IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Oksana Evdokimova
27 January 2023

Outline

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

Executive Summary

- Summary of methodologies
  - API for SpaceX data
  - Data collection via webscraping (Python / BeautifulSoup)
  - Data wrangling & analysis (Python / SQL)
  - Exploratory data analysis with Python, SQL and visulizations
  - Folium for map visualizations of data
  - Python Machine Learning for predictive analysis
- Summary of all results
  - The SpaceX Falcon 9 first stage rocket is predicted to land with high confidence after evaluating multiple ML models

Introduction

- Project background and context
  - Predicting the Falcon 9 first stage landing success using collected data of previous launches and machine learning models
- Problems you want to find answers
  - The chances of SpaceX meeting their advertised cost of launching and reusing rockets over time

Section 1

Methodology

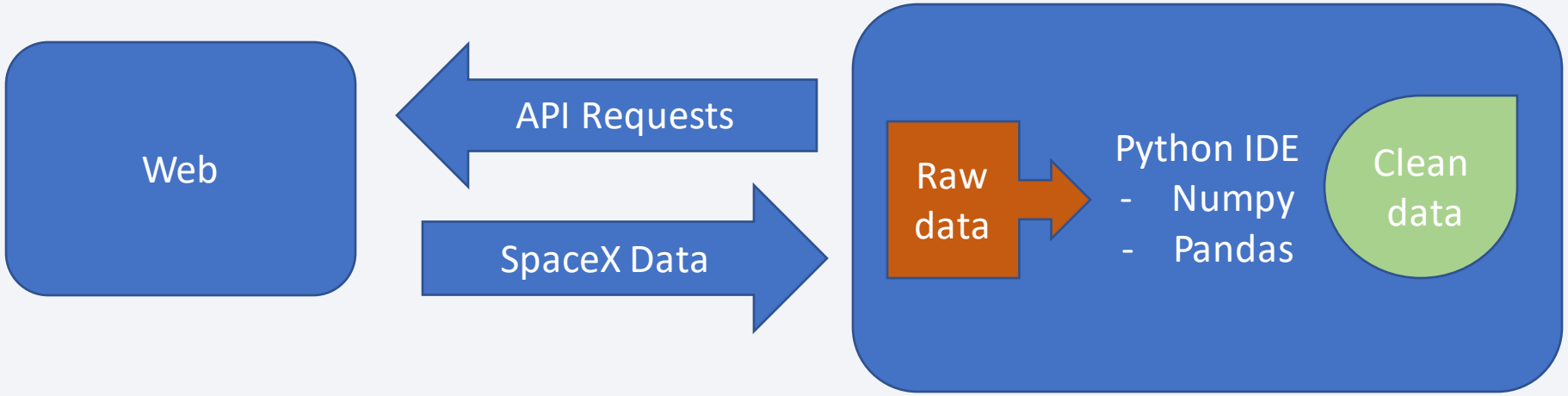
Methodology

Executive Summary

- Data collection methodology:
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

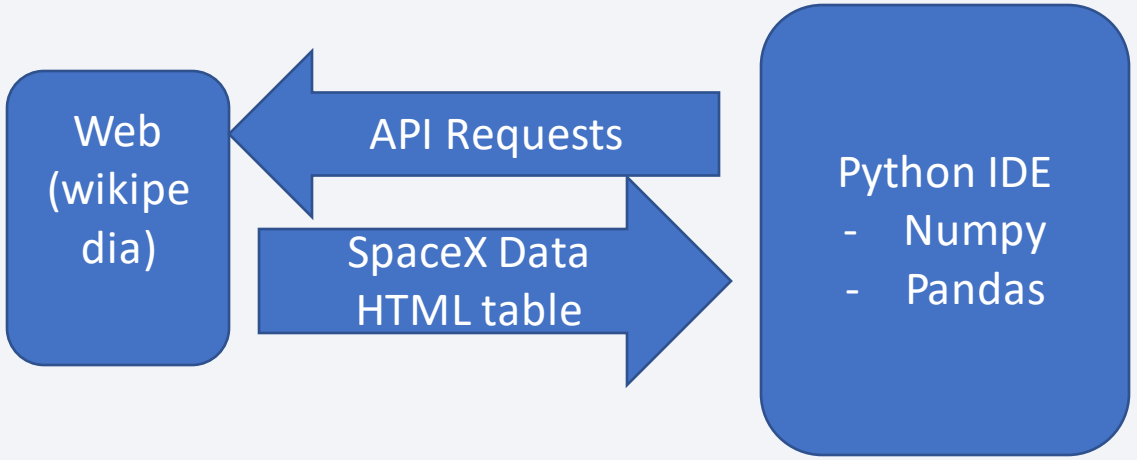
Data Collection

- Describe how data sets were collected.
- Using API and webscraping via Python to collect data in Python IDEA
- Numpy and pandas libraries to clean and format data set



Data Collection – SpaceX API

- https://github.com/jessedub/applied-data-science-capstone/blob/main/Data%20Collection%20API%20Lab.ipynb

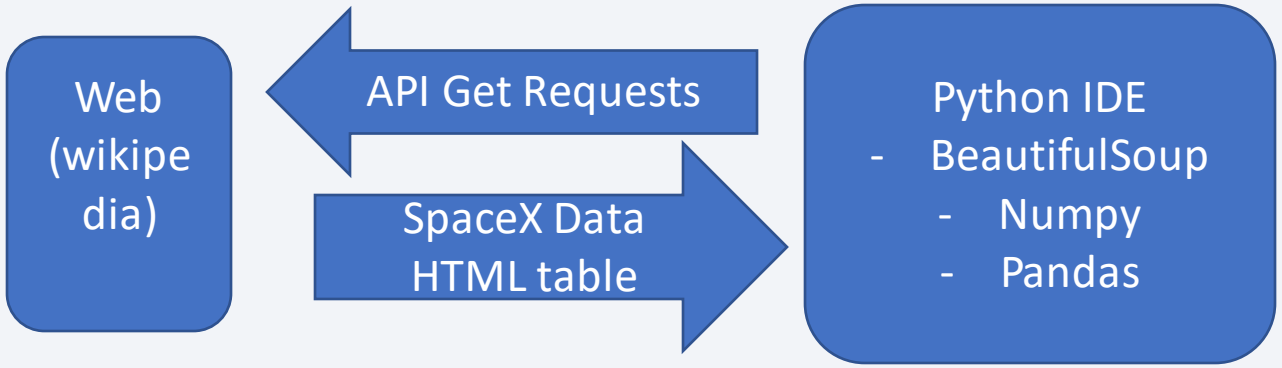


Example Python syntax for get requests

```
def getBoosterVersion(data):
    for x in data['rocket']:
       if x:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

Data Collection - Scraping

- https://github.com/jessedub/applied-data-science-capstone/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb



HTML table

```
<table class="wikitable plainrowhe
<tbody><tr>
<th scope="col">Flight No.
</th>
<th scope="col">Date and<br/>time
</th>
<th scope="col"><a href="/wiki/Lis
href="#cite_note-booster-11">[b]</
</th>
<th scope="col">Launch site
</th>
<th scope="col">Payload<sup class=
</th>
<th scope="col">Payload mass
```

Dataframe

BoosterVersion	PayloadMass	Orbit	LaunchSite
Falcon 9	6104.959412	LEO	CCAFS SLC 40
Falcon 9	525.000000	LEO	CCAFS SLC 40
Falcon 9	677.000000	ISS	CCAFS SLC 40
Falcon 9	500.000000	PO	VAFB SLC 4E
Falcon 9	3170.000000	GTO	CCAFS SLC 40

Data Wrangling

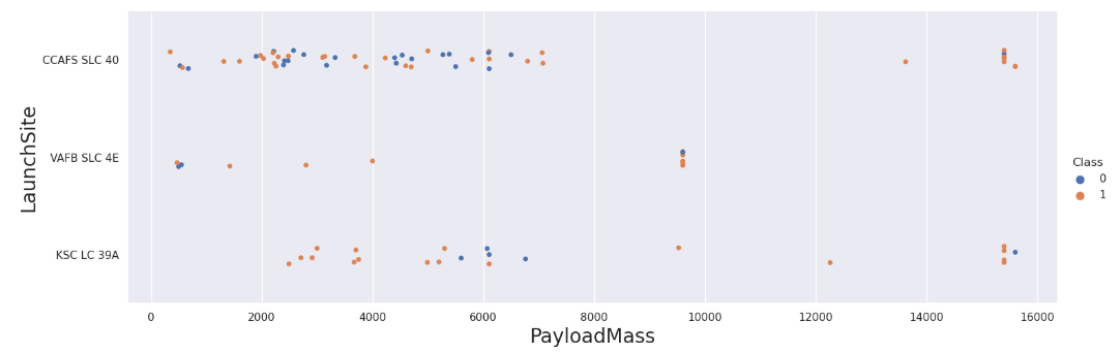
- Describe how data were processed
- Data was processed using Python libraries – pandas and numpy
- Value counts for launches at each site
- Occurences of each orbit
- Mission outcomes per orbit type
- Landing outcome classification

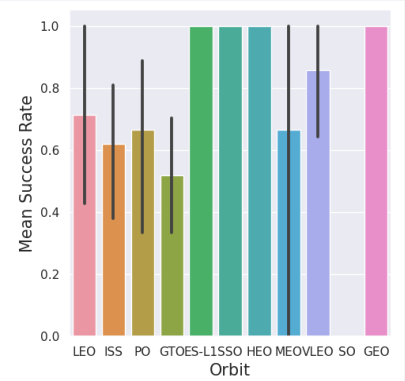
```
landing_class = df['Outcome'].map(lambda x: 0 if x in bad_outcomes else 1)

df['Class']=landing_class
df[['Class']].head(8)
```

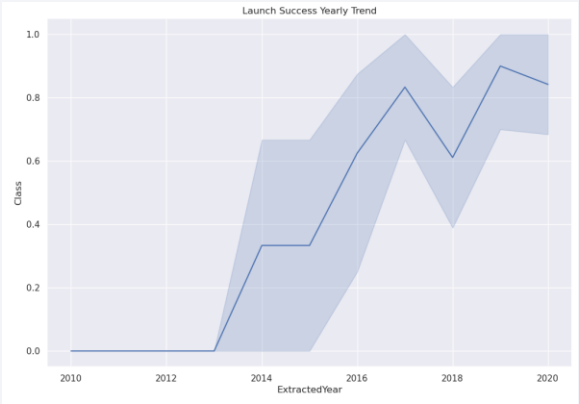
- https://github.com/jessedub/applied-data-science-capstone/blob/main/Data%20Wrangling.ipynb

EDA with Data Visualization

- Utilized matplotlib and seaborn for visualizations
- Catplots, bar charts and line plots







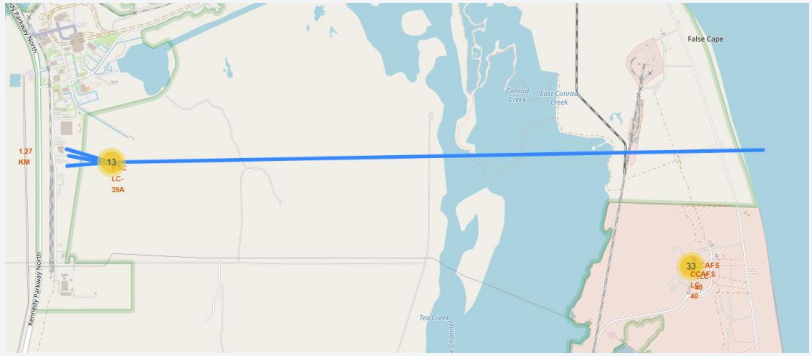
- https://github.com/jessedub/applied-data-science-capstone/blob/main/EDA%20with%20Visualization.ipynb

EDA with SQL

- Utilized IBM Db2 database to execute SQL queries and understand data set
- Queried:
  - Payload mass caried by boosters lauched by NASA
  - Average payload mass carried by booster F9 v1.1
  - Total number of successful and failure mission outcomes
  - + more
- https://github.com/jessedub/applied-data-science-capstone/blob/main/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Map objects we're added to give visual aid to Parametrics:
  - Coordinates via Mouse Position
  - MarkerCluster
  - Launch Site anchors
  - Distance markers



- https://github.com/jessedub/applied-data-science-capstone/blob/main/Data%20Visualization%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- ** Plotly Dash was optional in my Capstone Project

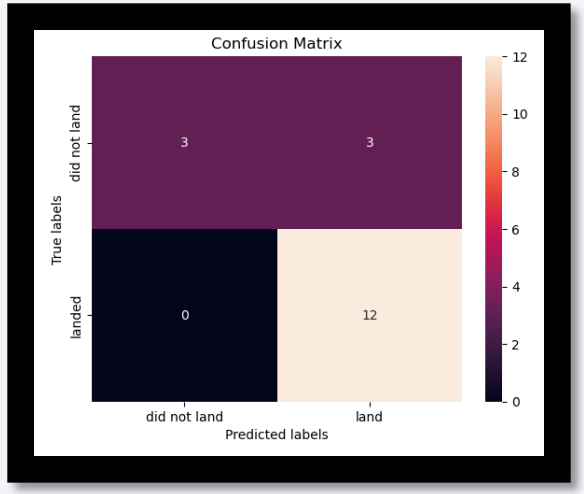
- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- Split data into training and test sets
- Exercised Hyperparameter for SVM, Classification Trees and Logistic Regression to find the most accurate predictor
- Discovered Decision Tree method performs best with test data

```
models = {'KNeighbors':knn_cv.best_score_,
          'DecisionTree':tree_cv.best_score_,
          'LogisticRegression':logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print(bestalgorithm, "method performs the best.")
```

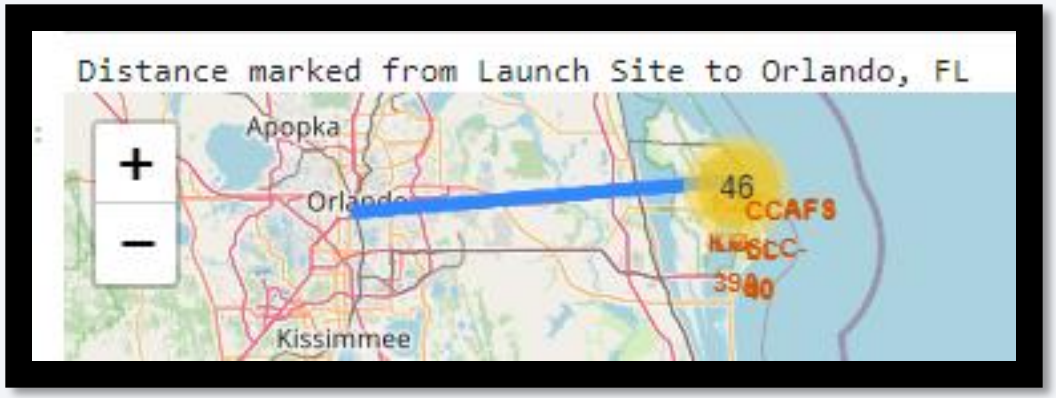
```
DecisionTree method performs the best.
```

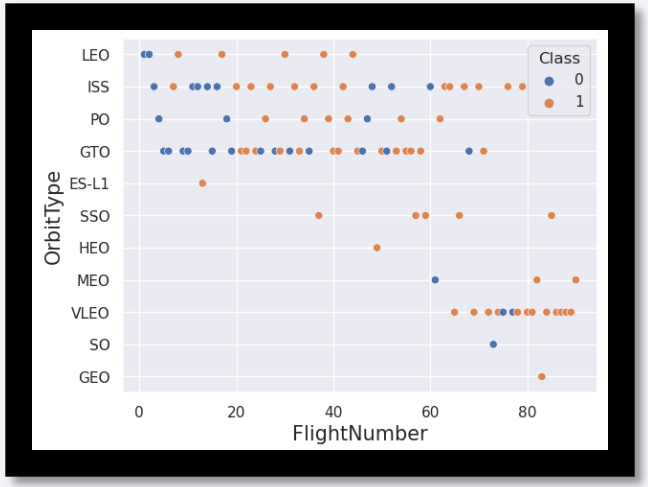


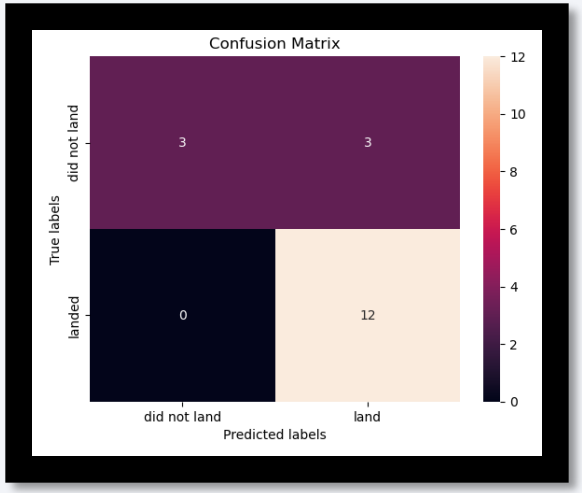
- https://github.com/jessedub/applied-data-science-capstone/blob/main/ML%20Prediction%20Lab.ipynb

Results

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





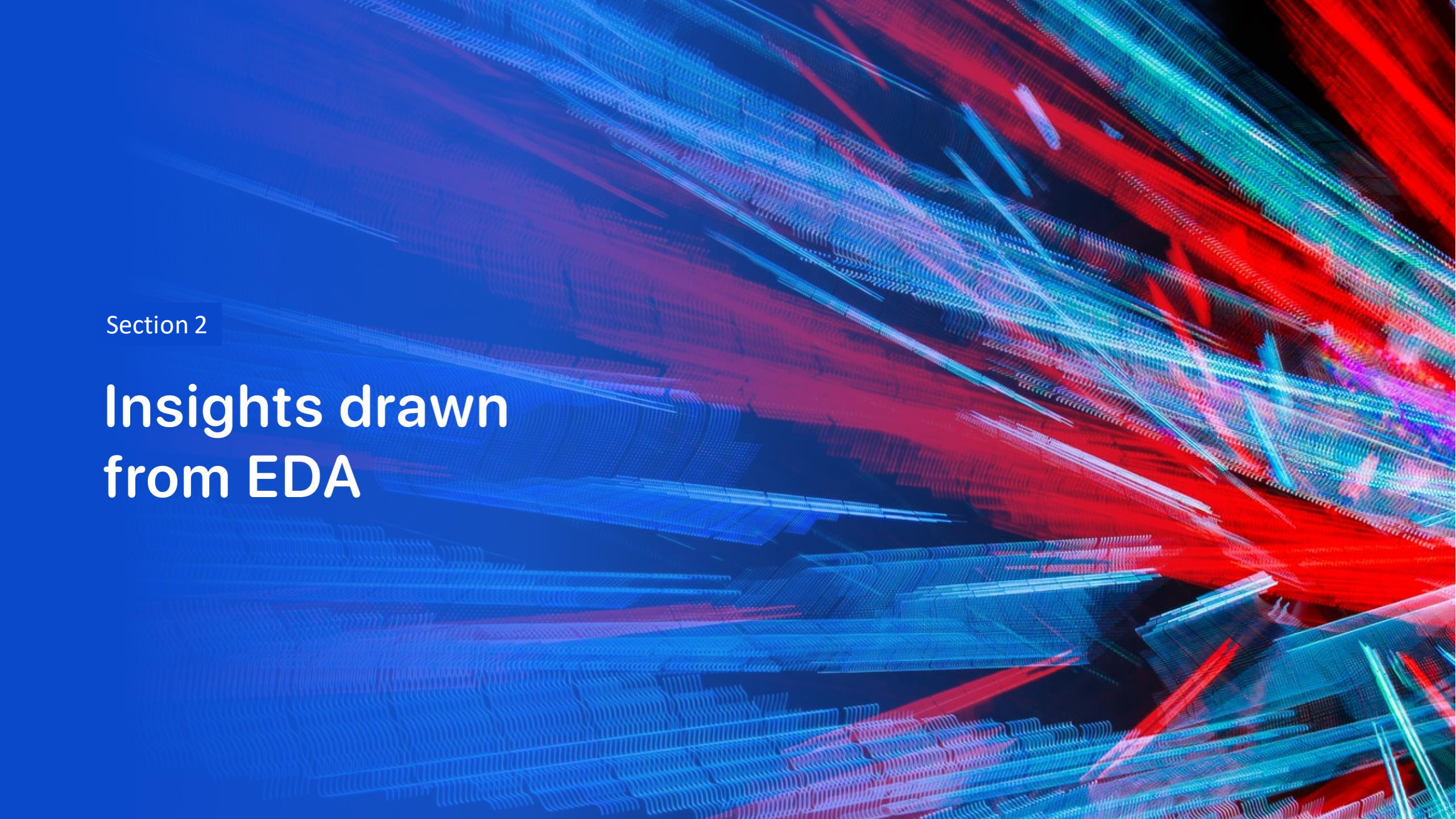



```
print("Test data accurary using 'tree_cv' 'score' method:", tree_cv.score(X_test, Y_test))
```

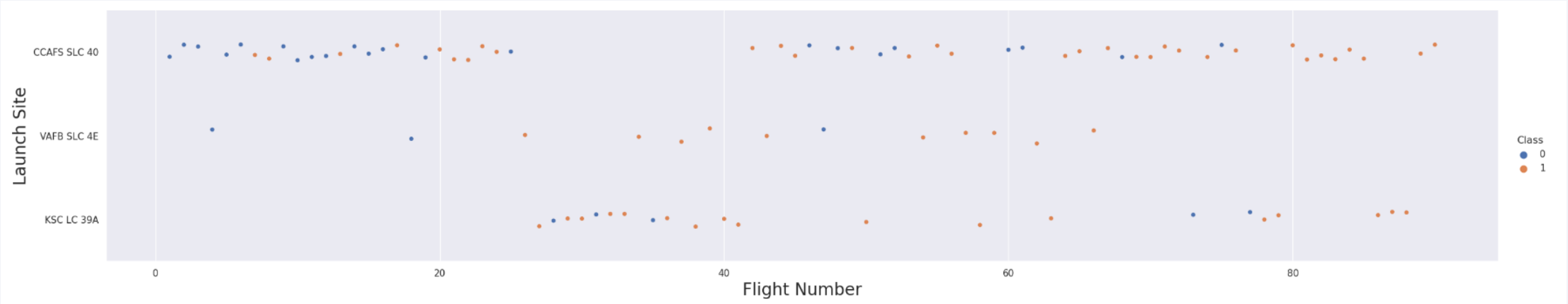
```
Test data accurary using 'tree_cv' 'score' method: 0.8888888888888888
```

Section 2

Insights drawn from EDA

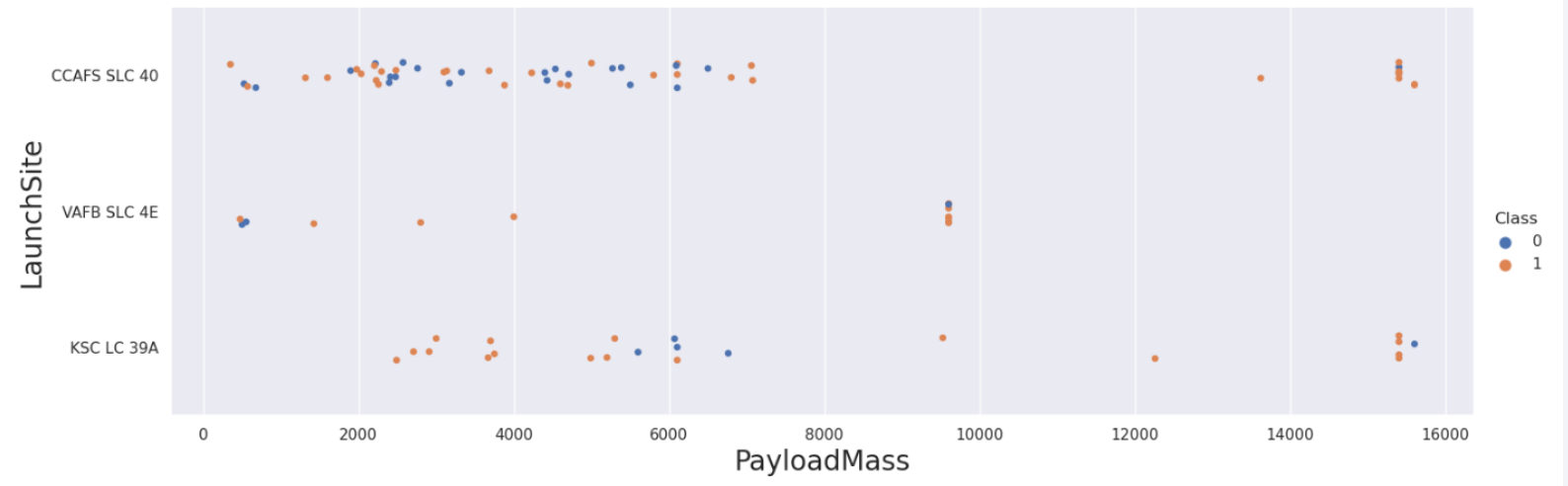


Flight Number vs. Launch Site



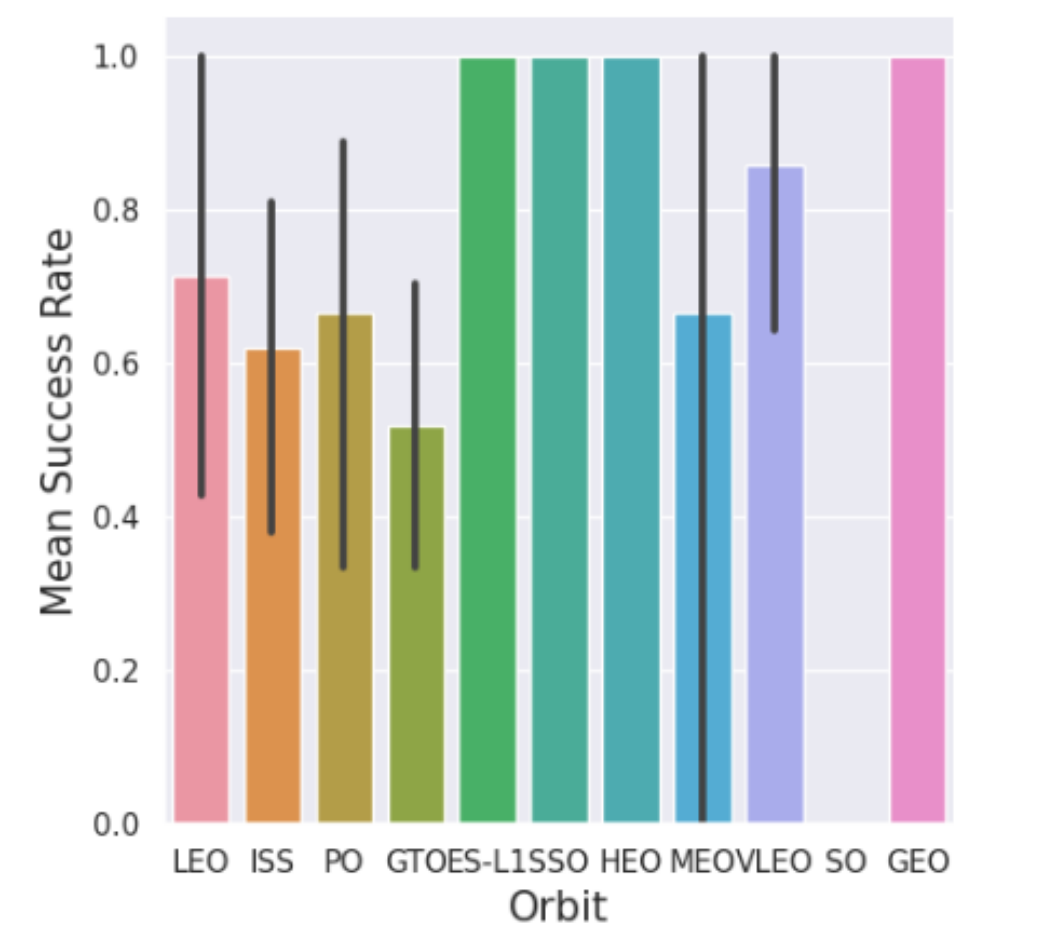
- VAFB SLC 4E has least number of Flights
- KSC LC 39A accumulated the most flights between flights 27 and 42
- CCASF SLC 40 had a flight hiatus during this time

Payload vs. Launch Site

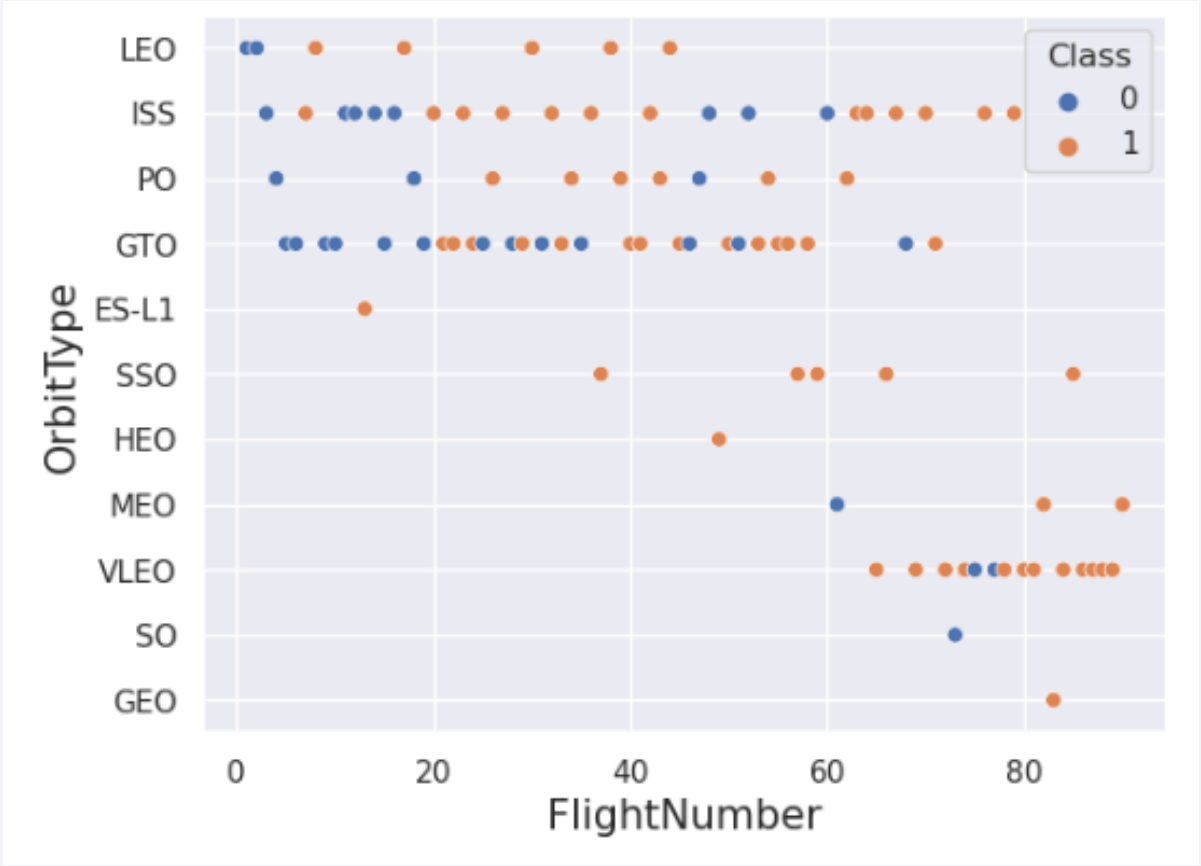


- VAFB SLC 4E had the least payload mass at less than 10,000 (units mass)
- KSC LC 39A has several successful launches with payload masses greater than 15,000 (units mass)
- CCASF SLC 40 had the most launches with payload mass less than 8000 (units mass)

Success Rate vs. Orbit Type

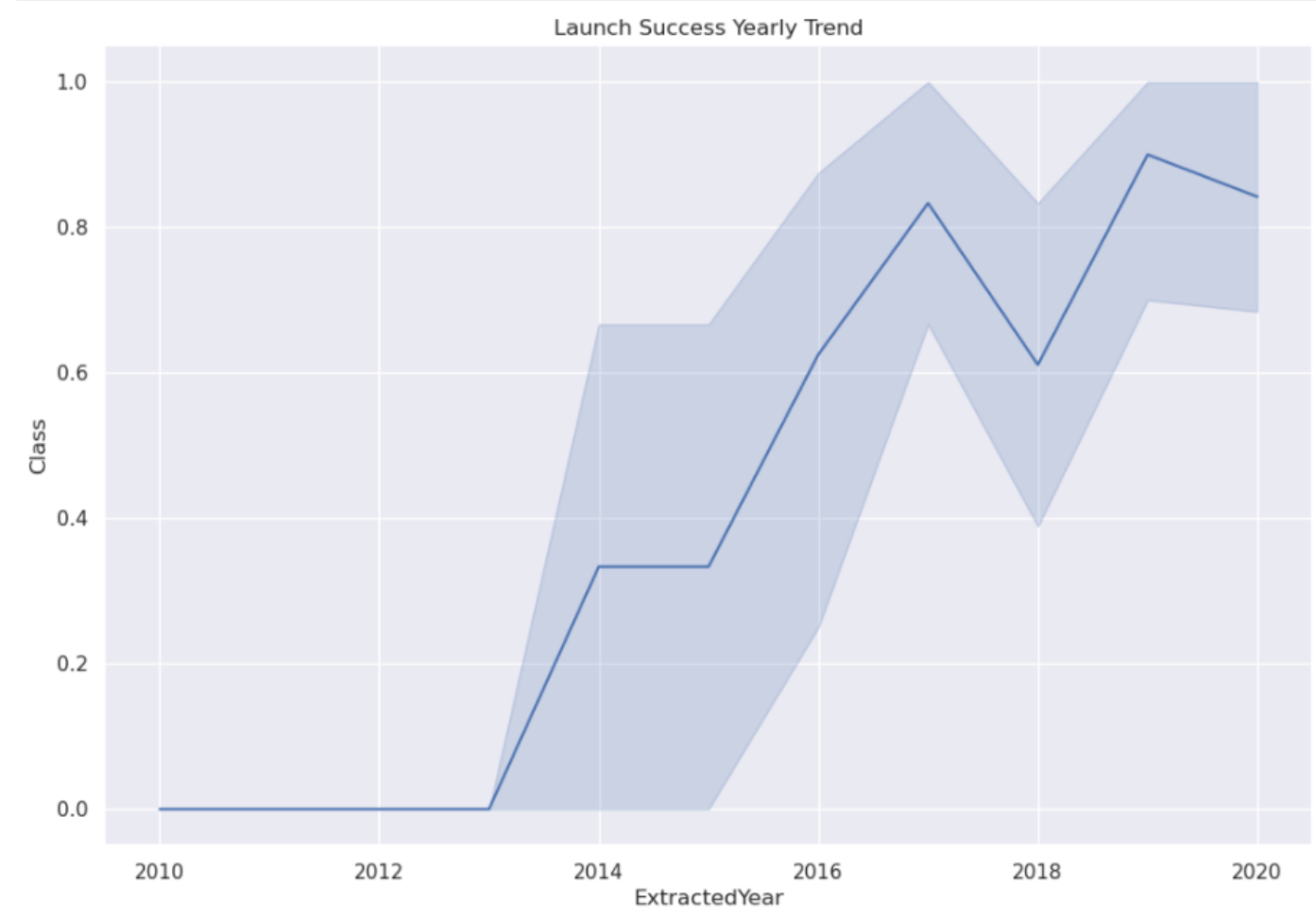


Flight Number vs. Orbit Type



Payload vs. Orbit Type

Launch Success Yearly Trend



All Launch Site Names

- IBM Db2 not working for me at this time. I'm resolving issue with IT support.

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEX;
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

Launch Site Names Begin with 'CCA'

- IBM Db2 not working for me at this time. I'm resolving issue with IT support.

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

```
%sql SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

Total Payload Mass

- IBM Db2 not working for me at this time. I'm resolving issue with IT support.

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

Average Payload Mass by F9 v1.1

- IBM Db2 not working for me at this time. I'm resolving issue with IT support.

Display average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEX \
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

First Successful Ground Landing Date

- IBM Db2 not working for me at this time. I'm resolving issue with IT support.

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad" FROM SPACEX \
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- IBM Db2 not working for me at this time. I'm resolving issue with IT support.

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

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

Total Number of Successful and Failure Mission Outcomes

- IBM Db2 not working for me at this time. I'm resolving issue with IT support.

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';


%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

Boosters Carried Maximum Payload

- IBM Db2 not working for me at this time. I'm resolving issue with IT support.

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX \
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

2015 Launch Records

- IBM Db2 not working for me at this time. I'm resolving issue with IT support.

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

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```

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

- IBM Db2 not working for me at this time. I'm resolving issue with IT support.

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

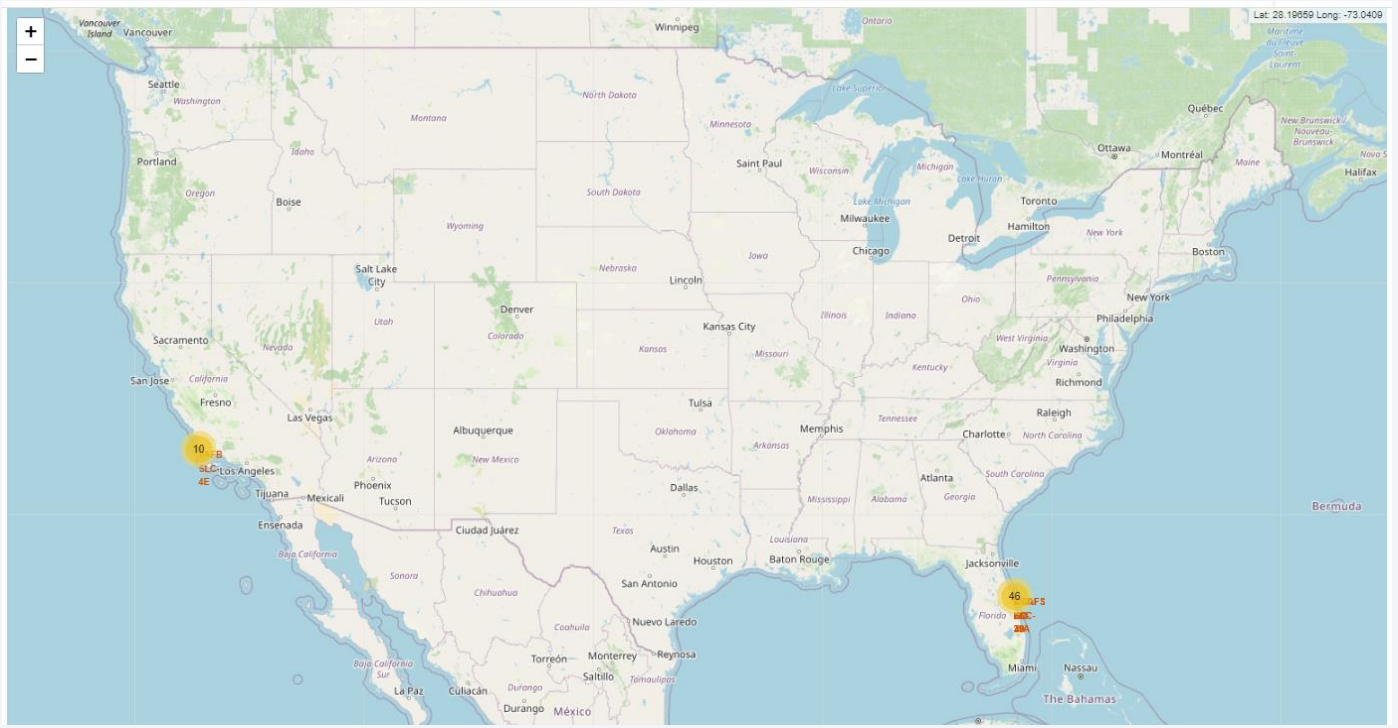
```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
Environment variable $DATABASE_URL not set, and no connect string given.
Connection info needed in SQLAlchemy format, example:
               postgresql://username:password@hostname/dbname
               or an existing connection: dict_keys([])
```


Section 3
Launch Sites
Proximities Analysis

Folium Map – USA Launch Sites

- Launch sites located in SW and SE corners of continental 48 (USA)
- Clustered sites identified in red text

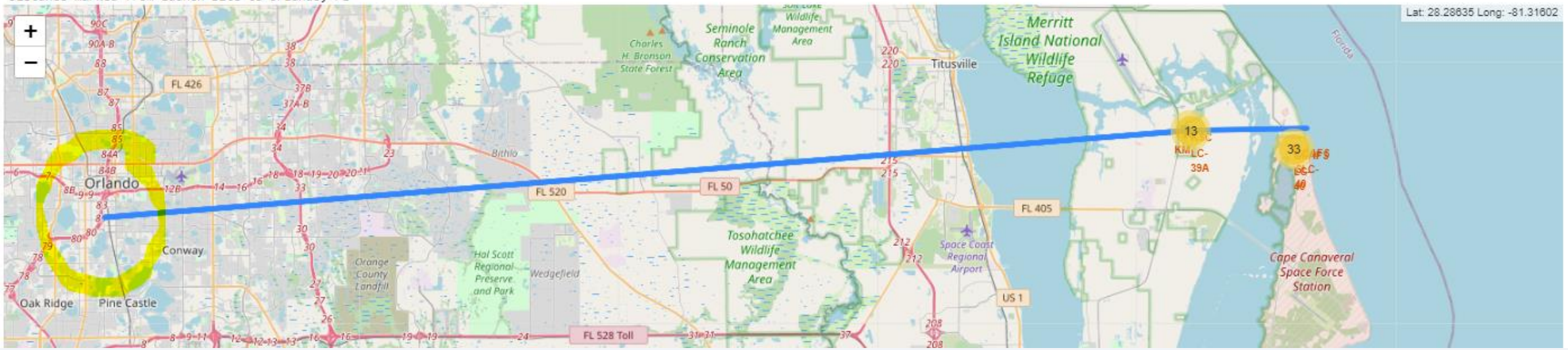


FL coast launch site distance to Orlando

- Blue line indicates scaled distance between launch site and major city

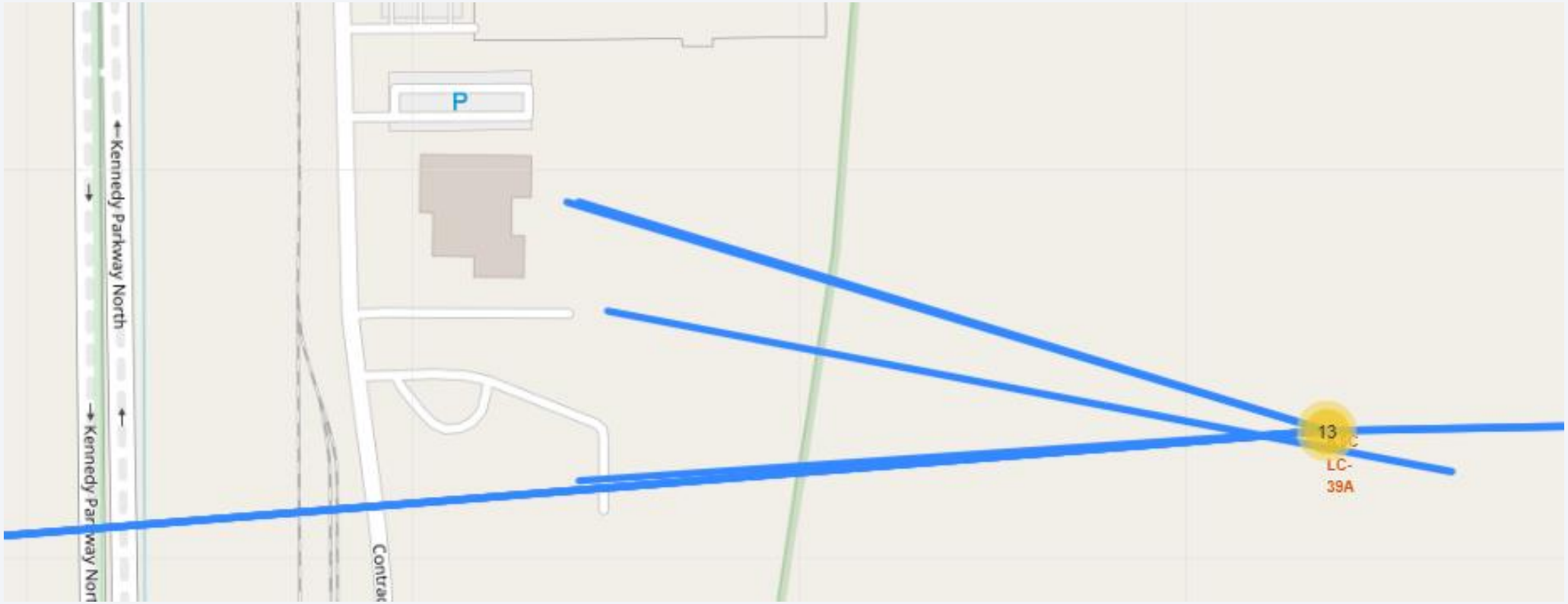
```
print('Distance marked from Launch Site to Orlando, FL')
coordinates=[[28.5218,-81.38397],[28.573255 ,-80.646895]]
lines=folium.PolyLine(locations=coordinates, weight=5)
site_map.add_child(lines)
```

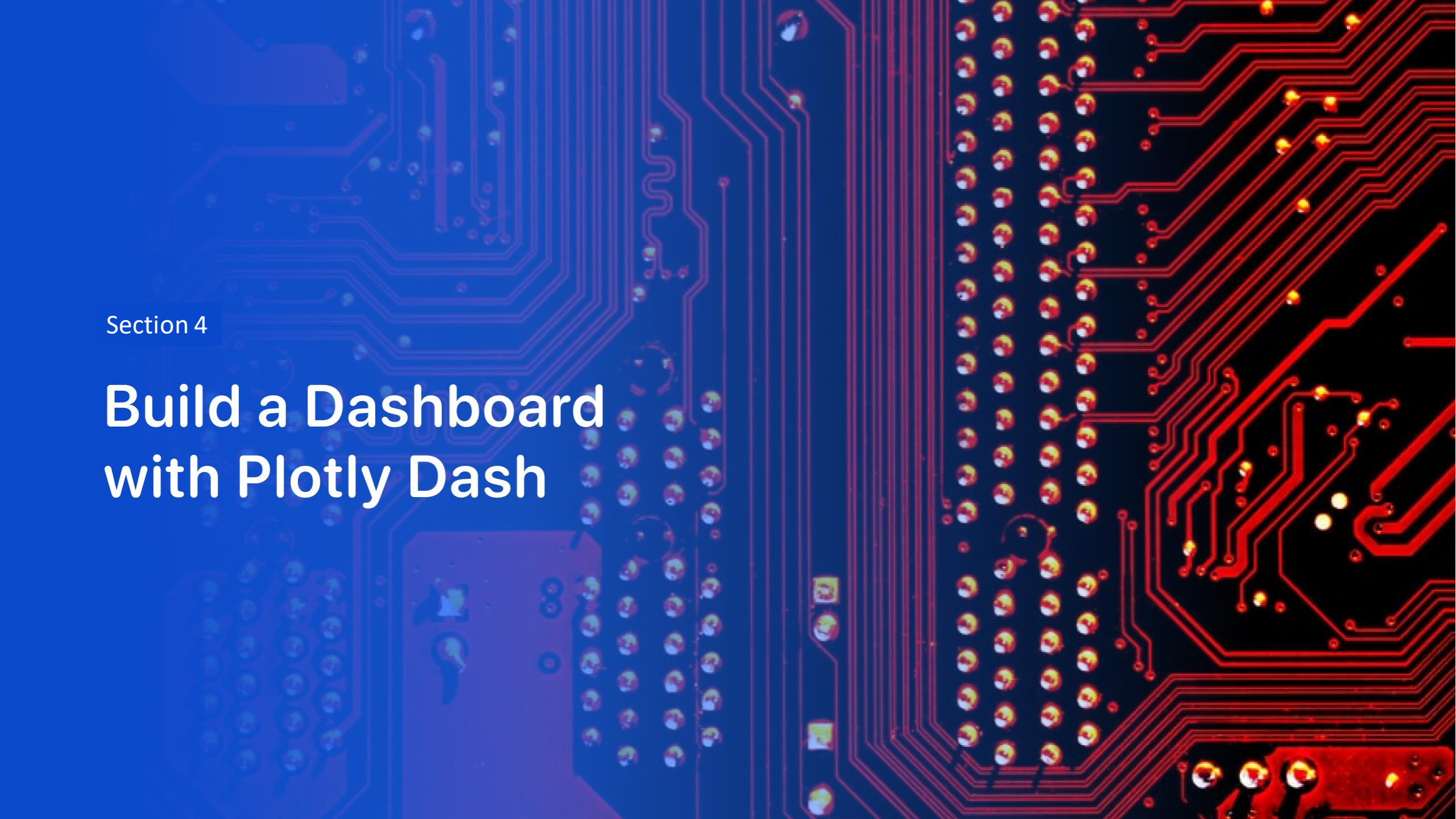
Distance marked from Launch Site to Orlando, FL



Visual scaled distance to pedestrian areas

- Scaled distances marked from launch site to nearby building & highway




Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- ** Plotly Dash was optional in my Capstone Project
- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- ** Plotly Dash was optional in my Capstone Project
- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- ** Plotly Dash was optional in my Capstone Project
- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The Decision Tree model had the best accuracy score and performed the best

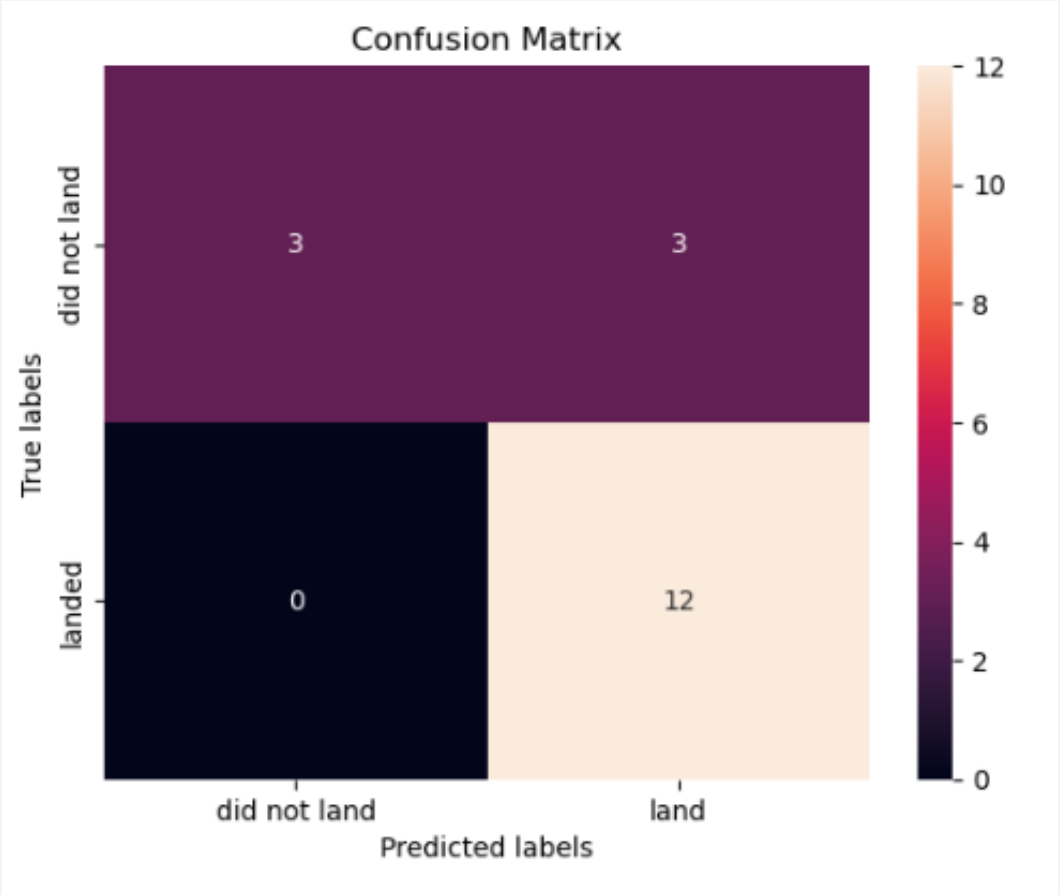
Find the method performs best:

```
models = {'KNeighbors':knn_cv.best_score_,
                'DecisionTree':tree_cv.best_score_,
                'LogisticRegression':logreg_cv.best_score_,
                'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print(bestalgorithm, "method performs the best.")
```

```
DecisionTree method performs the best.
```

Confusion Matrix

- 12 Count for True Positive
- 3 Count for True Negative



Conclusions

- Many launches from multiple US sites
- Payloads varying between roughly 1000 – 15,000 (units mass)
- Visualizations indicate successes & failures of mission success
- Predictive analysis is highly confident of future mission success
- Will analyze more data and refine model with more launches

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!