IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Megala Anbarasu
03/10/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data was collected from SpaceXdata API
  - Data was collected from scraped from Wikipedia
  - Data wrangling process included converting Missing Outcomes to classes
  - Exploratory Data Analysis with SQL and data visualizations (plots and maps)
  - Predictive analysis to train model
- Summary of all results
  - Exploratory data analysis results
  - Logistic Regression, SVM (Support Vector Machine), and KNN (k-Nearest Neighbors) results

Introduction

- Project background and context

  SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars. Other providers cost up to165 million dollars each. SpaceX savings and price difference are because SpaceX can reuse the first stage. Space Y that would like to compete with SpaceX founded by Billionaire industrialist Allon Musk. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch

- Problems you want to find answers

  - How features such as payload mass, launch site, number of flights, and orbits affect successful landing of the first stage.
  - Determine price of each launch.
  - Train machine learning model and use public information if the first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

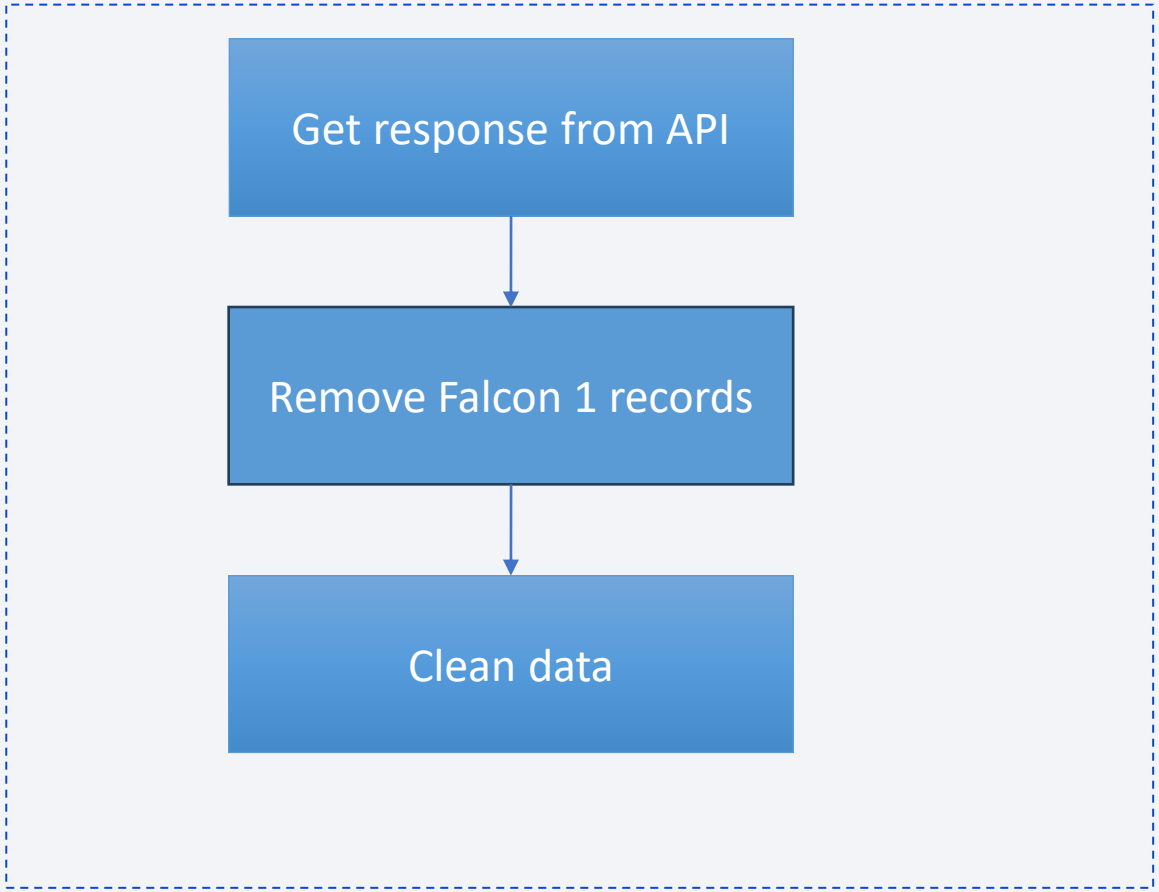
- Data collection methodology:
  - Using SpaceX API and web scraping Wikipedia web page
- Perform data wrangling
  - Data was filtered, included converting Missing Outcomes to classes, convert missing values and apply one hot encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tested Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors. Reviewed the output of the confusion matrix. Decision Tree has the highest accuracy of 87.5%

Data Collection

- Data was collected using API calls to the SpaceX API and Web Scraping the Wikipedia page. We decoded the data and adding to a data frame.

Data Collection – SpaceX API

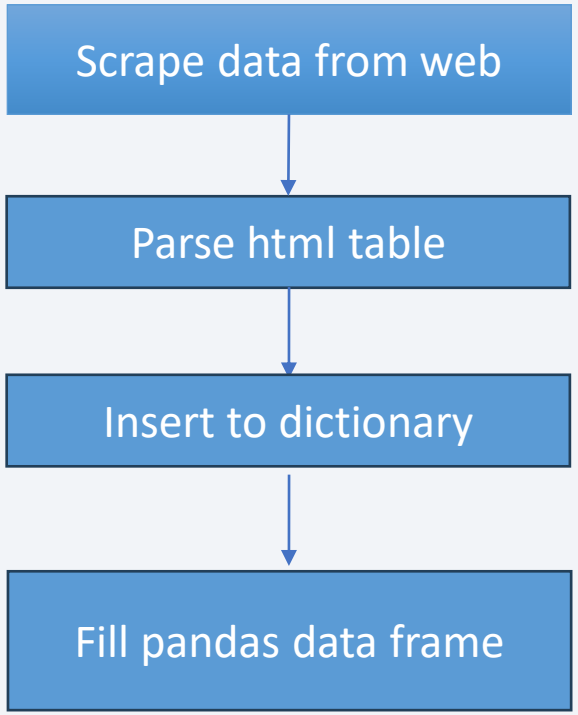
- We got the data with SpaceX REST calls.
- We made subsequent calls to get other information. i.e Launch site information from ID
- Store date in lists
- Remove Falcon 1 records
- Remove nulls



GIT URL: https://github.com/megalaanbarasu/Applied-Data-Science-Capstone/blob/main/LAB1-WEEK1-jupyter-labs-spacex-data-collection-api%20(4).ipynb

Data Collection - Scraping

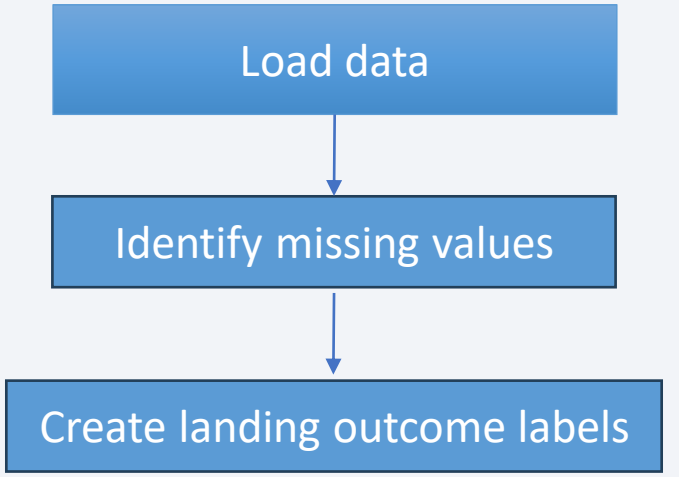
- Scraped data from Wiki page
- Parse the table
- Insert to dictionary
- Convert to Panda data frame



GIT URL: https://github.com/megalaanbarasu/Applied-Data-Science-Capstone/blob/main/LAB2-WEEK1-jupyter-labs-webscraping%20(1).ipynb

Data Wrangling

- We retrieved the data and performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.



GIT URL: https://github.com/megalaanbarasu/Applied-Data-Science-Capstone/blob/main/LAB3-WEEK1-labs-jupyter-spacex-Data%20wrangling%20(2).ipynb

EDA with Data Visualization

Multiple graphs was plotted with Matplotlib to see determine relationship between various features.

GIT URL: https://github.com/megalaanbarasu/Applied-Data-Science-Capstone/blob/main/LAB2-WEEK2-jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(2).ipynb

EDA with SQL

- Various SQL was performed to the datasets to get booster information, payload, outcomes by dates and years

GIT URL: https://github.com/megalaanbarasu/Applied-Data-Science-Capstone/blob/main/LAB1-WEEK2-jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- Created and plotted map objects with markers, circles, lines using folium map. This helps look at markers, clusters of similar coordinates grouped by, and drawing lines to closest city, railway and highway as well as how location affects launch outcomes.

GIT URL: https://github.com/megalaanbarasu/Applied-Data-Science-Capstone/blob/main/LAB1-WEEK3-lab_jupyter_launch_site_location.jupyterlite%20(1).ipynb

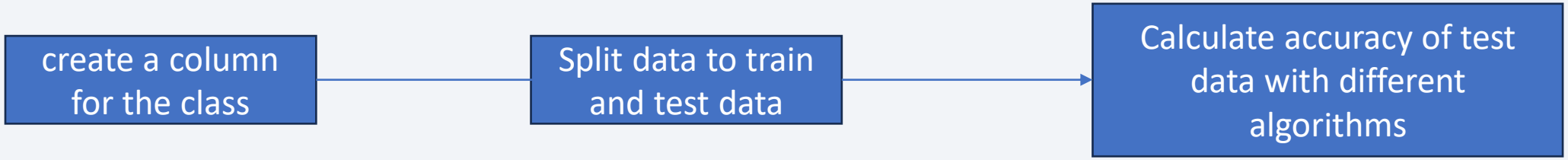
Build a Dashboard with Plotly Dash

- We created a dashboard to show success rates for launch sites and how payload range changes can affect success count by booster version. This interactive dashboard show changes as options are changed.
- We wanted to look at how each launch sites shows success rates and why payload may be a feature in launch outcomes.

GIT URL: https://github.com/megalaanbarasu/Applied-Data-Science-Capstone/blob/main/LAB2-WEEK3-spacex_dash_app%20(1).py

Predictive Analysis (Classification)

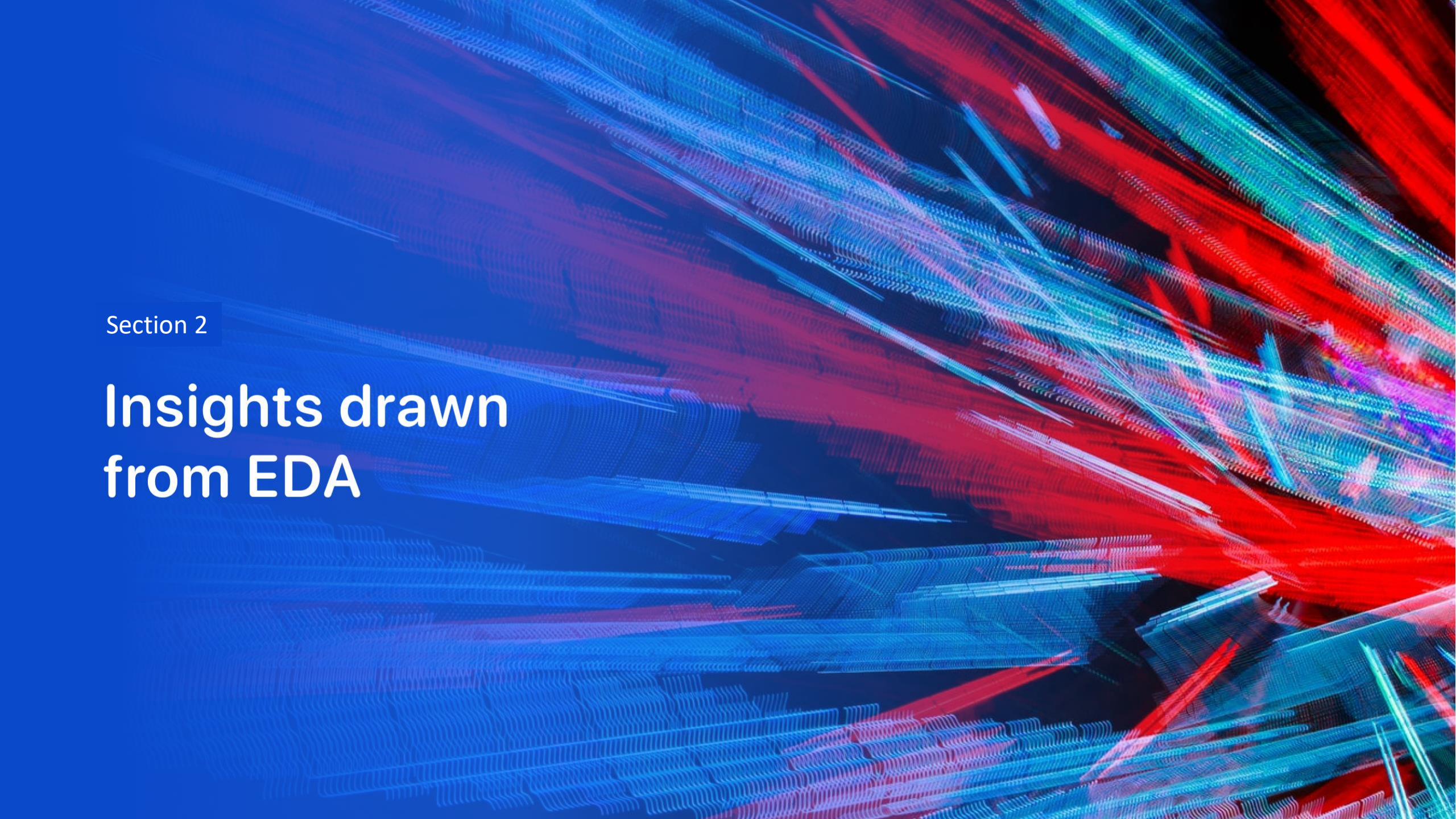
- We loaded data into data frames and created a machine learning pipeline to predict if the first stage will land given the data.
- We tested with SVM, KNN, Logical regression and decision tree algorithms and listed the accuracies from most accurate to least accurate.



GIT URL: https://github.com/megalaanbarasu/Applied-Data-Science-Capstone/blob/main/LAb1-WEEK4-SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

- Exploratory data analysis results
  - KSC LC-39A has the highest success rate among landing sites
  - Orbits Types ES-L1,GEO, HEO and SSO has 100% success of landing
- Interactive analytics demo in screenshots
  - Most launch sites are in proximity to the Equator line
  - Most launch sites are in very close proximity to the coast
- Predictive analysis results
  - All 4 models performed really well and are close in accuracies. Decision tree performed slightly more well.

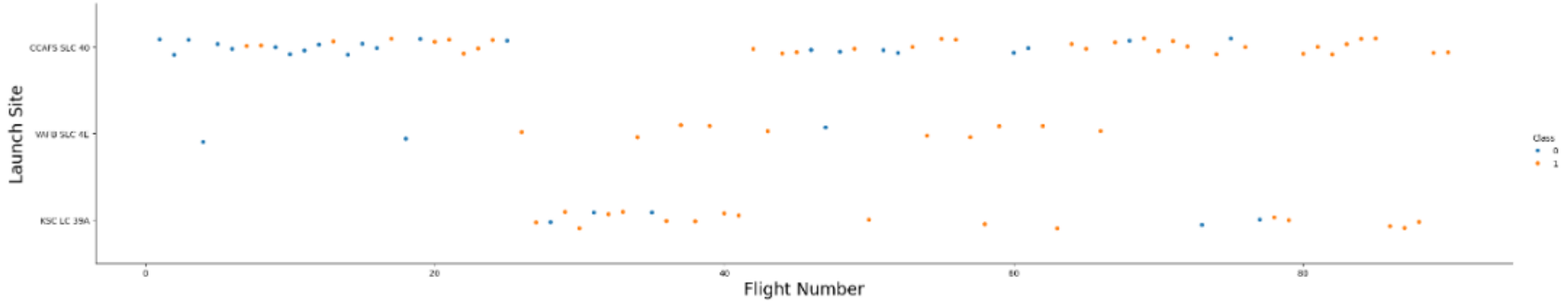

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- Launches from site CCAFS SLC 40 are higher than others
- More flight numbers increases the success rate, with more oranges to the right of the chart.

Next, let's drill down to each site visualize its detailed launch records.

[5]:
```
### 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()
```

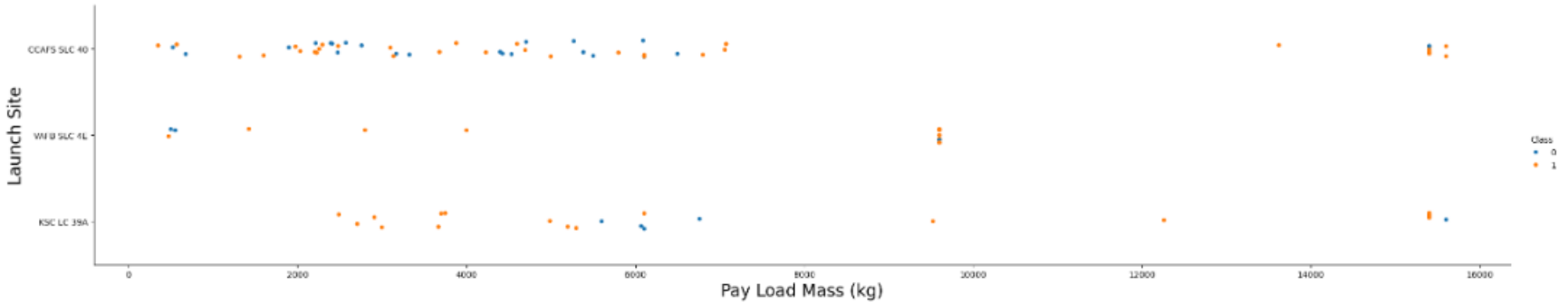


Catplot to show relationship between FlightNumber and Launch Site.

Payload vs. Launch Site

- More launches happens within the 0 to 7000 kg payload
- Scattered success rate for higher payload

```
[7]: ### 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 (kg)",fontsize=20)
     plt.ylabel("Launch Site",fontsize=20)
     plt.show()
```



Catplot to show relationship between PayloadMass and Launch Site

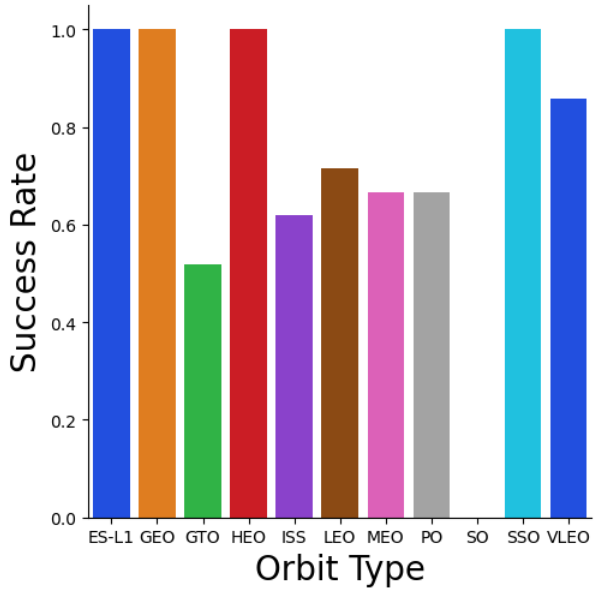
Success Rate vs. Orbit Type

- 100% successful rates for orbit types ES-L1, GEO, HEO and SSO.
- 0 success for orbit type SO
- 50 -85% for the other orbit types

Let's create a `bar chart` for the sucess rate of each orbit

```
[28]: # HINT use groupby method on Orbit column and get the mean of Class column

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



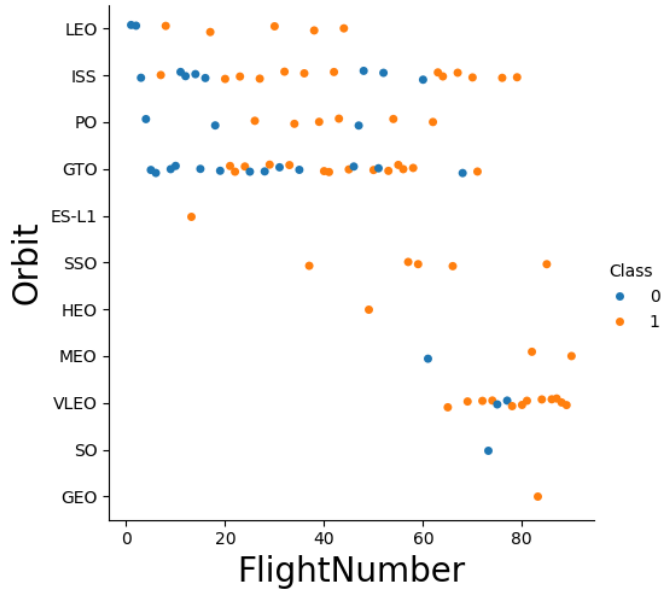
Catplot to show relationship between Orbit Type and Success Rate

Flight Number vs. Orbit Type

- Success rates typically increases with flight numbers are evident for LEO and ISS.

Analyze the ploted bar chart try to find which orbits have high sucess rate.

[21]:
```
### TASK  4: Visualize the relationship between FlightNumber and Orbit type
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

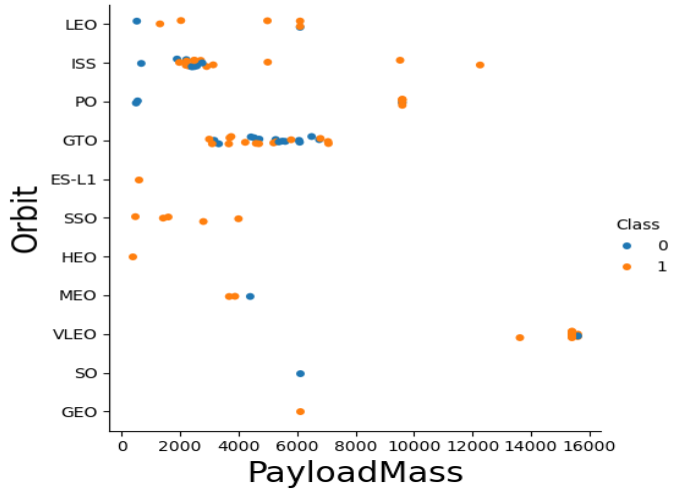


Catplot to show relationship between FlightNumber and Orbit

Payload vs. Orbit Type

- LEO, ISS and PO has more success with heavier payload
- ES-L1, SSO has success rates with lower pay loads
- The rest does not seem to have a correlation

```
[12]: ### TASK  5: Visualize the relationship between Payload and Orbit type
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

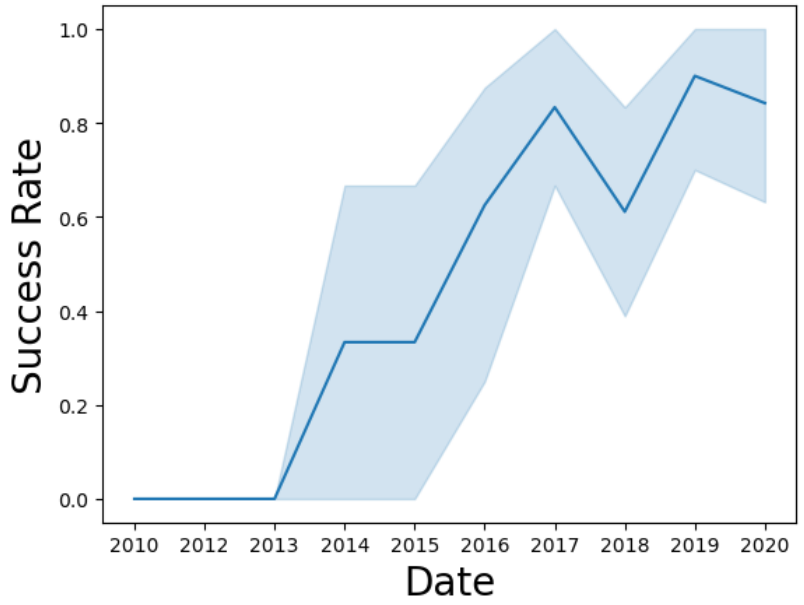


Catplot to show relationship between PayloadMass and Orbit

Launch Success Yearly Trend

- No success rate in the early 2010s
- Success rate increased consistently with plateau in 2014, and a dip in 2018 and 2020

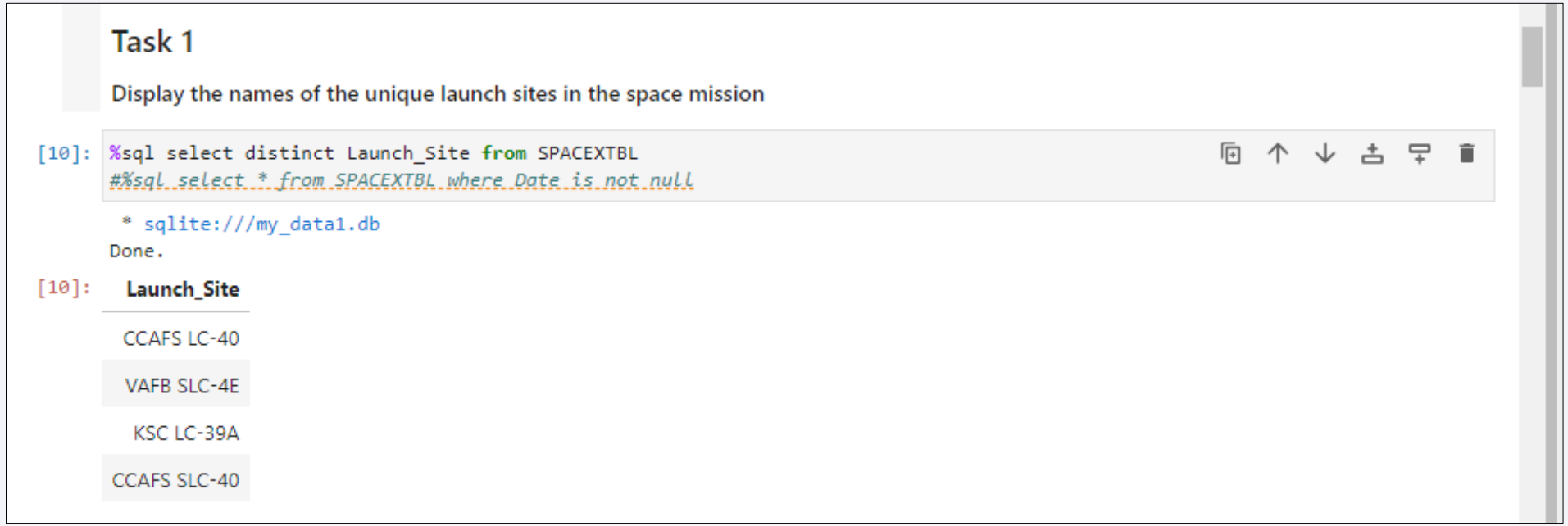
```
[14]: # Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(data=df, x="Date", y="Class")
plt.xlabel("Date",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



Catplot to show relationship between Date(Year) and SuccessRate

All Launch Site Names

The 4 unique sites are displayed here using the distinct keyword to filter unique values



Task 1

Display the names of the unique launch sites in the space mission

```
[10]: %sql select distinct Launch_Site from SPACEXTBL
      #%sql select * from SPACEXTBL where Date is not null
```

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

[10]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

Launch Sites starting with CCA was queried using the like query

```
[38]: %sql  Select * from SPACEXTBL where Launch_Site like 'CCA%' LIMIT 5
 * sqlite:///my_data1.db
Done.
```

[38]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcom
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attemp
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attemp
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attemp

Total Payload Mass

Get the total payload mass carried by NASA CRS using the SUM keyword

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

```
[20]: %sql  Select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where customer = 'NASA (CRS)'
```

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

[20]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Get the average payload mass carried by Booster F9 v1.1 using the SUM keyword.

Task 4

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

```
[21]: %sql Select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version like 'F9 v1.1%'
```

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

[21]:

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

To find the first successful ground landing date, we use the min date keyword with the lading outcome column filtered to ‘Success (ground pad)’

Task 5

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

Hint:Use min function

[24]:
```
#%sql Select Date from SPACEXTBL where Landing_Outcome = 'Success (ground pad)' order by Date asc 2015-12-22
%sql Select MIN(Date) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)' #2015-12-22
```

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

[24]:

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

To find the successful landing for drone ship between 4000 and 6000 kg, we use the distinct booter version with filter for landing outcome column and between keyword for the payload mass column.

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

```
[29]: %sql Select distinct(Booster_Version) from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ B
```

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

[29]:

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

Total Number of Successful and Failure Mission Outcomes

To get the count for success and failure, we would need to group by the Mission Outcome column and count the Mission Outcomes.

We see the outcomes are 99 true success, 1 success with unknown payload status and 1 failure

Task 7

List the total number of successful and failure mission outcomes

```
[31]: %sql Select DISTINCT COUNT(Mission_Outcome), Mission_Outcome  from SPACEXTBL group by Mission_Outcome
```

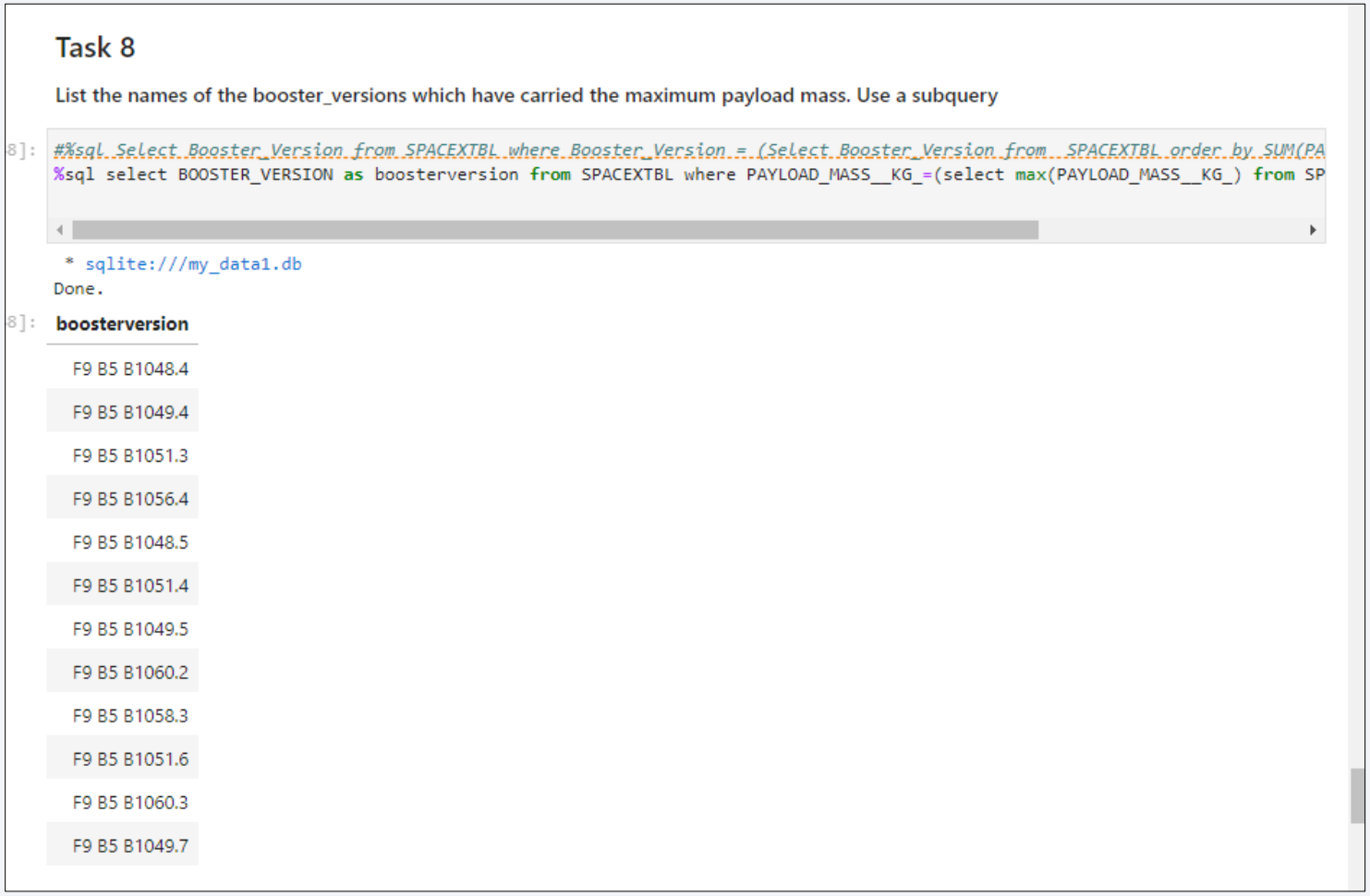
```
 * sqlite:///my_data1.db
Done.
```

[31]:

COUNT(Mission_Outcome)	Mission_Outcome
1	Failure (in flight)
98	Success
1	Success
1	Success (payload status unclear)

Boosters Carried Maximum Payload

To get the list of boosters that caried max payload, we get the max payload in a subquery and use that in the first query where clause



Task 8

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

```
[8]: #%sql Select Booster_Version from SPACEXTBL where Booster_Version = (Select Booster_Version from  SPACEXTBL order by SUM(PA
     %sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SP
```

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

[8]:

boosterversion
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

We get the dates for year 2015 by filtering the date column with substring and the landing outcome for drone ship. We get 2 dates in 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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[11]:
```
%sql Select substr(Date, 6,2), Date, Landing_Outcome, Booster_Version, Launch_Site  from SPACEXTBL where \
Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5)='2015'
```

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

[11]:

substr(Date, 6,2)	Date	Landing_Outcome	Booster_Version	Launch_Site
01	2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

To get the outcomes between 2010 and 2017, we use the between keyword on the date column and do a count on the landing outcomes ordered by highest count at the top. 10 is the highest outcome but without outcome status.

Task 10

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.

[46]:
```
%sql Select COUNT(Landing_Outcome),  Landing_Outcome from SPACEXTBL where Date between '2010-06-04' and '2017-03-20' group
```

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

[46]:

COUNT(Landing_Outcome)	Landing_Outcome
10	No attempt
5	Success (drone ship)
5	Failure (drone ship)
3	Success (ground pad)
3	Controlled (ocean)
2	Uncontrolled (ocean)
2	Failure (parachute)
1	Precluded (drone ship)


Section 3
Launch Sites
Proximities Analysis

All Launch Sites Markers

- Created and plotted map objects with markers, circles, lines using folium map. This helps look at markers, clusters of similar coordinates grouped by, and drawing lines to closest city, railway and highway as well as how location affects launch outcomes.
- We can see the sites are closer to the coasts

Cluster Markers

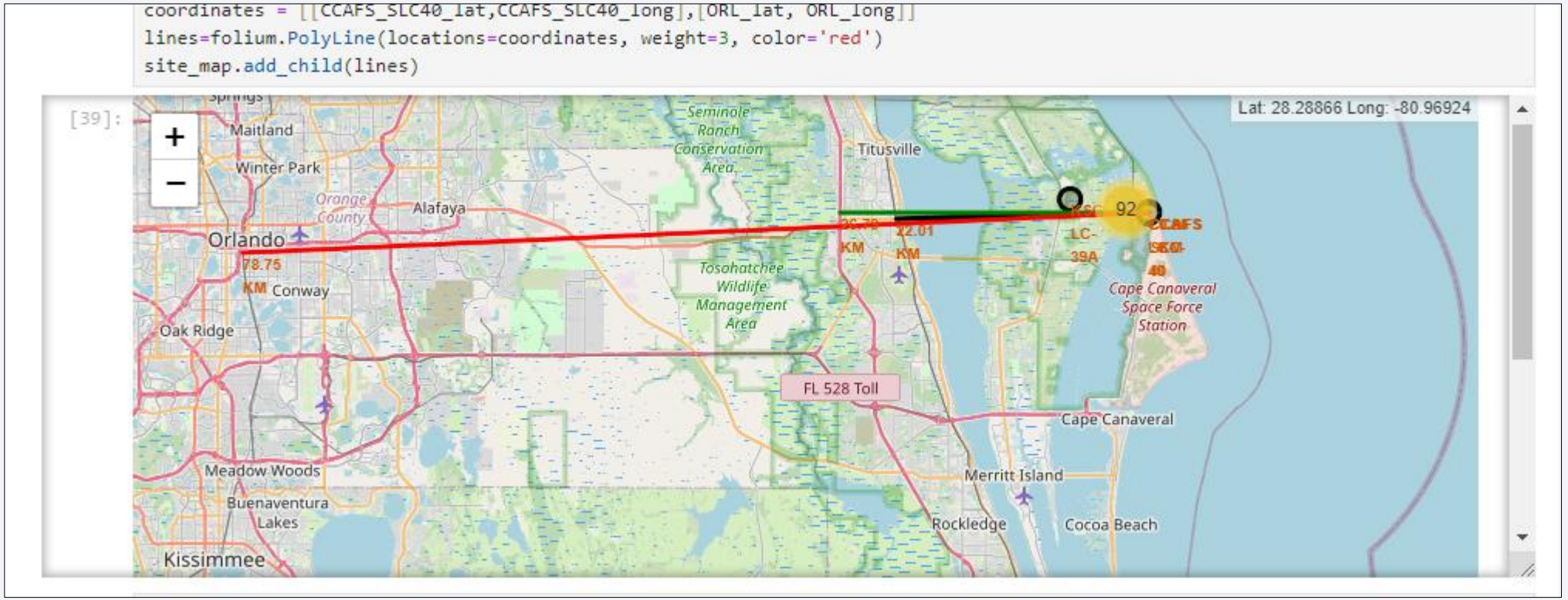
We can clearly see the numbers and on click the spread of success and failed outcomes for those coordinates with the marker clusters.

Marker cluster when clicked and zoomed out

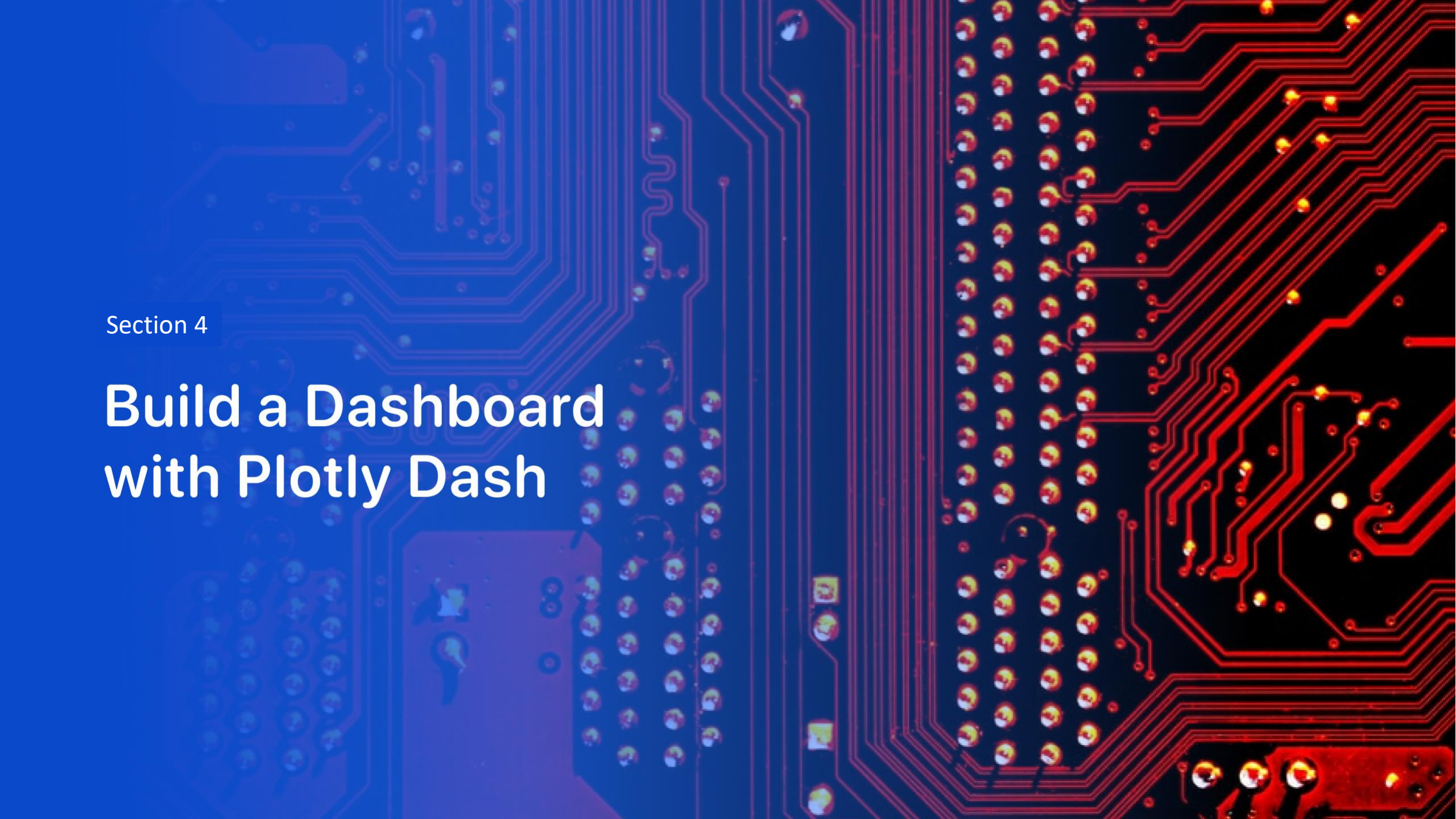
Marker cluster when clicked with launch outcomes color

Launch site proximities

Here we can see the 3 lines, red, green and black showing distances between the launch sites to nearest landmarks in kilometers.

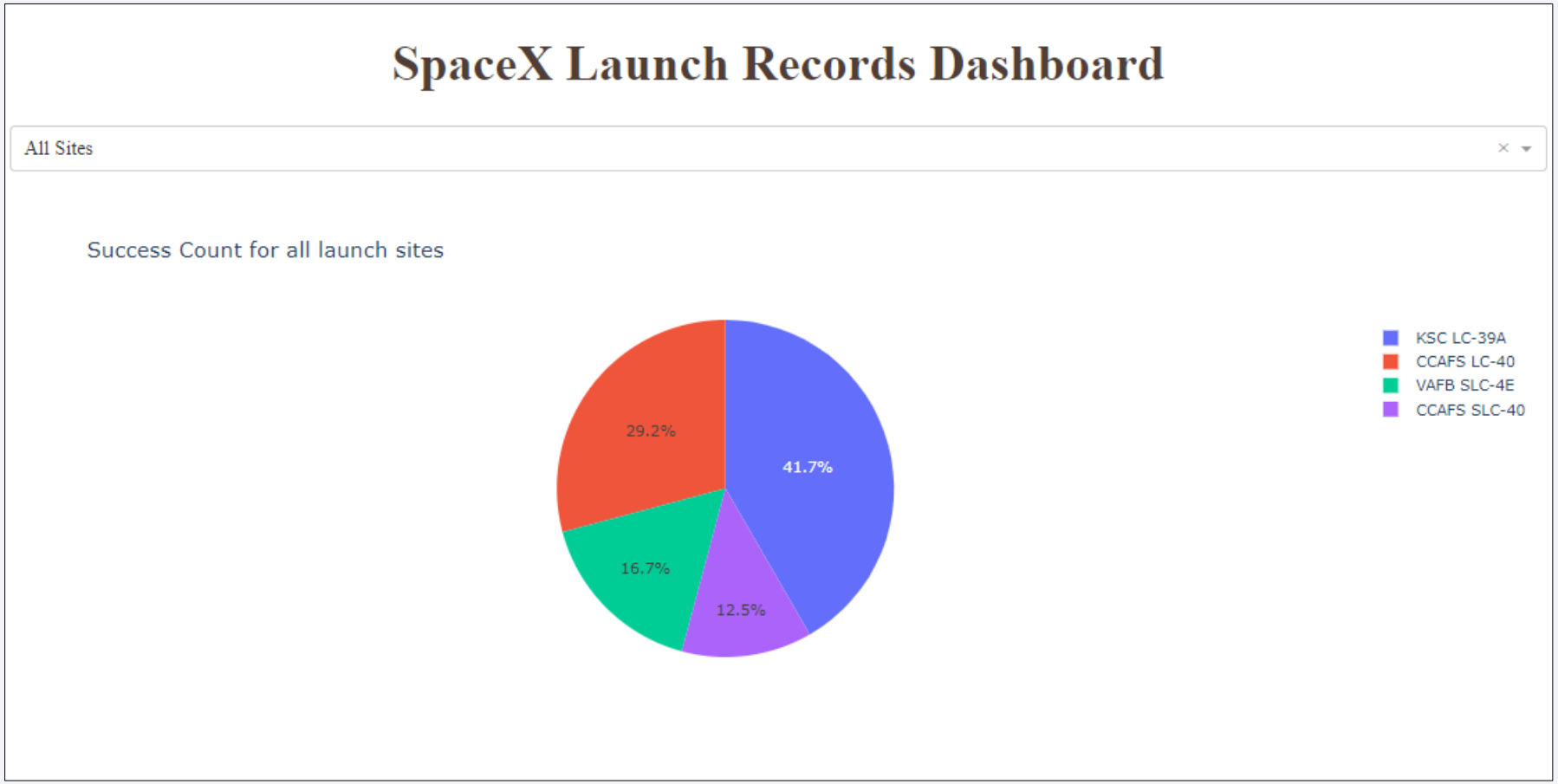


Polyline showing launch sites to its closest city, highway


Section 4
Build a Dashboard with Plotly Dash

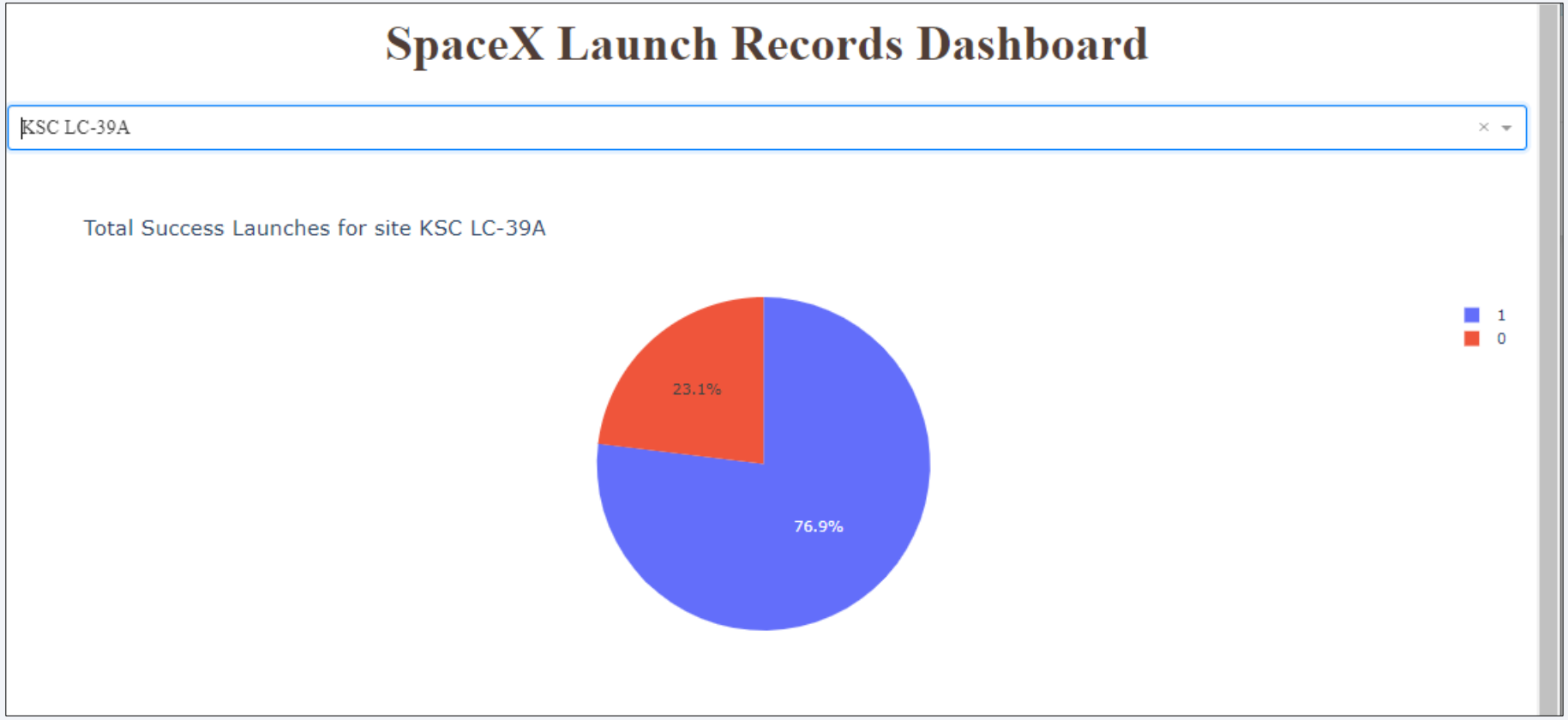
Dashboard – Launch success for all sites

In this pie chart, we can see KSC LC-39A has the highest success for launches



Dashboard – Highest launch rate

KSC LC-39A site show below with success and failure rates. Among all sites, this is the site with most success.



Dashboard – payload selector

- In these images, we can payload between 2000 and 600 has highest success.
- Higher payloads has less success.

Payload range with high success

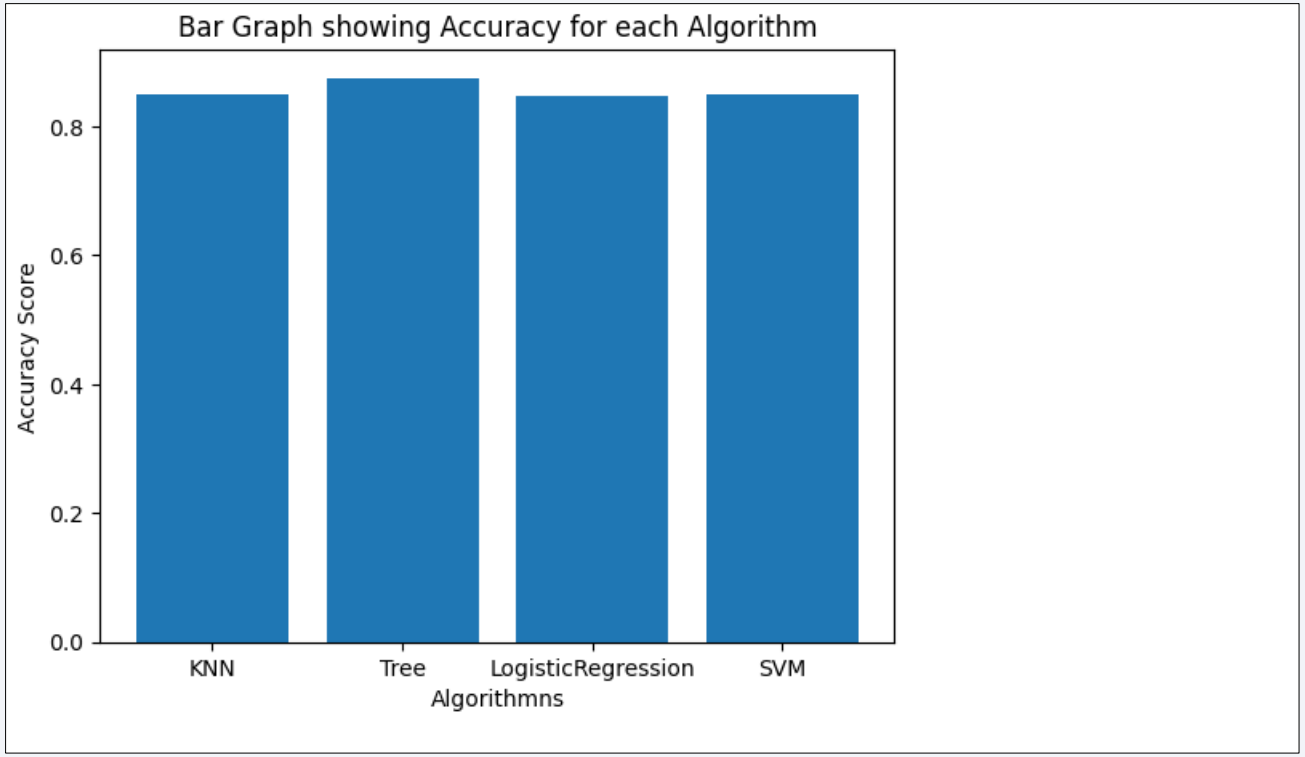
Payload range with low success

Section 5

Predictive Analysis (Classification)

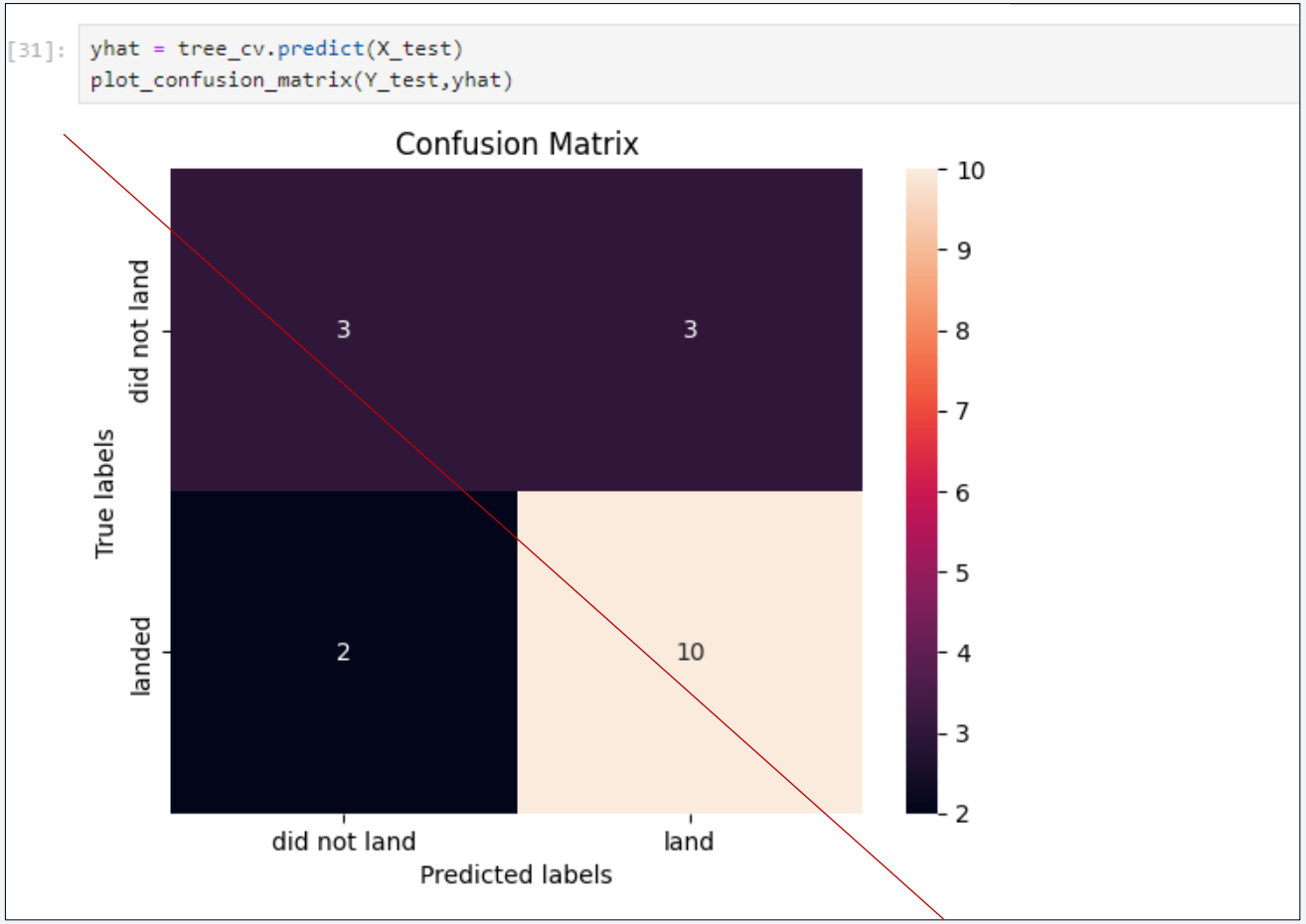
Classification Accuracy

Bar chat below shows the accuracy comparison for all algorithms. All 4 has pretty close rates. However, tree has the highest accuracy of 87%.



Confusion Matrix

The best model is the decision tree and here is the confusion matrix for that model. The diagonal line shows the true positive and true negative values.



Conclusions

- There is a correlation between number of flights and success rates. As the number of flights increase, success increases.
- 100% successful rates for orbit types ES-L1, GEO, HEO and SSO .
- Launch site KSC LC-39A has the highest success for launches.
- Decision tree has the best accuracy for prediction and is the best model for this data set.

Appendix

All notebooks, screenshots and data sets are available on my GitHub Repository.

GIT REPO URL: https://github.com/megalaanbarasu/Applied-Data-Science-Capstone

Thank you!