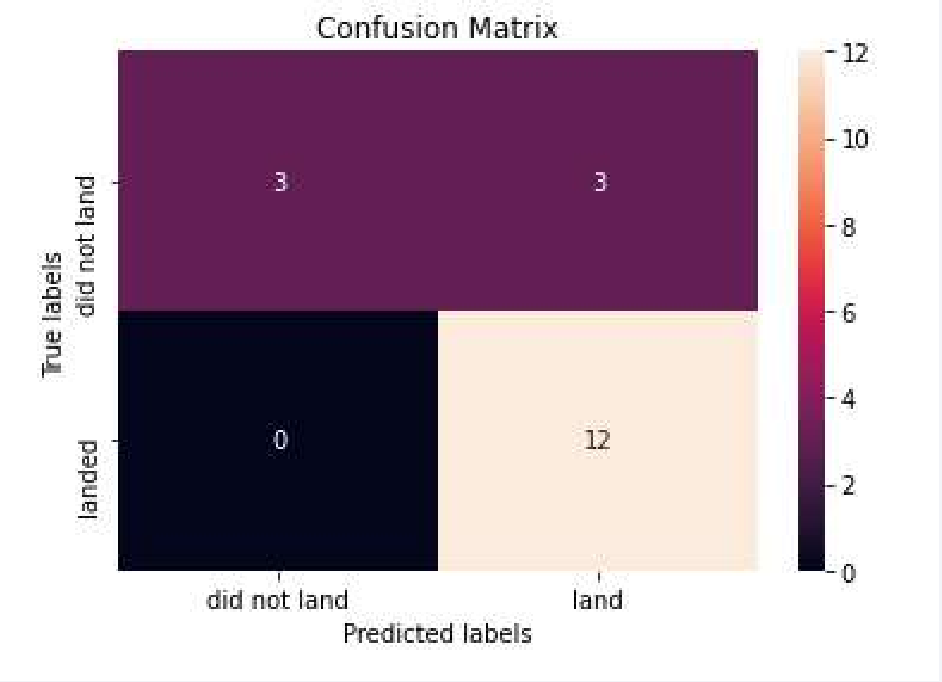
```python
print('Accuracy for Logistics Regression method:', logreg_cv.best_score_)
print( 'Accuracy for Support Vector Machine method:', svm_cv.best_score_)
print('Accuracy for Decision tree method:', tree_cv.best_score_)
print('Accuracy for K nearsdt neighbors method:', knn_cv.best_score_)
```

✓ 0.4s Python

```
Accuracy for Logistics Regression method: 0.8464285714285713
Accuracy for Support Vector Machine method: 0.8482142857142856
Accuracy for Decision tree method: 0.875
Accuracy for K nearsdt neighbors method: 0.8482142857142858
```

Confusion Matrix

- Attached is the confusion matrix for the decision tree classification model using the best parameters.



Conclusions

In conclusion the following can be gleaned from the data:

- The best launch site is LC-39A at Kennedy space center.
- If the target orbit is ES-L1, GEO, HEO, SSO, or VLEO then the booster has a greater chance of reusability.
- We can calculate the probability of success to 87.5% using the decision tree classification machine learning model.

Knowing this we will be able to adjust our pricing to remain competitive and profitable based on the input parameters.

Thank you!