Winning Space Race with Data Science

Twesigye Ronald Karakire

30th May, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis (EDA)
  - Predictive Analysis (Classification)
    - Logistic Regression
    - Support Vector Machine (SVM)
    - Decision Tree Classifier
    - K-Nearest Neighbour (KNN)
- Summary of all results
  - KSC LC-39A has highest launch success rate
  - Overall launch success rate has been improving since 2013
  - Decision Tree Classifier generated best accuracy score

Introduction

- Background
  - Determine cost of launch
  - Provide information for bid against SpaceX for rocket launch
- Problems
  - Predict whether Falcon 9 first stage will land successfully

Section 1

Methodology

Methodology

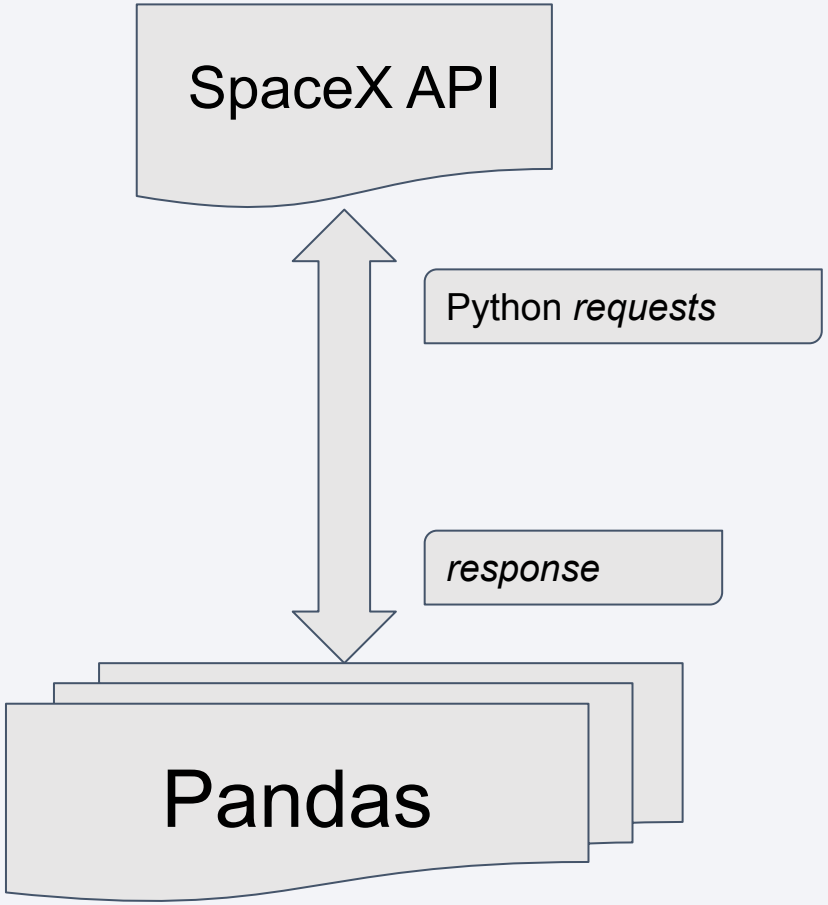
Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web Scraping (Wikipedia)
- Perform data wrangling using Python Pandas
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Standardization (Scikit-Learn Standard Scaler)
  - Train-Test Split
  - Hyperparameter tuning (GridSearchCV)
  - Train & test scores

Data Collection

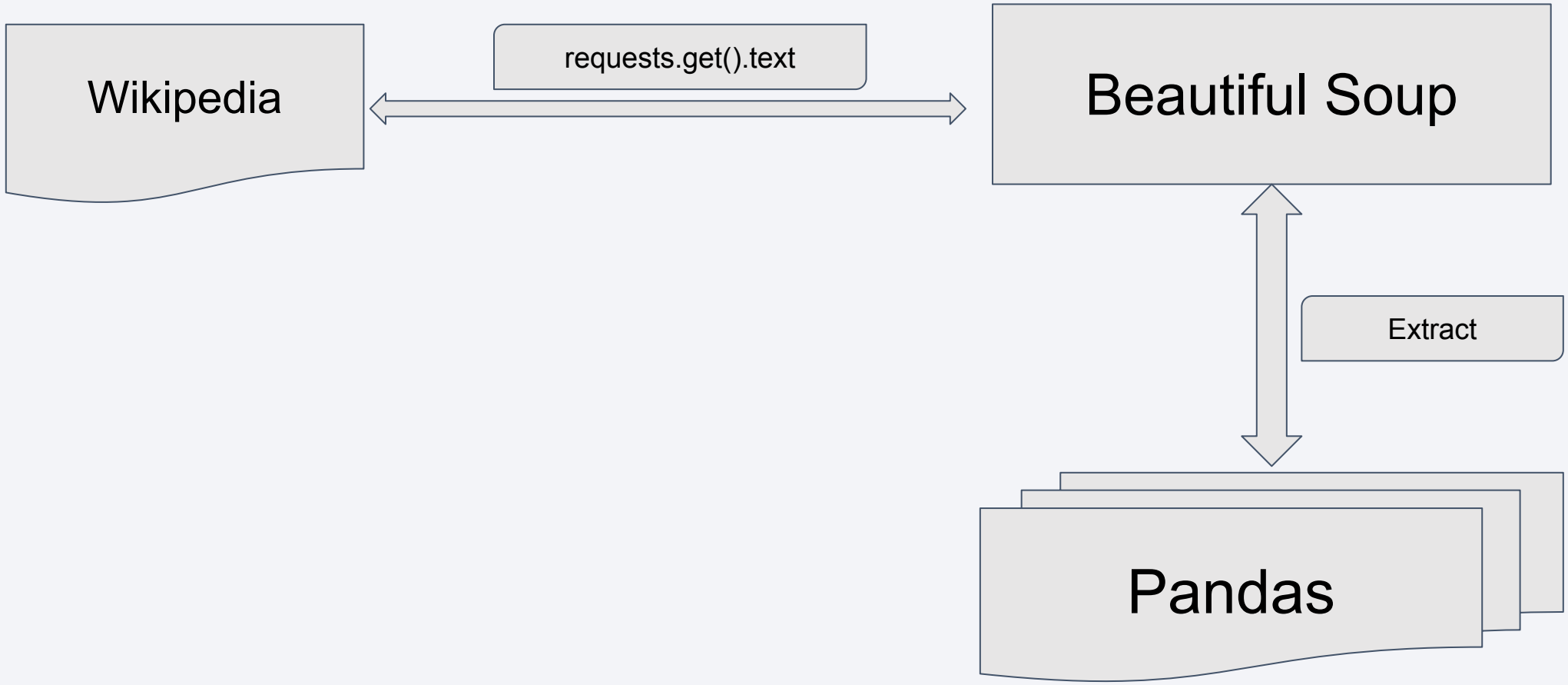
- Datasets were collected from two(2) sources:
  - SpaceX API
    - Using python requests library
  - Wikipedia
    - Using python BeautifulSoup web scraping library

Data Collection – SpaceX API



https://github.com/twesigyeronaldk/coursera-ibm-data-science/blob/master/Data%20Collection%20API.ipynb

Data Collection - Scraping



https://github.com/twesigyeronaldk/coursera-ibm-data-science/blob/master/Data%20Collection%20with%20Web%20Scraping.ipynb

Data Wrangling

- Tools
  - Pandas
- Key points
  - Null value counts
  - Column data types
  - Class mean
  - Columns analysed - Launch Sites, Orbit, Outcome, Class
- https://github.com/twesigyeronaldk/coursera-ibm-data-science/blob/master/EDA.ipynb

EDA with Data Visualization

- Graphs plotted
  - Scatter plot
    - Launch Site vs Flight Number
    - Launch Site vs Payload Mass (kg)
    - Orbit vs Flight Number
    - Orbit vs Payload Mass (kg)
  - Bar graph
    - Success Rate vs Orbit
  - Line graph
    - Success Rate vs Years
      - Visualize launch success yearly trend
- https://github.com/twesigyeronaldk/coursera-ibm-data-science/blob/master/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

- Queries
  - SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;
  - SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 0, 5;
  - SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
  - SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
  - SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';
  - SELECT (BOOSTER_VERSION) FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
  - SELECT COUNT(*), MISSION_OUTCOME FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
  - SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_ FROM SPACEXTBL ORDER BY PAYLOAD_MASS__KG_ DESC;
  - SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE LIKE '2015%';
  - SELECT LANDING__OUTCOME, COUNT(*) AS COUNT FROM SPACEXTBL WHERE DATE > '2010-06-04' AND DATE < '2017-03-20' GROUP BY LANDING__OUTCOME;

- https://github.com/twesigyeronaldk/coursera-ibm-data-science/blob/master/EDA%20with%20SQL.ipynb

- **Note: I run my sql queries directly inside IBM DB2 environment not jupyter notebooks**

Build an Interactive Map with Folium

- Map objects
  - Circle
    - For drawing circles
  - Marker
    - For identifying particular coordinate points on the map
  - FeatureGroup
    - For grouping a particular feature (for example a collection of markers) on the map
  - MousePosition
    - To get coordinate of current mouse position over the map
- https://github.com/twesigyeronaldk/coursera-ibm-data-science/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

- Graphs plotted
  - Pie charts
    - Success launches by site
      - For all launch sites
      - Per launch site
  - Scatter plot
    - Class vs Pay load mass (kg)
- Interactions implemented
  - Laun sites drop down menu
    - Used to switch between launch sites for target figure (pie chart)
  - Payload mass (kg) slider
    - Used to select payload mass range (kg) for target figure (scatter plot)
- https://github.com/twesigyeronaldk/coursera-ibm-data-science/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- Scikit-Learn was used for the machine learning process
- After importing the data into pandas, we had to first get X, that is, the independent variables and y, the dependent variable
- We then had to first scale X using scikit-learn Standard Scaler
- We then split the data into train & test data
- To get the best hyperparameters, we used GridSearchCV for the various estimators
    - Logistic Regression
    - Support Vector Machine
    - Decision Tree Classifier
    - K-Nearest Neighbour Classifier
- We got the score of each classifier on the test data using the score method
- We also drew th confusion matrix for each of the classifiers

- https://github.com/twesigyeronaldk/coursera-ibm-data-science/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results - EDA(1)

- All columns have zero (0) null values except LandingPad
- LandingPad has 40.625% null values
- LaunchSite value counts
  - CCAFS SLC 40 55
  - KSC LC 39A 22
  - VAFB SLC 4E 13

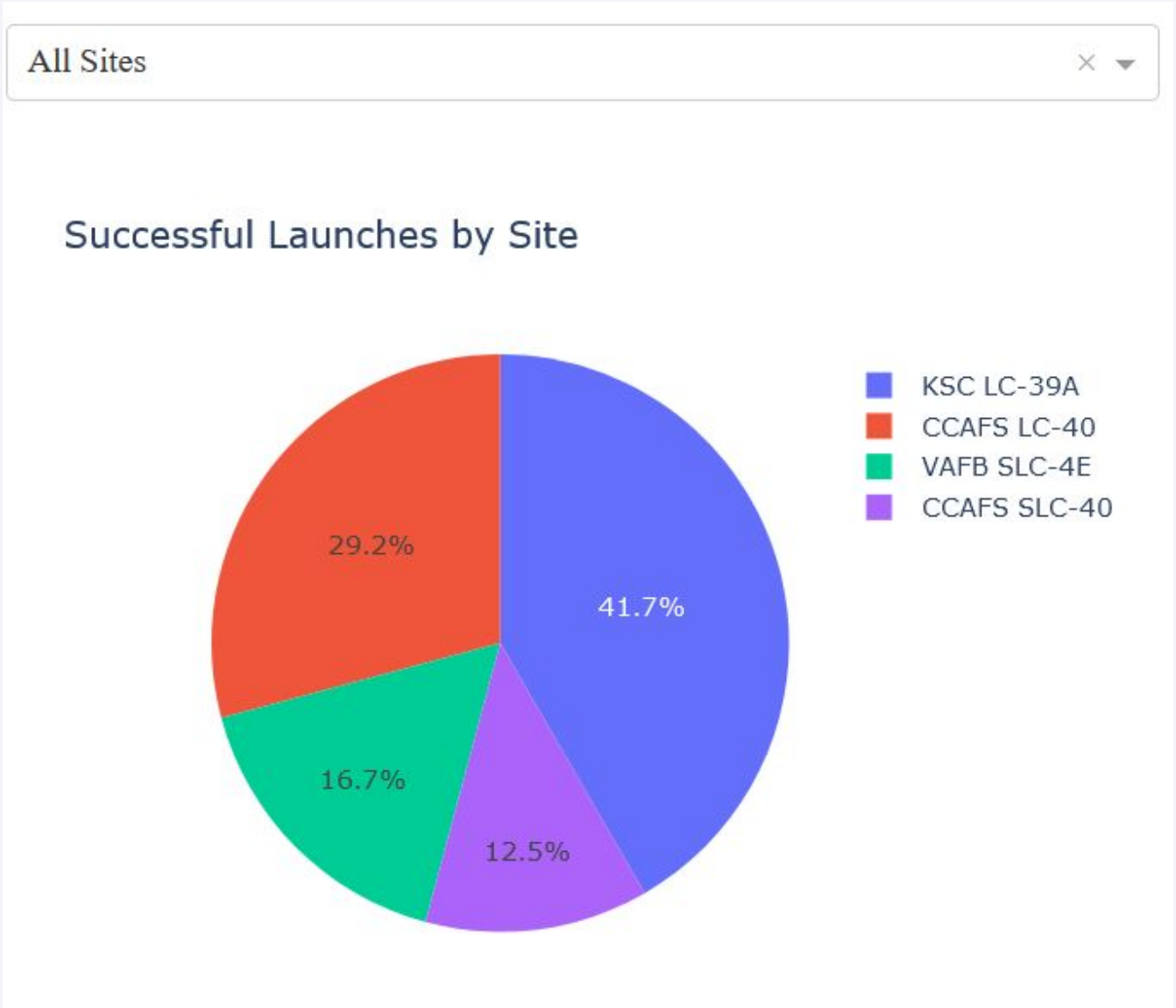
Results - EDA(2)

- Orbit value counts
  - GTO 27
  - ISS 21
  - VLEO 14
  - PO 9
  - LEO 7
  - SSO 5
  - MEO 3
  - ES-L1 1
  - HEO 1
  - SO 1
  - GEO 1

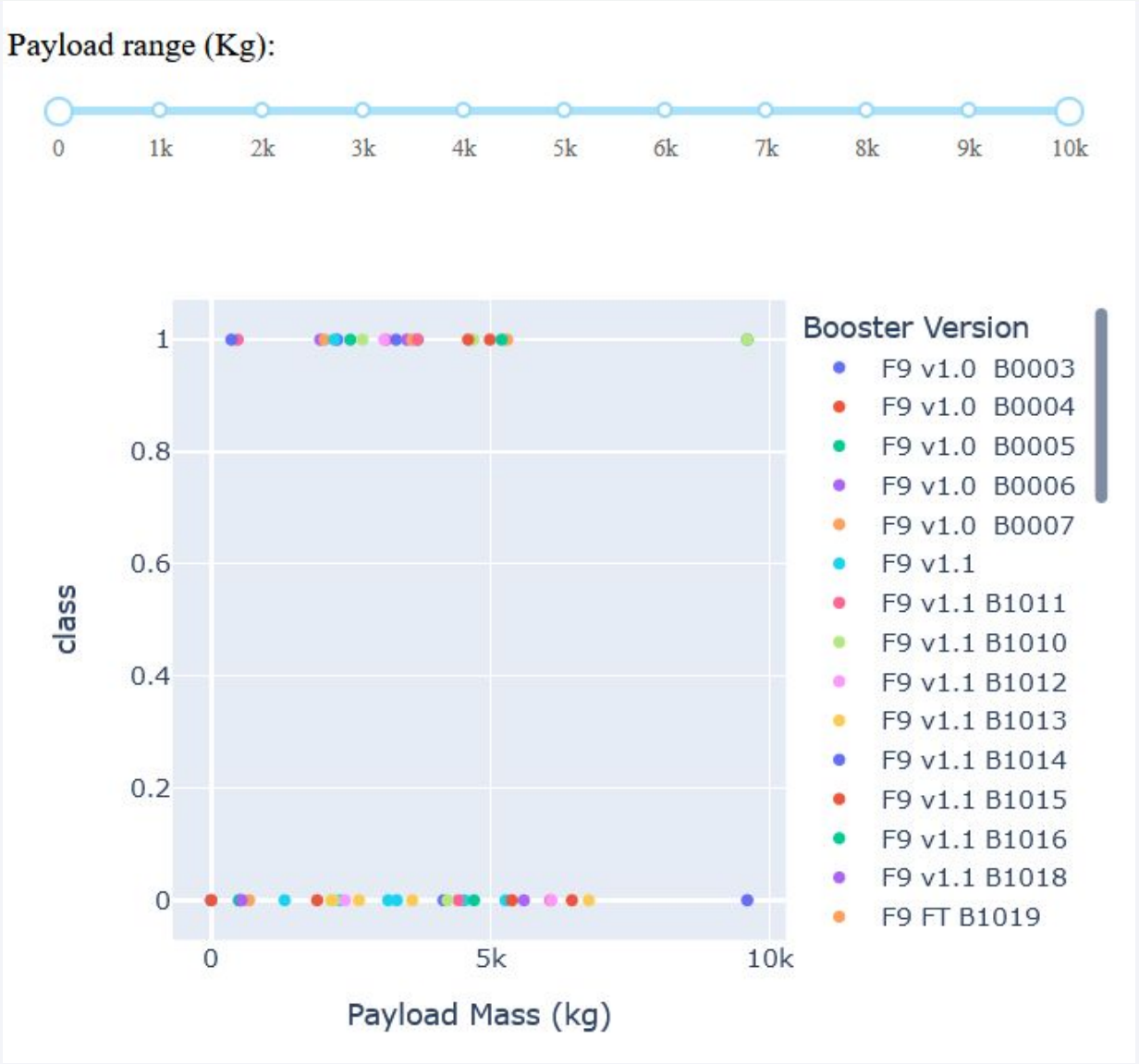
Results - EDA(3)

- Landing outcomes value counts
  - True ASDS 41
  - None None 19
  - True RTLS 14
  - False ASDS 6
  - True Ocean 5
  - False Ocean 2
  - None ASDS 2
  - False RTLS 1

Results - Dash Plotly(1)



Results - Dash Plotly(2)

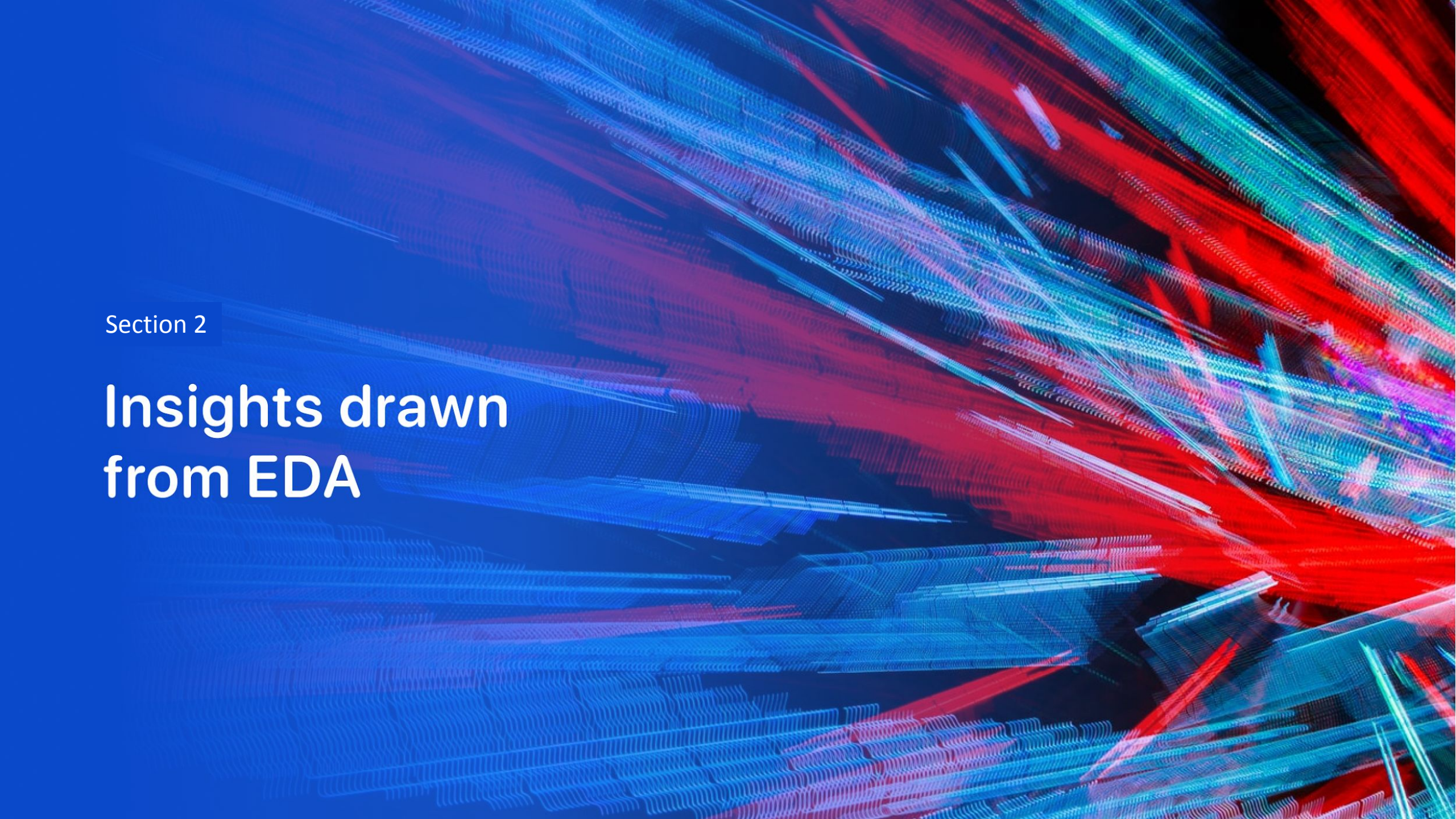


Results - Predictive Analysis

	Accuracy	
Classifier	**Train**	**Test**
Logistic Regression	0.84	0.85
Support Vector Machine	0.85	0.83
Decision Tree Classifier	0.9	0.78
KNN	0.85	0.83

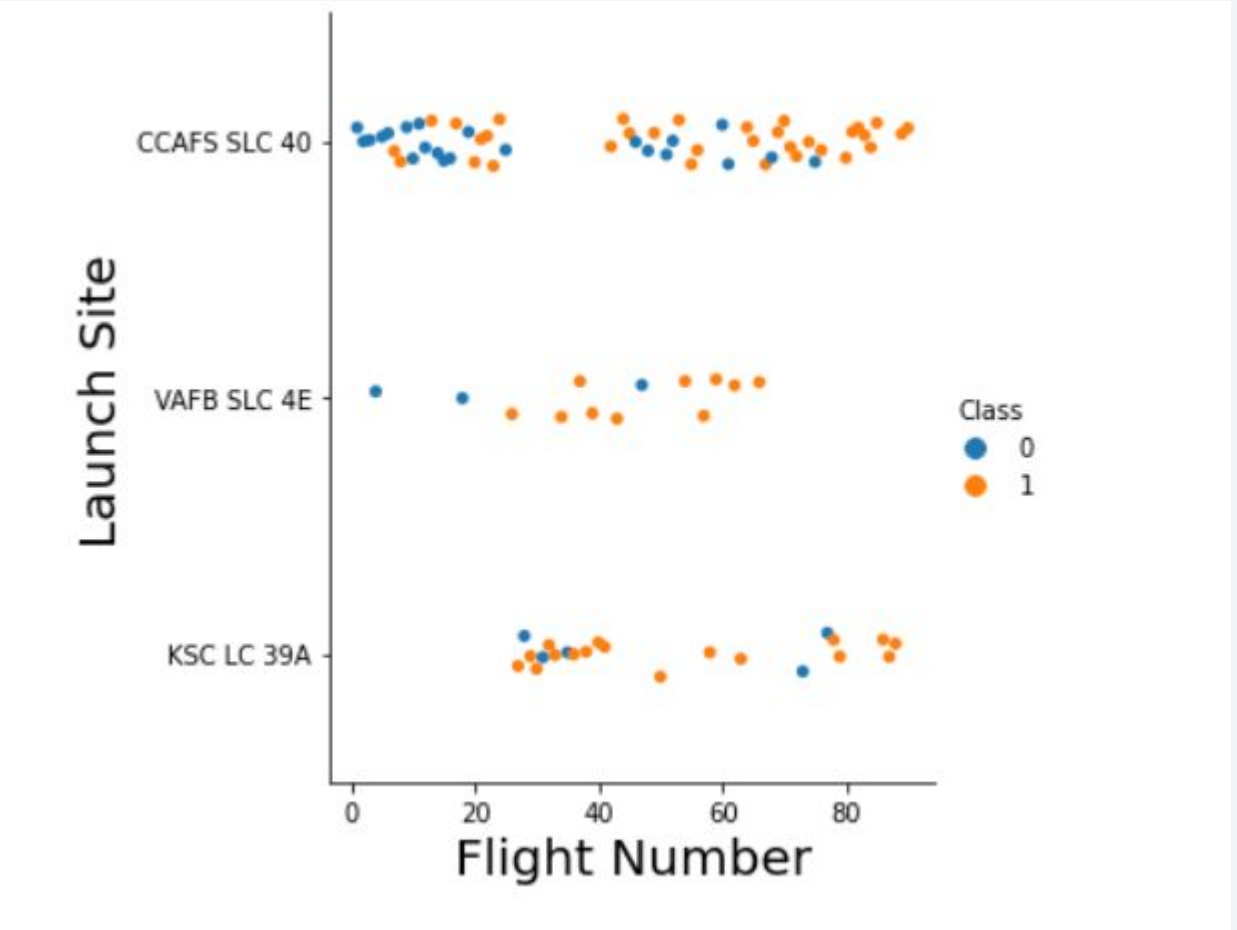
Section 2

Insights drawn from EDA



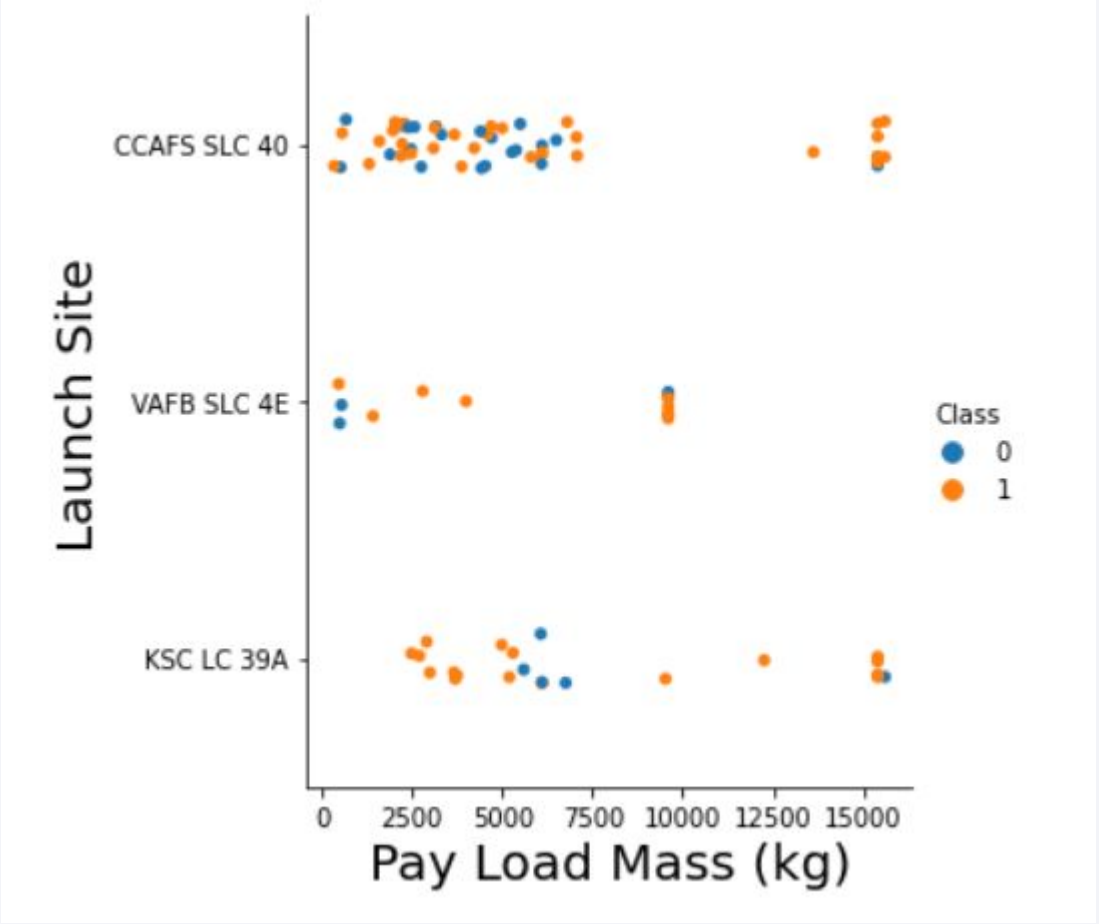
Flight Number vs. Launch Site

- Majority of the first flights (flight number < 40) where a fail for CCAFS SLC 40 launch site
- Majority of later flights (flight number > 40) where a success for CCAFS SLC 40 launch site
- Launch site VAFB SLC 4E had only three(3) fails
- Majority of flights from KSC LC 39A where a success



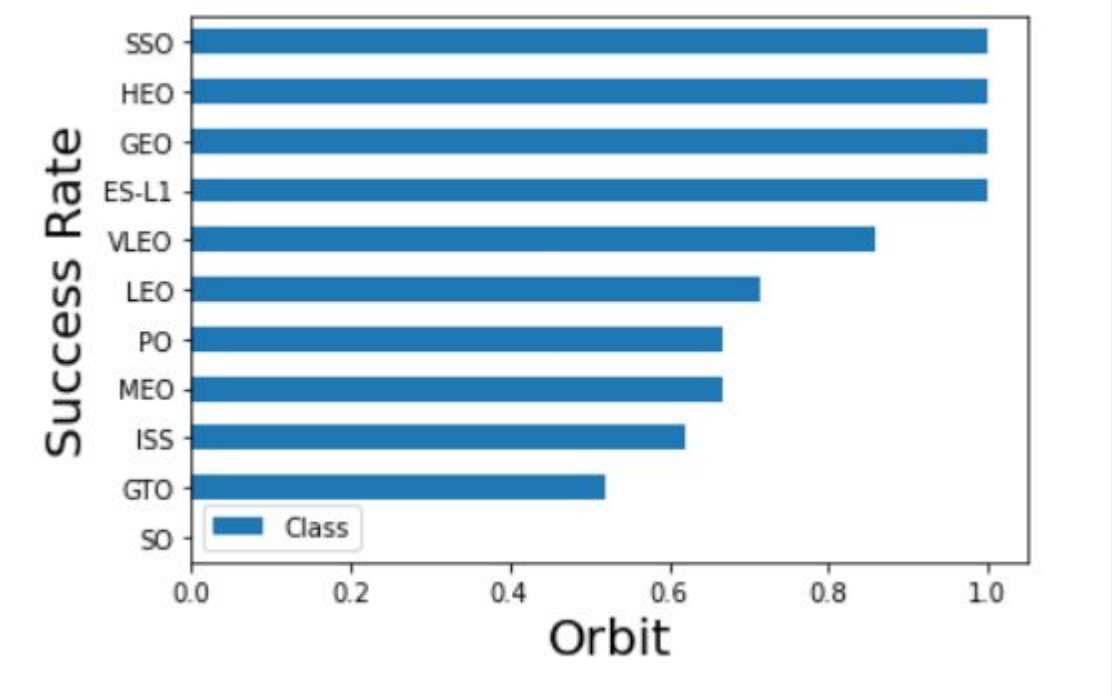
Payload vs. Launch Site

- Majority of payloads at CCAFS SLC 40 where less than 10 tonnes and most where fails
- CCAFS SLC 40 launches with payload > 12.5 tonnes where mostly successful
- VAFB SLC 4E had very few launches. All had a maximum payload of about 10 tonnes and most were successful
- KSC LC 39A had payloads between 2.5 and about 150 tonnes. Majority of launches were succesfull



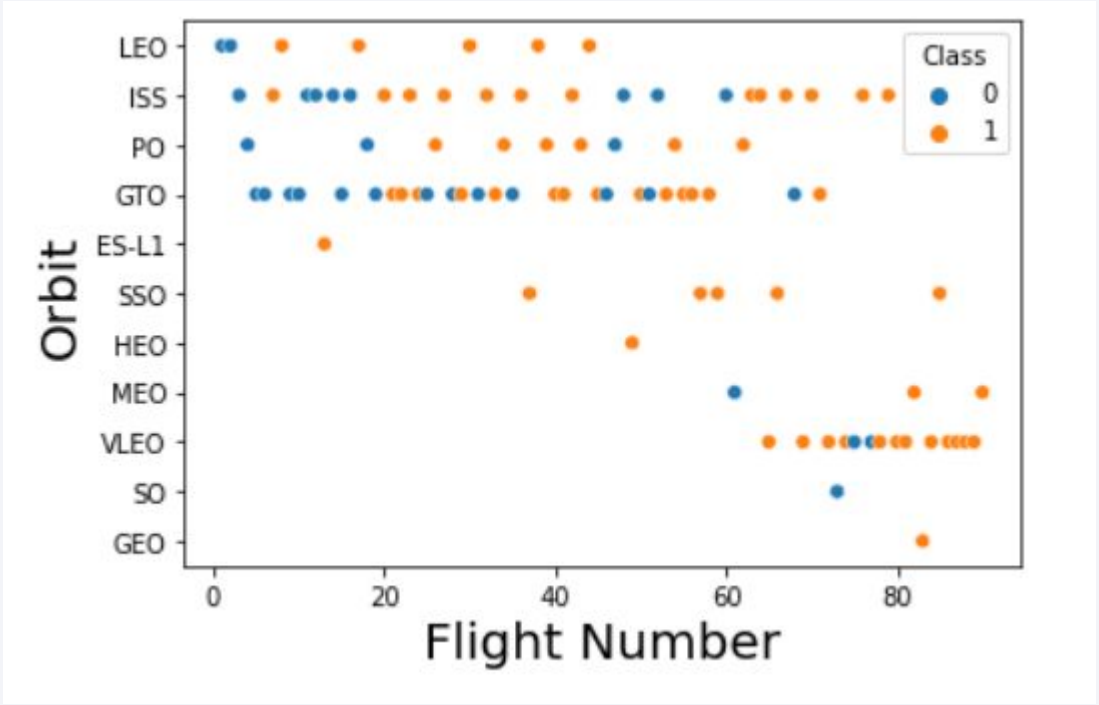
Success Rate vs. Orbit Type

- SSO, HEO, GEO and ES-L1 orbits all had 100% success rates
- SO has to lowest success rate



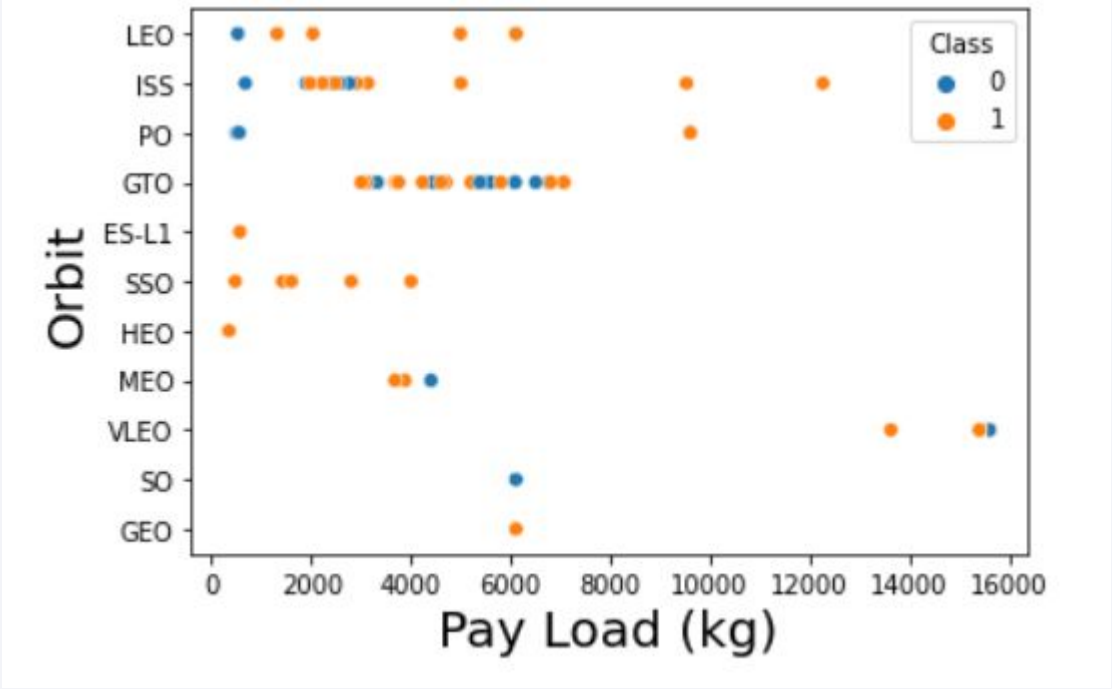
Flight Number vs. Orbit Type

- Majority of launches where conducted for LEO, ISS, PO and GTO orbits



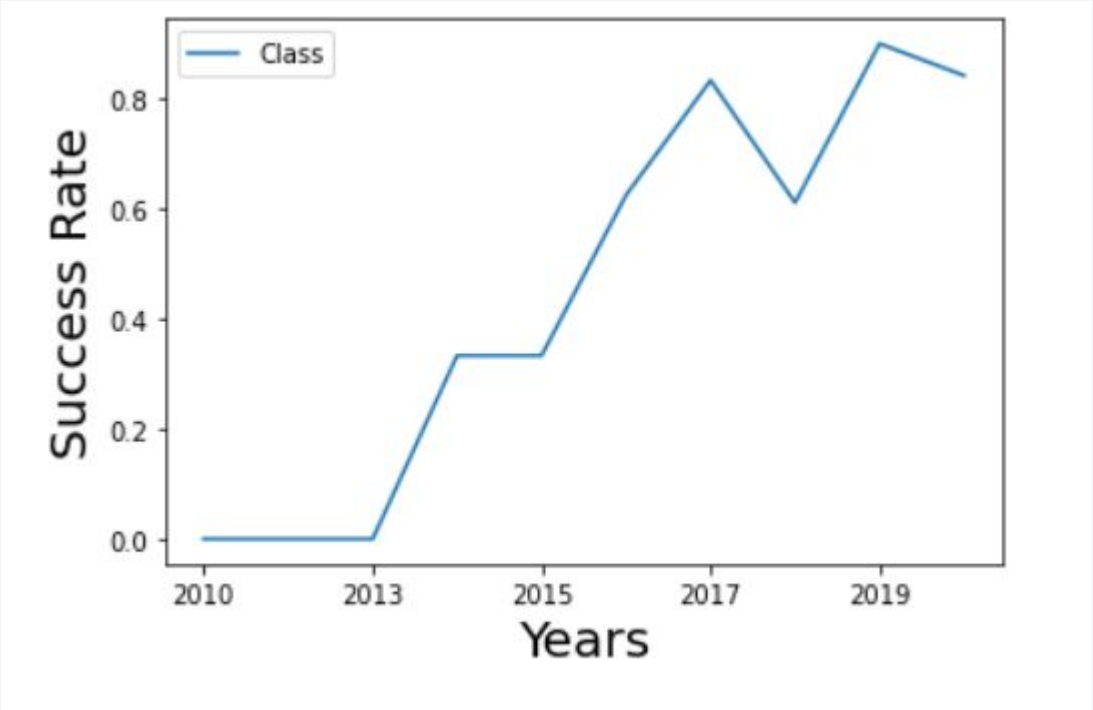
Payload vs. Orbit Type

- Majority of orbits had a payload less than 10 tonnes
- Only ISS, PO and VLEO had payloads that where greater than 10 tonnes



Launch Success Yearly Trend

- Between 2010 and 2013, there was a zero(0) percent success rate
- From 2013 upwards, the success rate began to steadily increase
- There was a slight drop in success rate during 2018 but it improved the next year



All Launch Site Names

- Launch sites
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E
- SQL Query
  - SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;

Launch Site Names Begin with 'CCA'

- Query
  - SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 0, 5;

Total Payload Mass

- Total payload carried by boosters from NASA
  - 45596
- Query
  - SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL
    WHERE CUSTOMER = 'NASA (CRS)';

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1
  - 2928
- Query
  - SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';

First Successful Ground Landing Date

- Date of first successful landing outcome on ground pad
  - 2015-12-22
- Query
  - SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
  - F9 FT B1022
  - F9 FT B1026
  - F9 FT B1021.2
  - F9 FT B1031.2
- Query
  - SELECT (BOOSTER_VERSION) FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes
  - Failure (in flight) 1
  - Success 99
  - Success (payload status unclear)1
- Query
  - SELECT MISSION_OUTCOME, COUNT(*) AS COUNT FROM SPACEXTBL GROUP BY MISSION_OUTCOME;

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass
  - 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
- Query
  - SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_ FROM SPACEXTBL ORDER BY PAYLOAD_MASS__KG_ DESC;

2015 Launch Records

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - F9 v1.1 B1012 CCAFS LC-40
  - F9 v1.1 B1015 CCAFS LC-40
- Query
  - SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE LIKE '2015%';

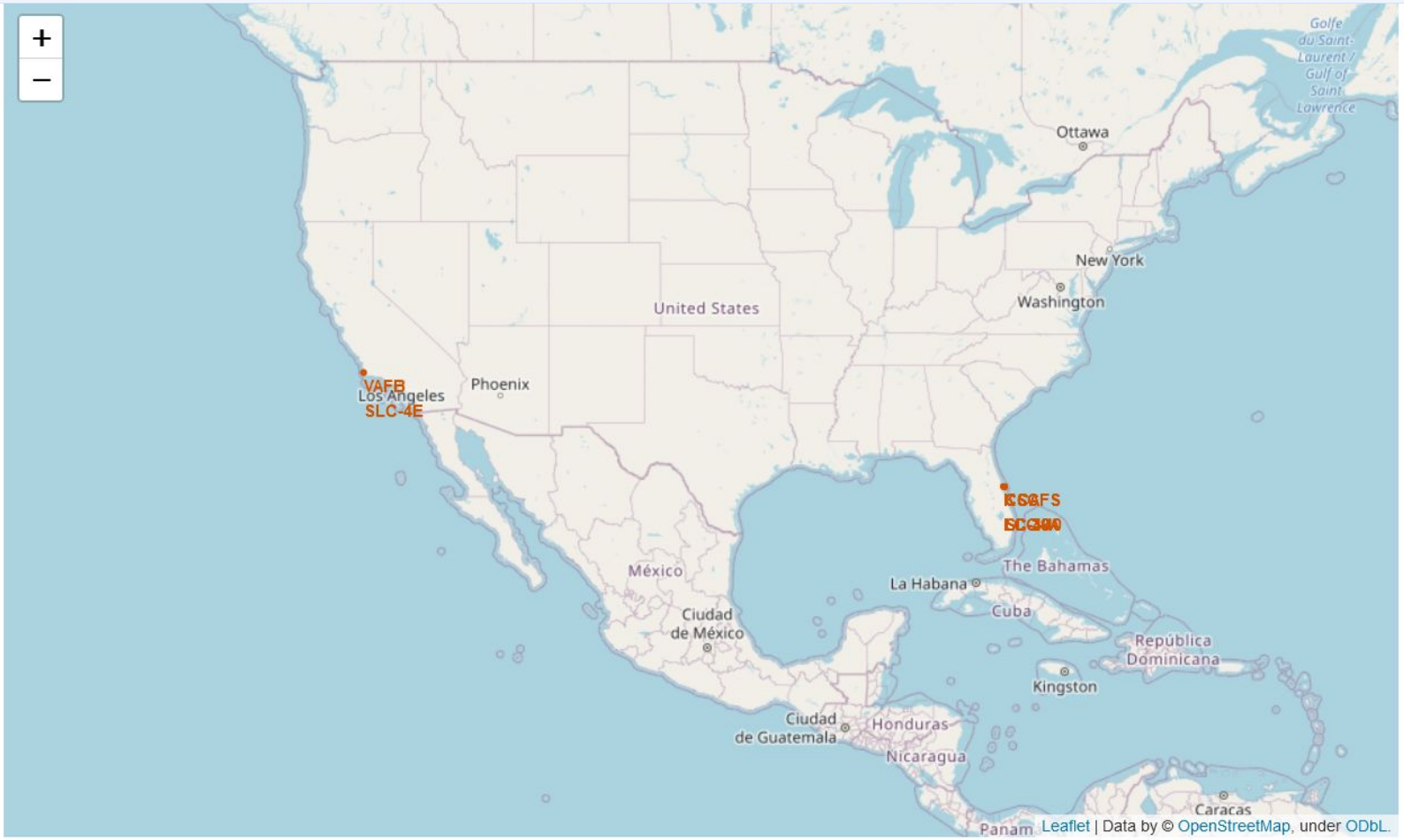
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
  - No attempt 10
  - Failure (drone ship) 5
  - Success (drone ship) 5
  - Controlled (ocean) 3
  - Success (ground pad) 3
  - Uncontrolled (ocean) 2
  - Failure (parachute) 1
  - Precluded (drone ship) 1
- Query
  - SELECT LANDING__OUTCOME, COUNT(*) AS COUNT FROM SPACEXTBL WHERE DATE > '2010-06-04' AND DATE < '2017-03-20' GROUP BY LANDING__OUTCOME;


Section 3
Launch Sites
Proximities Analysis

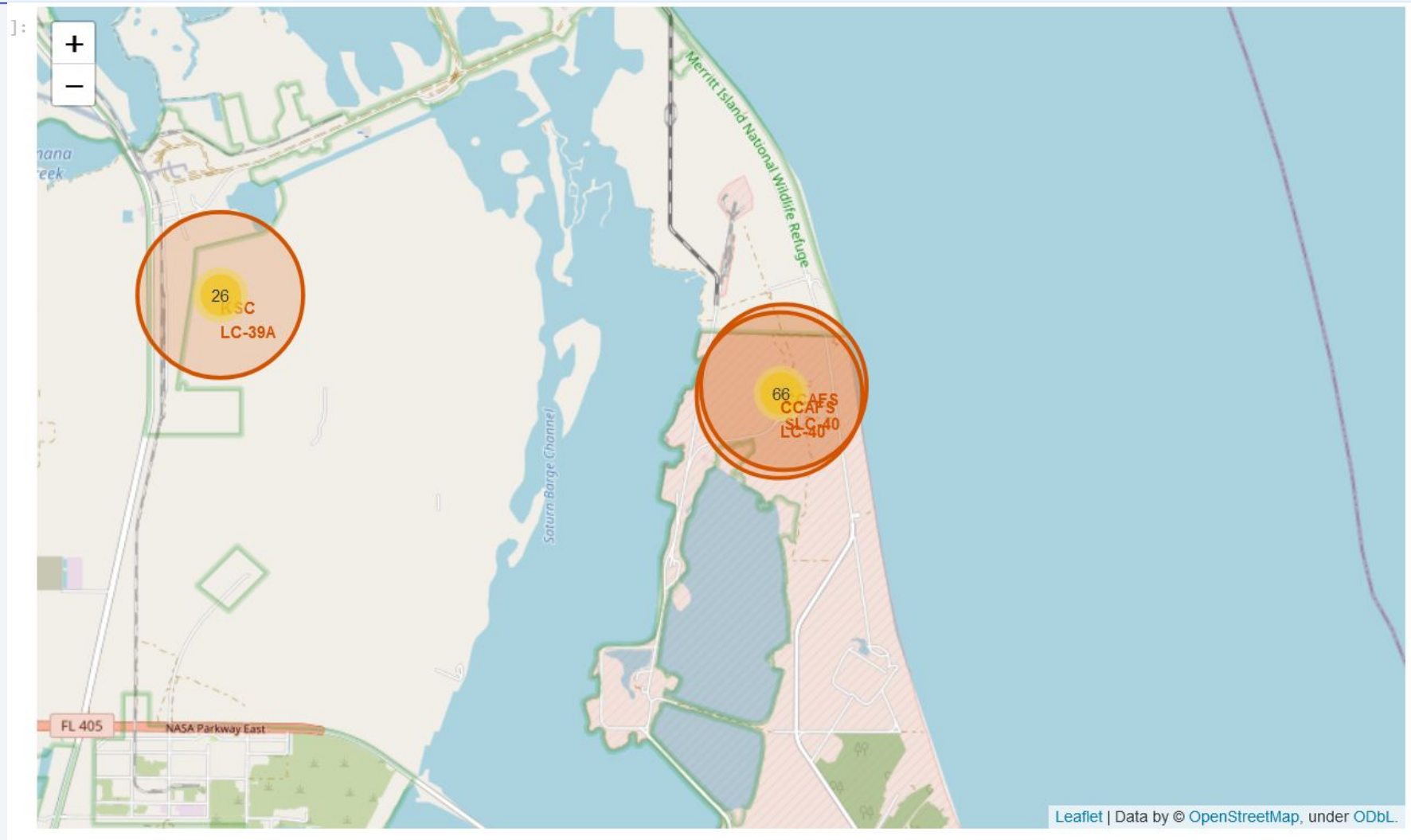
Map of Launch Site Locations

[9]:

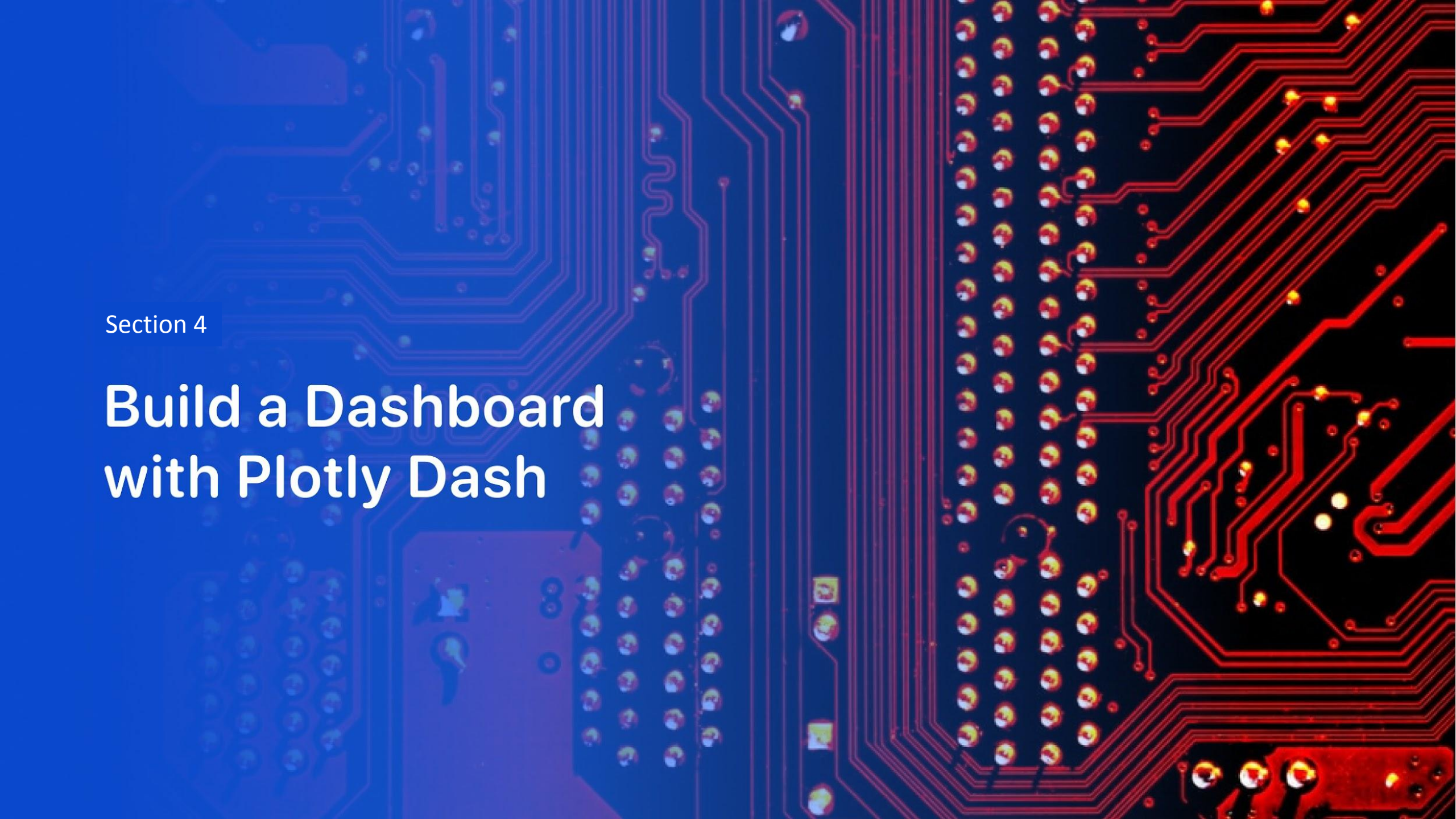


- The different launch sites are located on the east and west coast of the United States

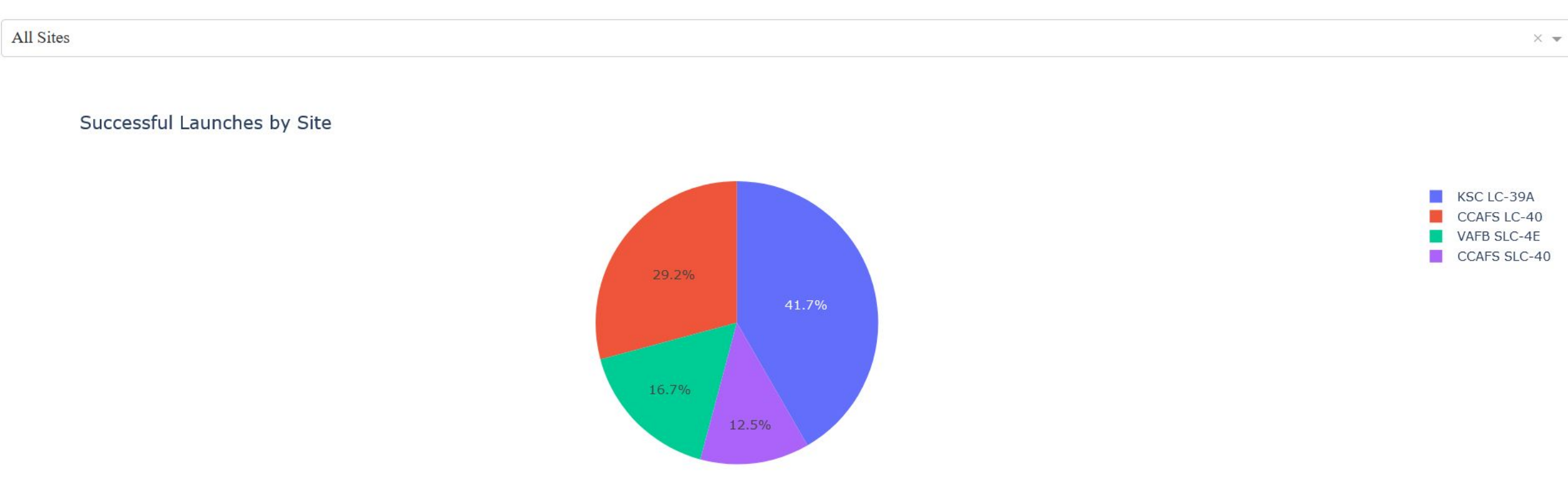
Map showing successful launches per site



- The above map narrows down to the launch sites on the east coast

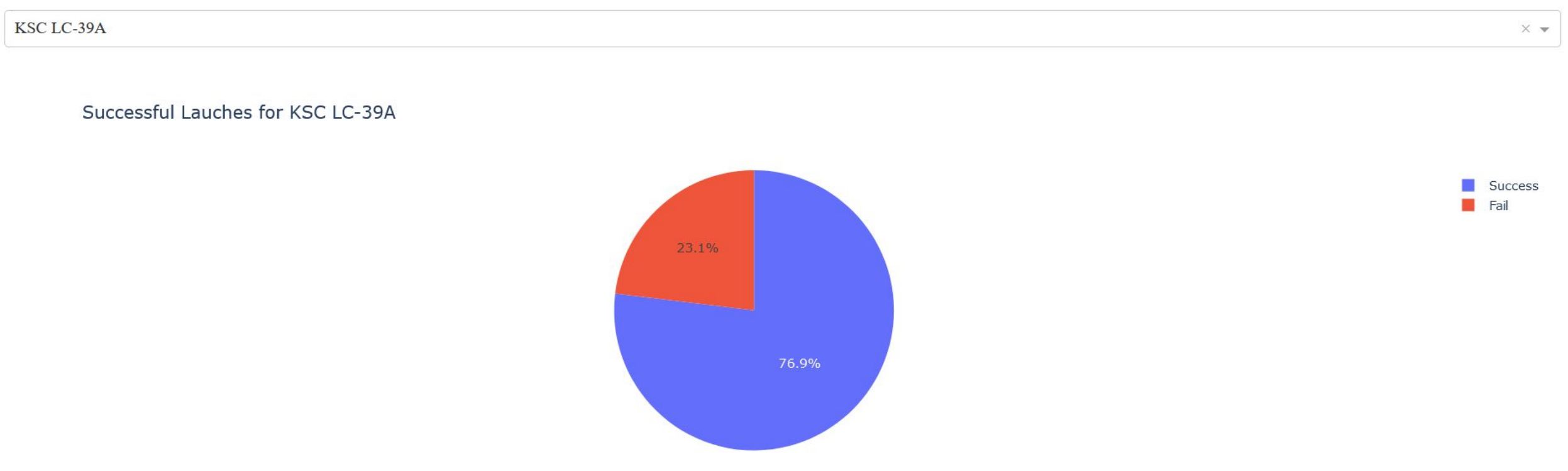

Section 4
Build a Dashboard with Plotly Dash

SpaceX Launch Records Dashboard - Pie Chart(1)



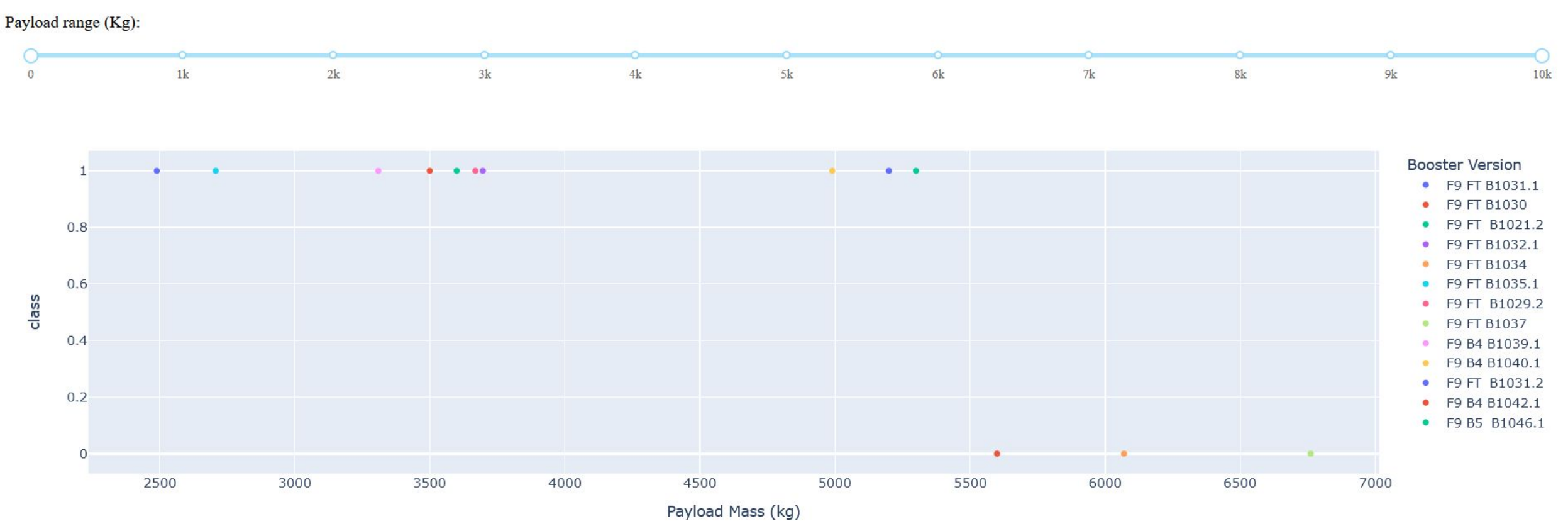
- The above screenshot shows “All Sites” selected
- KSC LC-39A has the highest success rate of 41.7%
- CCAFS SLC-40 has the lowest success rate of 12.5%

SpaceX Launch Records Dashboard - Pie Chart(2)



- The above screenshot shows “KSC LC-39A” selected
- KSC LC-39A has an individual 76.9% success rate
- KSC LC-39A has an individual 23.1% failure rate

SpaceX Launch Records Dashboard - Scatter Plot



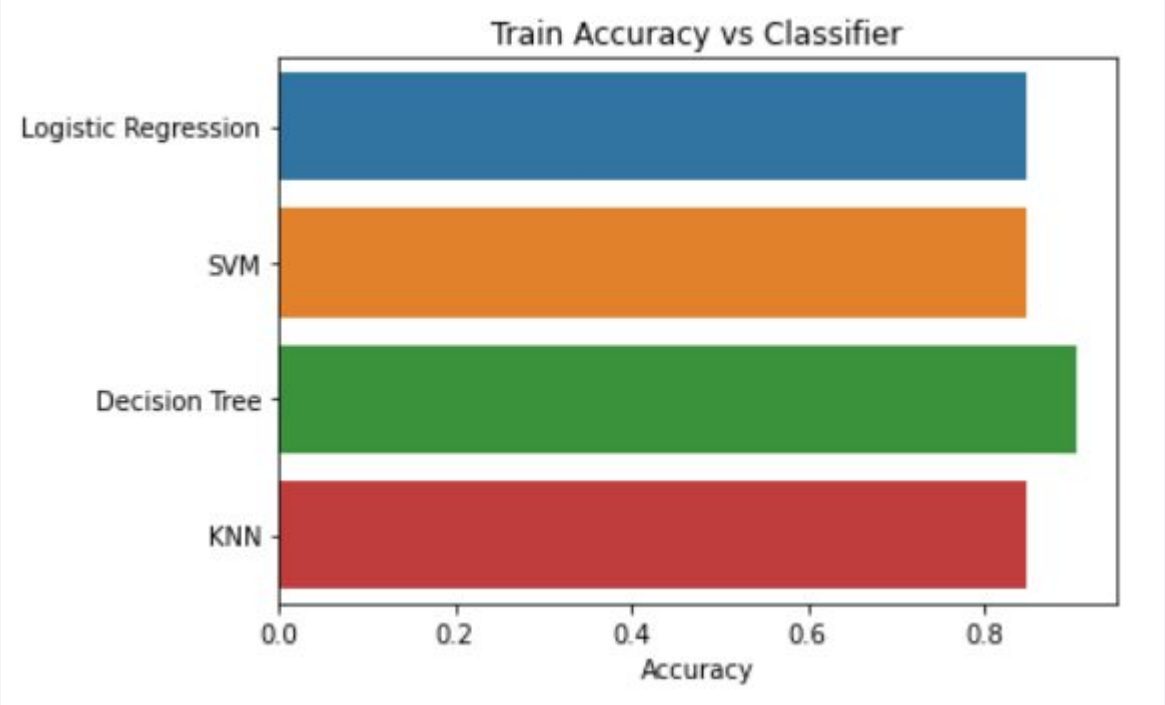
- Scatter plot showing relationship between payload mass (kg) and class

Section 5

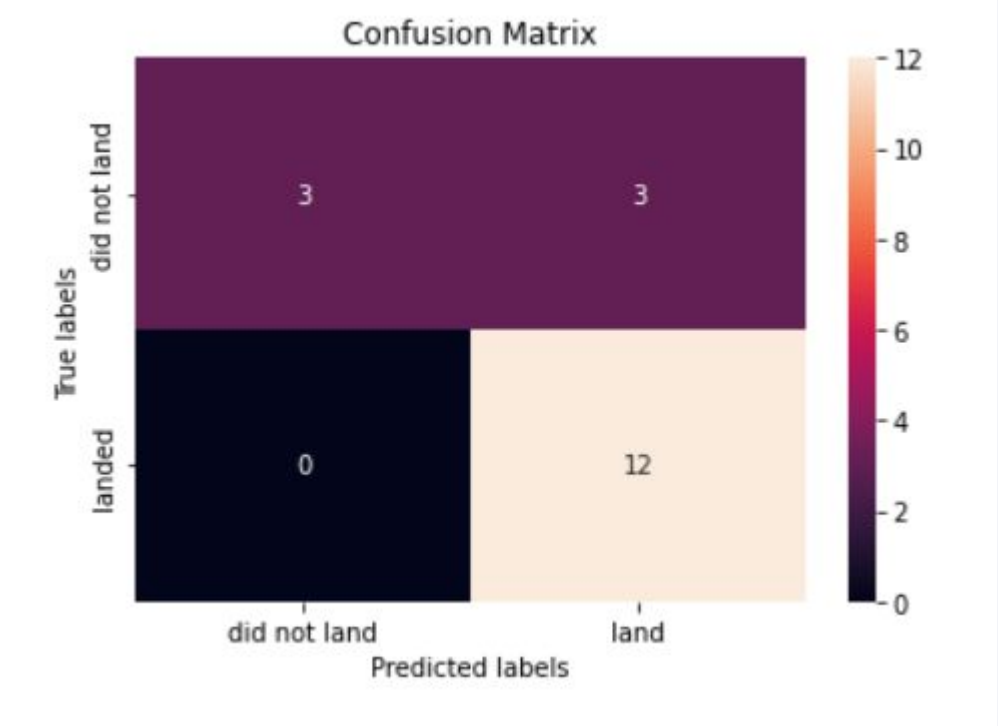
Predictive Analysis (Classification)

Classification Accuracy

- The Decision Tree Classifier has the highest training accuracy



Confusion Matrix



- True Positives 12
- True Negatives 3
- False Positives 0
- False Negatives 3

Conclusions

- Based on the result of our classifier, we will have an over 80% accuracy in making our predictions of determining whether Falcon 9 will launch successfully.

Appendix

- Dash Plotly Dropdown

```
from dash import Dash, dcc, html, Input, Output

app = Dash(__name__)
app.layout = html.Div([
    dcc.Dropdown(['NYC', 'MTL', 'SF'], 'NYC', id='demo-dropdown'),
    html.Div(id='dd-output-container')
])


@app.callback(
    Output('dd-output-container', 'children'),
    Input('demo-dropdown', 'value')
)
def update_output(value):
    return f'You have selected {value}'


if __name__ == '__main__':
    app.run_server(debug=True)
```

Appendix

- Dash Plotly Simple Slider

```
from dash import dcc, html, Input, Output

external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = Dash(__name__, external_stylesheets=external_stylesheets)

app.layout = html.Div([
    dcc.Slider(0, 20, 5,
               value=10,
               id='my-slider'
    ),
    html.Div(id='slider-output-container')
])

@app.callback(
    Output('slider-output-container', 'children'),
    Input('my-slider', 'value'))
def update_output(value):
    return 'You have selected "{}"'.format(value)

if __name__ == '__main__':
    app.run_server(debug=True)
```

Thank you!