IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Partson Pedzisayi
08/07/2024

Outline

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

Executive Summary

- The goal of this research was to predict if falcon 9's first stage launch will land. It is done using machine learning by following specific stages:
  - data collection , data wrangling , data exploration, data analysis and come up with the best predictive models to achieve our goal.
- After analyzing the data, three models were built, refined, and evaluated to come up with the best predictive model. The models are SVM (82%), Decision Tree(86%) and KNN(84%).
- After considering the three models and their accuracy scores it was determined that the best predictive model could be built using a decision tree.

Introduction

- In this space age, companies like Blue Origin, SpaceX, and Rocket Lab strive to make space exploration affordable for everyone, with spaceX being the most inexpensive.
- Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars, other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage.
- The first stage sometimes lands successfully, sometimes it doesn't. If the re-usability of the first stage is unknown, so is the cost , therefore we want to know whether or not the launch will land or not in order to determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch.

Section 1

Methodology

Methodology

Executive Summary

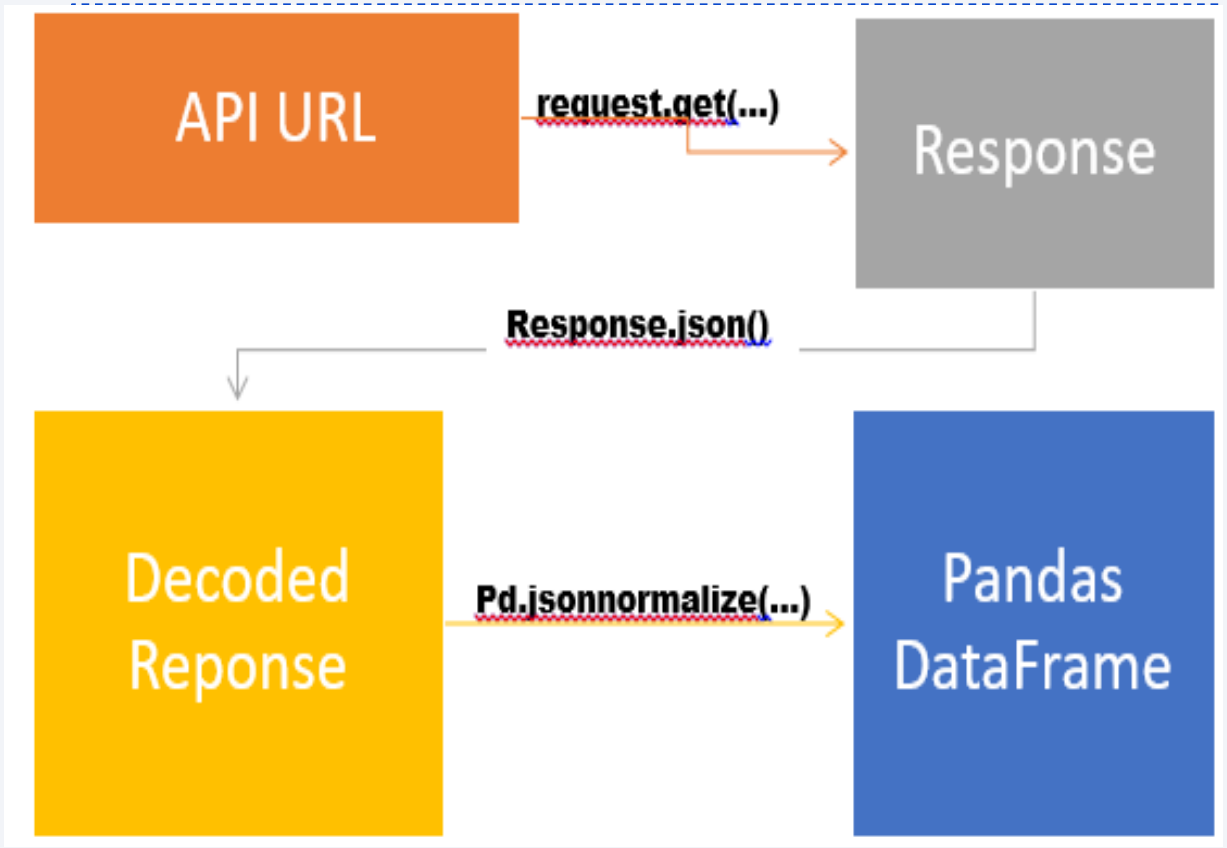
- Data collection methodology:
  - **A spaceX API is to be used to collect the data from the space X website.**
- Perform data wrangling
  - The collected was tabulated using a pandas dataframe and was cleaned by checking and dealing with missing values.
- Perform exploratory data analysis (EDA) using 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

- Data was collected using SpaceX API
- A series of helper functions was used to aid using the API to extract information using identification numbers in the launch data.
- Requested and parsed the SPaceX launch data using Get request.
- The response content was decoded as a Json using .json() and turned it nto a Pandas dataframe using .json_normalize()
- Some web scrapping was done for Falcon 9 launches from wikipedia into a pandas dataframe using beautiful soup.

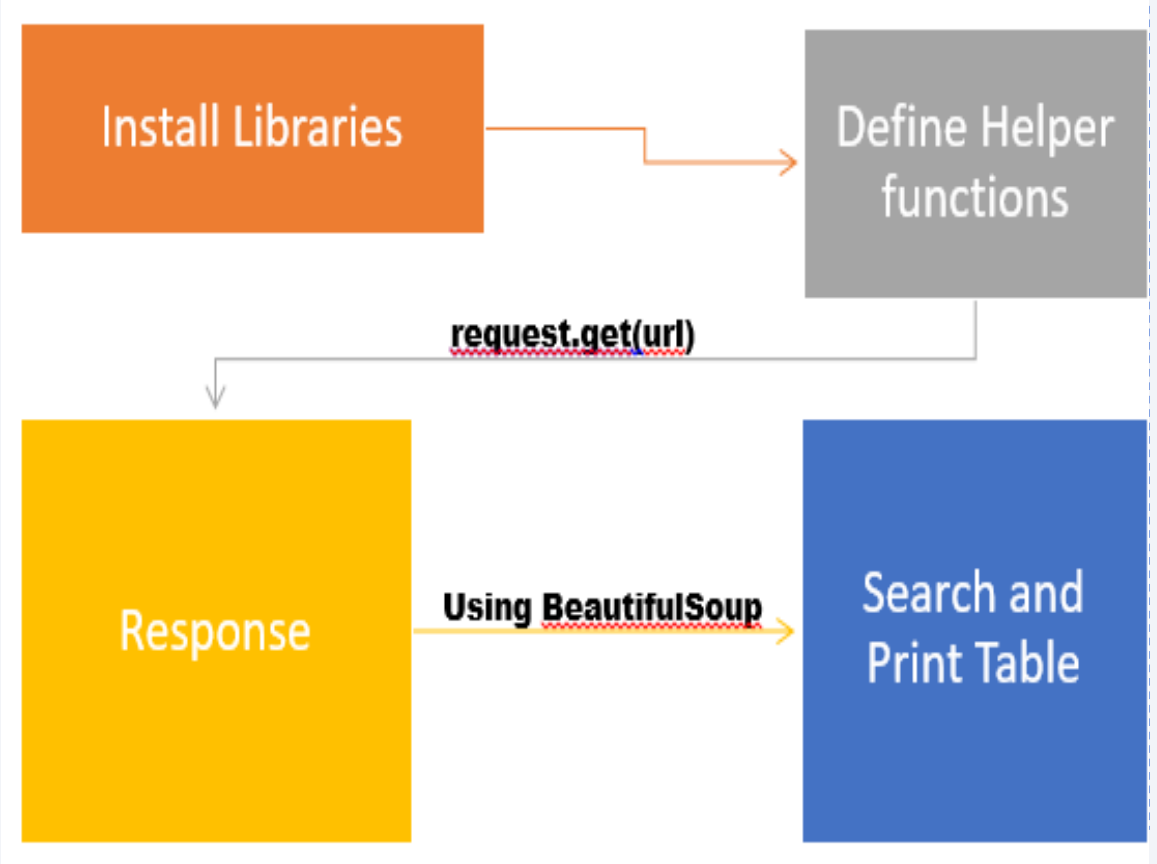
Data Collection – SpaceX API

- Made a request to the api url using requests.get(spacex_url)
- Stored the content as “response”
- Confirmed if request was successful using response.status_code
- Decoded the content using ,json() and turned it to a Pandas dataframe using .json_normalize()



Data Collection - Scraping

- Install Libraries, including BeautifulSoup
- Defined helper functions to process web scraped HTML table
- Made a request to the url using requests.get(url)
- Stored the content as “response”
- Created a BeautifulSoup object BeautifulSoup (response, 'html.parser')
- Searched tables from the page using soup.find_all('table') and the printed our target table
- Link to Web Scrapping file



Data Wrangling

- Data was processed using pandas and numpy.
- We loaded the last 10 rows to get an outlook of what the data looks like.
- Checked for null values
- Calculated number of launches on each site, number and occurrence of each orbit, number and occurence of mission outcome of the orbits to get some insights on the data
- Creating a landing outcome label from Outcome column to be used for prediction.
- GitHub Link to Data Wrangling file

EDA with Data Visualization

- A catplot of FlightNumber vs LaunchSite : was used to see how many number of flights it took before we started having a constant number of successful launches on each site.From the plot it seems, it a minum of 30 launches to start having constant successful launches.
- Payload and Launch Site: to observe if there is any relationship between the two.WE observed that for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).
- Success rate of each orbit type (bar chart): we want to visually check if there are any relationship between success rate and orbit type.
- Flight Number vs Orbit type:For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type. in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit
- Payload and Orbit type: to reveal the relationship between Payload and Orbit type. With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.
- launch success yearly trend: To visualize the average yearly success trend. the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.
- https://github.com/PedzisayiP/Applied-Data-Science-Cap-stone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Listed the total number of successful and failure mission outcomes

```
%sql Select "Mission_Outcome", count("Mission_Outcome") from SPACEXTABLE group by "Mission_Outcome"
```

- Listed the names of the booster_versions which had carried the maximum payload mass. Using a subquery [*%sql Select "Booster_Version", "PAYLOAD_MASS__KG_" from SPACEXTABLE where "PAYLOAD_MASS__KG_" = (Select Max("PAYLOAD_MASS__KG_") from SPACEXTABLE)]*
- Listed the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015. *[%sql SELECT substr(Date, 6, 2) "Month", substr(Date, 0, 5) "Year", "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTABLE where "Landing_Outcome" = "Failure (drone ship)" and Date like "2015%"]*
- Ranked 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 Count("Landing_Outcome"), Date from SPACEXTABLE where "Landing_Outcome" = "Failure (drone ship)" and Date like "2015%"*]
- https://github.com/PedzisayiP/Applied-Data-Science-Cap-stone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.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
- Marked all launch sites on the map using their coordinates with small circles and text labels to be able to tell each location's name
- Highlighted each launch as success/failure using marker clusters to see whther a launch was succsful or not.
- Used Mouseposition to see each coordinates on mouse hover, and drew polylines to be able to calculate the distance.
- https://github.com/PedzisayiP/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Added Launch Site Drop-downs for interactive selection of desired reports/graphs
- Created and defined the callback functions to update the input container based on the selected statistics and the output containers
- Created pie charts, line graphs and bar chart for recession and yearly report Statistics
- https://github.com/PedzisayiP/Applied-Data-Science-Cap-stone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Trained the data using train_test_split function, by separating the data into training and testing set.
- Built a SVM object and then built a GridSearhCV object using dictionary parameters to fine tune the gridsearch.
- Fitted the object to find the best parameters from the dictionary parameters.
- Found the best parameters by using the method .best_params_
- Evaluated the accuracy of the SVM model using .best_score method.
- Repeated the above steps to build different models, and then compared the models using their accuracies to come up with the best classification model
- https://github.com/PedzisayiP/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results (eda)

- We were able to List the date when the first succesful landing outcome in ground pad was achieved which is 2015-12-22.
- 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 SPACEXTABLE Where "Landing_Outcome" = 'Success (drone ship)' and "PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

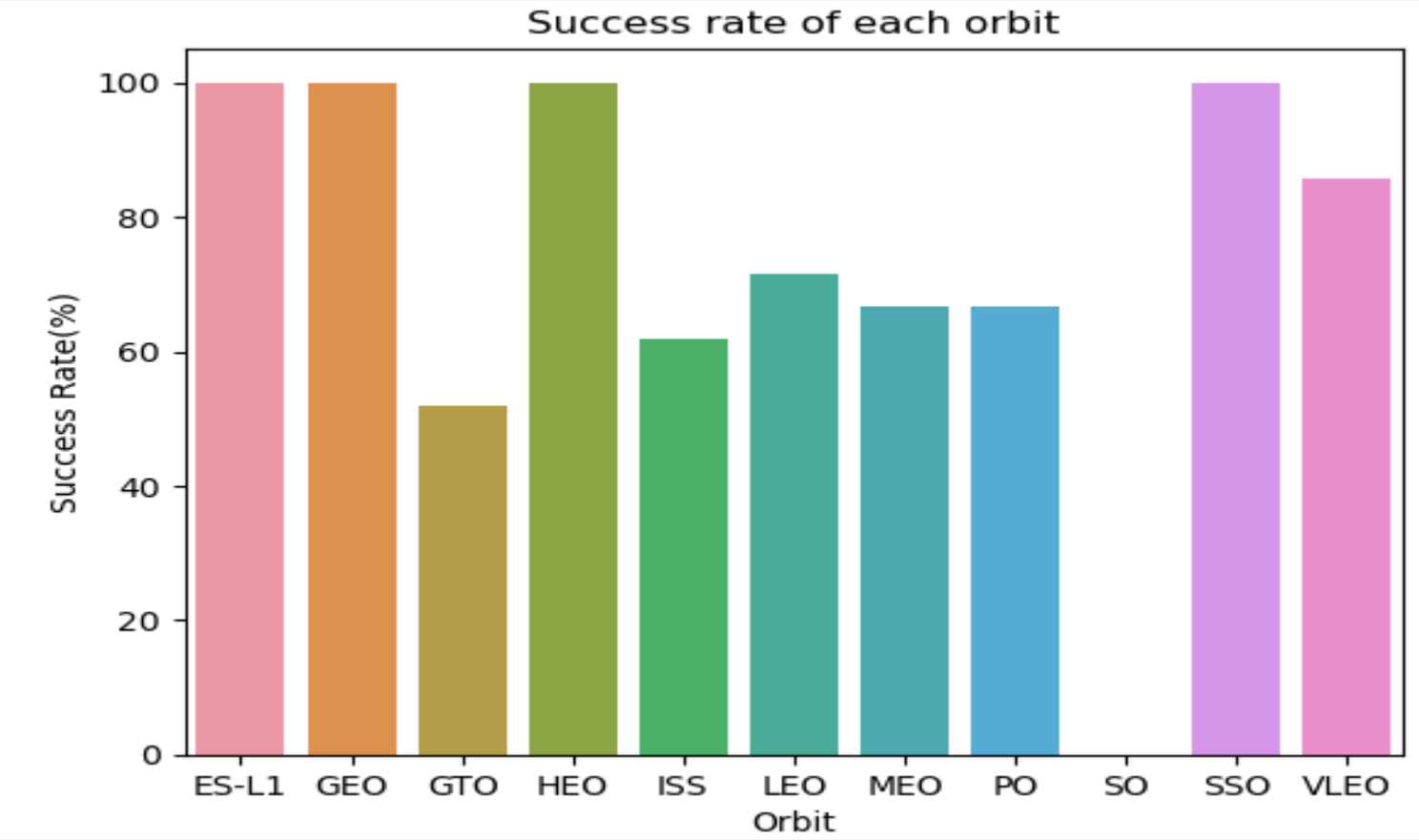
Results(eda) cont..

- We were able to list the total number of successful and failure mission outcomes

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

Results(eda) cont..

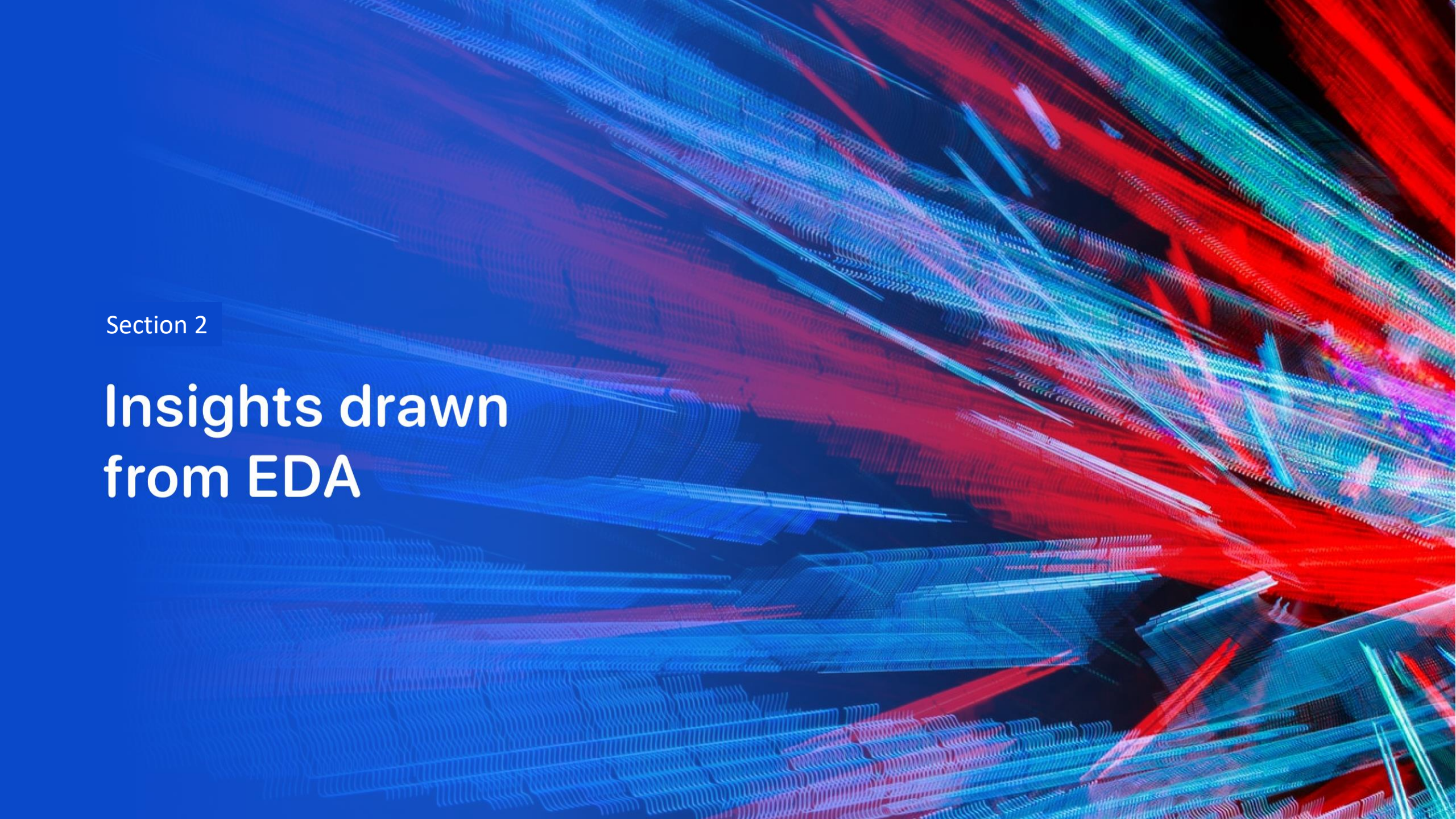
- We were able visualize the relationship between success rate of each orbit type



Results(predictive analysis)

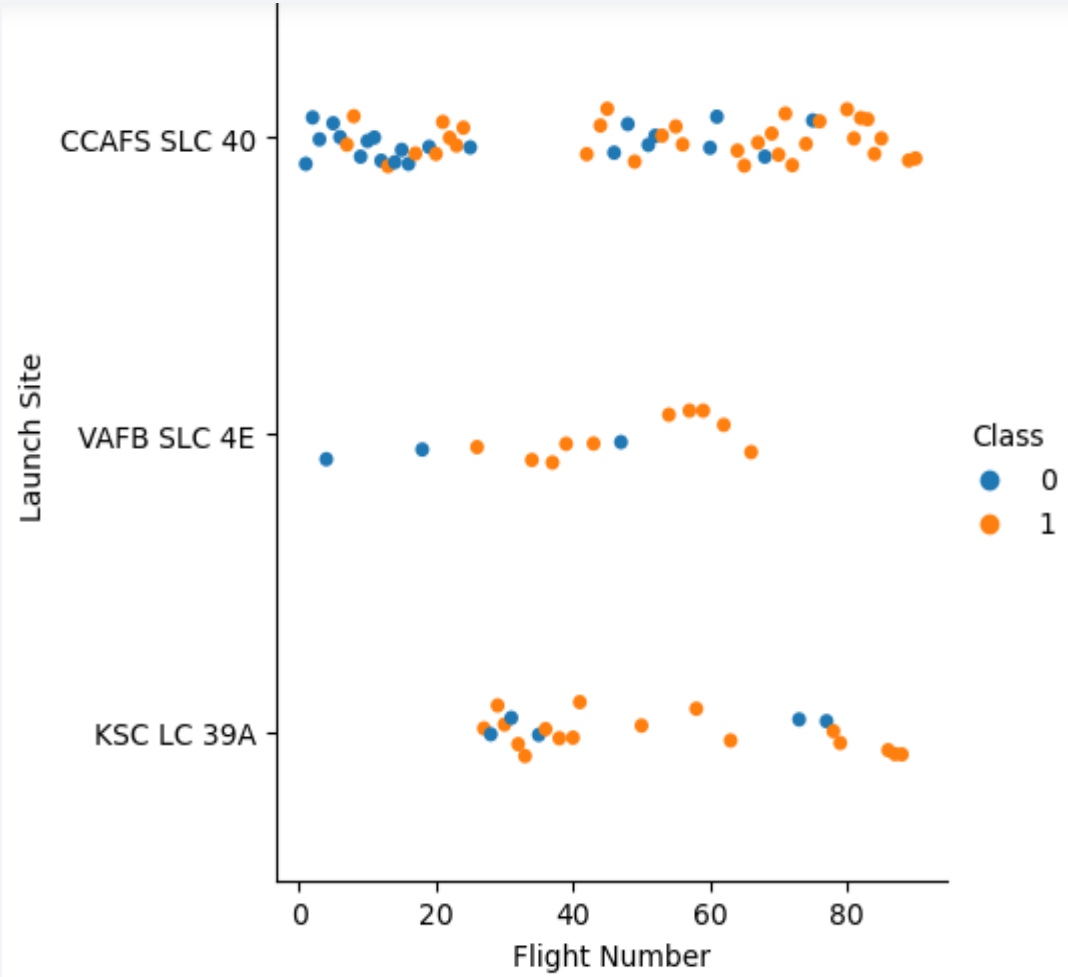
```
print("SVM best score:", svm_cv.best_score_)
print("Decision Tree best score:", tree_cv.best_score_)
print("KNN best score:", knn_cv.best_score_)


if svm_cv.best_score_ >= tree_cv.best_score_ and svm_cv.best_score_ >= knn_cv.best_score_:
    print("SVM is the best model.")
    best_model = svm_cv.best_estimator_
elif tree_cv.best_score_ >= svm_cv.best_score_ and tree_cv.best_score_ >= knn_cv.best_score_:
    print("Decision Tree is the best model.")
    best_model = tree_cv.best_estimator_
else:
    print("KNN is the best model.")
    best_model = knn_cv.best_estimator_
```

```
SVM best score: 0.8222222222222223
Decision Tree best score: 0.8666666666666668
KNN best score: 0.8444444444444444
Decision Tree is the best model.
```

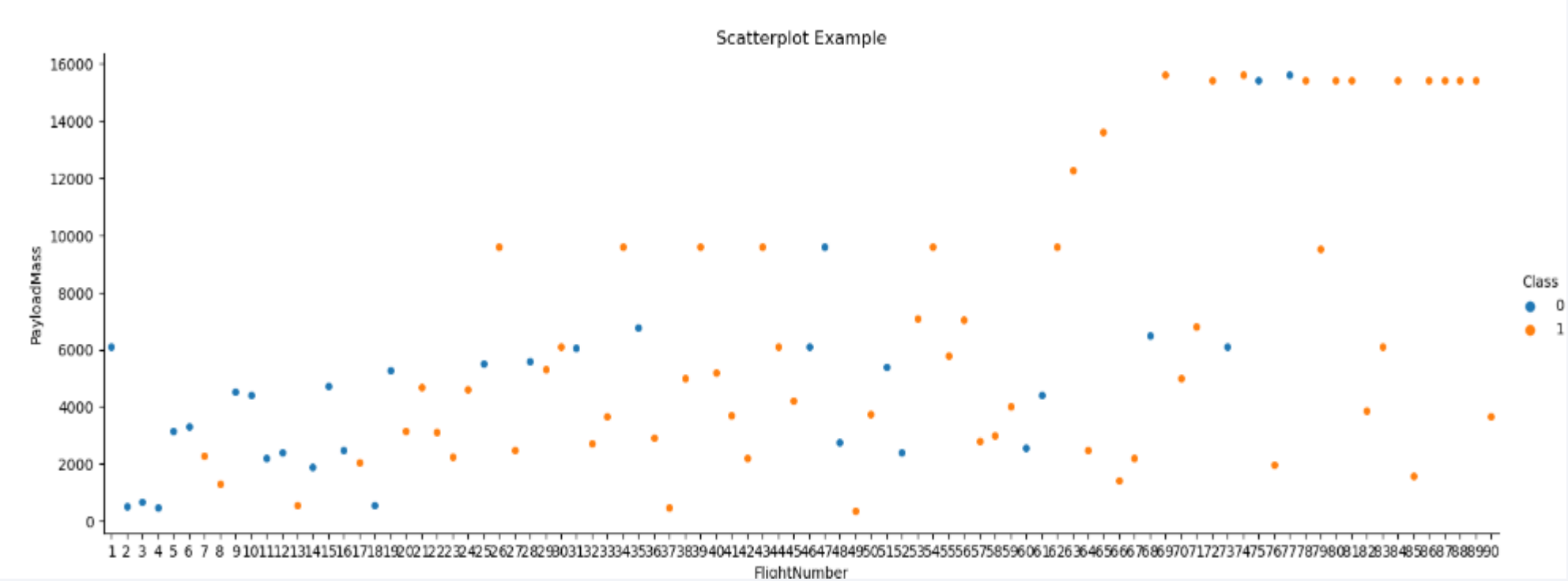

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- The first several launches before the 20th flight did not land successfully.
- Started having a significant number of successful landings after around the 40th flight in all launch sites.
- VAFB SLC 4E Has the least number of unsuccessful landings. Most of them were a success.
- Most launches were tested on CCAFS SLC 40, which explains the most unsuccessful first landings as compared to other launch sites.



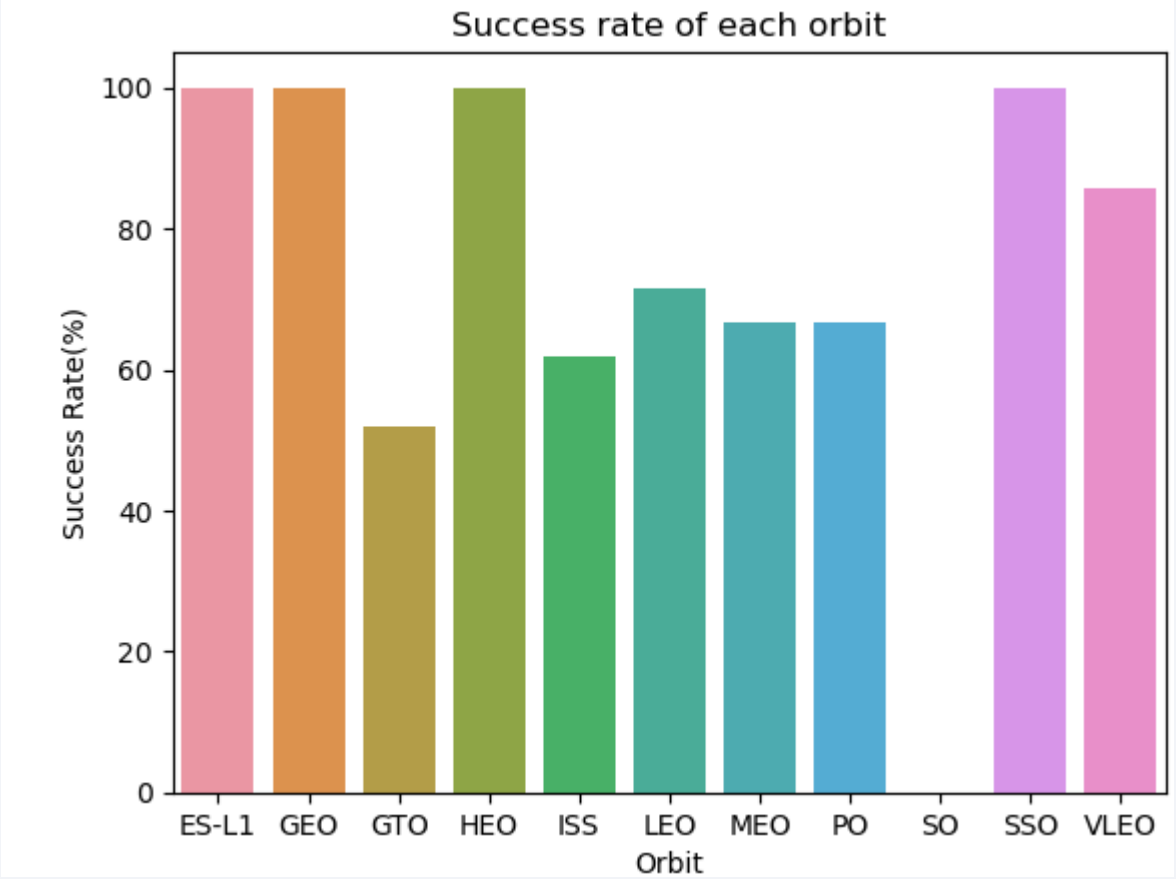
Payload vs. Launch Site



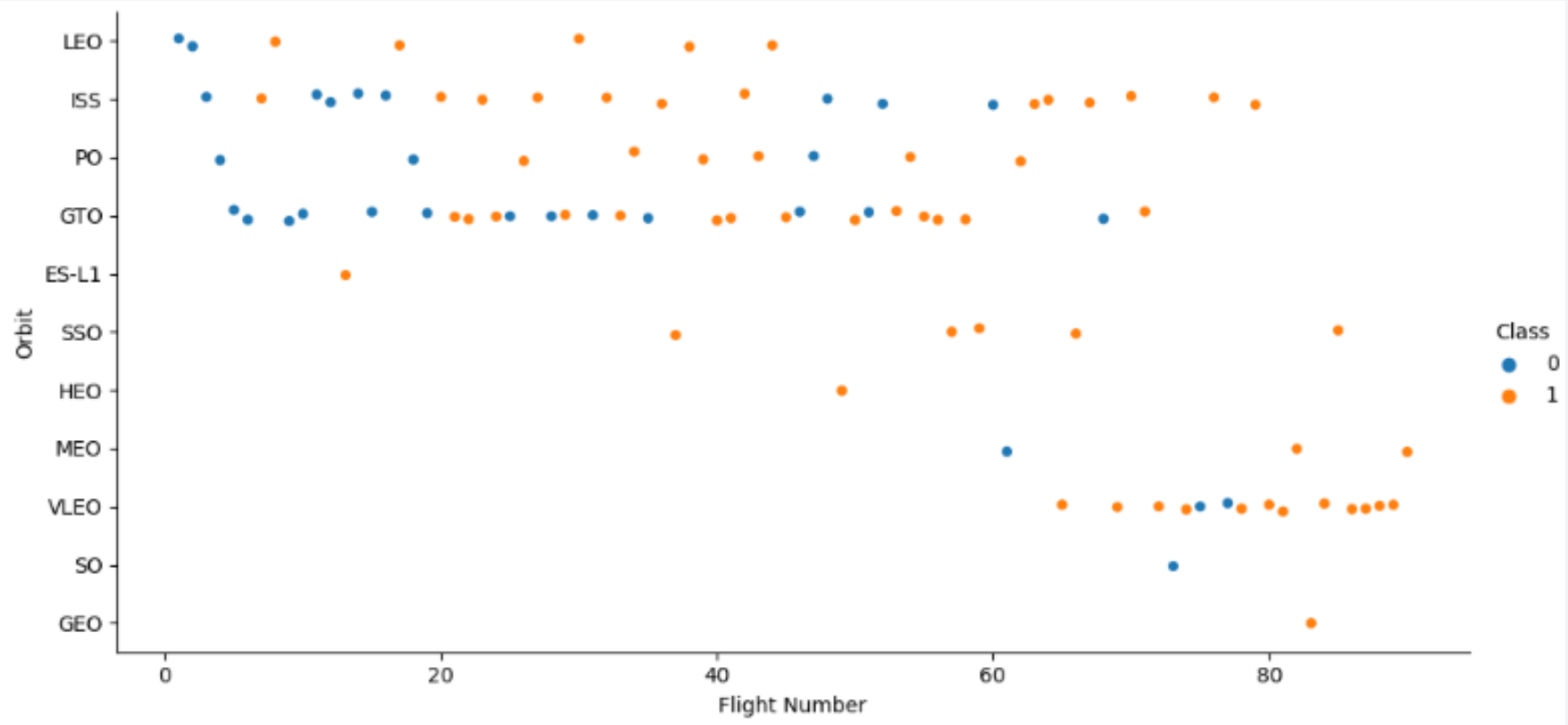
- There are no rockets launched for heavy payload mass(greater than 10000) for the VAFB-SLC launchsite

Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO have a higher success rate

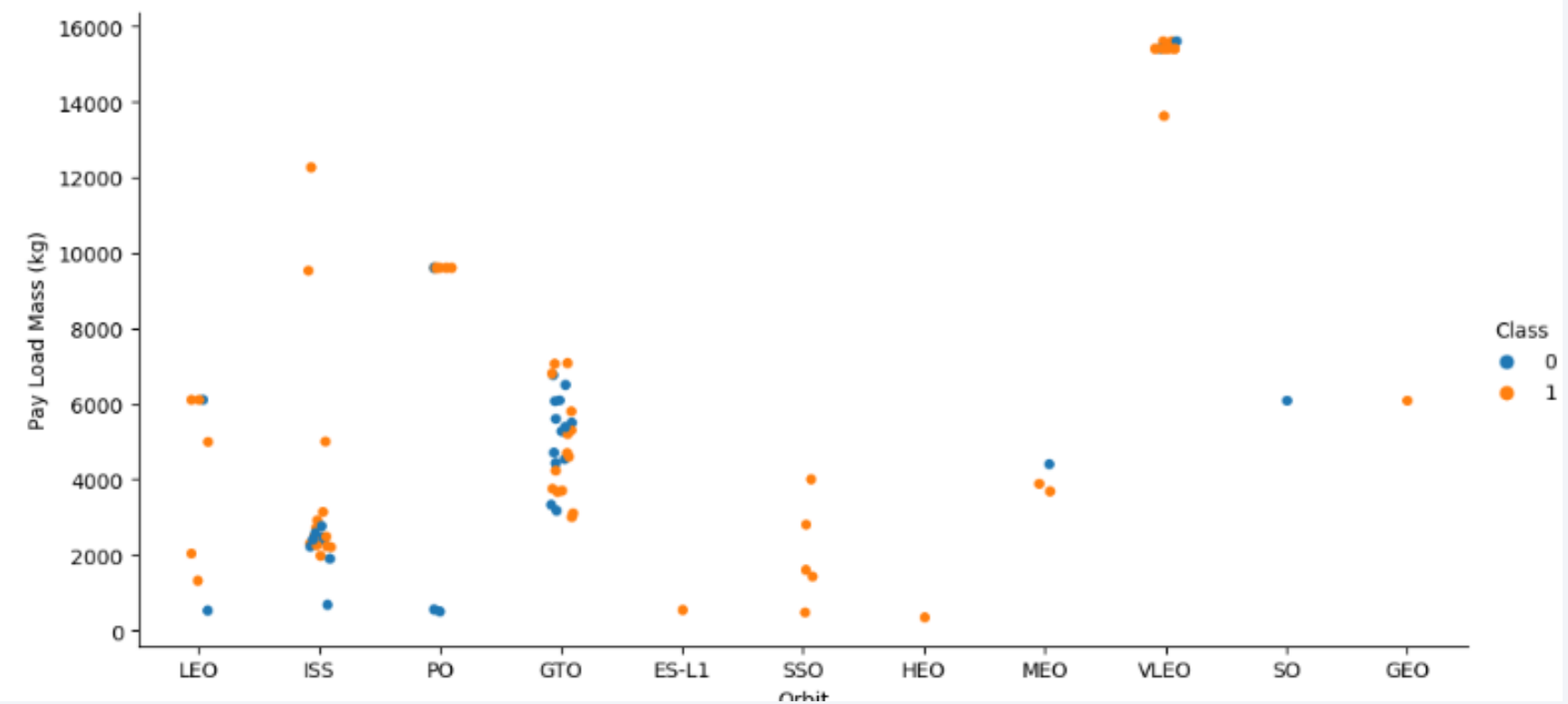


Flight Number vs. Orbit Type



- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

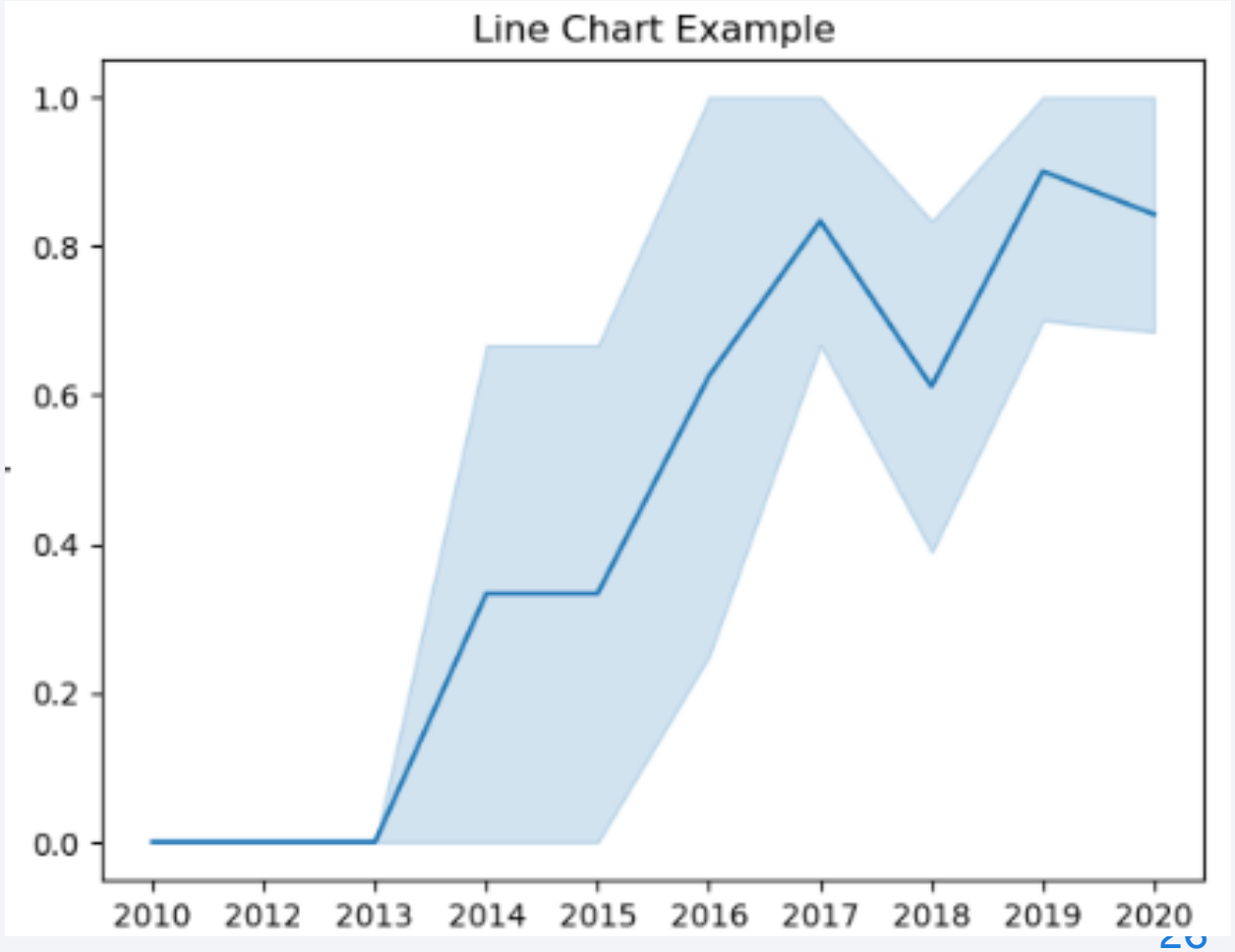
Payload vs. Orbit Type



- The graphs shows a positive relationship between a payload mass and success rate for all orbits except for GTO and SO.
- In the orbit GTO there is no relationship between the success rate and payload mass.

Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- There is a steady rise in success rate since 2013
- There was a sharp decline between 2018 and 2019 which recovered quickly within the same period



All Launch Site Names

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

- I selected all my Launch site names and used ‘Distinct’ to print only unique records.

Launch Site Names Begin with 'CCA'

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
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 attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Selected all records with a condition using the ‘WHERE” clause to select only the records ending with CCA by using the “LIKE” wildcard.

Reduced the number of records to 5 using “LIMIT”.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

Out[29]:

Total Payload
48213

- Calculated the total payload mass carried by boosters launched by NASA (CRS) using “SUM” and the “LIKE” wildcard to select the ones launched by NASA(CRS)

Average Payload Mass by F9 v1.1

Out[21]:

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

- Calculated the average using “AVG” function from booster version F9 v1.1 using the “where” clause and the “LIKE” wildcard.

```
%sql Select AVG("PAYLOAD_MASS__KG_") from SPACEXTABLE  Where "Booster_Version" like "F9 v1.1%"
```

First Successful Ground Landing Date

Out[38]:

Min(Date)
2015-12-22

- Selected the first date using “Min” function with a condition of the Landing outcome being a success using the “WHERE” clause,

```
%sql Select Min(Date) from SPACEXTABLE Where "Landing_Outcome" = 'Success (ground pad)'
```

Successful Drone Ship Landing with Payload between 4000 and 6000

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

- Selected records “WHERE” Landing outcome was a success and payload mass was greater than 4000 and less than 6000

Total Number of Successful and Failure Mission Outcomes

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

- Selected Mission outcomes and count their occurences using “count” and then group them using “group by”

```
%sql Select "Mission_Outcome", count("Mission_Outcome") from SPACEXTABLE group by "Mission_Outcome"
```

Boosters Carried Maximum Payload

- Selected Booster version and payload mass “WHERE” we had max payload mass on each booster version.
- Used “Max” function in a subquery to select maximum value for each booster version.

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600


```
: l Select "Booster_Version", "PAYLOAD_MASS__KG_" from SPACEXTABLE where "PAYLOAD_MASS__KG_" = (Select Max("PAYLOAD_MASS__KG_") fro
```

2015 Launch Records

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Used “substr” to etraxt the month and year and then a wild card “like” to output where the year was 2015.

```
%ELECT substr(Date, 6, 2) "Month", substr(Date, 0, 5) "Year",   "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEX
```

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

Count(Landing_Outcome)	Date	Landing_Outcome
10	2012-05-22	No attempt
5	2016-04-08	Success (drone ship)
5	2015-01-10	Failure (drone ship)
3	2015-12-22	Success (ground pad)
3	2014-04-18	Controlled (ocean)
2	2013-09-29	Uncontrolled (ocean)
2	2010-06-04	Failure (parachute)
1	2015-06-28	Precluded (drone ship)

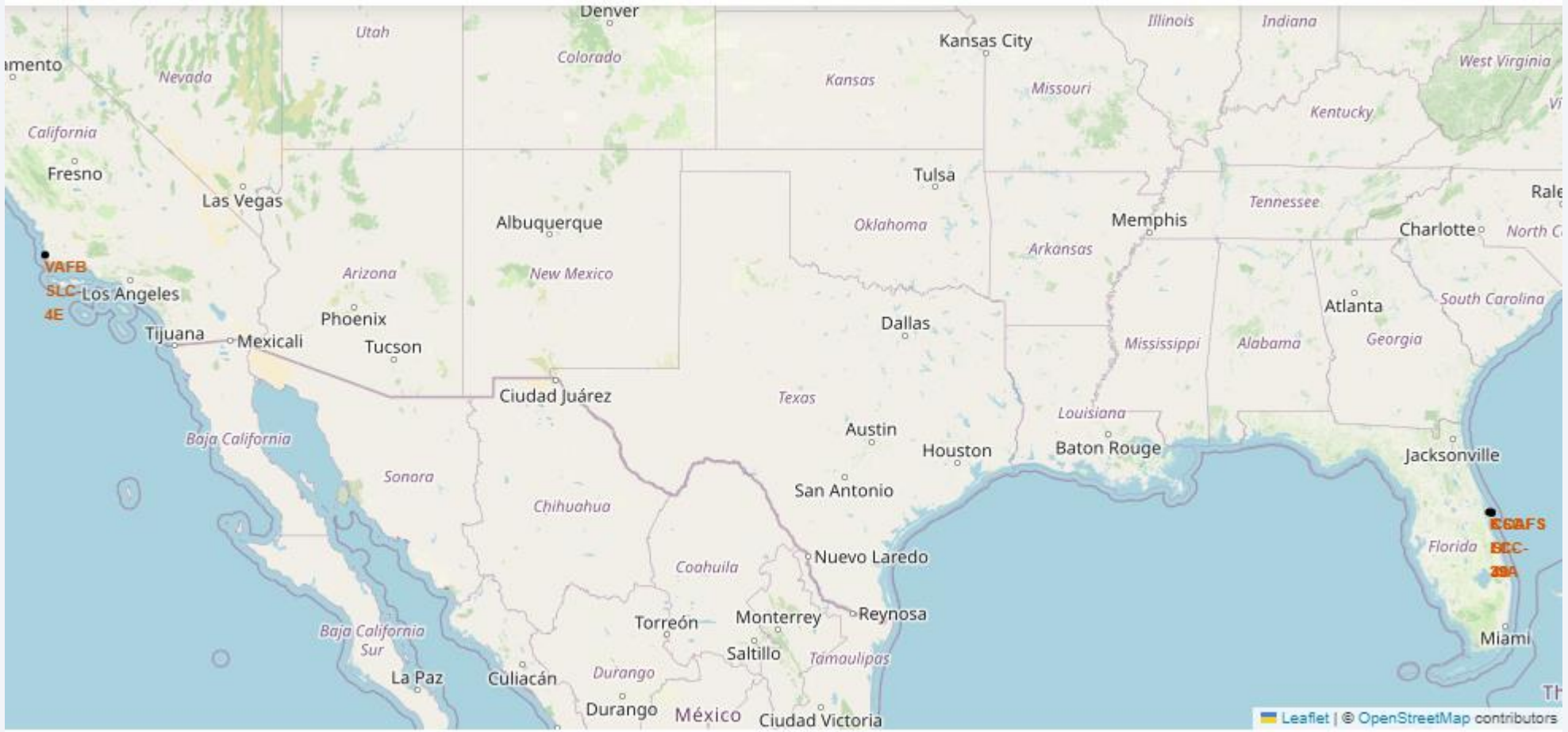
```
In [20]: PACEXTABLE where (date BETWEEN '2010-06-04' and '2017-03-20') group by  "Landing_Outcome" order by Count( "Landing_Outcome") DESC
```

- Used “Count” to get total landing outcomes for each date using “group by” and sorted results using “order by”

Section 3

Launch Sites Proximities Analysis

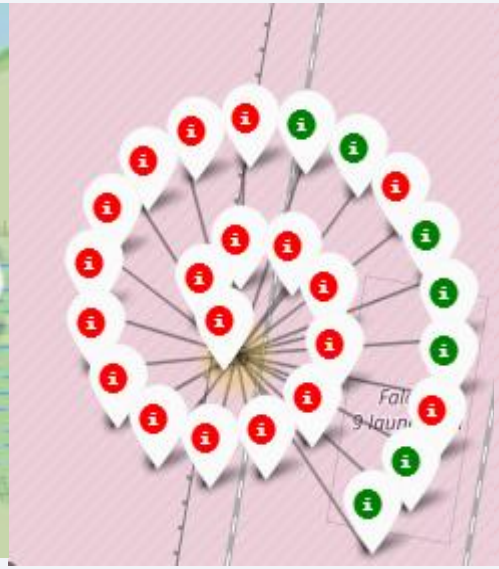
Launch Sites Locations



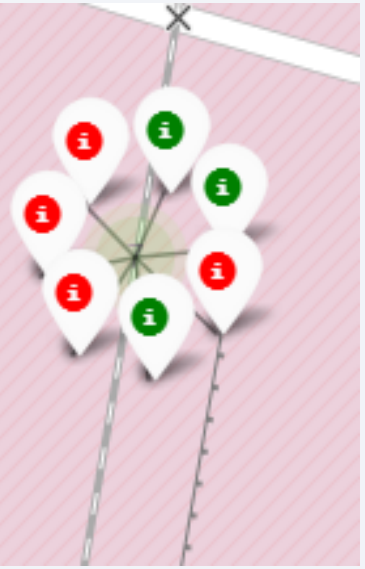
- All launch sites are in very close proximity to the coast and they are close to the equator as well.

Success/failed Launches on each Location

VAFB SLC-4E



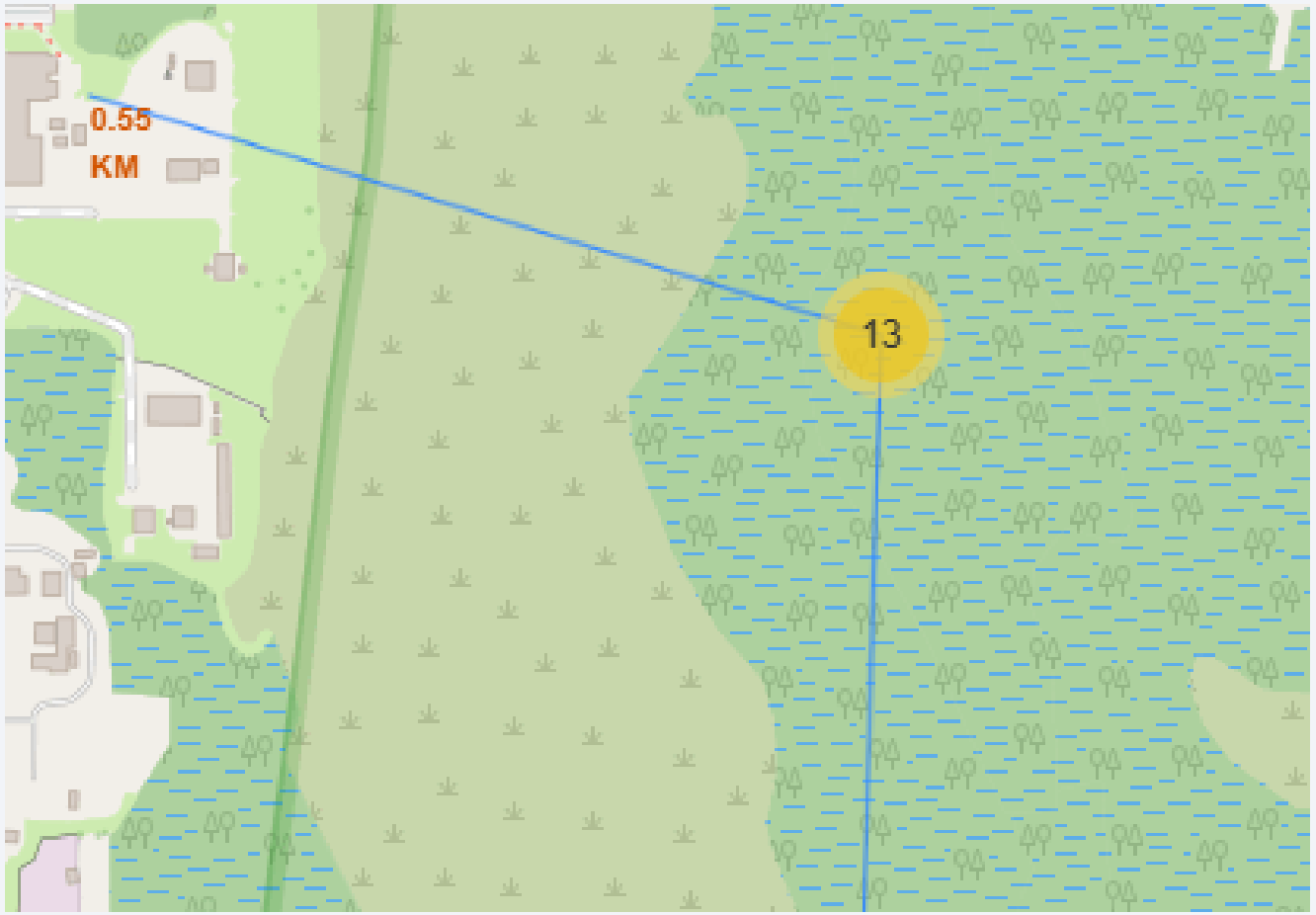
KSC LC-39A

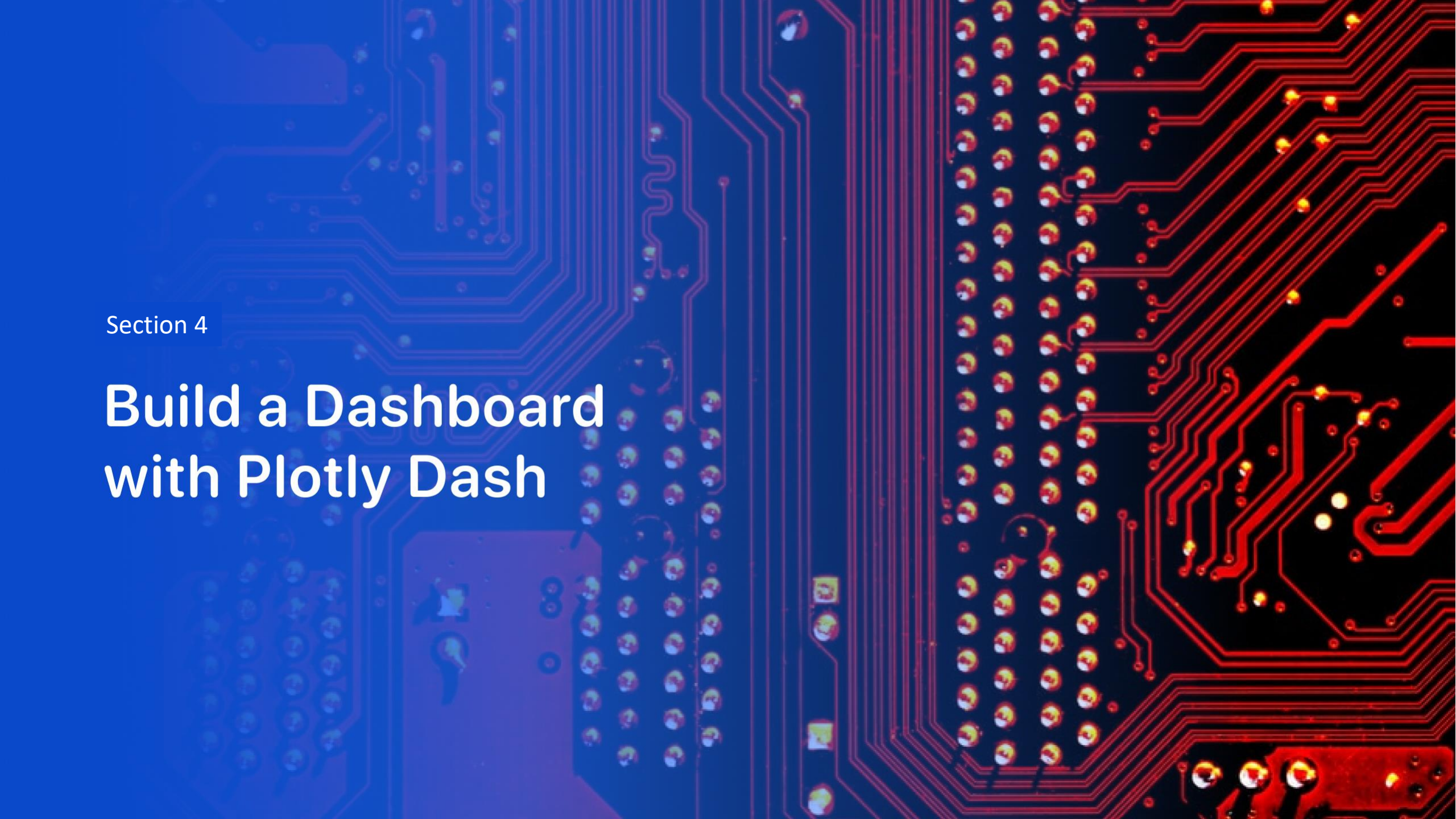


CCAFS LC-40

- On CCAFS LC-40 we had a few launches, hence little success rate.
- On VAFB SLC-4E we had more success rate as compared to other sites
- On KSC LC-39A we had more failure than success rate, this was probably the first site to used for launching.

Site Distance to its Proximities




Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- 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>

- 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

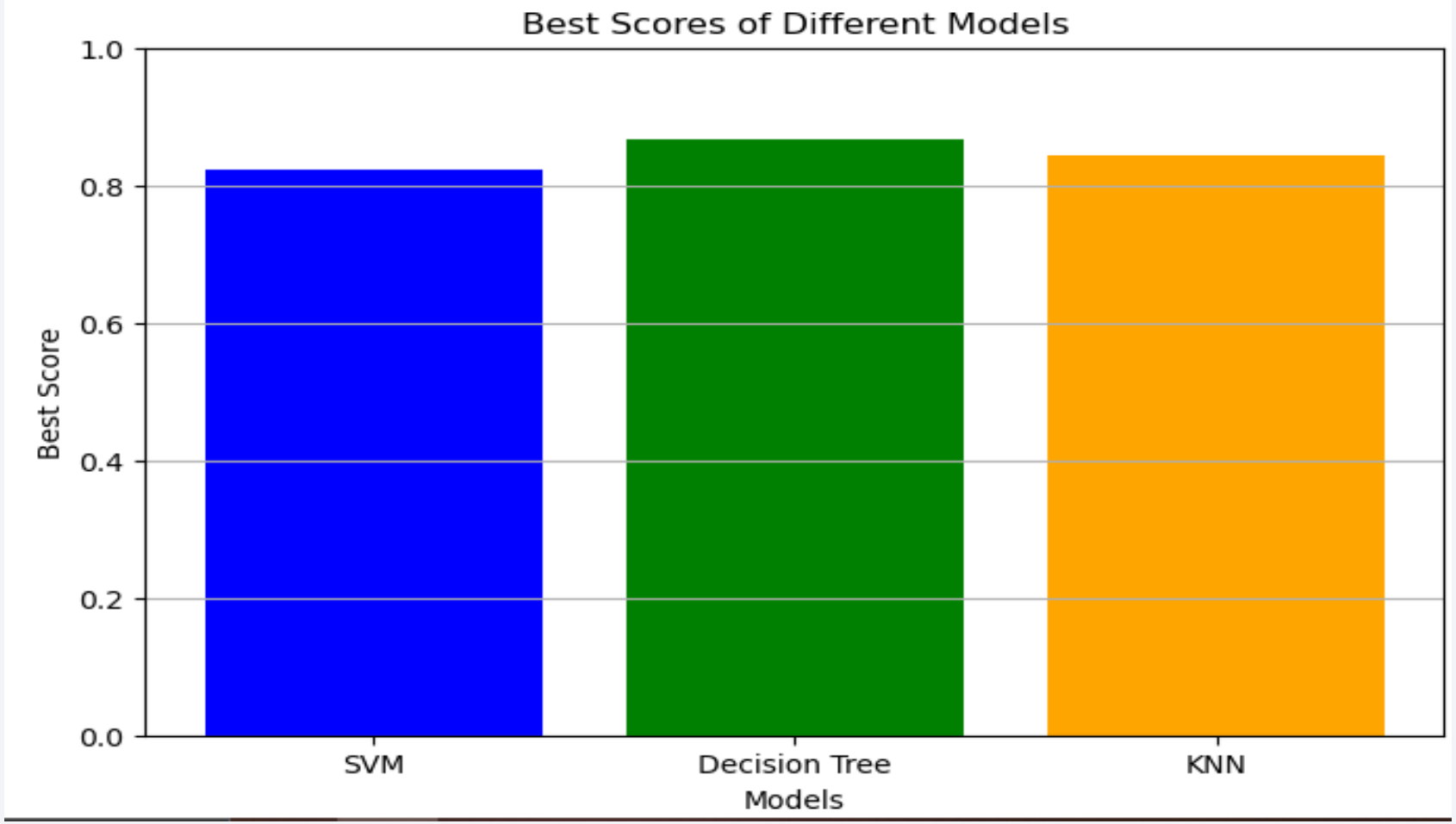
<Dashboard Screenshot 3>

- 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

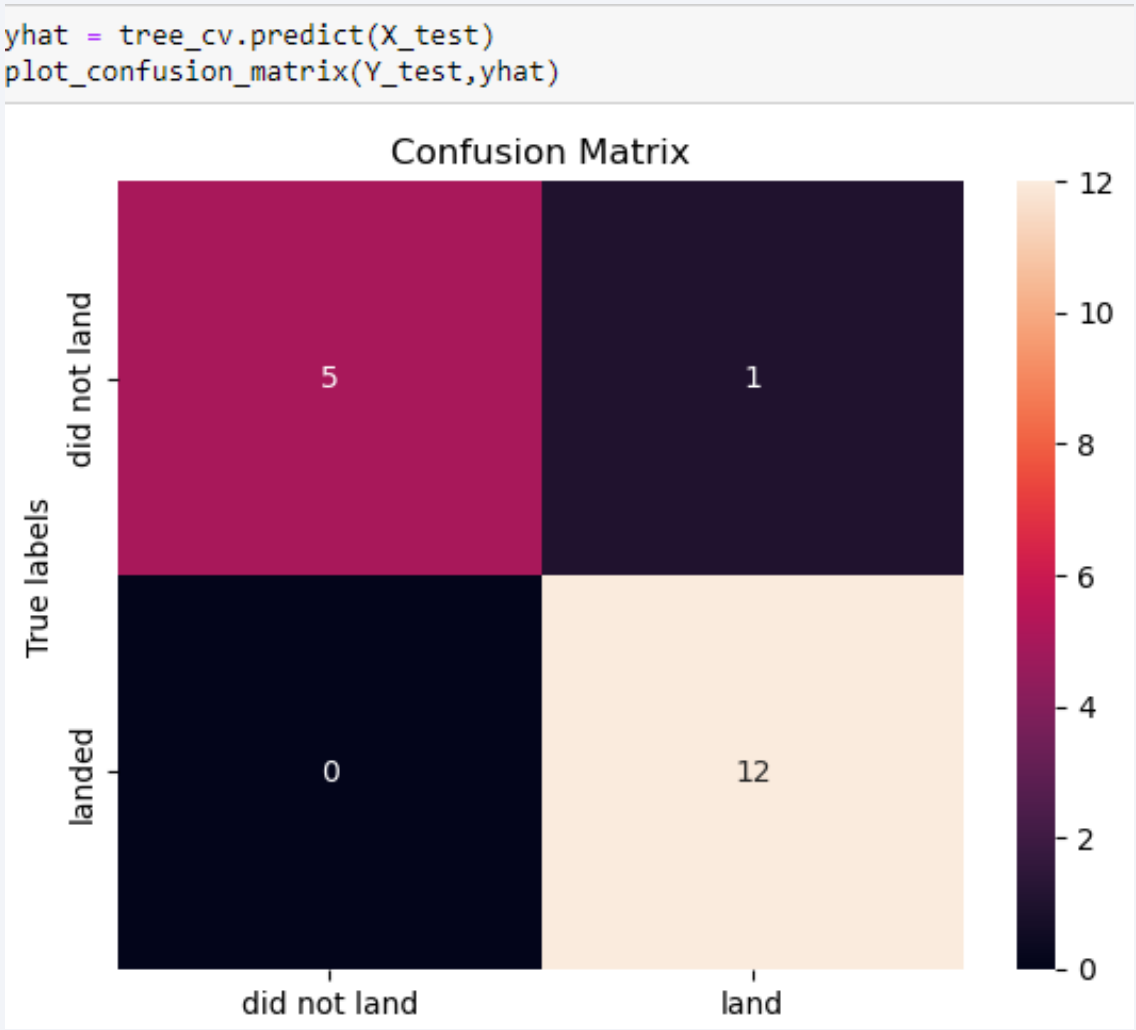


- Decision tree is the most acurate

Confusion Matrix

- The matrix shows very few misclassified instances (0 and 1) and more correctly classified instances (5 and 12)
- This shows a high accuracy rate for the model

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- The first few flights were unsuccessful and the success rate increased with flight number, a higher payload of 8000 increased chances of success.
- There was a higher success rate in ES-L1, GEO, HEO and SSO orbits
- The success rate was oon a steady rise since 2013
- Decision Tree is the best model classifier for this project

Thank you!