IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chris Jones
Friday, April 22, 2022

Outline

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

Executive Summary

- The mission of this data dive was to hypothesize whether or not the Falcon 9 spacecraft of SpaceX, would successfully land during the first stage.
- By using IBMCP we were able to collect data from Wikipedia and a REST API using webscraping, build a dashboard to analyze launch records, build an interactive map to analyze the launch site proximity with Folium, and then use machine learning to determine if the first stage of Falcon 9 will land successfully.
- Based on the results, SpaceX could provide a cheaper and more efficient option for space exploration in the future.
- We were able to use 4 machine learning models: Logistic regression, Support Vector Machine, Decision Trees, and K Nearest Neighbor, which, all produced similar results. The results being an accuracy rate around 83%, with the models seemingly predicting successful landings more of the time.

Introduction

- In this capstone, we will predict if the Falcon 9 first stage will land successfully.
- SpaceX advertises Falcon 9 rocket launches with a cost of $62 million.
- Other providers cost upward of $165 million each. Much of the savings is because SpaceX can reuse the first stage.
- If we can predict if the first stage will land, we can ultimately determine the cost.
- Other companies may like this information, because the information can be used to determine if the company wants to bid against SpaceX.
- What we hope to accomplish: Train some ML models to predict whether the Stage 1 landing will be successful or not.

Section 1

Methodology

Methodology

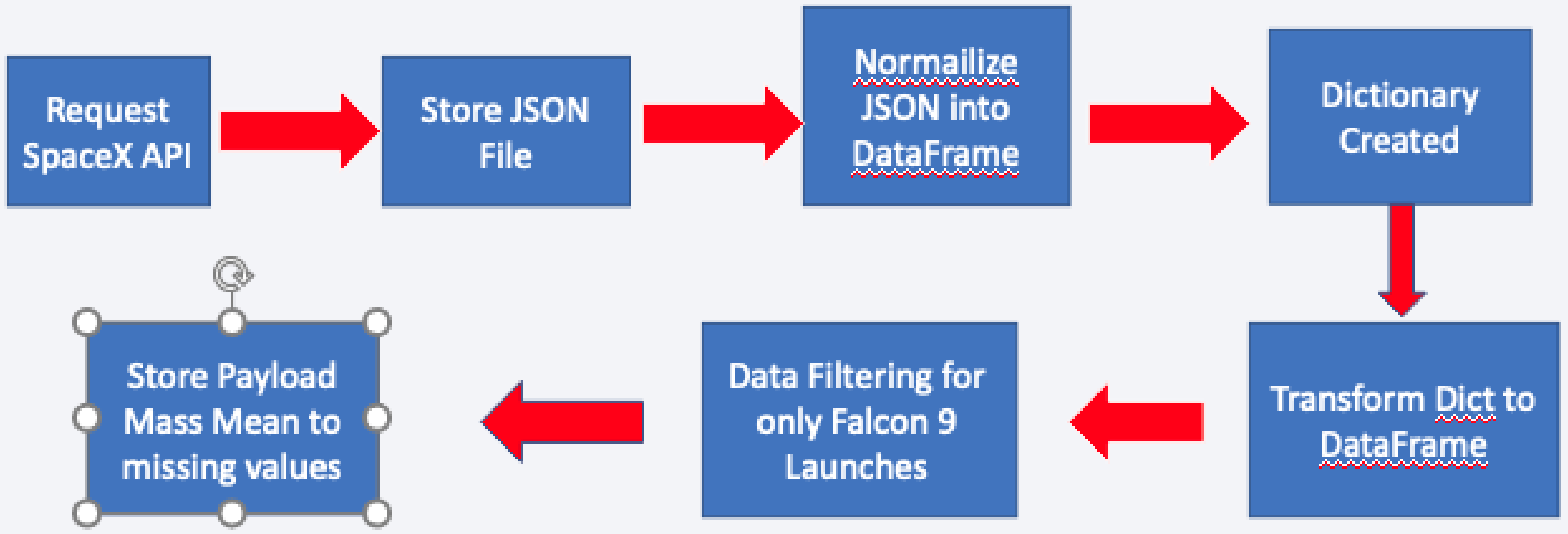
Executive Summary

- Data collection methodology:
  - Merged data from Space X Wikipedia and API
- Perform data wrangling
  - Identified landings as successful or unsuccessful, created a class column
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Using GridSearchCV we were able to fine tune models

Data Collection

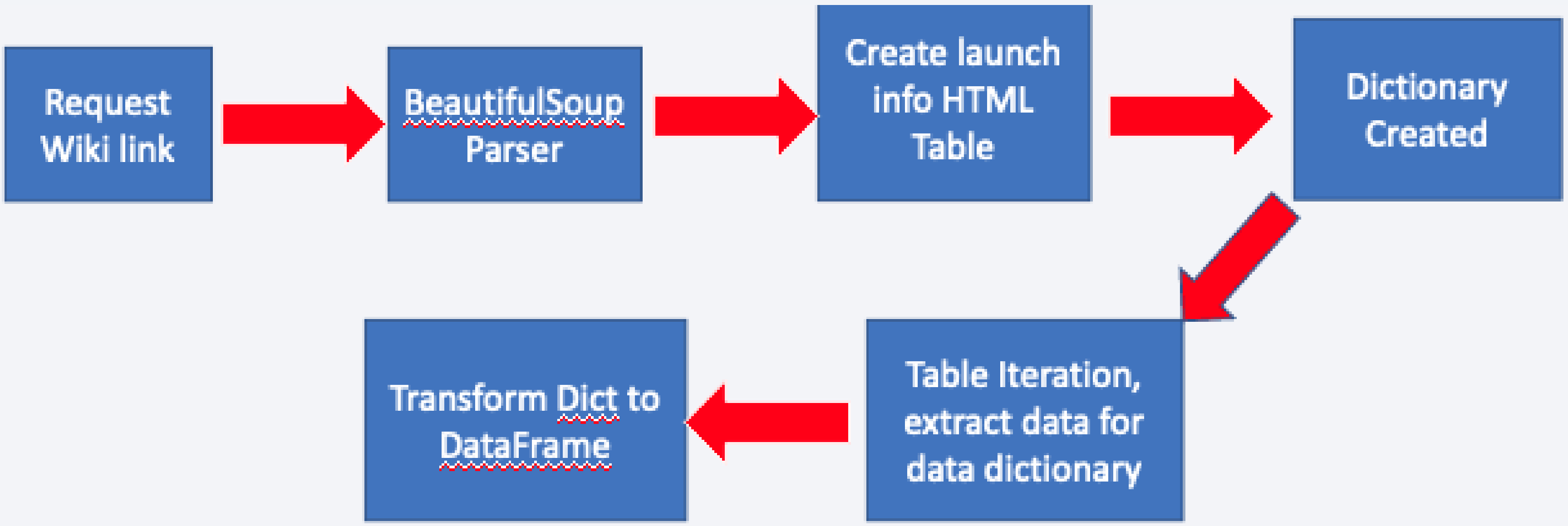
- Datasets were collected by combining Web scraping techniques on SpaceX's Wikipedia page, and API requests for SpaceX's API.
- In the next couple of slides, I have shown the flowcharts of the API data collection and Web Scraping data collection, along with the notebooks through Gitbhub.

Data Collection – SpaceX API



- https://github.com/cljones97/DataSci_IBM_Project/blob/master/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping



- https://github.com/cljones97/DataSci_IBM_Project/blob/master/jupyter-labs-webscraping.ipynb

Data Wrangling

- In this lab we mainly converted outcomes into Training Labels with 1 meaning the booster successfully landed, and 0 meaning it was unsuccessful.
- 1 = True ASDS, True RTLS, & True Ocean
- 0 = None None, None ASDS, False ASDS, False RTLS, & False Ocean

- https://github.com/cljones97/DataSci_IBM_Project/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts.
- Exploratory Analysis performed on columns Flight Number, Launch Site, Payload, Success Rate, Orbit Type, & Yearly Trends of success.
- Plots Used: Flight Number vs. Launch Site, Payload vs. Launch Site, Success Rate vs. Orbit Type, Flight Number vs. Orbit Type, Payload vs. Orbit Type. All Scatter plots to view potential correlation between variables.
- Line plot measuring annual success. Line plot was used to notice patterns in the data that may be consistent or inconsistent.
- https://github.com/cljones97/DataSci_IBM_Project/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Display all the unique Launch Sites
- 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 successful landing outcome in ground pad was achieved.
- List the total number of successful and failure mission outcomes
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- https://github.com/cljones97/DataSci_IBM_Project/blob/main/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- Mark all launch sites on a map. Next, mark the success/failed launches for each site on the map. Finally, Calculate the distances between a launch site to its proximities.
- Proximities include: Railways Highways, Coasts, and Cities close by.
- The markers also show successful and unsuccessful landings close to the launch site.
- These markers may give us a better understanding why these locations may be launch sites in the first place.
- https://github.com/cljones97/DataSci_IBM_Project/blob/main/lab_jupyter_launch_site_location.ipynb

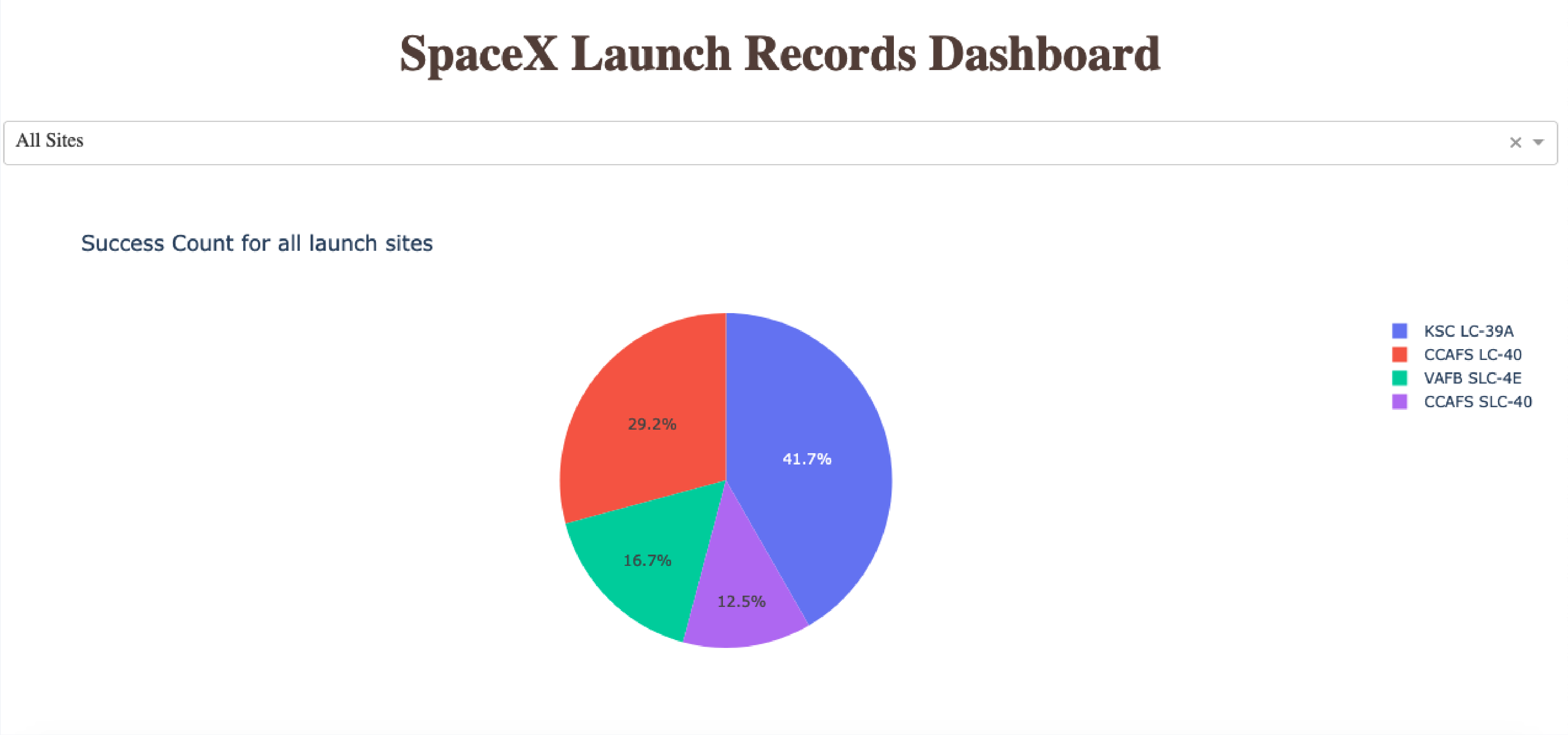
Build a Dashboard with Plotly Dash

- Dashboard includes Pie Charts and Scatter Plots.
- Scatter plot shows the relationship between Launch Sites and Payload. Payload mass can be slid to adjust values. Sites can be all selected or individually selected.
- The Pie Chart shows distribution of successful landings across launch sites. Individual Launch sites may be selected in order to show individua launch site success
- https://github.com/cljones97/DataSci_IBM_Project/blob/main/spaceX_dash.py

Predictive Analysis (Classification)

- To begin we Created a column for the class, Standardize the data, then Split into training data and test data
- Next, we set up parameters, in order for the model to learn from.
- Then, found best Hyperparameter for SVM, Classification, Trees and Logistic Regression
- Fitted and tuned the model with the training data in order to find best parameter and accuracy of model
- https://github.com/cljones97/DataSci_IBM_Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

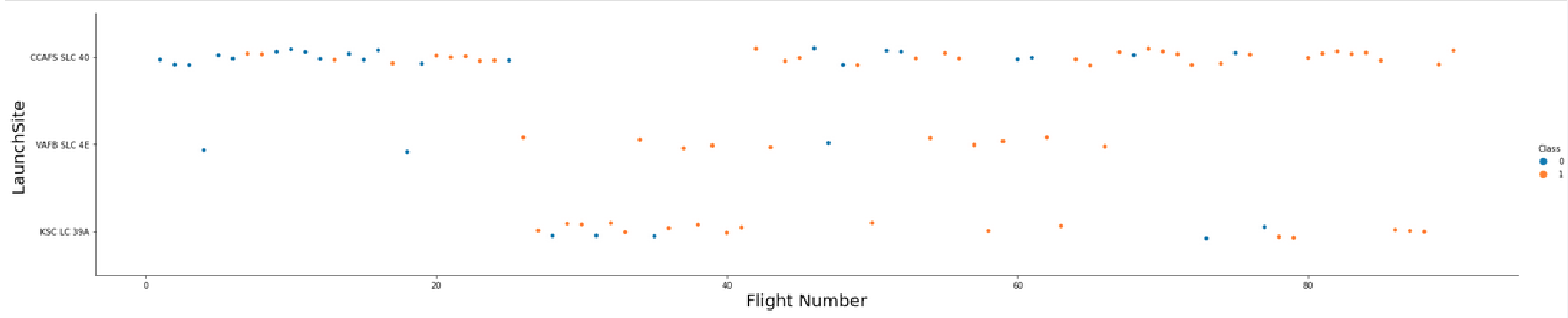
Results


SpaceX Launch Records Dashboard
All Sites
Success Count for all launch sites
29.2%
41.7%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40

Section 2

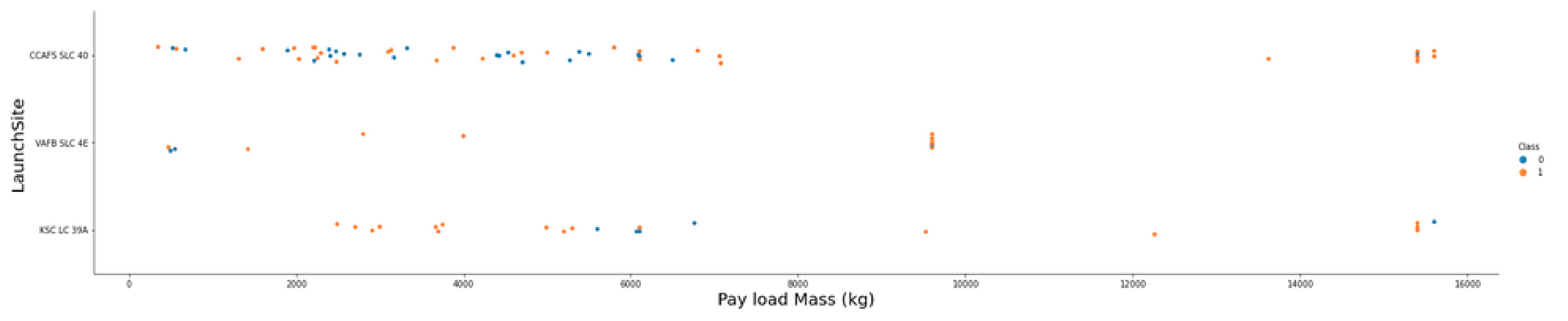
Insights drawn from EDA

Flight Number vs. Launch Site



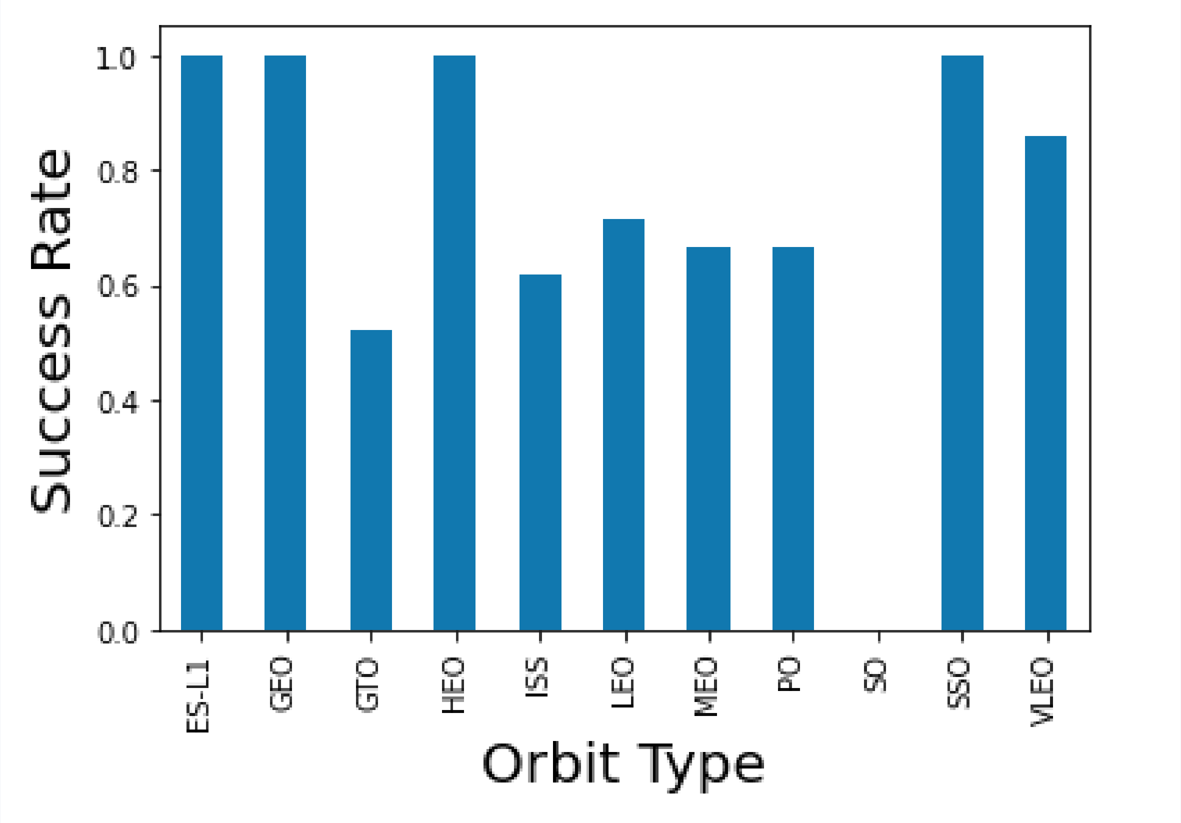
- Here we can see Flight Number vs. Launch Site. With 3 Launch Sites being listed. A pattern to notice is that as the flight number increases, the launches become more successful.

Payload vs. Launch Site



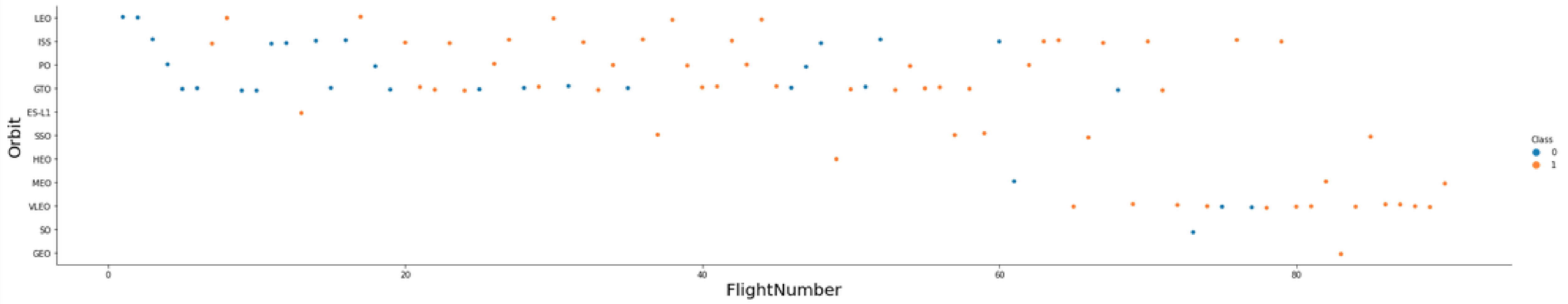
- Here we have Payload vs Launch Site. Some patterns to notice are payloads that approach MAX(Payload) tended to launch from CCAFS SLC 40 & KSC LC 39A. Also, payloads less than 8000 kg tended to fail at a higher rate when launched from CCAFS SLC 40

Success Rate vs. Orbit Type



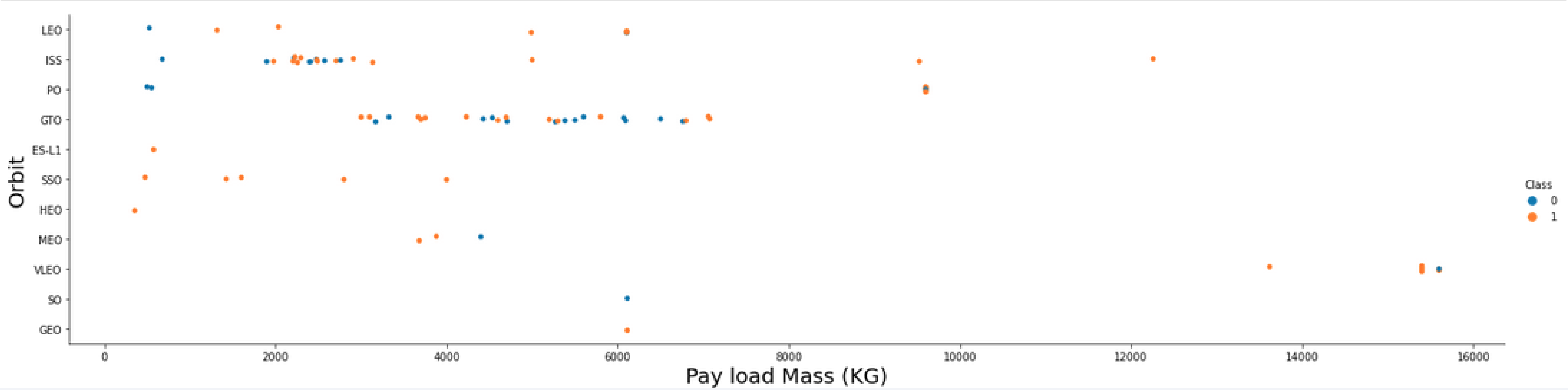
- Here we have Success Rate vs. Orbit Type. ES-L1, GEO, HEO, and SSO have the greatest success rates

Flight Number vs. Orbit Type



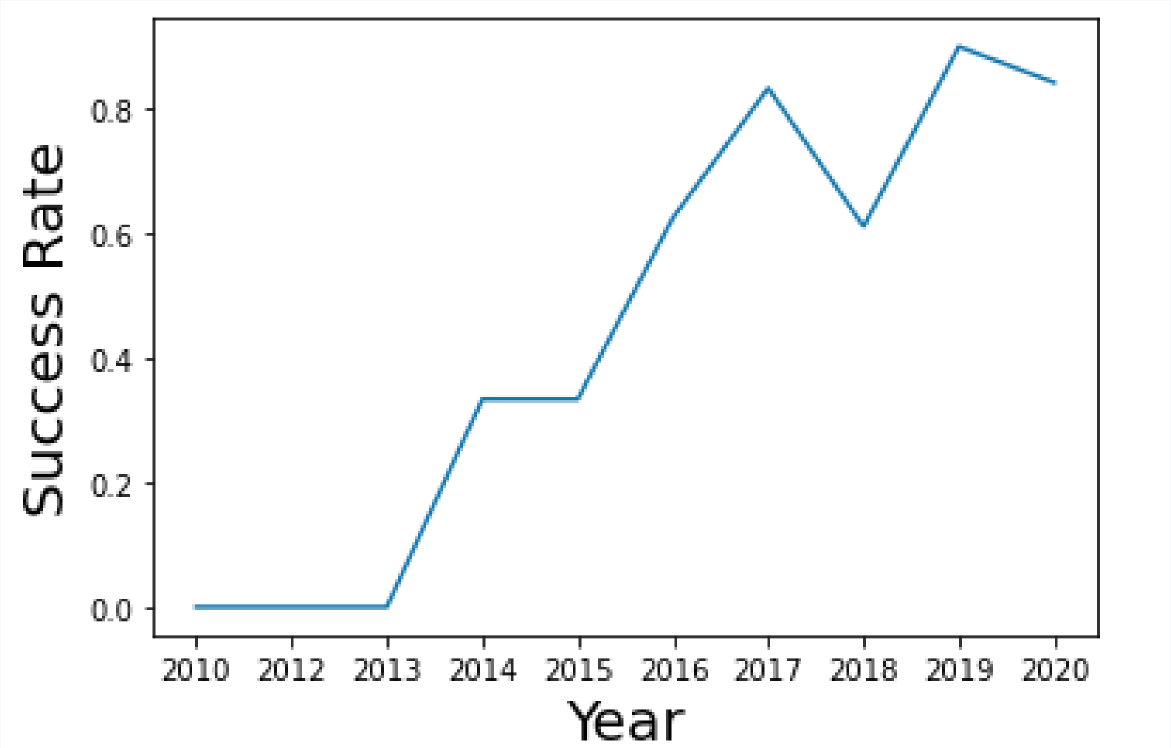
- Here we have Flight Number vs. Orbit Type. Not really a pattern to see although one could say, as the flight number increases there seems to be more of a success rate with some orbits, specifically LEO. But, I don't think this is evident enough in the other orbits.

Payload vs. Orbit Type



- Here we have Payload vs. Orbit Type. With Heavy payloads the successful landing are more for Polar, LEO and ISS.

Launch Success Yearly Trend



- Here we have the Success Rate vs. Year. This measures the success rate annually, and as you can see, ever since 2013, the success rate has been increasing. From 2017-2019 we see a little dip, but the success rate seems to recover after that.

All Launch Site Names

```
#Construct the query that retrieves all rows from the INSTRUCTOR table
selectQuery = "select UNIQUE(LAUNCH_SITE) from SPACEXTBL"

#Execute the statement
selectStmt = ibm_db.exec_immediate(conn, selectQuery)
tuple = ibm_db.fetch_tuple(selectStmt)

#Fetch the Dictionary (for the first row only) - replace ... with your code
while tuple != False:
    print ("The name is : ", tuple[0])
    tuple = ibm_db.fetch_tuple(selectStmt)
```

```
The name is :  CCAFS LC-40
The name is :  CCAFS SLC-40
The name is :  KSC LC-39A
The name is :  VAFB SLC-4E
```

- The unique launch sites for the SpaceX launches.

Launch Site Names Begin with 'CCA'

```
selectQuery2 = "SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;"

#Execute the statement
selectStmt2 = ibm_db.exec_immediate(conn, selectQuery2)
tuple = ibm_db.fetch_tuple(selectStmt2)

while tuple != False:
    print ("The name is : ", tuple[0])
    tuple = ibm_db.fetch_tuple(selectStmt2)
```

```
The name is :  CCAFS LC-40
The name is :  CCAFS LC-40
The name is :  CCAFS LC-40
The name is :  CCAFS LC-40
The name is :  CCAFS LC-40
```

- 5 records where the launch site started with CCA.

Total Payload Mass

```
selectQuery3 = "SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where (CUSTOMER) LIKE 'NASA (CRS)';"

#Execute the statement
selectStmt3 = ibm_db.exec_immediate(conn, selectQuery3)
tuple = ibm_db.fetch_tuple(selectStmt3)

while tuple != False:
    print ("The total mass is is : ", tuple[0])
    tuple = ibm_db.fetch_tuple(selectStmt3)
```

```
The total mass is is :  45596
```

- The total payload mass (kg) from NASA CRS (Commercial Resupply Service).

Average Payload Mass by F9 v1.1

```
selectQuery4 = "SELECT AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where (BOOSTER_VERSION) LIKE 'F9 v1.1';"

#Execute the statement
selectStmt4 = ibm_db.exec_immediate(conn, selectQuery4)
tuple = ibm_db.fetch_tuple(selectStmt4)

while tuple != False:
    print ("The average mass is is : ", tuple[0])
    tuple = ibm_db.fetch_tuple(selectStmt4)
```

```
The average mass is is :  2928
```

- The average payload mass (kg) for the F9 v1.1 booster was 2928 (kg). This happened to be on the lower end of our payload mass range we have shown in previous examples and may further explore.

First Successful Ground Landing Date

```
selectQuery5 = "select min(DATE) from SPACEXTBL where (LANDING__OUTCOME) LIKE 'Success (ground pad)';"

#Execute the statement
selectStmt5 = ibm_db.exec_immediate(conn, selectQuery5)
tuple = ibm_db.fetch_tuple(selectStmt5)

while tuple != False:
    print ("The first successful launch date: ", tuple[0])
    tuple = ibm_db.fetch_tuple(selectStmt5)
```

```
The first successful launch date:  2015-12-22
```

- The first successful ground landing was not until December 22nd, 2015...although other landings were recorded as early as 2014.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
selectQuery6 = "select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME = 'Success (drone ship)'
" and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;"

#Execute the statement
selectStmt6 = ibm_db.exec_immediate(conn, selectQuery6)
tuple = ibm_db.fetch_tuple(selectStmt6)

while tuple != False:
    print ("The names: ", tuple[0])
    tuple = ibm_db.fetch_tuple(selectStmt6)
```

```
The names:  F9 FT B1022
The names:  F9 FT B1026
The names:  F9 FT  B1021.2
The names:  F9 FT  B1031.2
```

- The successful booster that are in between the payload mass (kg) of 4,000 – 6,000 (kg)

Total Number of Successful and Failure Mission Outcomes

```
selectQuery7 = "select count(MISSION_OUTCOME), MISSION_OUTCOME from SPACEXTBL GROUP BY MISSION_OUTCOME;"

#Execute the statement
selectStmt7 = ibm_db.exec_immediate(conn, selectQuery7)
tuple = ibm_db.fetch_tuple(selectStmt7)

while tuple != False:
    print ("The count: ", tuple[0], "| The mission outcome: ", tuple[1])
    tuple = ibm_db.fetch_tuple(selectStmt7)
```

```
The count:  1 | The mission outcome:  Failure (in flight)
The count:  99 | The mission outcome:  Success
The count:  1 | The mission outcome:  Success (payload status unclear)
```

- The mission outcomes for SpaceX seem to be 99% success. There is 1 failed missing in flight, and 1 mission success with an unclear payload. But, what this really means is, there is a chance the failures SpaceX experienced were in fact intended/planned.

Boosters Carried Maximum Payload

```
selectQuery8 = "select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_)
"from SPACEXTBL);"

#Execute the statement
selectStmt8 = ibm_db.exec_immediate(conn, selectQuery8)
tuple = ibm_db.fetch_tuple(selectStmt8)

while tuple != False:
    print ("The Booster version: ", tuple[0])
    tuple = ibm_db.fetch_tuple(selectStmt8)
```

```
The Booster version:  F9 B5 B1048.4
The Booster version:  F9 B5 B1049.4
The Booster version:  F9 B5 B1051.3
The Booster version:  F9 B5 B1056.4
The Booster version:  F9 B5 B1048.5
The Booster version:  F9 B5 B1051.4
The Booster version:  F9 B5 B1049.5
The Booster version:  F9 B5 B1060.2
The Booster version:  F9 B5 B1058.3
The Booster version:  F9 B5 B1051.6
The Booster version:  F9 B5 B1060.3
The Booster version:  F9 B5 B1049.7
```

- All of the boosters that carried the maximum payload mass (kg), which was 15,600 (kg)

2015 Launch Records

```
selectQuery9 = "SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015';

#Execute the statement
selectStmt9 = ibm_db.exec_immediate(conn, selectQuery9)
tuple = ibm_db.fetch_tuple(selectStmt9)

while tuple != False:
    print ("The month: ", tuple[0], "| Mission Outcome: ", tuple[1], "| Booster: ", tuple[2], "Launch site:", tuple[3])
    tuple = ibm_db.fetch_tuple(selectStmt9)
```

```
The month:  1 | Mission Outcome:  Success | Booster:  F9 v1.1 B1012 Launch site: CCAFS LC-40
The month:  2 | Mission Outcome:  Success | Booster:  F9 v1.1 B1013 Launch site: CCAFS LC-40
The month:  3 | Mission Outcome:  Success | Booster:  F9 v1.1 B1014 Launch site: CCAFS LC-40
The month:  4 | Mission Outcome:  Success | Booster:  F9 v1.1 B1015 Launch site: CCAFS LC-40
The month:  4 | Mission Outcome:  Success | Booster:  F9 v1.1 B1016 Launch site: CCAFS LC-40
The month:  6 | Mission Outcome:  Failure (in flight) | Booster:  F9 v1.1 B1018 Launch site: CCAFS LC-40
The month:  12 | Mission Outcome:  Success | Booster:  F9 FT B1019 Launch site: CCAFS LC-40
```

- All the launches of the 2015 calendar year.

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

```
selectQuery10 = "SELECT landing__outcome, COUNT(landing__outcome), date FROM SPACEXTBL WHERE date BETWEEN '2010-06-04'
"AND '2017-03-20' GROUP BY (landing__outcome), date ORDER BY COUNT(landing__outcome), date DESC;"

#Execute the statement
selectStmt10 = ibm_db.exec_immediate(conn, selectQuery10)
tuple = ibm_db.fetch_tuple(selectStmt10)

while tuple != False:
    print ("The outcome: ", tuple[0], "| The count: ", tuple[1], "| Date: ", tuple[2])
    tuple = ibm_db.fetch_tuple(selectStmt10)
```

```
The outcome:  No attempt | The count:  1 | Date:  2017-03-16
The outcome:  Success (ground pad) | The count:  1 | Date:  2017-02-19
The outcome:  Success (drone ship) | The count:  1 | Date:  2017-01-14
The outcome:  Success (drone ship) | The count:  1 | Date:  2016-08-14
The outcome:  Success (ground pad) | The count:  1 | Date:  2016-07-18
The outcome:  Failure (drone ship) | The count:  1 | Date:  2016-06-15
The outcome:  Success (drone ship) | The count:  1 | Date:  2016-05-27
The outcome:  Success (drone ship) | The count:  1 | Date:  2016-05-06
The outcome:  Success (drone ship) | The count:  1 | Date:  2016-04-08
The outcome:  Failure (drone ship) | The count:  1 | Date:  2016-03-04
The outcome:  Failure (drone ship) | The count:  1 | Date:  2016-01-17
The outcome:  Success (ground pad) | The count:  1 | Date:  2015-12-22
The outcome:  Precluded (drone ship) | The count:  1 | Date:  2015-06-28
The outcome:  No attempt | The count:  1 | Date:  2015-04-27
The outcome:  Failure (drone ship) | The count:  1 | Date:  2015-04-14
The outcome:  No attempt | The count:  1 | Date:  2015-03-02
```

- Ranking landing outcomes.

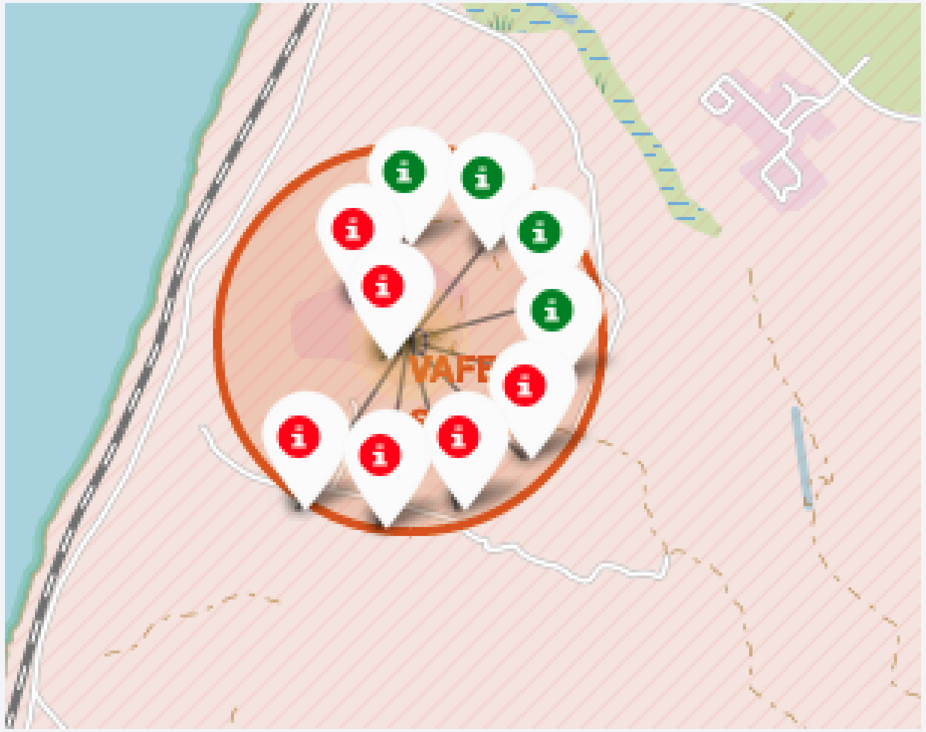
Section 3

Launch Sites Proximities Analysis

All Launch Sites

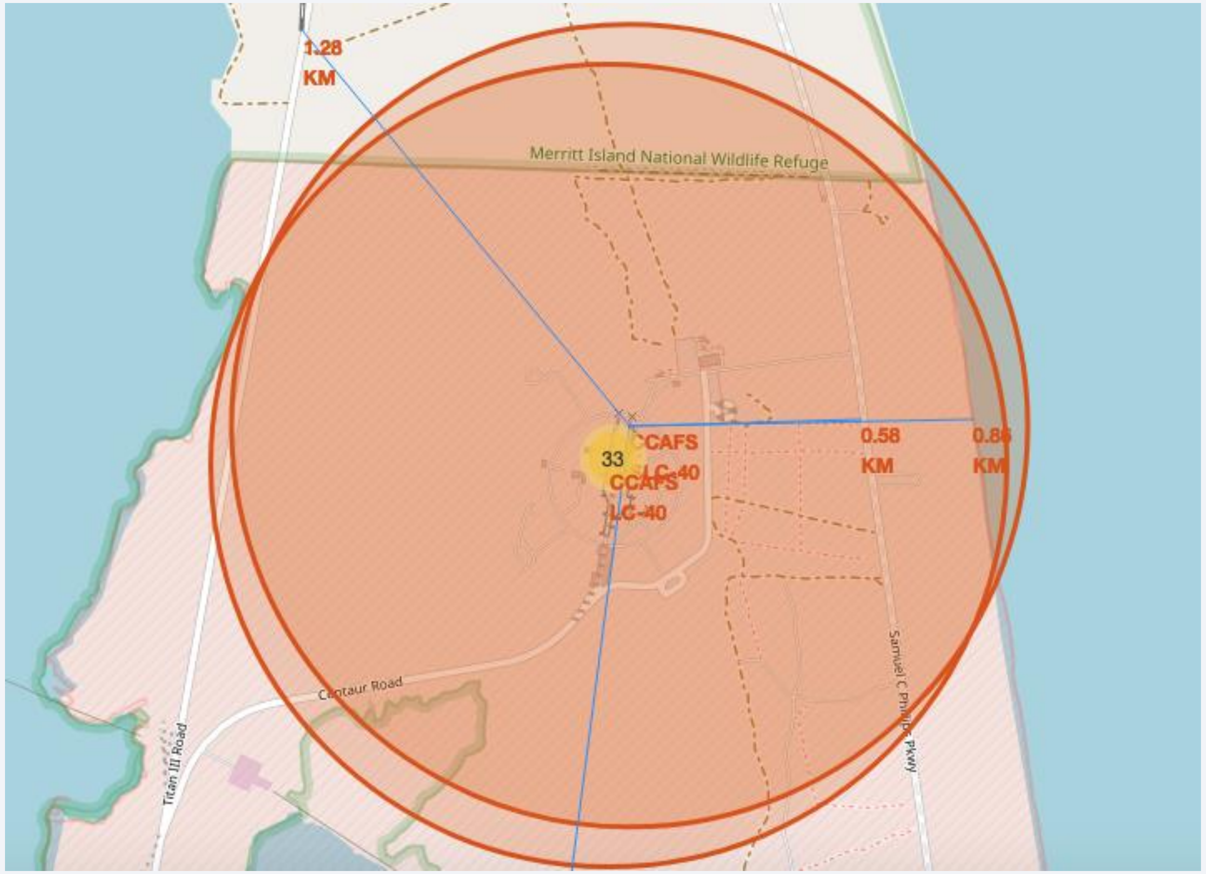
- Here you can see the 3 unique Launch Sites. 1 in California near the coast, and 2 in Florida near the coast.

Color Labeling Markers



- Here with Folium we can see the successful and unsuccessful landings. Green represents success and Red represents failure. Here we can see, out of 10 landings, only 4 were successful and 6 unsuccessful.

Proximities

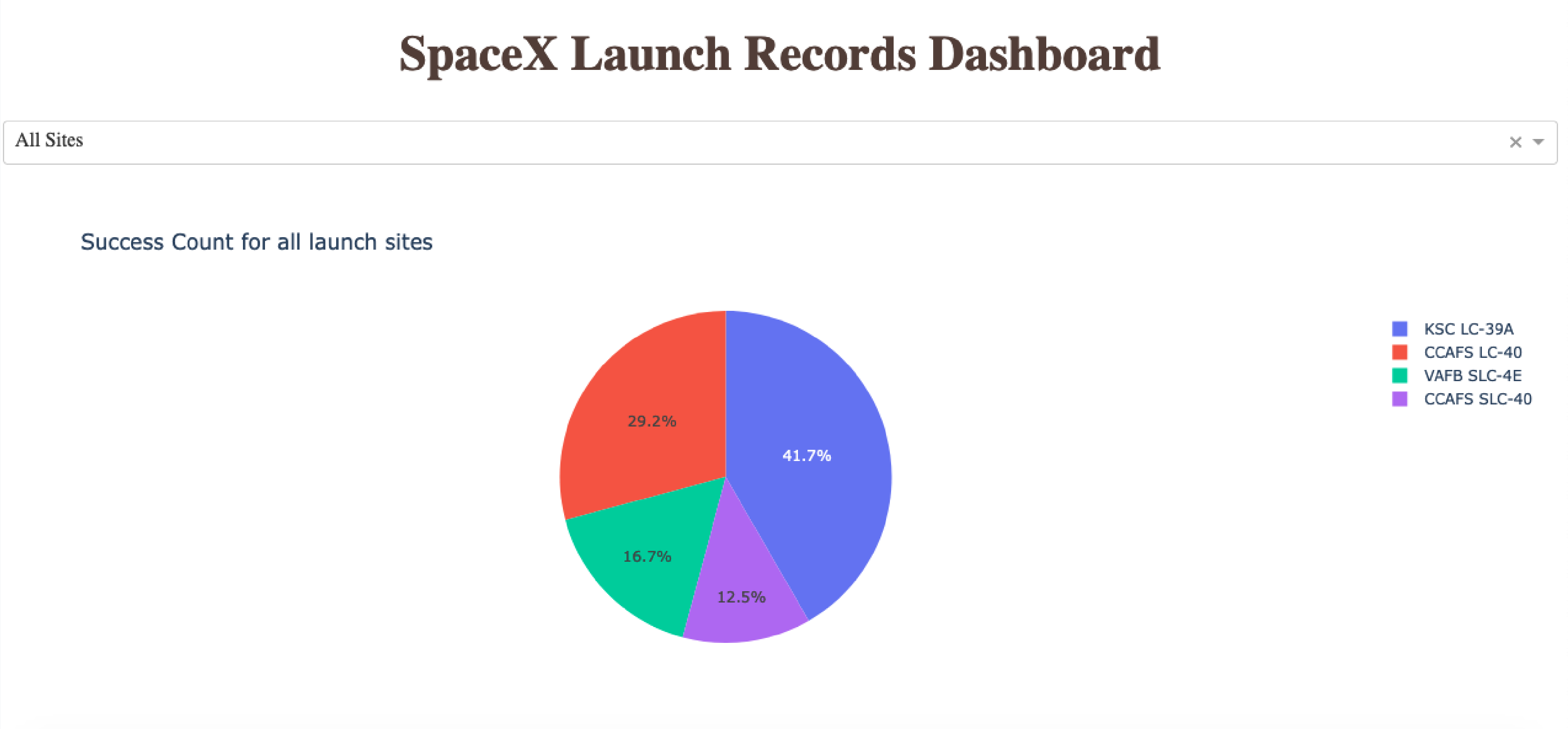


- Here we can see the proximities from a Florida launch site. About 0.58 km away from the highway, 1.28 km away from a railway, and 0.86 km away from the coastline.

Section 4

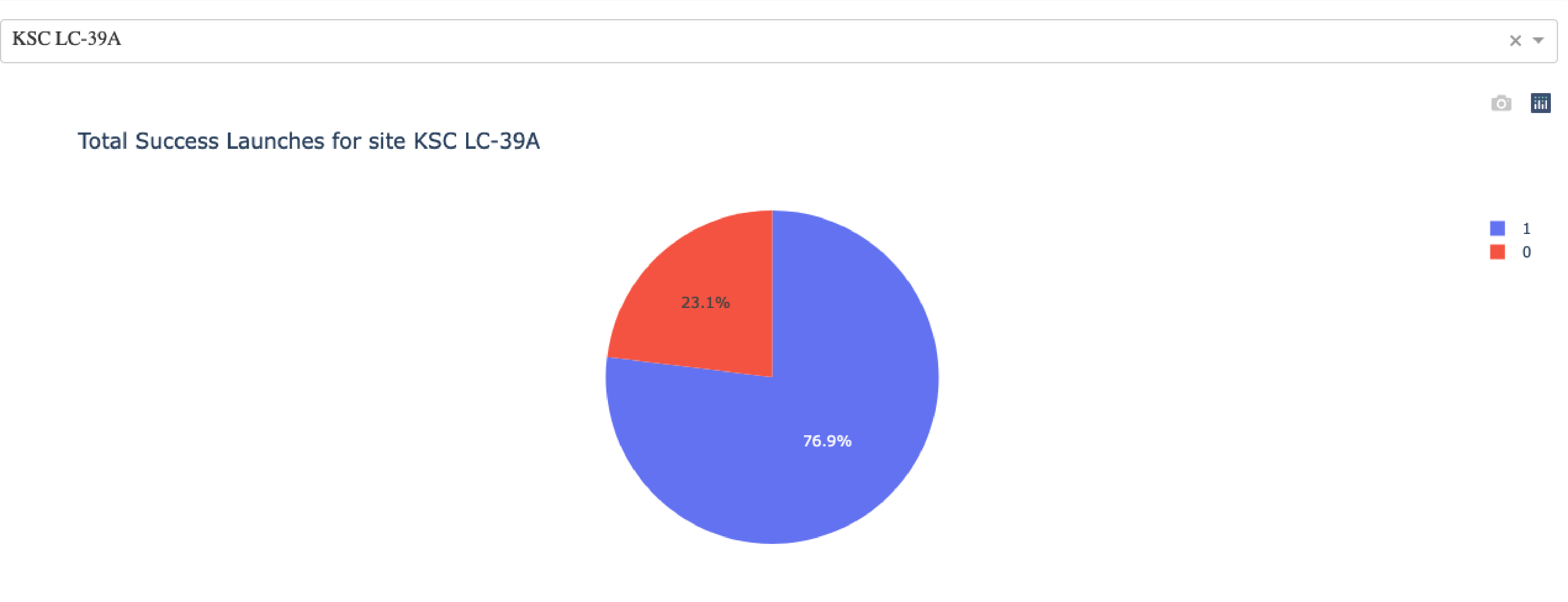
Build a Dashboard with Plotly Dash

Success Rates: All Launches



- Here we see that KSC LC-39A is the most successful launch site, being the only site having over 30% of its launches being successful.

Highest Success Rate: KSC LC-39A



- The highest success rate was at site KSC LC 39-A, which had 10 successful landings involving 3 different boosters and 3 unsuccessful landings involving one booster...all unsuccessful launches had a payload mass over 5500 (kg)

Payload vs. Class

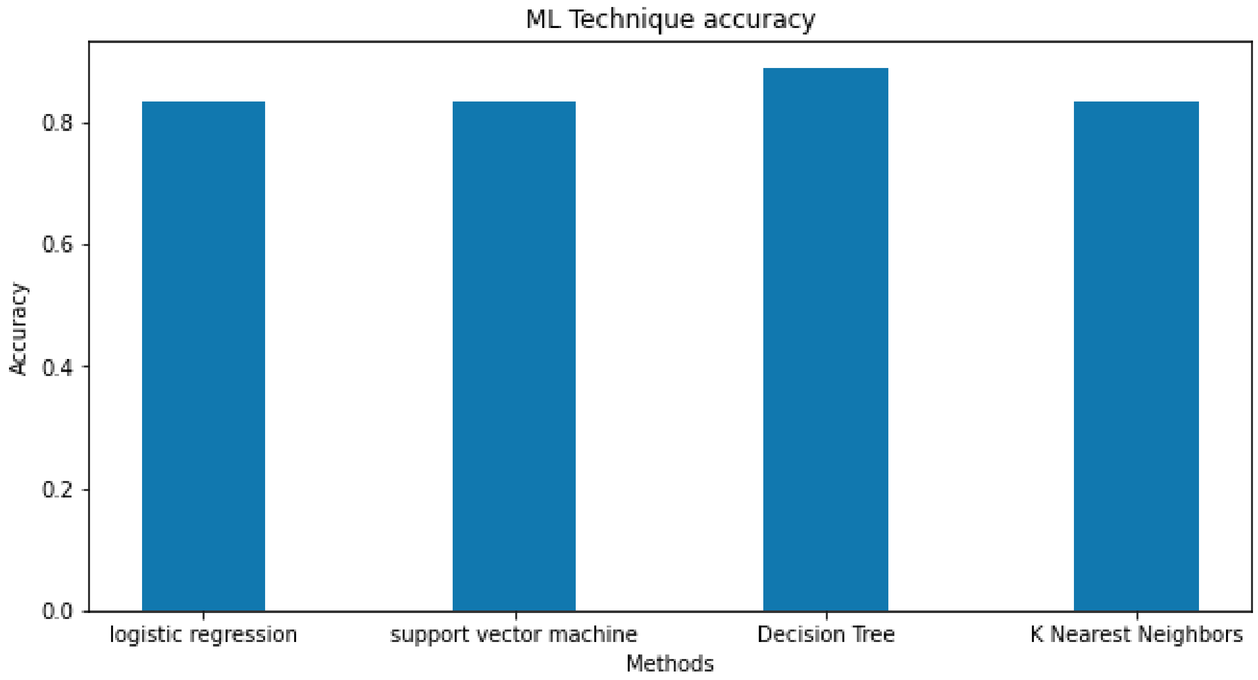
- Here you can see the scatter plot on our payload vs. Class graph. As you can see, there is a range slider we adjusted to a max payload of 6,000 kg, that is because most of the successful launches came between 2,000-4,000 kg. The most successful booster is the FT booster with 13 successful launches in this graph.

Section 5

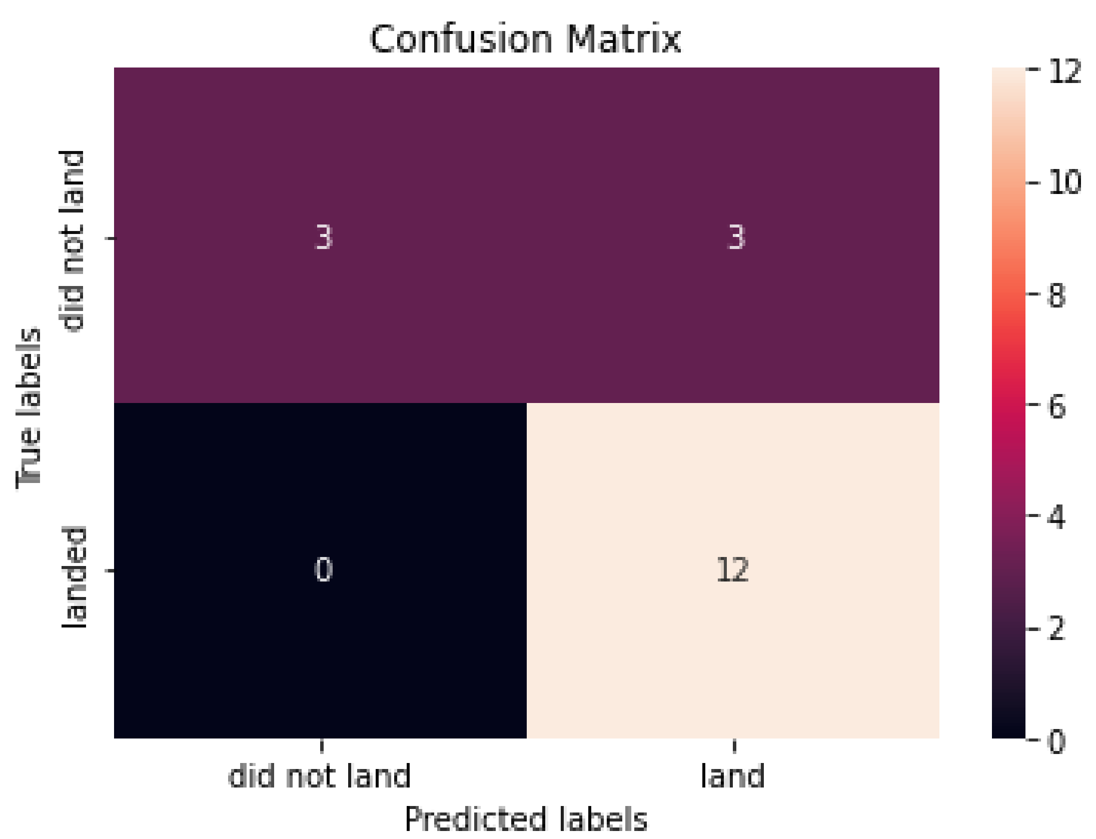
Predictive Analysis (Classification)

Classification Accuracy

- In my example specifically, Decision Trees had the highest accuracy, with an accuracy of about 87%, comparatively to the other ML techniques that had an accuracy of about 83%.



Confusion Matrix



- We were able to predict 12 landed, and 3 did not land, which were correct. "**true positive**" for correctly predicted event values. "**false positive**" for incorrectly predicted event values. "**true negative**" for correctly predicted no-event values. "**false negative**" for incorrectly predicted no-event values. Diagonally 3 and 12 are the true positive and true positive.

Conclusions

- The most successful Launches we see come out of KSC LC-39A, which is a site in Florida
- We see the most successful launches come when the payload mass is between 2,000 and 4,000 kg
- As of 2013, we also see the success rate of launches increase, as well as the success rate of launches increases with the amount of flights recorded
- The ML model we created was able to have an accuracy of 83% as well as creating a confusion matrix correctly predicting the amount of spacecrafts that would land and did, and predicting the amount of spacecrafts that would not land and did not.
- With all of this I do think there is more data needed, but with the data with have and how we were able to manipulate it, I would say the Falcon 9 will successfully land at the first stage.

Appendix

- https://github.com/cljones97/DataSci_IBM_Project

Thank you!