IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Yifan Jiang>
<01/02/2022>

Outline

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

Executive Summary

- Summary of
  - ✓ Collecting Data from SpaceX API
  - ✓ Collecting Data from Web Scraping
  - ✓ Data Wrangling
  - ✓ EDA with Visualization & SQL
  - ✓ Building interactive map with Folium
  - ✓ Building Dashboard using Plotly Dash
  - ✓ Predictive analysis - Classification
- Summary of all results
  - ✓ EDA resulting (Visualization & SQL)
  - ✓ Interactive analytics results
  - ✓ Predictive analysis results

Introduction

- Project background and context

In this project, we will forecast if the Falcon 9 first stage will successfully land. SpaceX advertises Falcon 9 rocket launches on its website for 62 million dollars; other companies charge upwards of 165 million dollars apiece; a large portion of the savings is due to SpaceX's ability to reuse the first stage. As a result, if we can predict whether the first stage will land, we can estimate the cost of a launch. This data can be used if another firm wishes to compete with SpaceX for a rocket launch. This module will give you an overview of the problem as well as the tools you'll need to finish the course.

- Problems you want to find answers

  - ✓ What factors determine whether or not the rocket lands successfully?
  - ✓ The influence that each relationship with specific rocket characteristics will have on the success rate of a successful landing.
  - ✓ What requirements must SpaceX meet in order to obtain the greatest results and ensure the highest rate of rocket landing success?

Section 1

Methodology

Methodology

- Data collection methodology:
  - Collect SpaceX launch data from SpaceX API
  - Another approach of data collecting is web scraping (Wikipedia) using BeautifulSoup Package
- Perform data wrangling
  - Calculate the number of launches on each site, the number and occurrence of each orbit, and the number and occurrence of mission outcome per orbit type
  - Create a landing outcome label from Outcome 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
  - Evaluate the models with comparing their accuracy scores
  - Look at the confusion Matrix for additional insights

Data Collection – Space X API

Data Collect Flow Chart

GitHub URL

1. Requesting rocket launch data from SpaceX API

```
response = requests.get(spacex_url)
response
```

2. Convert the json result into a dataframe

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Apply function to clean data

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4. Apply function to clean data

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Data Collection - Scraping

GitHub URL

1. Request the Falcon9 Launch HTML

```
page = requests.get(static_url)
page.status_code
```

2. Create a BeautifulSoup object from response text content

```
soup = BeautifulSoup(page.text, 'html.parser')
```

3. Find Tables

```
html_tables = soup.find_all('table')
```

4. Get Column Names

```
column_names = []

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

5. Create Dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Append Table

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikit
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corres
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

7. Dictionary to dataframe to .CSV

```
df=pd.DataFrame(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling - Summary

- In data wrangling part, we performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

Data Wrangling - Flowchart

GitHub URL

1. **Calculate the number of launches on each site**

```
df["LaunchSite"].value_counts()
```

2. **Calculate the number and occurrence of each orbit**

```
df["Orbit"].value_counts("Orbit")
```

3. **Calculate the number and occurence of mission outcome per orbit type**

```
landing_outcomes = df["Outcome"].value_counts()
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

4. **Create a landing outcome label from Outcome column**

- Assign bad outcomes to variable landing_class

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

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

- Success Rate

```
df["Class"].mean()
```

```
0.6666666666666666
```

- Export to .CSV

```
df.to_csv("dataset_part_2.csv", index=False)
```

EDA with Data Visualization

GitHub URL

1. Plot Scatter Point Chart

- Payload Mass VS. Flight Number
- Flight Number VS. Launch Site
- Payload and Launch Site
- Flight Number VS. Orbit type
- Payload VS. Orbit type

2. Analyze the plotted bar chart try to find which orbits have high success rate.

- Result: GEO, HEO, SSO, ES-L1 has the top 4 success rate.

3. Plot a line chart of Success Rate VS. Year

- Result: The success rate since 2013 kept increasing till 2020

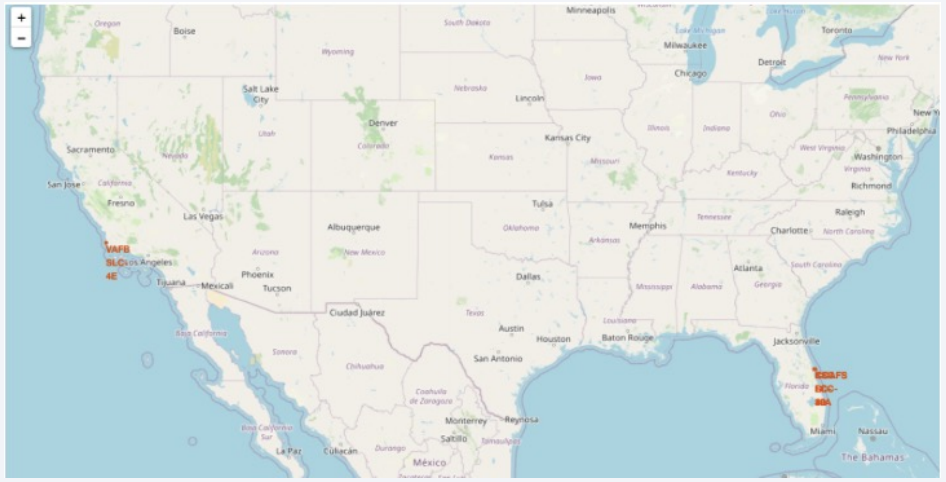
EDA with SQL

- Understand the Spacex DataSet
- Load the dataset into the corresponding table in a Db2 database
- Execute SQL queries to answer assignment questions
  - ✓ 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 successful landing outcome in ground pad was achieved
  - ✓ List the names of the boosters which have success in drone ship and have payload mass greater than 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. Use a subquery
  - ✓ List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
  - ✓ Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

GitHub URL

Build an Interactive Map with Folium 1

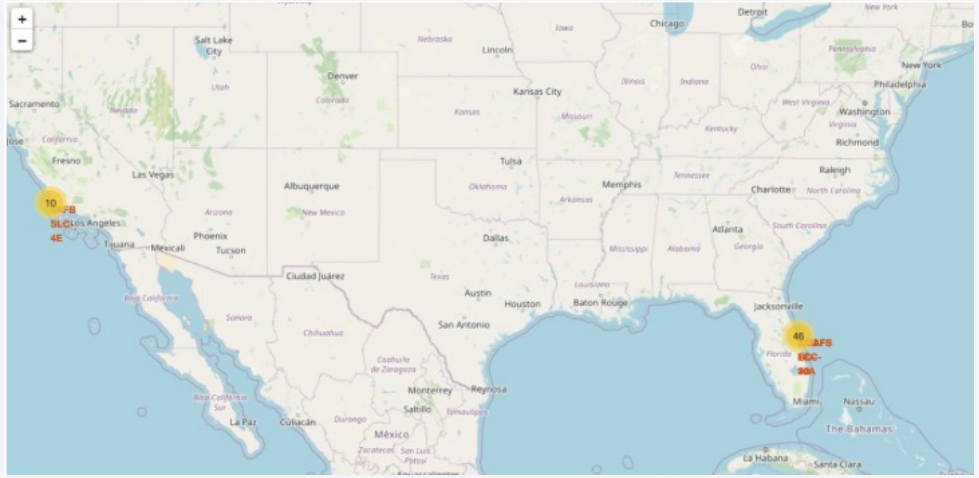
- Mark all launch sites on a map



- Why?
  - Try to answer the following questions:
    - Are all launch sites in proximity to the Equator line?
    - Are all launch sites in very close proximity to the coast?

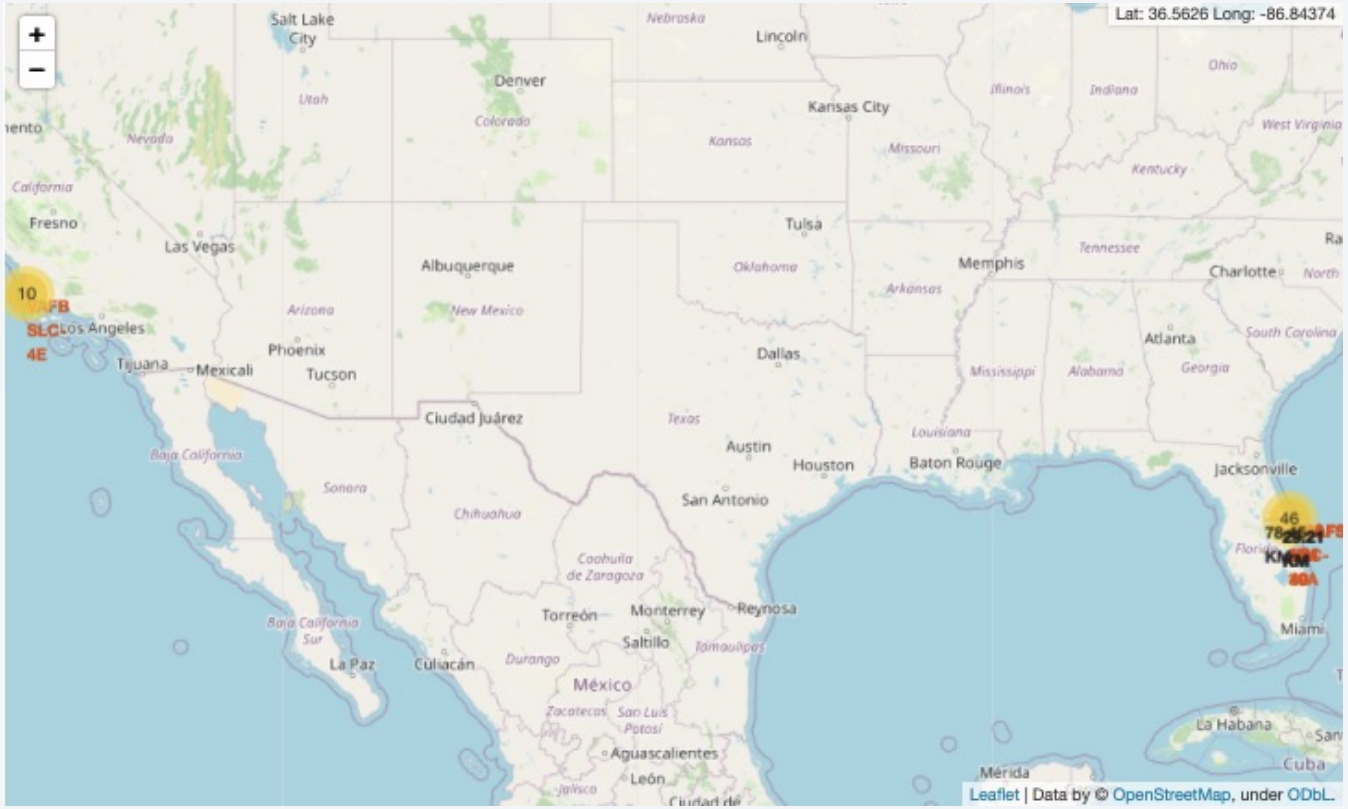
GitHub URL

- Mark the success/failed launches for each site on the map



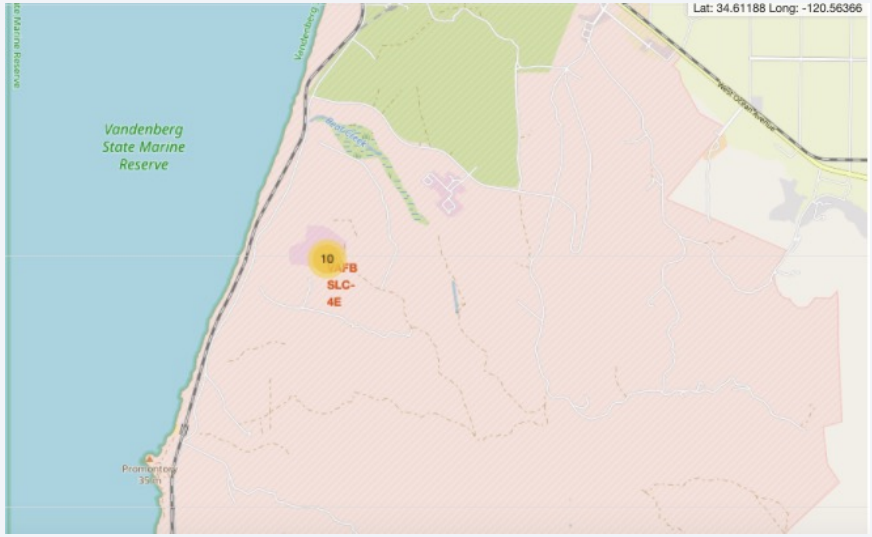
Build an Interactive Map with Folium 2

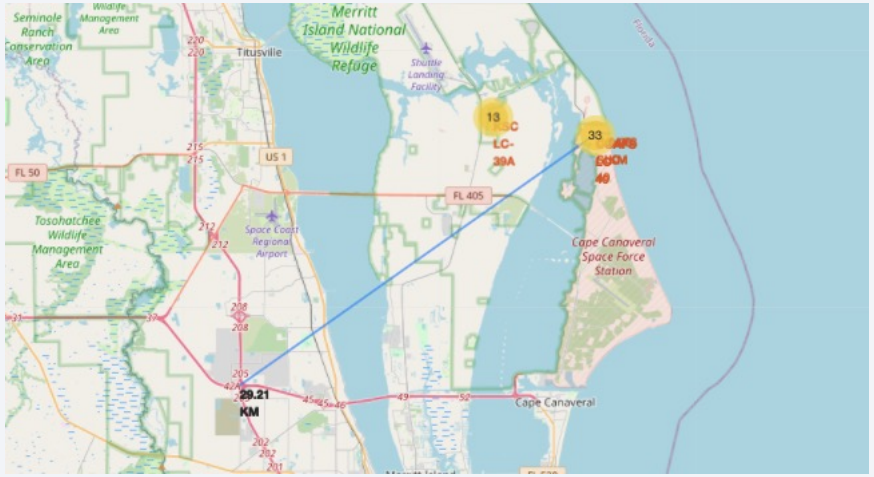
- Calculate the distances between a launch site to its proximities
  - After obtained its coordinate, create a folium.Marker to show the distance



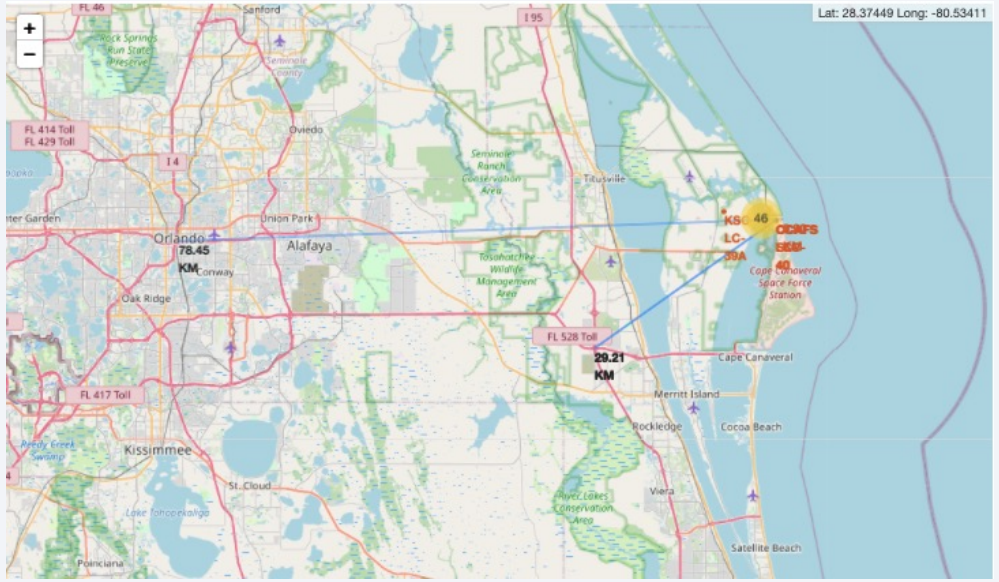
Build an Interactive Map with Folium 3

- Distance to Highway





- Distance to Florida



- **Answers of previous questions**
  - Are launch sites in close proximity to railways? (NO)
  - Are launch sites in close proximity to highways? (NO)
  - Are launch sites in close proximity to coastline? (YES)
  - Do launch sites keep certain distance away from cities? (YES)

Build a Dashboard with Plotly Dash

GitHub URL

- 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)

GitHub URL

- Building Model
  - Create a NumPy array from the column Class
  - Standardize the data in X then reassign it to X
  - Split the data X and Y into training and test data
  - Create a logistic regression object then create a GridSearchCV object
  - Create a support vector machine object then create a GridSearchCV object
  - Create a decision tree classifier object then create a GridSearchCV object
  - Create a k nearest neighbors object then create a GridSearchCV object
- Evaluating Model
  - Calculate the accuracy on the test data using the method score for each model
  - Plot the confusion matrix
  - Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Improving Model
  - Feature Engineering
  - Algorithm Tuning
- Finding the Best Performing Model (Classification)
  - Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
  - The model with the highest accuracy score is declared the best performing model

Results

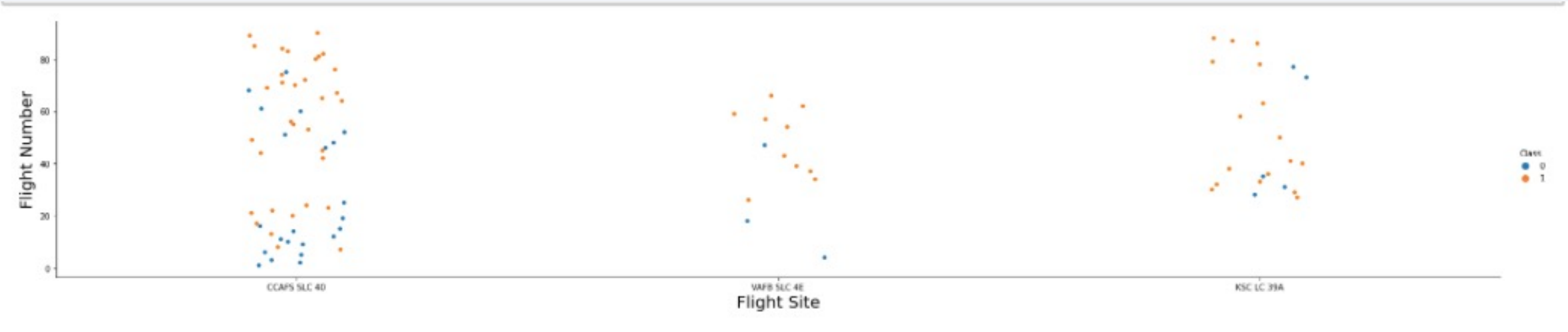
- Insights drawn from EDA visualization (Section 2)
- EDA with SQL queries (Section 3)
- Launch Sites Proximities Analysis – Interactive Folium (Section 4)
- Dashboard with Plotly Dash (Section 5)
- Predictive Analysis – Classification (Section 6)

Section 2

Insights drawn from EDA (Visualization)

Flight Number vs. Launch Site

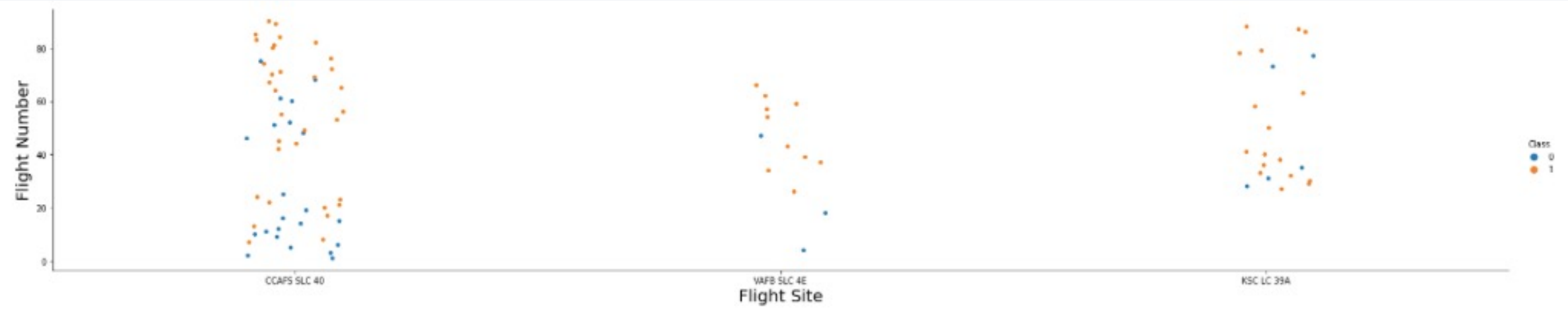
- Scatter plot of Flight Number vs. Launch Site



Explanation: The more the number of flights at a launch site, the higher the success rate.

Payload vs. Launch Site

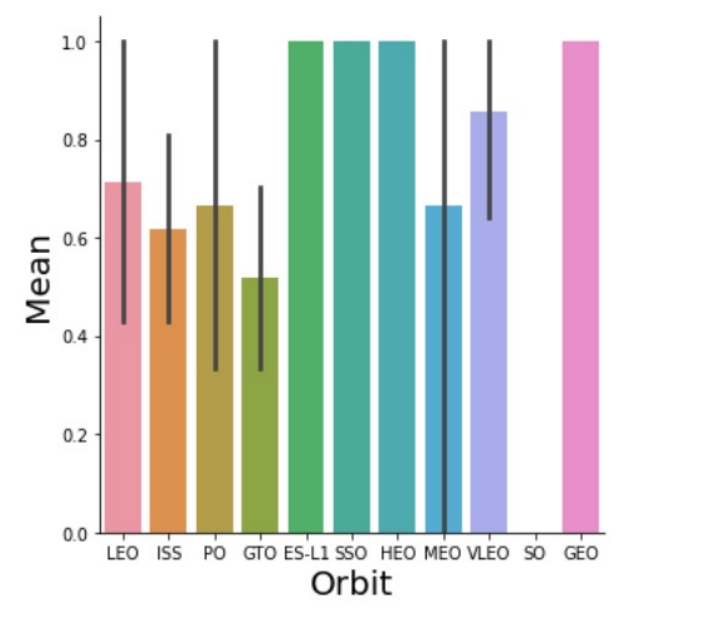
- Scatter plot of Payload vs. Launch Site



Explanation: The larger the payload mass for Launch Site CCAFS SLC 40, the better the Rocket's success rate.

Success Rate vs. Orbit Type

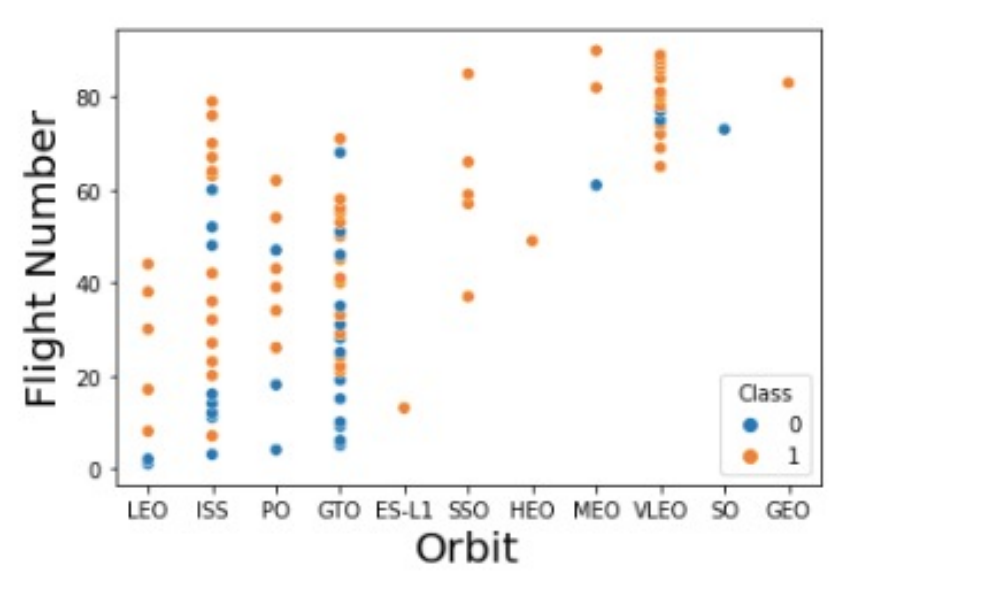
- Bar chart of Success Rate vs. Orbit Type



Explanation: GEO, HEO, SSO, ES-L1 has the top 4 success rate.

Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type

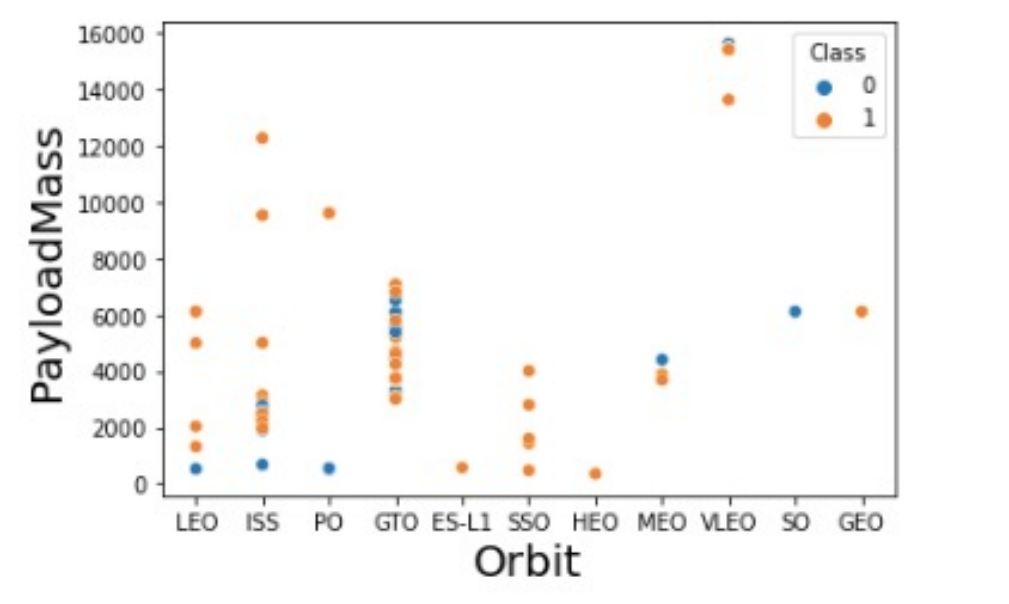


✓ The number of flights in LEO appears to be proportional to success.

✓ When in GTO orbit, there appears to be no correlation between flight numbers.

Payload vs. Orbit Type

- Scatter point of Payload vs. Orbit type

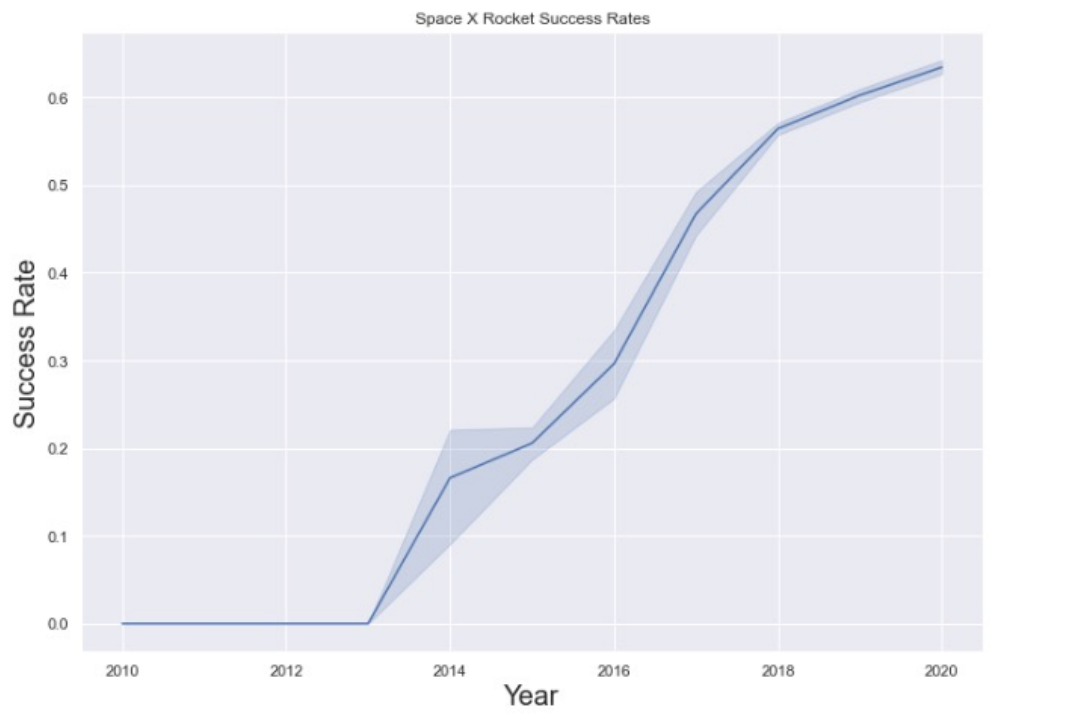


✓ For Polar, LEO, and ISS, the successful or positive landing rate is higher with large payloads.

✓ It's hard to clearly define the relationship for GTO, since both positive landing rate and negative landing rate (unsuccessful mission) are present.

Launch Success Yearly Trend

- Scatter point of Payload vs. Orbit type



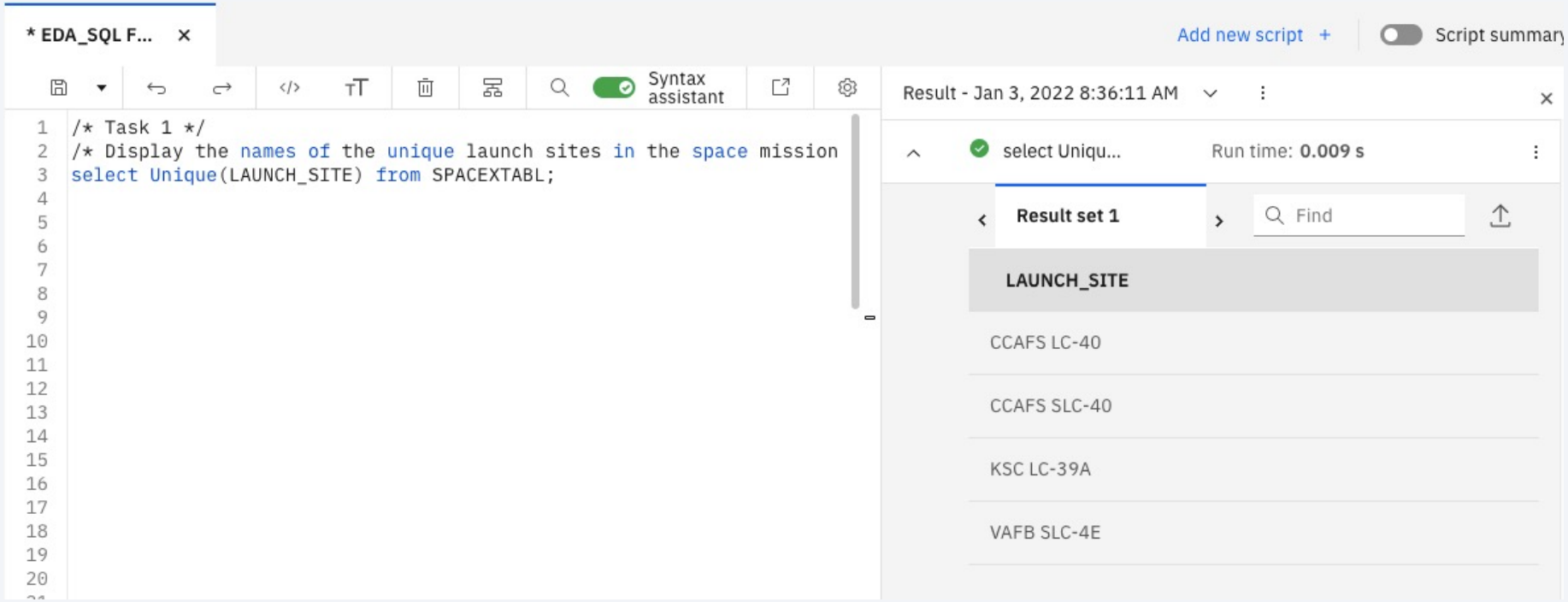
✓ Since 2013, the success rate has been steadily increasing till 2020.

Section 2

Insights drawn from EDA (SQL)

All Launch Site Names

- SQL Queries



- Explanation: Unique argument in the query only shows unique values in Launch_Site

Launch Site Names Begin with 'CCA'

- SQL Queries

```
/* Task 2 */
/* Display 5 records where launch sites begin with the string 'CCA' */
SELECT LAUNCH_SITE from SPACEXTABL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

/* Task 2 */ /* Display 5 records where launch site... Run time: 0.010 s

Result set 1 Find

LAUNCH_SITE
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- Explanation: LIMIT 5 shows 5 records from the database SPACEXTANL; and LIKE %CCA shows Launch_Sites with CCA

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
/* Task 3 */
/* Display the total payload mass carried by boosters launched by NASA (CRS) *
select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTABL;
```

/* Task 3 */ /* Display the total payload mass carri...

Result set 1

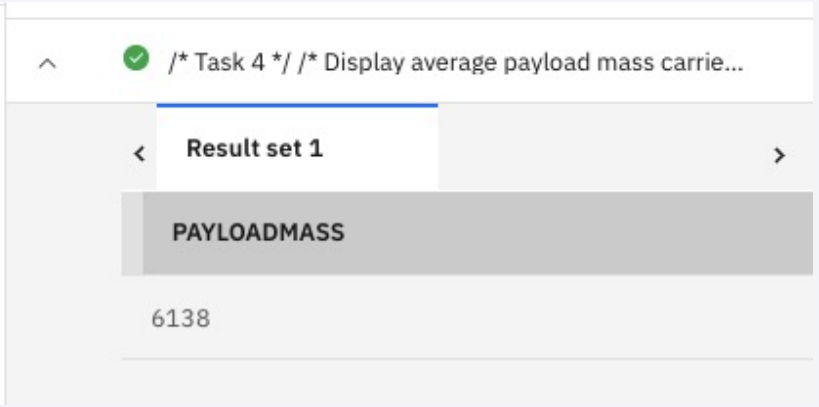
PAYLOADMASS
619967

- Sum argument sums the total in the column of PAYLOAD_MASS_KGL; as rename the returns as payloadmass

Average Payload Mass by F9 v1.1

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

```
/* Task 4 */
/* Display average payload mass carried by booster version F9 v1.1 */
select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTABL;
```



- avg argument calculates the average number in the column of PAYLOAD_MASS_KGL; as rename the returns as payloadmass

First Successful Ground Landing Date

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

```
/* Task 5 */
/* List the date when the first succesful landing outcome in ground pad was acl
select min(DATE) from SPACEXTABL;
```

/* Task 5 */ /* List the date when the first succesfu...

Result set 1

1
2010-06-04

- min statement returns the minimum/earliest date from DATE column of the SPACEXTABL database

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
/* Task 6 */
/* List the names of the boosters which have success in drone ship a
select BOOSTER_VERSION from SPACEXTABL
where LANDING__OUTCOME='Success (drone ship)'
and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
```

select BOOSTER_VERSION...

Result set 1

BOOSTER_VERSION
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- Where statement filters the the Booster_Version whose Landing_Outcome is “Success”; and statement adds on a condition of Payload_Mass_Kg between 4000 and 6000 to be satisfied

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
/* Task 7 */
/* List the total number of successful and failure mission outcomes */
select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTABL GROUP BY MISSION_OUTCOME;
```

select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTABL G...

Result set 1

MISSIONOUTCOMES
1
99
1

- Count counts the total entries of MISSSION_OUTCOME, Group By MISSION_OUTCOME

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
/* Task 8 */
/* List the names of the booster_versions which have carried the maximum payload mass. Use a subque
SELECT DISTINCT Booster_Version, max(PAYLOAD_MASS__KG_) as Maximum_Payload_Mass FROM SPACEXTABL
Group By BOOSTER_VERSION Order By Maximum_Payload_Mass DESC;
```

SELECT DISTINCT Booster_Version, max(PAYLOAD_MASS__KG_

Result set 1

BOOSTER_VERSION	MAXIMUM_PAYLOAD_MASS
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600

Result set is truncated, only the first 97 rows have been loaded. Select "View al rows.

- Using the word DISTINCT in the query means that it will only show Unique values in the Booster_Version column from SPACEXTABL
- GROUP BY puts the list in order set to a certain condition
- DESC means its arranging the dataset into descending order

2015 Launch Records

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

```
/* Task 9 */
/* List the records which will display the month names, failure landing_outcomes in drone ship ,boos
SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTABL
where EXTRACT(YEAR FROM DATE)='2015';
```

SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_... Run time: 0.006 s

Result set 1

Find

1	MISSION_OUTCOME	BOOSTER_VERSION	LAUNCH_SITE
1	Success	F9 v1.1 B1012	CCAFS LC-40
2	Success	F9 v1.1 B1013	CCAFS LC-40
3	Success	F9 v1.1 B1014	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40
4	Success	F9 v1.1 B1016	CCAFS LC-40

Result set is truncated, only the first 7 rows have been loaded. Select "View all loaded data" on the right top of the result to view all loaded rows.

- The month function returns the month of the date
- The extract function nested in the where statement extract the year and set it to equal to 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

```
/* Task 10 */
/* Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in desc
SELECT COUNT(LANDING__OUTCOME) FROM SPACEXTABL
WHERE (LANDING__OUTCOME LIKE '%Success%')
AND (Date >'2010-06-04')
AND (Date <'2017-03-20');
```

- Function COUNT counts records in column WHERE filters data
  - LIKE ('Conditions')
  - AND ('Conditions')
  - AND ('Conditions')

Section 4

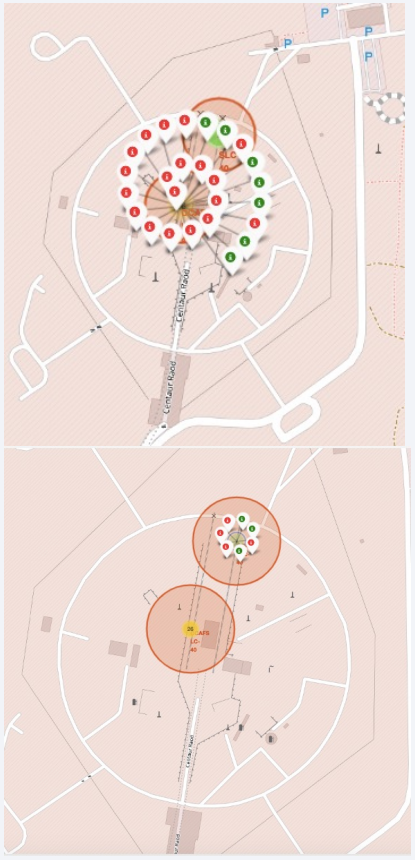
Launch Sites Proximities Analysis

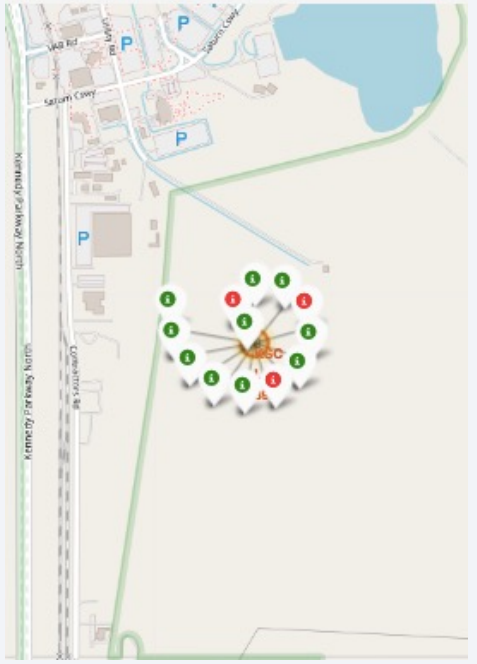
All launch sites' location markers on a global map

GitHub URL

- The Launch sites are located at U.S.A. east/west coasts, which is Cal and Florida

Color-labeled launch outcomes on the map (Florida)



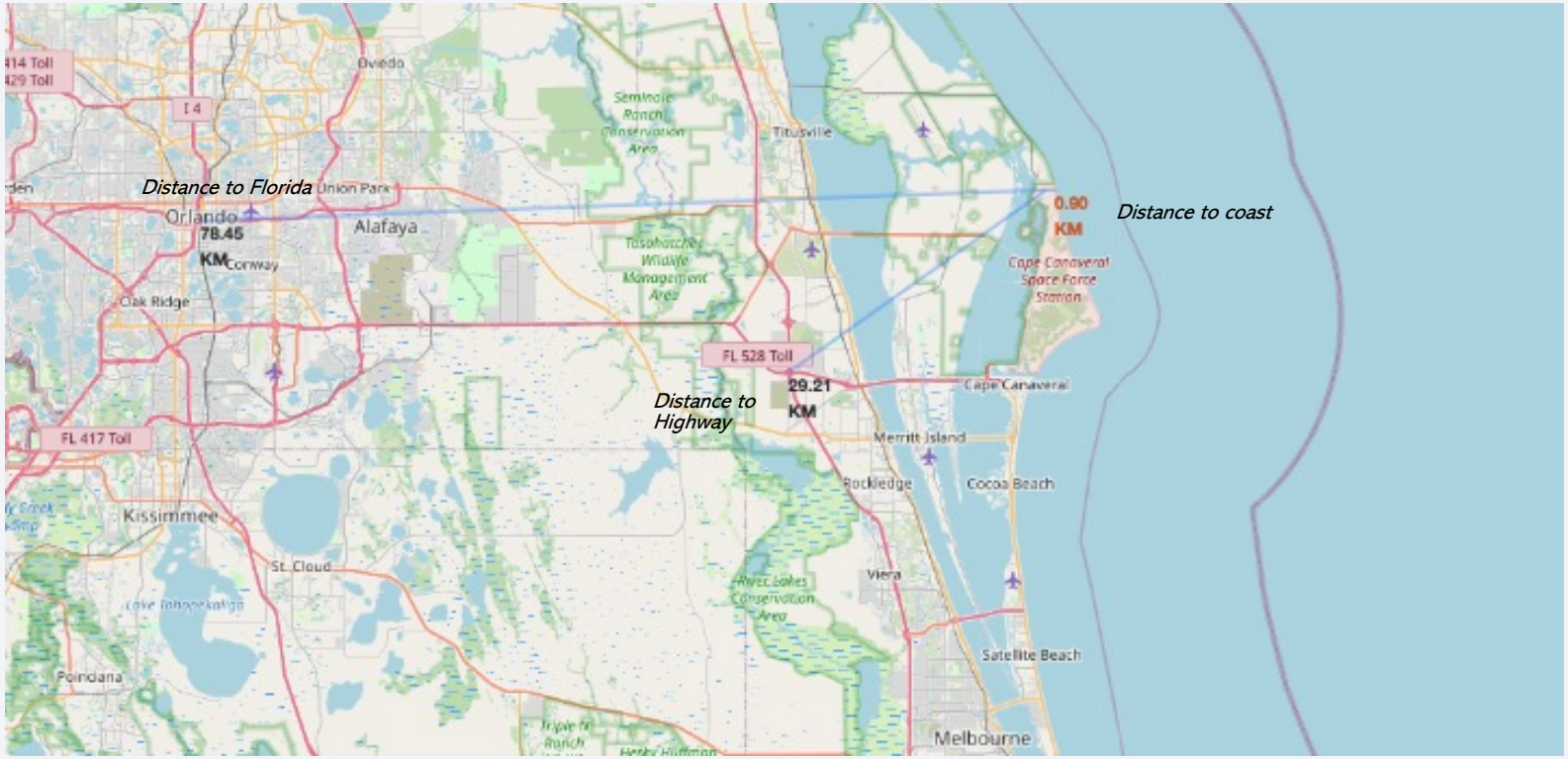


- Green – Success Launch
- Red – Failed Launch

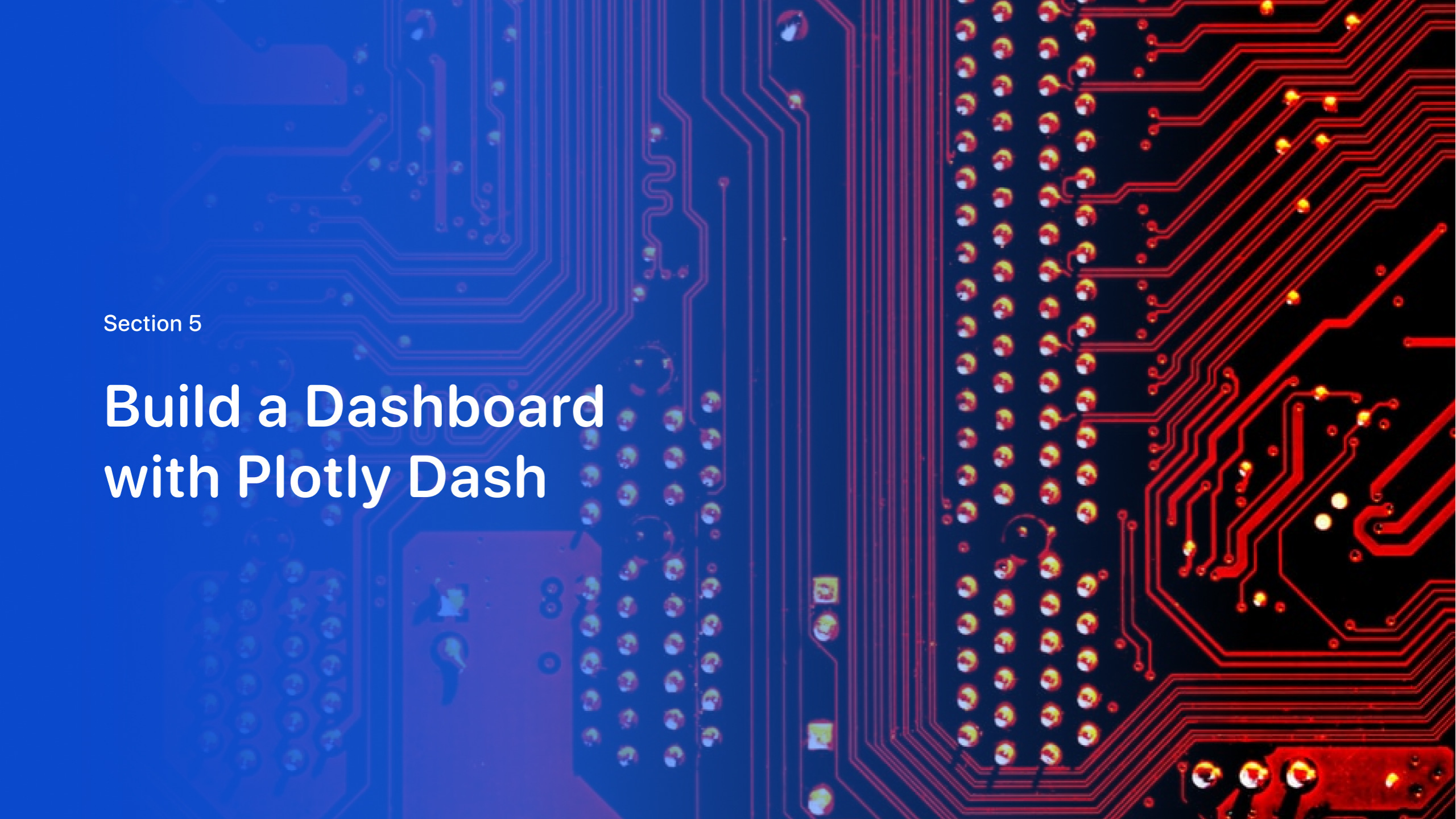
Color-labeled launch outcomes on the map (California)

- Green – Success Launch
- Red – Failed Launch

Distance to different objects



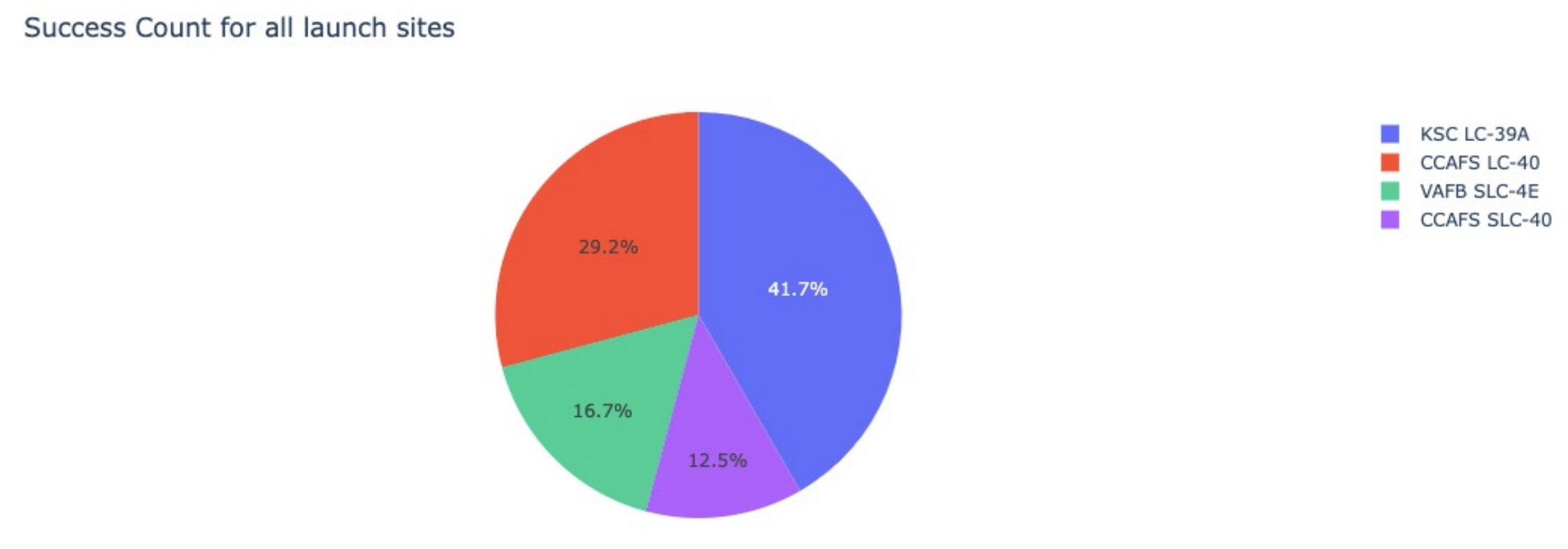
- Are launch sites in close proximity to railways? No
- Are launch sites sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance sway from cities? Yes



Section 5

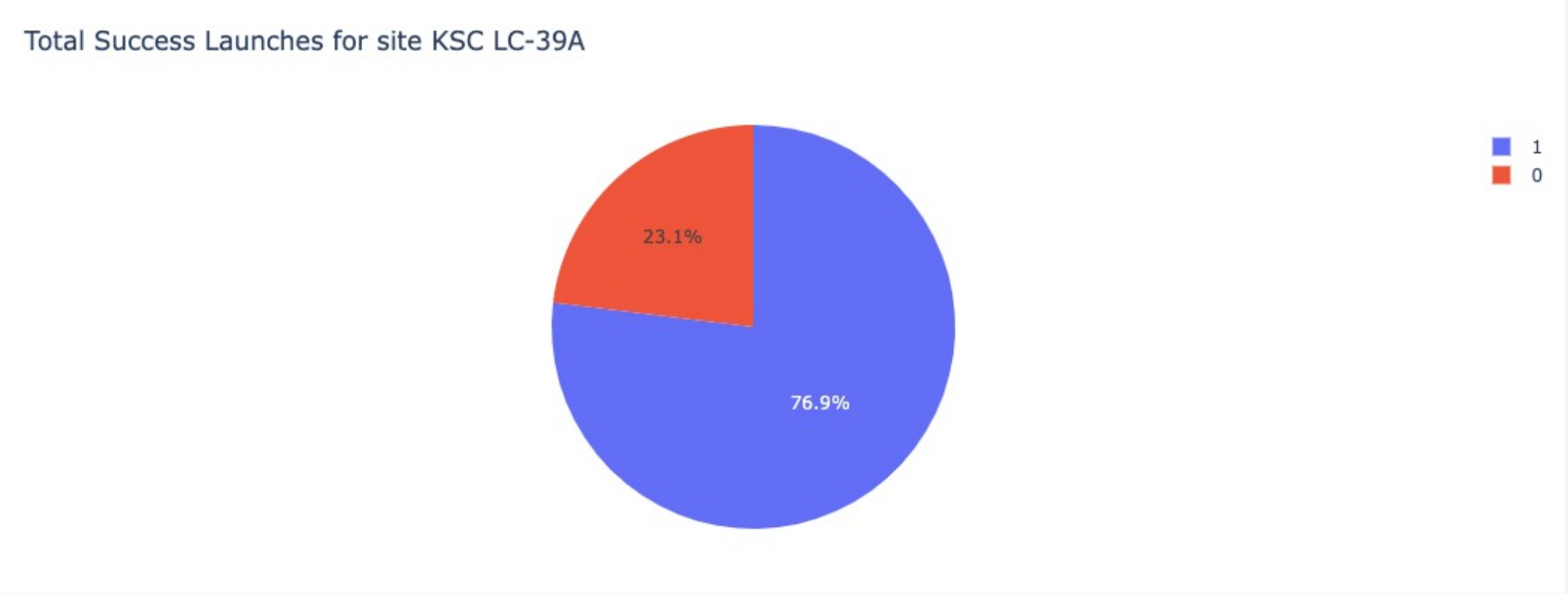
Build a Dashboard with Plotly Dash

Pie chart of launch success count for all sites



- Selected ”all sites” in the dropdown list
- 41.7% percentage of launch sites is KSC LC-39A, which has the most success launch sites
- 12.5% percentage of launch sites is CCAFS SLC-40, which has the lowest success launch sites

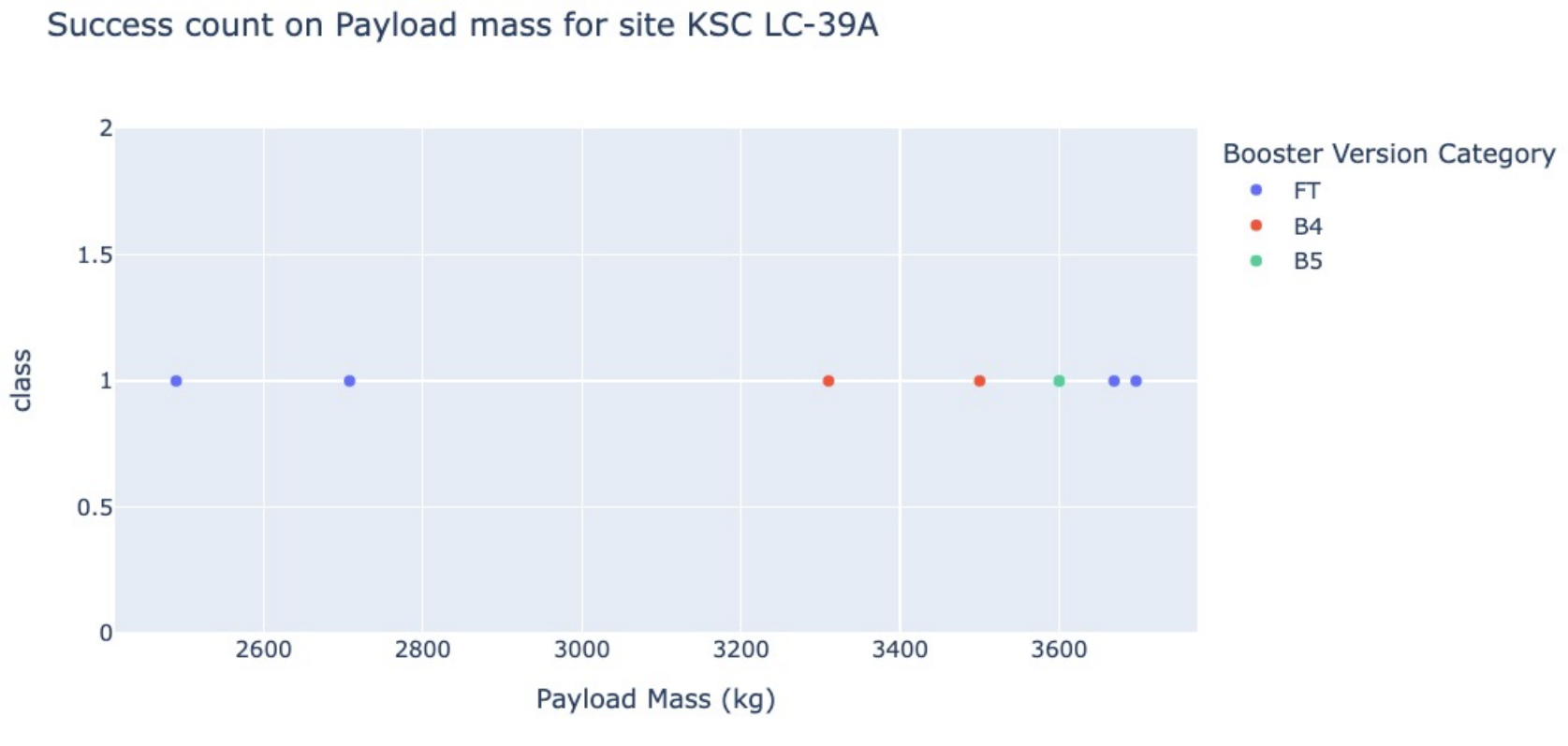
Pie chart of launch success count for KSC LC-39A



- Previously we learnt that KSC LC-39A has the most launches
- Blue means success so 76.9% of the launches of KSC LC-39A is success
- Red means fail so 23.1% of the launches of KSC LC-39A failed

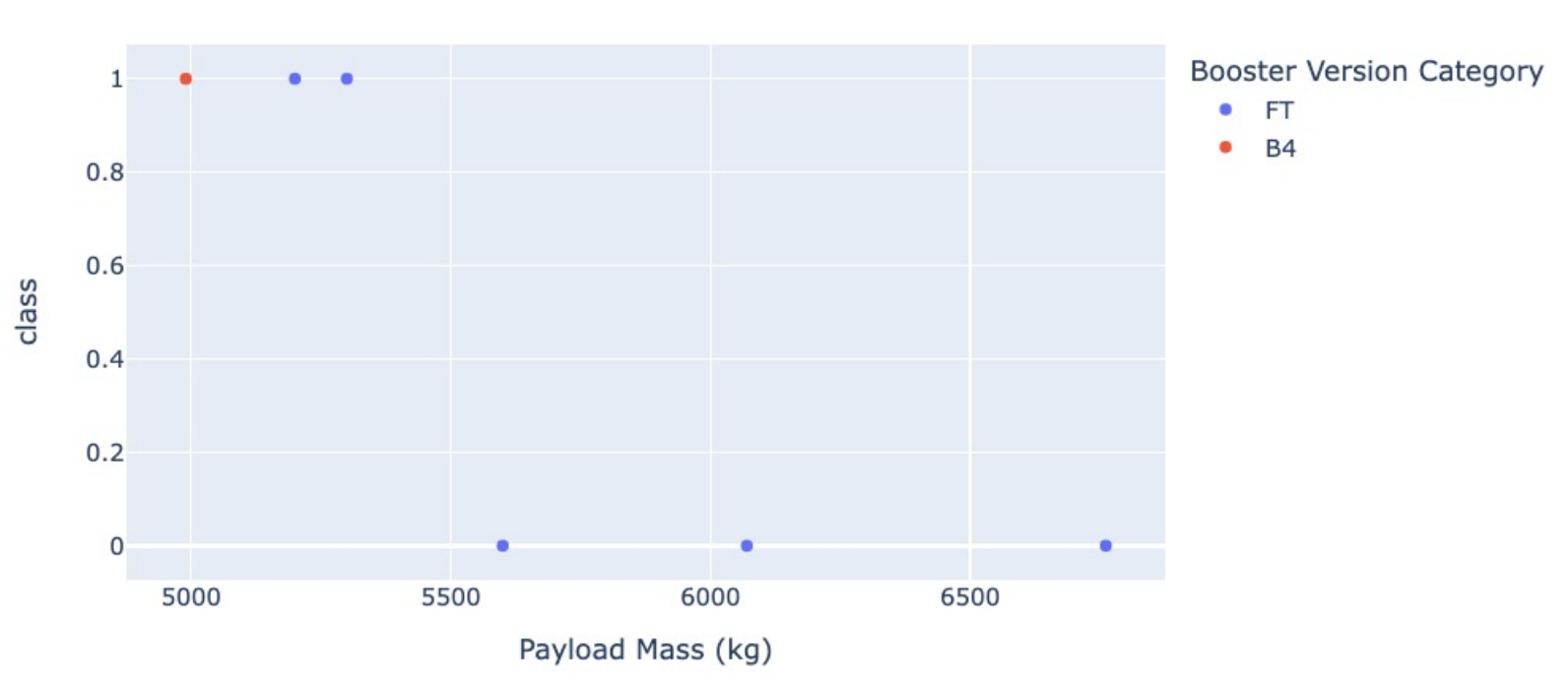
Scatter plot of Payload Mass (Kg) vs. Launch Outcomes

- Payload range (Kg) 0~4000



Scatter plot of Payload Mass (Kg) vs. Launch Outcomes

- Payload range (Kg) 4000~10000



- Conclusion:

  ✓ When payload mass is lower (0~4000), the success rate is higher (more 1s') compares with that of 4000~10000 Kg

Section 6

Predictive Analysis (Classification)

Classification Accuracy

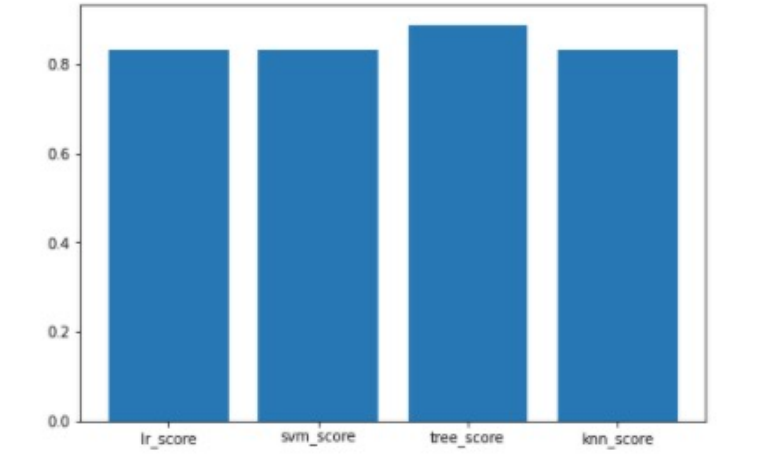
URL

- Visualize the built model accuracy for all built classification models, in a bar chart

```
scores = [lr_score,svm_score,tree_score,knn_score]
print(scores)
print(scores.index(max(scores)))
```

```
[0.8333333333333334, 0.8333333333333334, 0.8888888888888888, 0.8333333333333334]
2
```

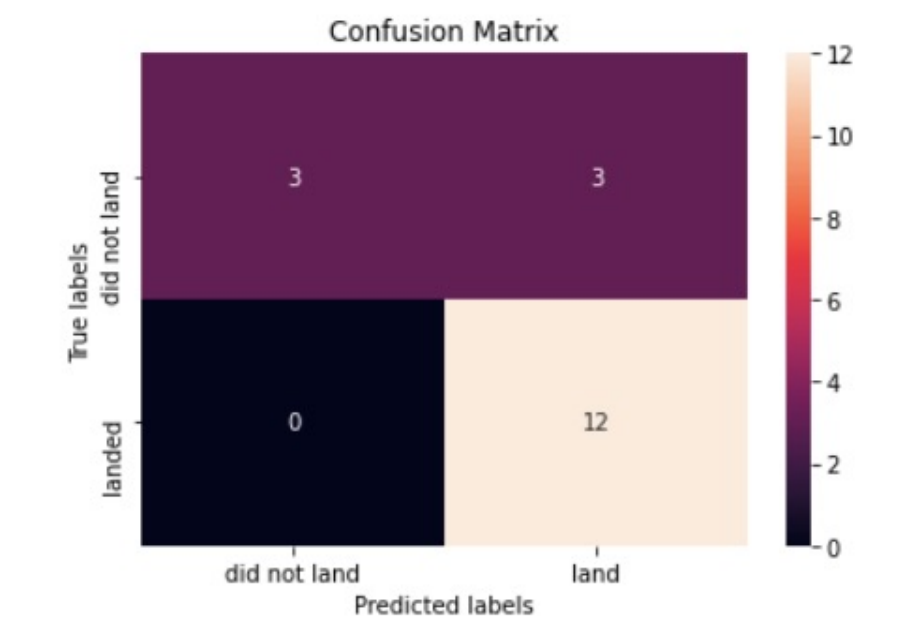
```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
name = ['lr_score', 'svm_score', 'tree_score', 'knn_score']
scores = [lr_score,svm_score,tree_score,knn_score]
ax.bar(name,scores)
plt.show()
```



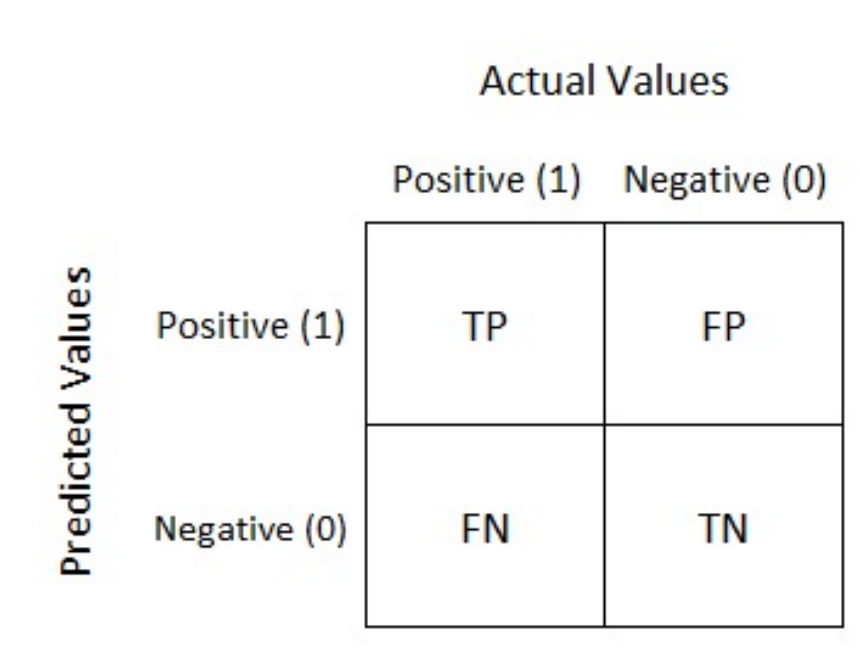
- From the bar char and the table above, we can see that the tree model gets the highest accuracy score, which is 0.8888….
- Which means that we achieved 88.89% accuracy with the tree model

Confusion Matrix

- Confusion matrix of the best performing model, which is the Tree model in this case



- Confusion matrix definition
- Explanation: The biggest problem is with false positive (FP)



Conclusions

- GEO, HEO, SSO, ES-L1 has the top 4 success rate.
- Tree Classification Model is the best for Machine Learning & Predictive Analysis
- KSC LC-39A had the top 1 successful launches among all sites
- Low weighted payloads performs better
- The success rate of SpaceX launches is directly proportional to the period in years it will take them to perfect the launches.

Thank you!