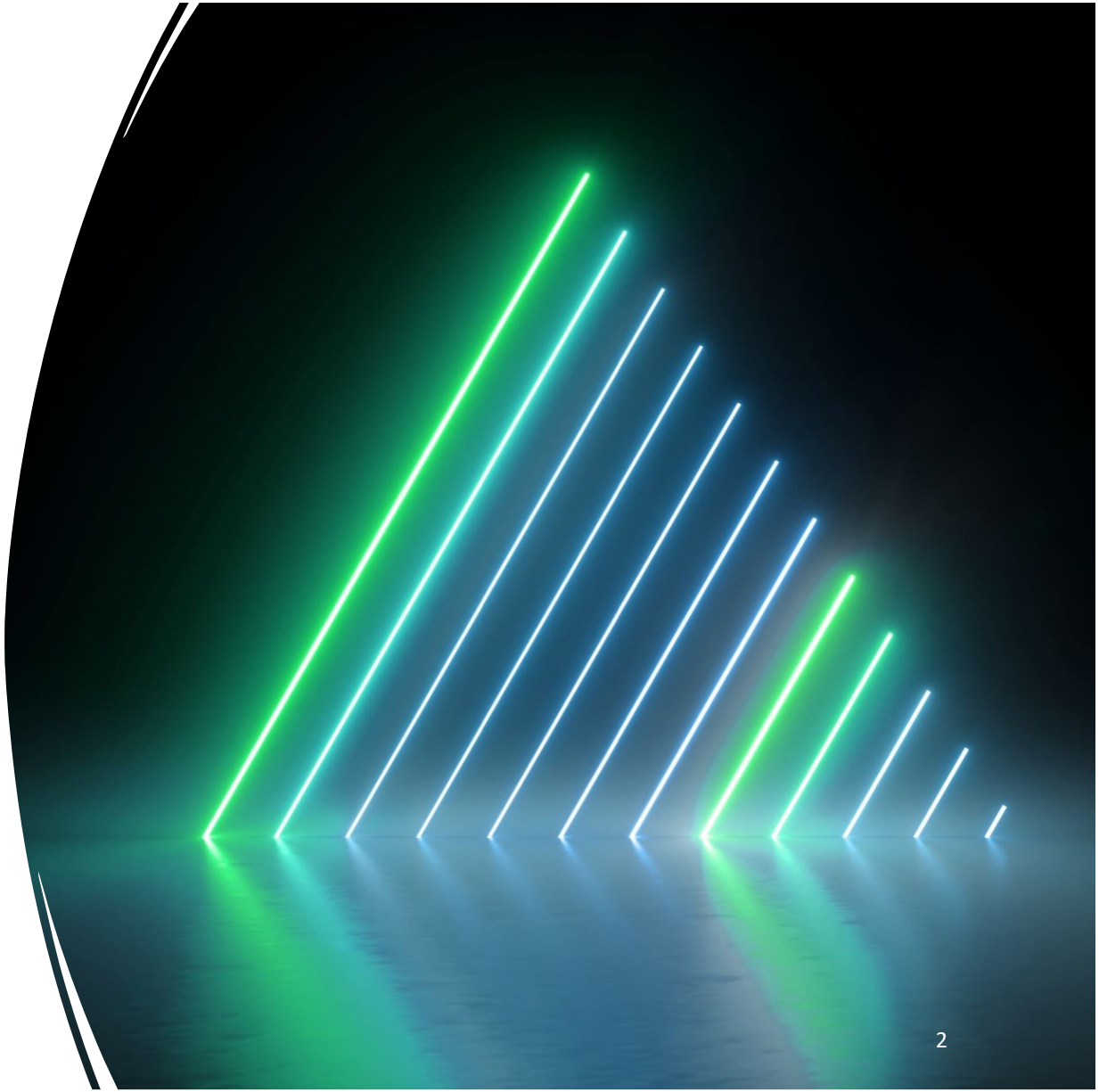
Winning Space Race with Data Science

Rebecca Dudek
15/06/2023

Outline

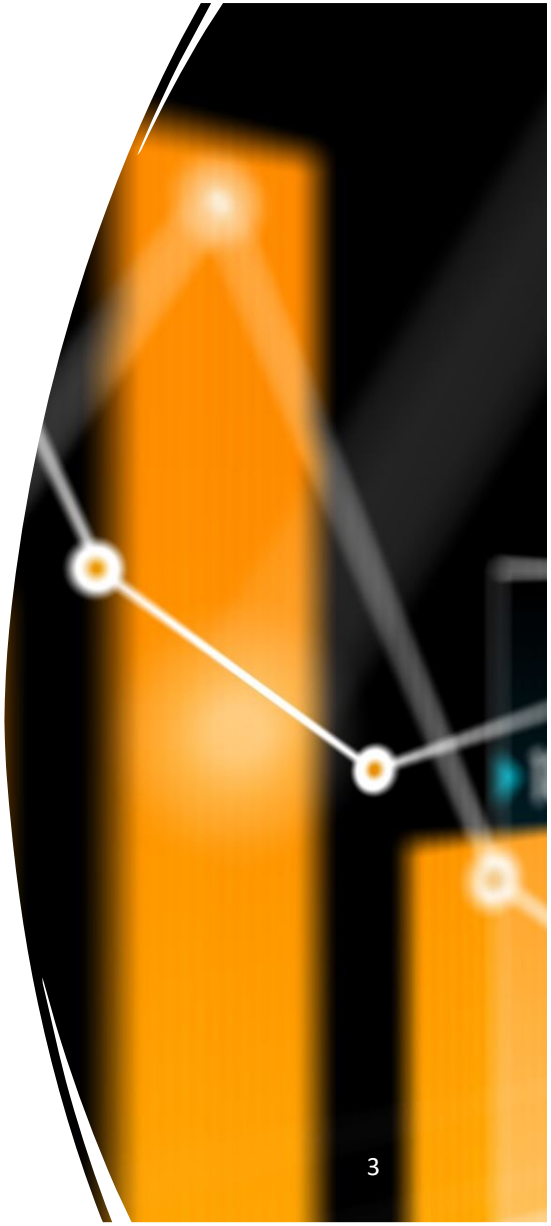
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix



Executive Summary

- The following methodologies were used to analyse the data:
  - Data Collection using web scraping and the SpaceX API
  - Exploratory Data Analysis (EDA), wrangling and data viz
  - Machine Learning Prediction

- Summary of all results
  - We then look through the results of our analysis and machine learning predictions.



Introduction

- **SpaceX** advertises **rocket launche**s on their website with a cost of **62 million dollars**.
- Much of the savings is because SpaceX can reuse the first stage of the rocket and therefore we want to predict if the first stage will land, as this will provide most of the savings.
- In this project we are collecting, wrangling, analyzing and visualizing the rocket data from the SpaceX website and creating models to determine probability of the first stage being reused.

Section 1

Methodology

Methodology

- **Data Collecting**: First we use the SpaceX API and webscraping using the BeautifulSoup class to download and clean the SpaceX data form the website.
- **Data Wrangling**: Then we use the pandas class to create a Landing_Class which has value 0 for unsuccessful landings and 1 for successful landings.
- **Explore:** Next we used SQL to explore the data and generate valuable insights.
- **Visualise:** Folium and Plotly were then used to explore the geographical data and also to visualize that data in a dashboard.
- **Build Models**: Lastly we tested four different types of models and assessed their accuracy for prediction of the Falcon9 landing.

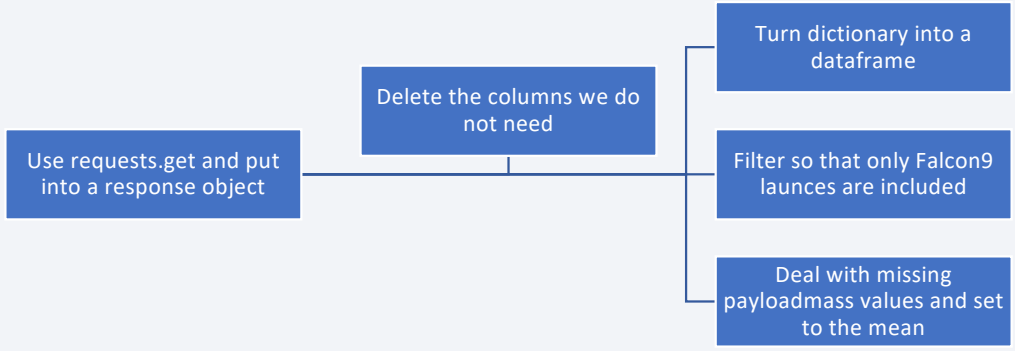
Data Collection

- The data sets were collected using two methods.
- The first method used the SpaceX API available publicly on their website using requests.get to retrieve the data from past launches.
- The other method was completed using webscraping with the Beautiful Soup class, the data was downloaded from the SpaceX Wikipedia page and then parsed into a pandas dataframe for cleaning.

Data Collection – SpaceX API

Using the SpaceX API provided on their website, use the requests module to download the data from past launches

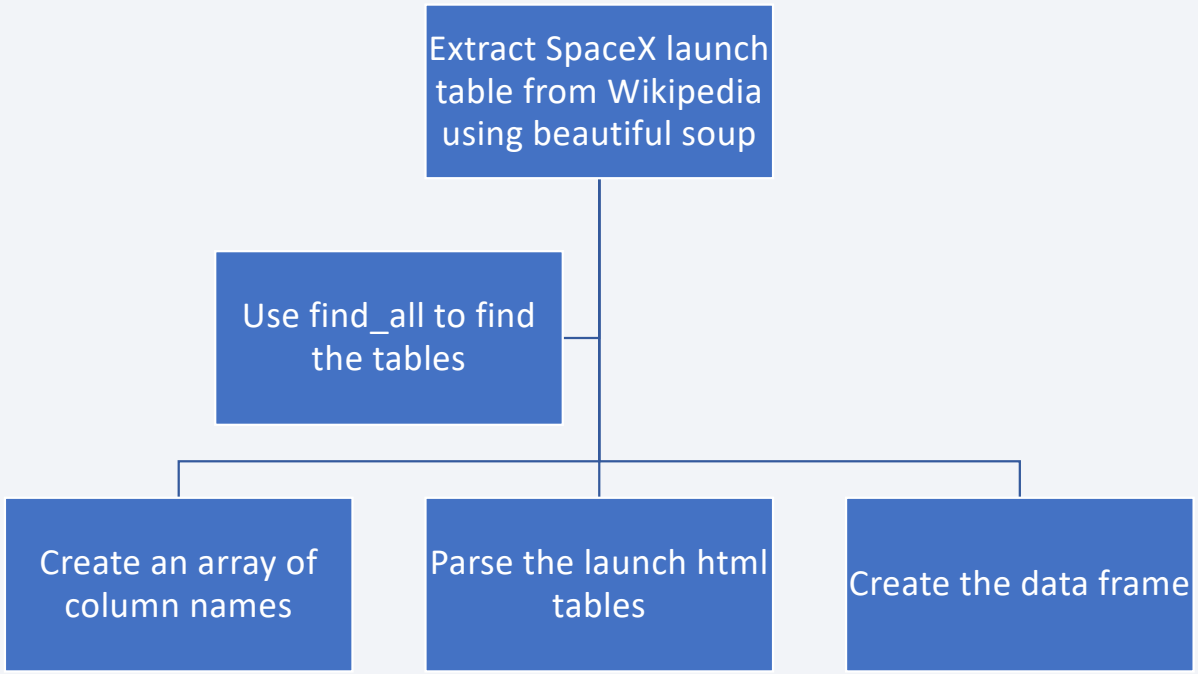
https://github.com/BecDudek/IBM-Data-Science-Capstone/blob/e57dee6a8f8e830eb89edfacbab2178b8dbbf99a/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Aim: To extract the Falcon9 launch records form a html Wikipedia table using Beautiful Soup

https://github.com/BecDudek/IBM-Data-Science-Capstone/blob/e57dee6a8f8e830eb89edfacbab2178b8dbbf99a/jupyter-labs-webscraping.ipynb



Data Wrangling

Performed Exploratory Data Analysis (EPA) to find some patterns in the data and determine what would be the labels for training supervised models.

Converted outcomes to training labels 0 (unsuccessful) and 1 (successful)

We calculated the percentage of missing values for each attribute

We used df.types to check the types of each attribute

We then created a list called landing_class which had a 1 for successful landings and a 0 for unsuccessful landings

https://github.com/BecDudek/IBM-Data-Science-Capstone/blob/e57dee6a8f8e830eb89edfacbab2178b8dbbf99a/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

We plotted payload mass and launch site against Flight Number, launch site against payload mass, orbit against flight number, and lastly payload against orbit.

We explored the above relationships using scatterplots and bar charts.

https://github.com/BecDudek/IBM-Data-Science-Capstone/blob/e57dee6a8f8e830eb89edfacbab2178b8dbbf99a/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

Exploratory Data Analysis was performed using the following SQL queries:

- %sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
- %sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' limit 5
- %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer == "NASA (CRS)”
- %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version == "F9 v1.1”
- %sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome == "Success (ground pad)"
- %sql SELECT * FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND Landing_Outcome == "Success (drone ship)"
- %sql SELECT Mission_Outcome, COUNT(*) FROM SPACEXTBL GROUP BY Mission_Outcome
- %sql SELECT Booster_Version, MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL GROUP BY Booster_Version
- %sql SELECT substr(Date, 4, 2) AS Month_Names, Booster_Version, Launch_Site FROM SPACEXTBL WHERE substr(Date,7,4)='2015' AND Landing_Outcome == "Failure (drone ship)"
- %sql SELECT * FROM SPACEXTBL WHERE Landing_Outcome == "Success" AND substr(Date,7,4) == '2010' ORDER BY Date DESC

https://github.com/BecDudek/IBM-Data-Science-Capstone/blob/e57dee6a8f8e830eb89edfacbab2178b8dbbf99a/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Using the Folium package we added circles and markers around the SpaceX launch sites.
- Then we added a column called ‘marker_color’ in our dataframe to be set to green for successful launches and red for unsuccessful launches, to easily visualize the data.
- Next we added lines from the nearest coastline using MousePosition to the launch sites

https://github.com/BecDudek/IBM-Data-Science-Capstone/blob/e57dee6a8f8e830eb89edfacbab2178b8dbbf99a/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

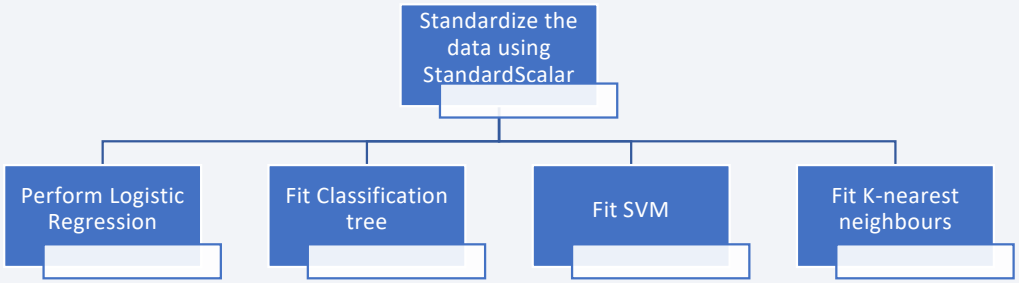
Build a Dashboard with Plotly Dash

- Plotly was used to create visuals such as pie charts to easily determine probability of successful launches of Falcon9.

https://github.com/BecDudek/IBM-Data-Science-Capstone/blob/e57dee6a8f8e830eb89edfacbab2178b8dbbf99a/spacex_dash_app.py

Predictive Analysis (Classification)

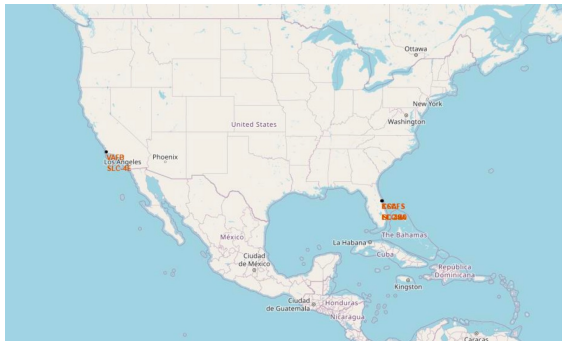
- Objective: Find the best hyperparameter for SVM, Classification Trees and Logistic Regression
- Firstly we normalized the data set using 'X = preprocessing.StandardScaler().fit(X).transform(X)' from the sklearn library
- Then we performed Logistic Regression, KNN, Classification trees and SVM

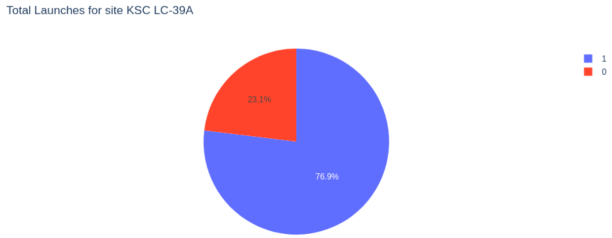


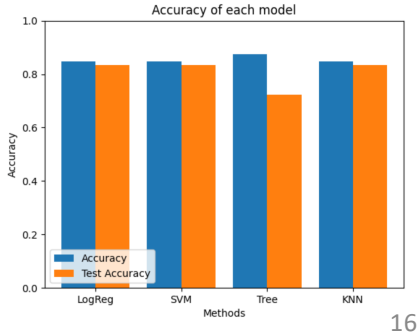
https://github.com/BecDudek/IBM-Data-Science-Capstone/blob/cf32e5f5cbdfb24827a830e433d7c064c0fcc5ae/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

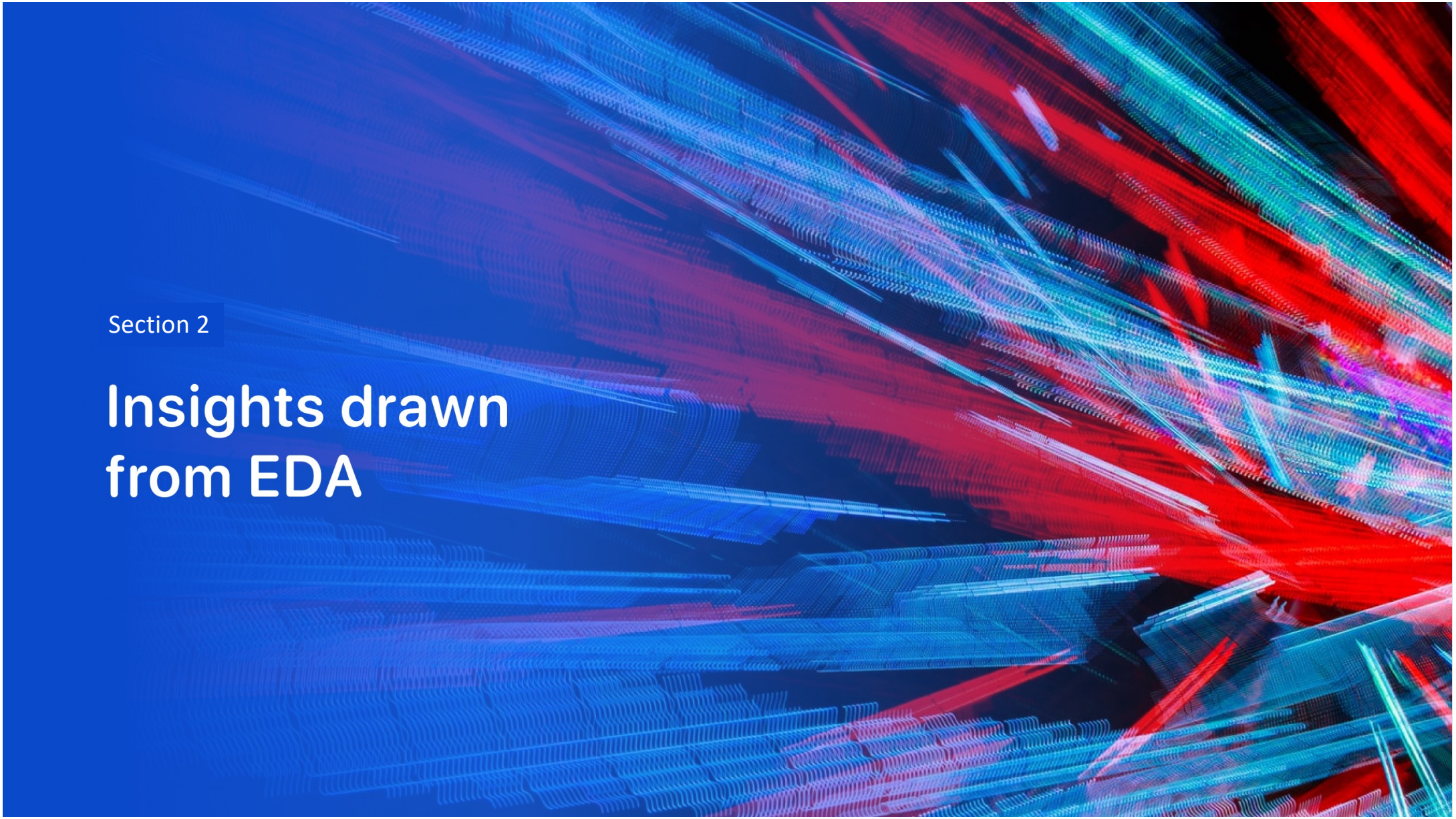
Results

- Using Folium we discovered that all of the launches were in coastal areas.
- Using Plotly we discovered that 79% of launches were successful.
- Predictive analysis results – we determined that the best model was the classification tree.





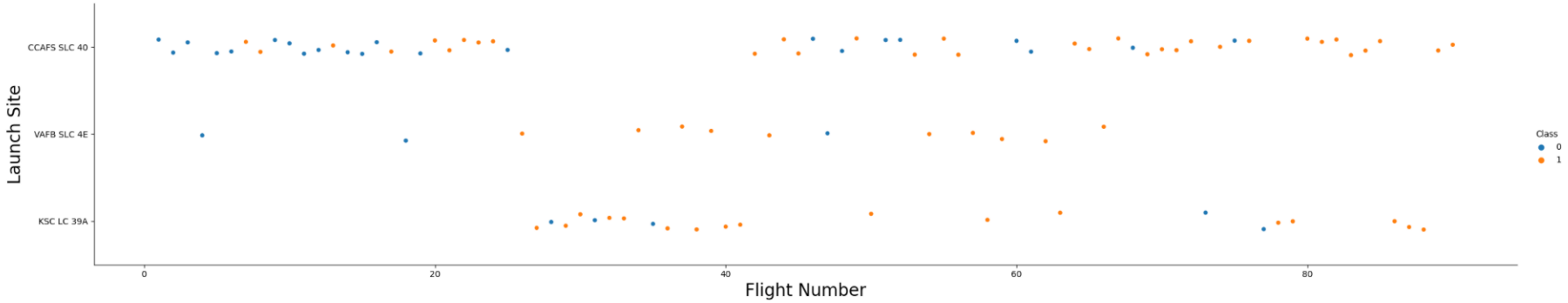



Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- We see a scatter plot of Flight Number vs. Launch Site for three different launch sites.
- We can see that the earlier launches all took place at the CCAFS SLC 40 launch site.

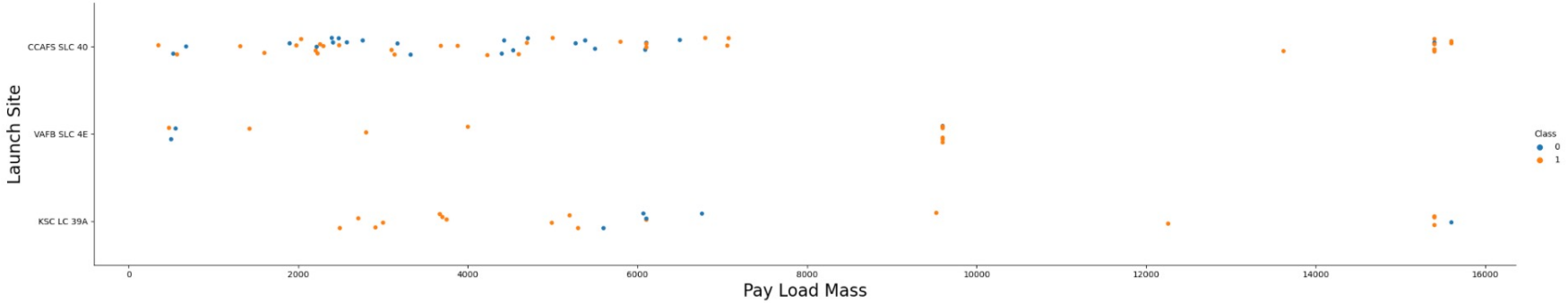
```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Payload vs. Launch Site

- We can see a scatter plot of Payload Mass vs. Launch Site
- We can see that most of the Pay Load Masses were below 7000kg

```
### TASK 2: Visualize the relationship between Payload and Launch Site
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay Load Mass",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

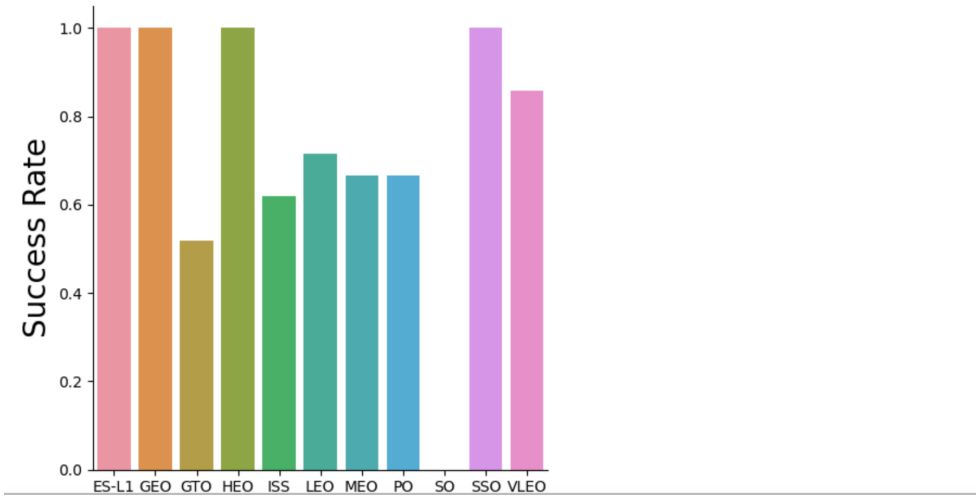


Success Rate vs. Orbit Type

We see a bar chart for the success rate of each orbit type

- The bar chart shows us that four of the launch sites all have a 100% success rate
- VLEO has the next highest success rate before the rest of the launch sites which perform much poorer.

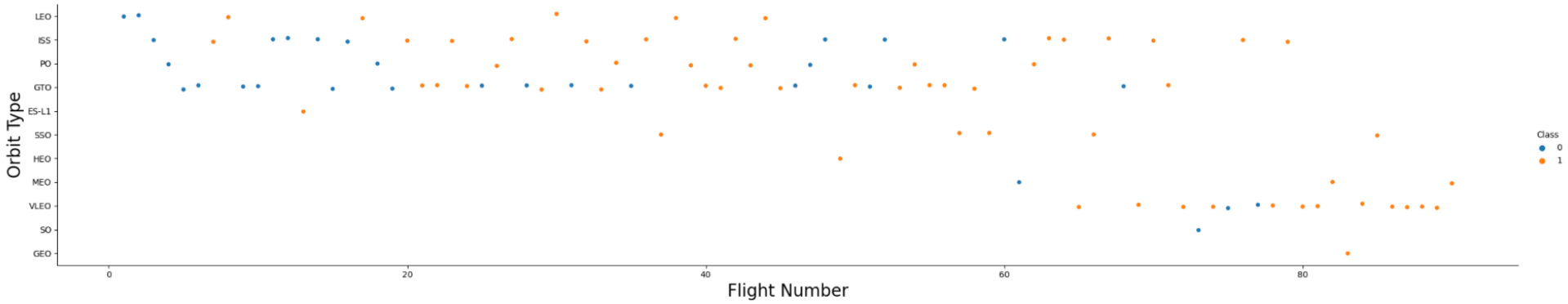
```
sns.catplot(x= 'Orbit', y = 'Class', data = df.groupby('Orbit')['Class'].mean().reset_index(), kind = 'bar')
plt.xlabel('Orbit Type',fontsize=20)
plt.ylabel('Success Rate',fontsize=20)
plt.show()
```



Flight Number vs. Orbit Type

- We see a scatter point of Flight number vs. Orbit type
- We see that earlier flights have orbits of LEO, ISS, PPO and GTO whereas later flights have mainly VLEO orbits

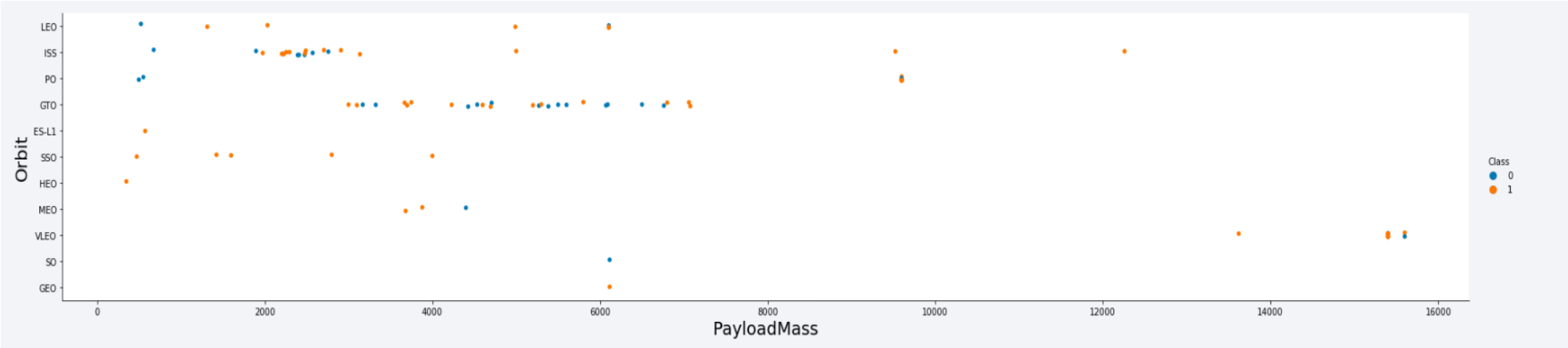
```
### TASK  4: Visualize the relationship between FlightNumber and Orbit type
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit Type",fontsize=20)
plt.show()
```



Payload vs. Orbit Type

We see a scatter point of payload mass vs. orbit type

- Again we see that most payload masses are below 7000.
- GTO orbits have a payload mass of between 3000 and 7000.

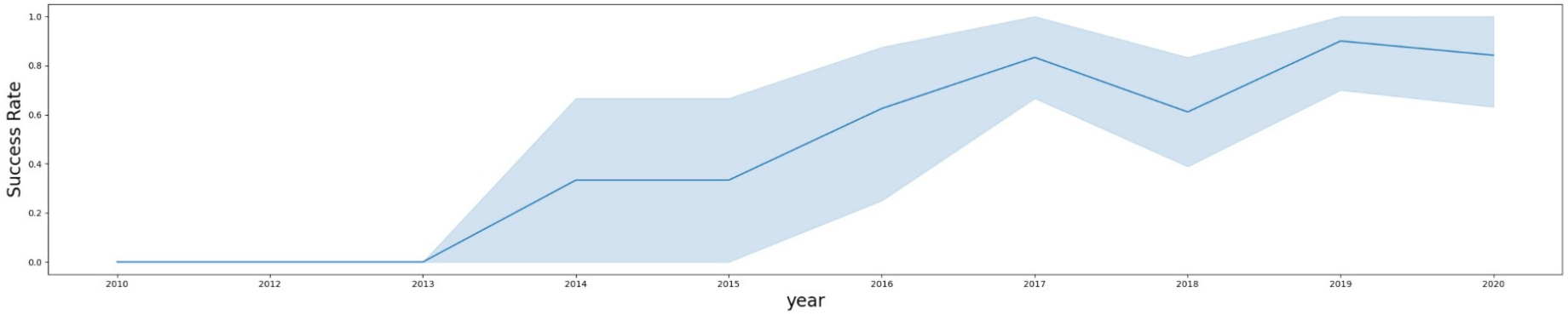


Launch Success Yearly Trend

We see a line chart of yearly average success rate

- We can see that the success rate is directly proportional to the year where the success rate improves over time.

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(data=df, x="Date", y="Class")
plt.xlabel("year",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



All Launch Site Names

The unique launch site names were found using the query below

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

The 5 records where launch sites begin with `CCA' are below.

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

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' limit 5
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	N
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	N
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

The total payload carried by boosters from NASA

Task 3

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

```
[28]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer == "NASA (CRS)"
```

```
 * sqlite:///my_data1.db
Done.
```

[28]:

SUM(PAYLOAD_MASS__KG_)
45596.0

Task 4

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version == "F9 v1.1"
```

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

The date of the first successful landing outcome on ground pad

Task 5

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

Hint:Use min function

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome == "Success (ground pad)"
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(Date)
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

List of 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 and have payload mass greater than 4000 but less than 6000

```
%sql SELECT * FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND Landing_Outcome == "Success (drone ship)
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Out
05/06/2016	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696.0	GTO	SKY Perfect JSAT Group	Success	Success (
14/08/2016	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600.0	GTO	SKY Perfect JSAT Group	Success	Success (
30/03/2017	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300.0	GTO	SES	Success	Success (
10/11/2017	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200.0	GTO	SES EchoStar	Success	Success (

Total Number of Successful and Failure Mission Outcomes

The total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT Mission_Outcome, COUNT(*) FROM SPACEXTBL GROUP BY Mission_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	COUNT(*)
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

List of 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 subquery

```
%sql SELECT Booster_Version, MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL GROUP BY Booster_Version
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	MAX(PAYLOAD_MASS__KG_)
None	None
F9 B4 B1039.2	2647.0
F9 B4 B1040.2	5384.0
F9 B4 B1041.2	9600.0
F9 B4 B1043.2	6460.0
F9 B4 B1039.1	3310.0
F9 B4 B1040.1	4990.0
F9 B4 B1041.1	9600.0
F9 B4 B1042.1	3500.0
F9 B4 B1043.1	5000.0
F9 B4 B1044	6092.0
F9 B4 B1045.1	362.0

2015 Launch Records

List of 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 ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%sql SELECT substr(Date, 4, 2) AS Month_Names, Booster_Version, Launch_Site FROM SPACEXTBL WHERE substr(Date,7,4)='2015
```

```
 * sqlite:///my_data1.db
Done.
```

Month_Names	Booster_Version	Launch_Site
10	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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 04-06-2010 and 20-03-2017 in descending order.

```
%sql SELECT * FROM SPACEXTBL WHERE Landing_Outcome == "Success" AND substr(Date,7,4) LIKE '201%'
```

```
 * sqlite:///my_data1.db
Done.
```

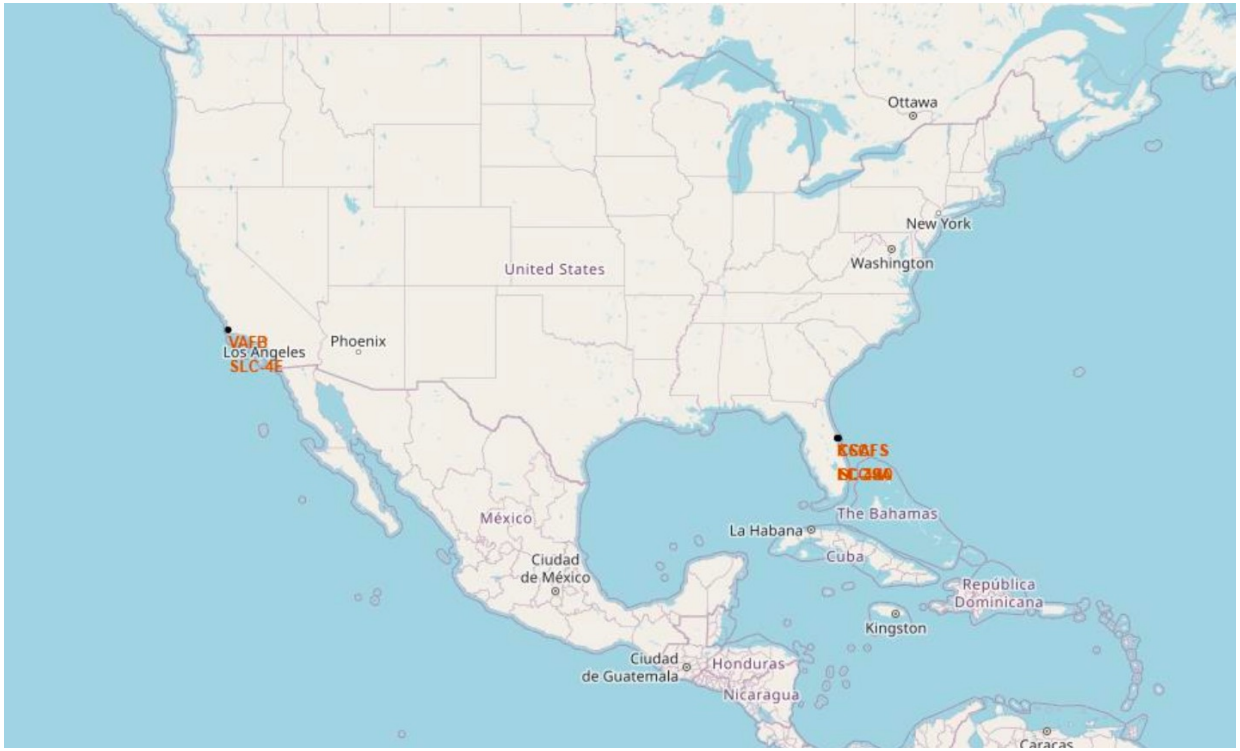
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	La
22/07/2018	5:50:00	F9 B5B1047.1	CCAFS SLC-40	Telstar 19V	7075.0	GTO	Telesat	Success	
25/07/2018	11:39:00	F9 B5B1048.1	VAFB SLC-4E	Iridium NEXT-7	9600.0	Polar LEO	Iridium Communications	Success	
08/07/2018	5:18:00	F9 B5 B1046.2	CCAFS SLC-40	Merah Putih	5800.0	GTO	Telkom Indonesia	Success	
09/10/2018	4:45:00	F9 B5B1049.1	CCAFS SLC-40	Telstar 18V / Apstar-5C	7060.0	GTO	Telesat	Success	
10/08/2018	2:22:00	F9 B5 B1048.2	VAFB SLC-4E	SAOCOM 1A	3000.0	SSO	CONAE	Success	
15/11/2018	20:46:00	F9 B5 B1047.2	KSC LC-39A	Es hail 2	5300.0	GTO	Es hailSat	Success	
12/03/2018	18:34:05	F9 B5 B1046.3	VAFB SLC-4E	SSO-A	4000.0	SSO	Spaceflight Industries	Success	

Section 3

Launch Sites Proximities Analysis

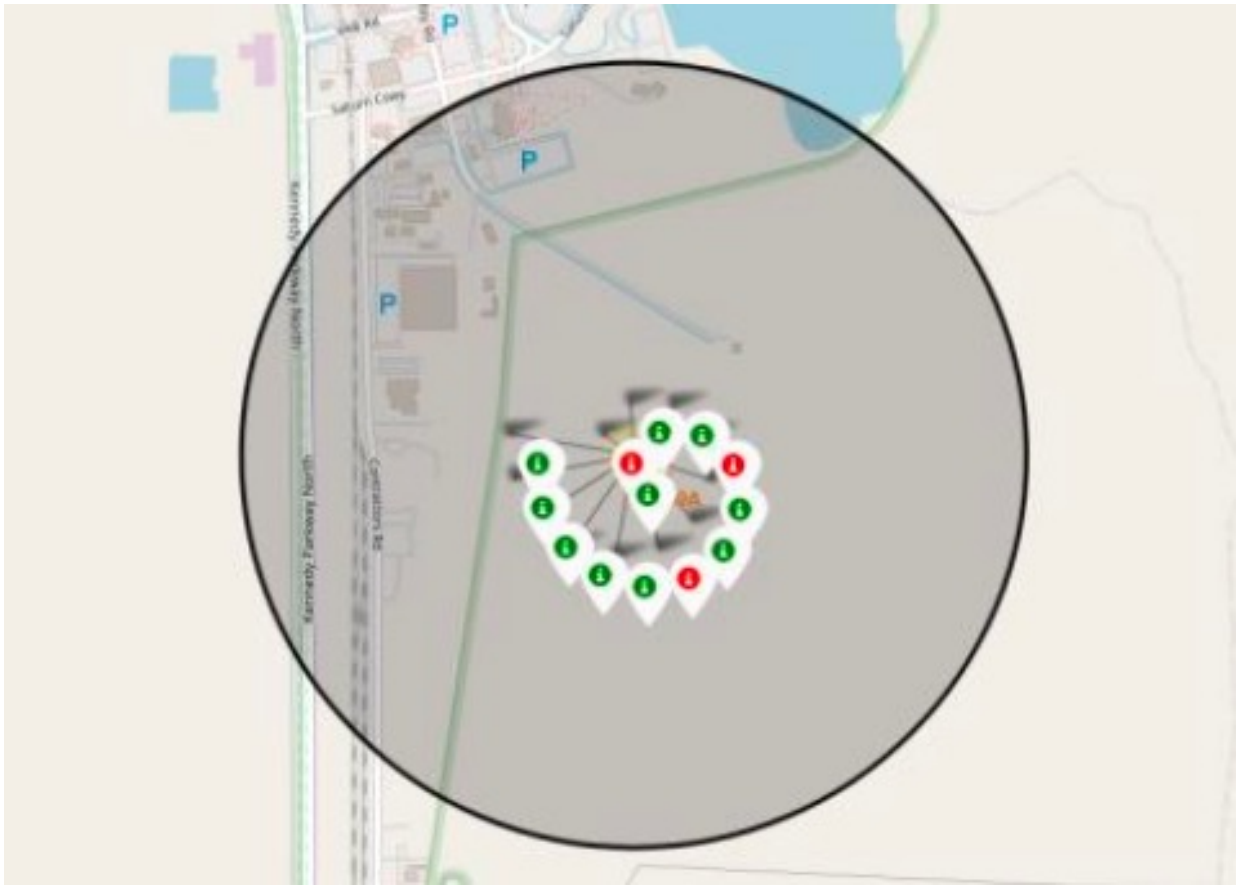
Launch sites generated using Folium

- We see that the launch sites are in two areas.
- The launch sites are on either the east or west coast.



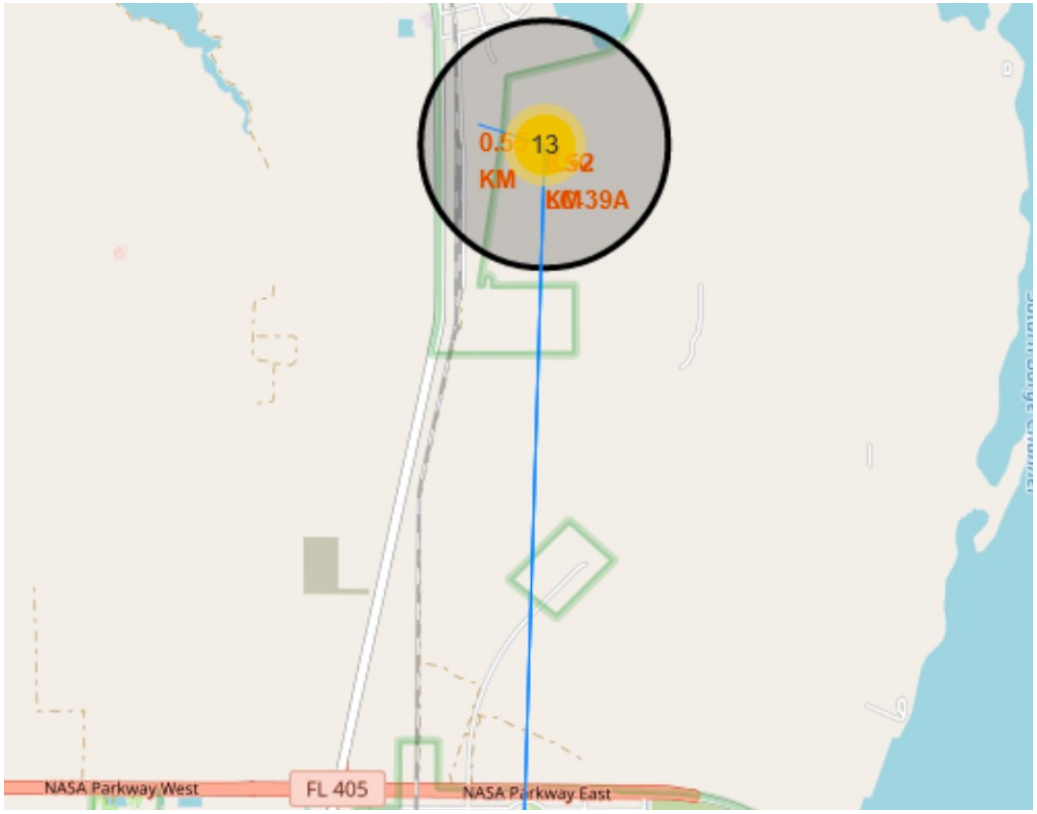
Successful Launch site icons generated using Folium

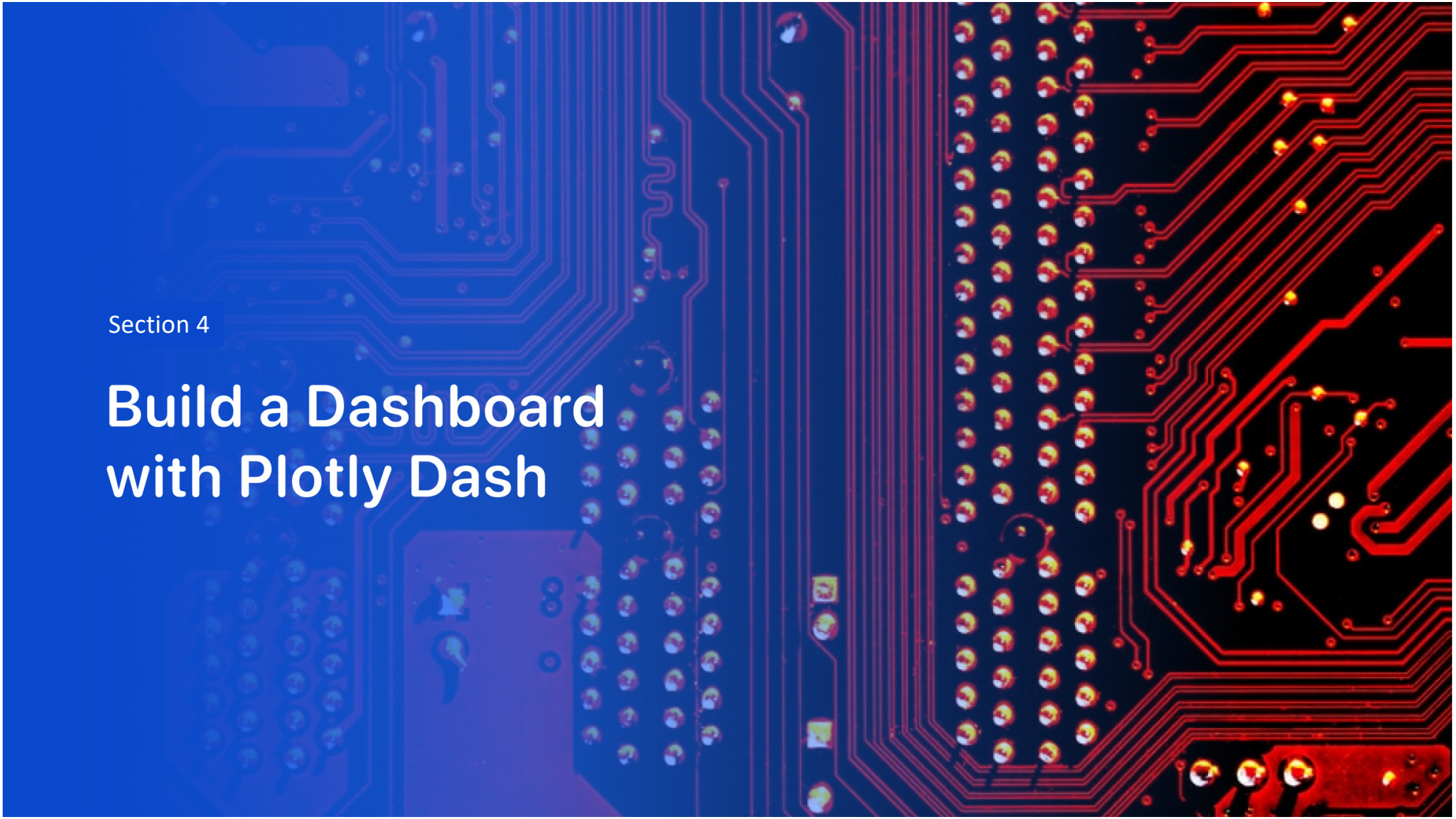
- We then added icons on to our map.
- A green icon denotes a successful landing and a red icon denotes an unsuccessful landing.



Folium map denoting distance to specific places

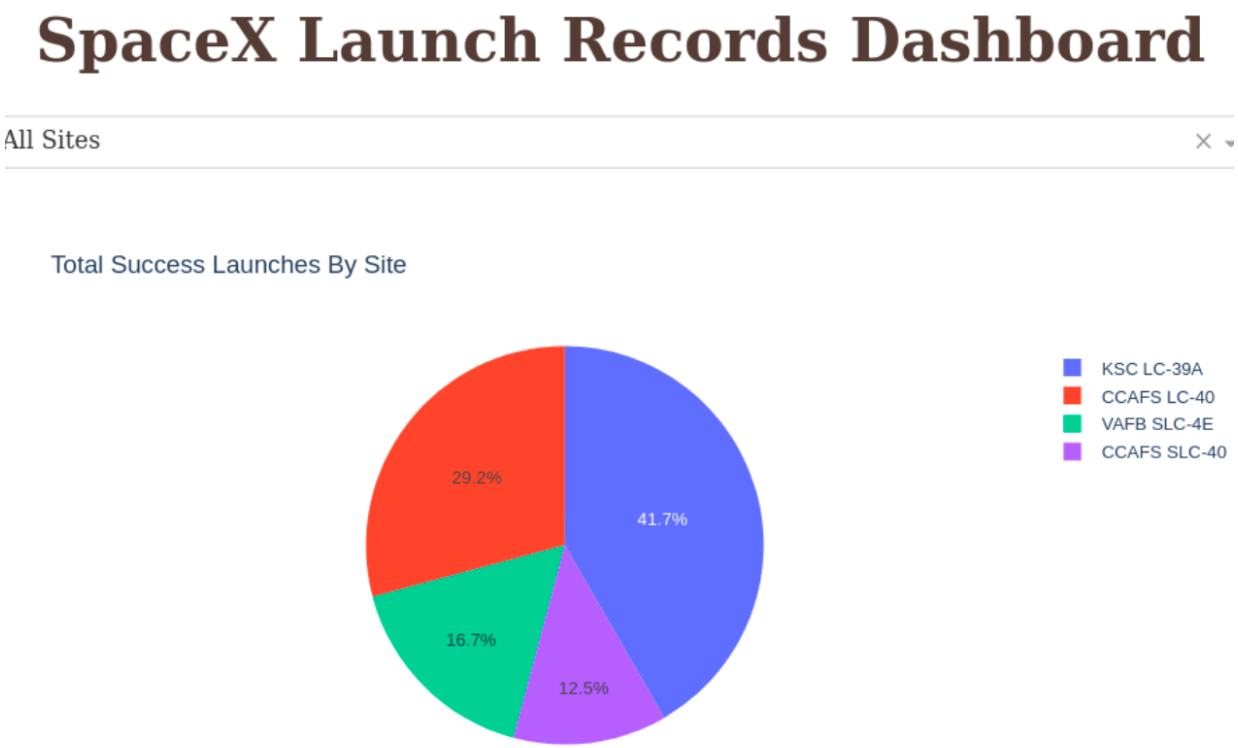
- We see KSC LC-39A is close to certain amenities and away from built up areas.




Section 4
Build a Dashboard with Plotly Dash

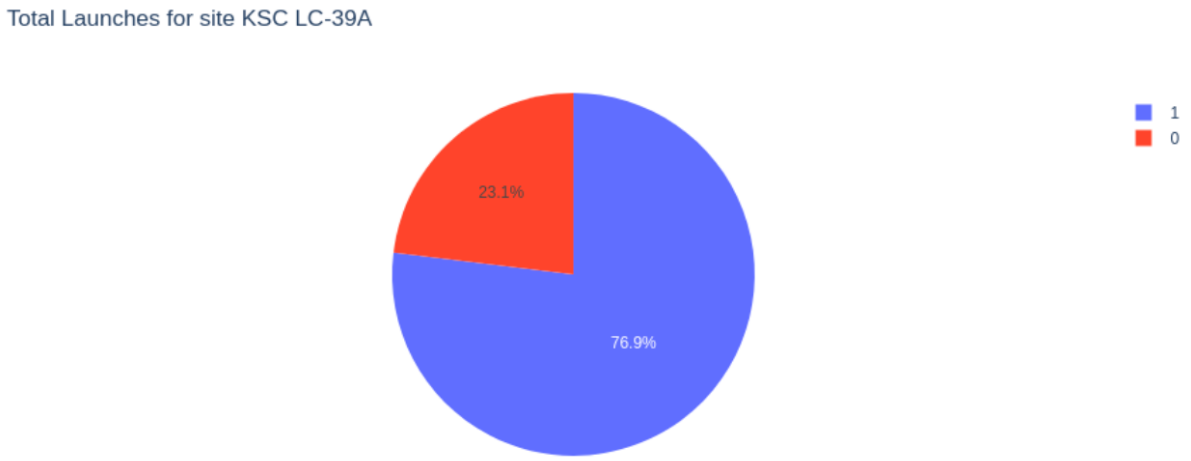
Space X Launch Records Dash using Plotly

- We have a pie chart generated using Plotly which shows the total successful launches by landing site.
- We see that KSC LC-39A is the most successful launch site whereas CCAFS SLC-40 is the least successful launch site.



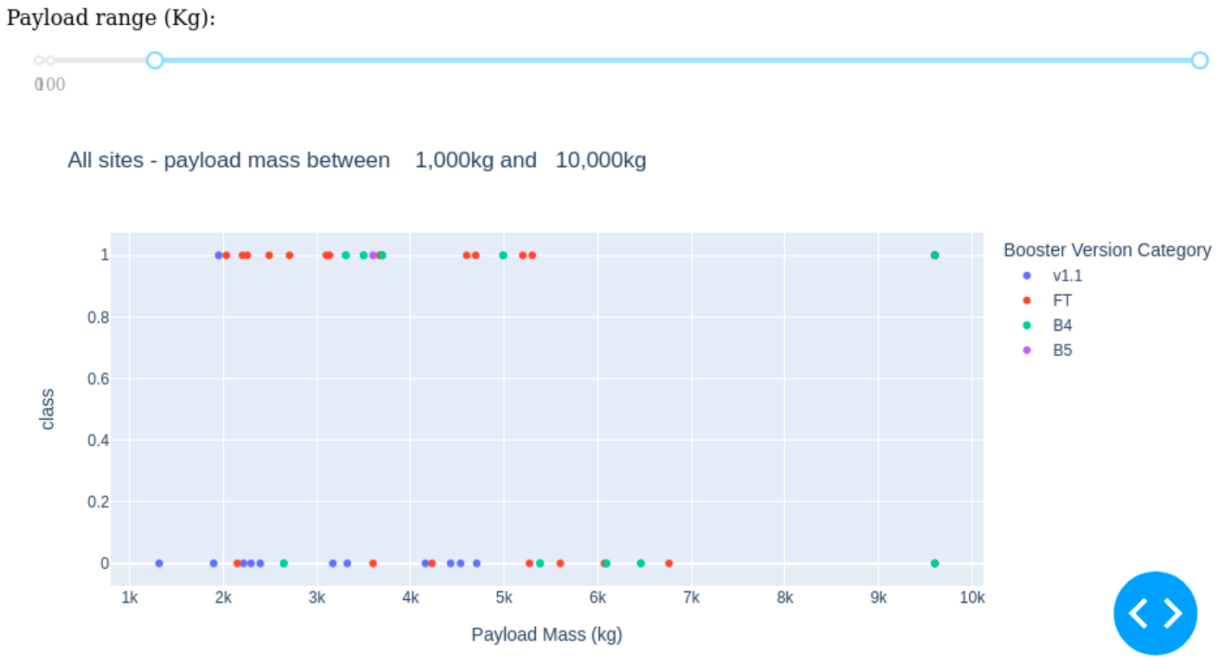
Total Launches for KSC LC-39A

- Plotly was used the visualize the successful launches for KSC LC-39A as a pie chart.
- We see that KSC LC-39A has a success rate of 76.9%.



Class vs Payload Mass visualised using Plotly

- We plotted the class against the Payload Mass in kg to visualize the relationship and draw insights.
- We see form the dot plot that unsuccessful launches have a higher variance of Payload Mass than does successful launches.

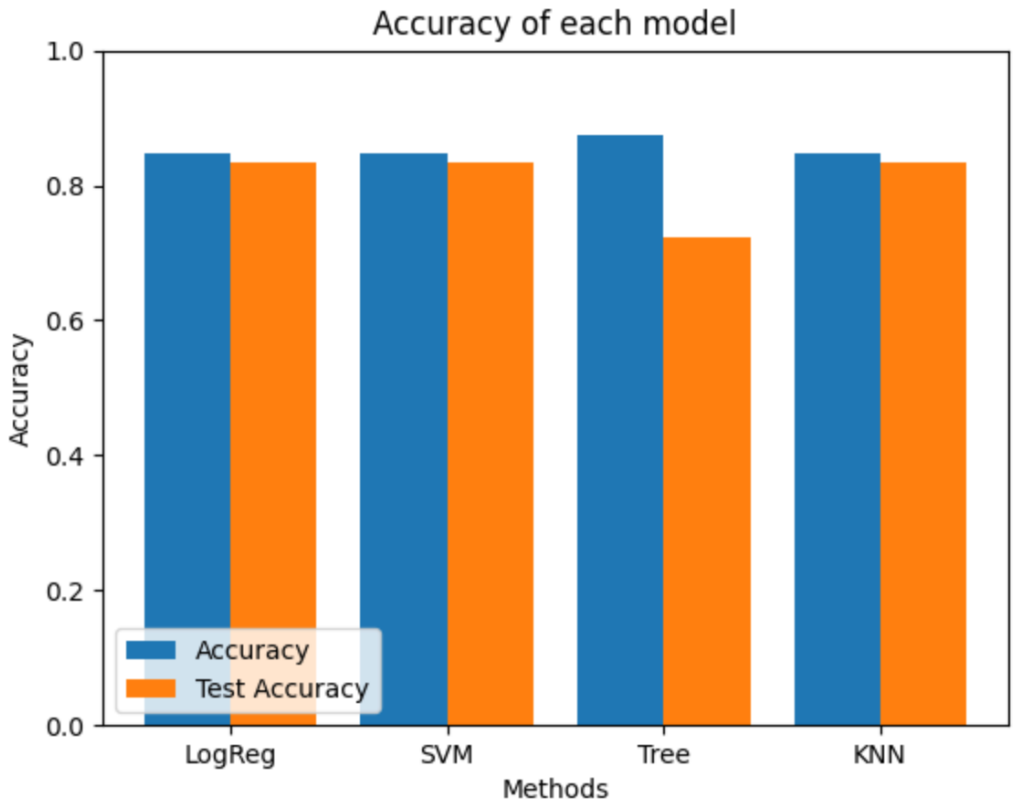


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- We trained and tested four different models to determine which performed the best on the test set.
- We can see from the bar chart that the Tree Classification model was the highest performing model.



Confusion Matrix

- The best performing model was tree classification with a score of 0.9444.
- We see a diagram of the confusion matrix for the tree classification model.

TASK 9

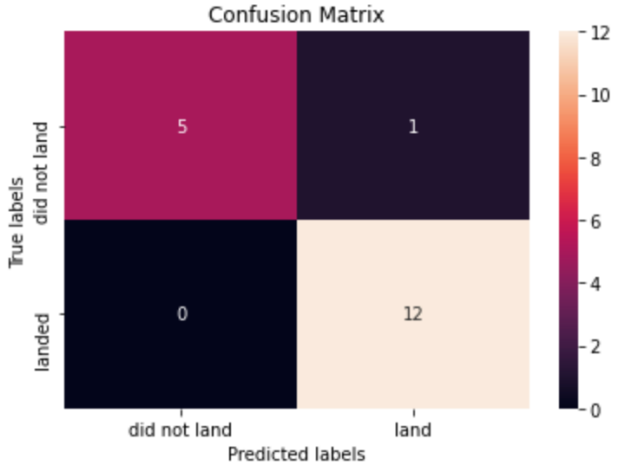
Calculate the accuracy of tree_cv on the test data using the method `score` :

```
[4]: print("accuracy :", tree_cv.score(X_test, Y_test))
```

```
accuracy : 0.9444444444444444
```

We can plot the confusion matrix

```
[5]: yhat = tree_cv.predict(X_test)
     plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- The success rate of launches improved over time.
- Pay Load masses which were under 7000 kg had a higher success rate.
- The most successful launch site is KSC LC-39A which we determined from the Plotly dashboards with a success rate of 76%.
- The Tree Classification model was the most accurate and can be used to predict future successful landings of the SpaceX rockets first stage.

Thank you!