IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rohan Jagdish Sali
2nd July 2025

Outline

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

Executive Summary

- In this capstone project , We are developing a model which will predict if the SpaceX Falcon9 first stage will land successfully or not based on various parameters related to First Stage of Rocket.
- We are performing Data Collection , Data Pre-processing and EDC on data using SQL and Visualization.
- Also, We are visualizing Launch sites on map using Folium and Preparing a Dashboard for Data Visualization using Plotly and Dash.
- The Graphical Visualizations shows that some features of rocket landings are correlated with successful landing outcome.
- We are building several ML models to find best for predicting landing outcome i.e. Success or Failure.

Introduction

- SpaceX advertises Falcon9 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 SpaceX can reuse the first stage. 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.
- We are mainly addressing a problem that whether Falcon9 first stage successfully lands or not?
- We are performing a data analysis to gather the efficient data which to used to predict if Falcon9 lands successfully or not .
- We are creating a machine learning classification model which will analyze the features and predict if Falcon9 first stage makes up to successful landing or not.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data is collected in two ways : Using SpaceX API and Web Scrapping of Wikipedia.
  - There are several API provided by SpaceX which gives Json data related to Space Operations performed and their details in timely manner.
  - Wikipedia is also storing all the information about Space Missions and can be scraped for our requirements.
- Perform data wrangling:
  - Data is consisting missing values , unwanted values , unexpected types.
  - These Missing Values can be replaced with appropriate values or removed.

Methodology (cont..)

Executive Summary

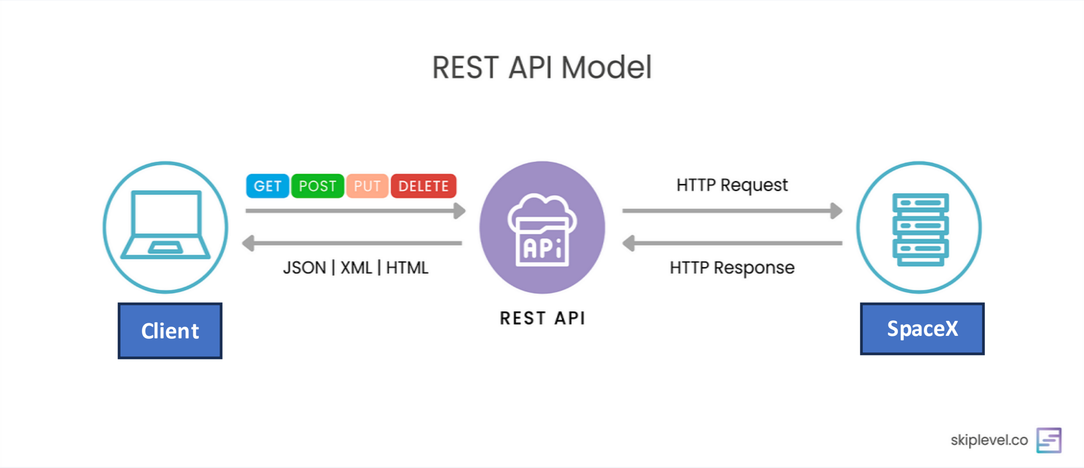
- Perform exploratory data analysis (EDA) using visualization and SQL:
  - Visual Graphs are used to analyze correlations between different parameters.
- Perform interactive visual analytics using Folium and Plotly Dash:
  - Folium is used to plot features graphically on World map.
  - Plotly and Dash are used to create an app Dashboard for Visualizations.
- Perform predictive analysis using classification models:
  - Machine learning models are trained on refined data.
  - Models like Logistic Regression , Decision Trees , SVM , KNN , etc. are trained on analyzed to make predictions and one with good accuracy score is selected.

Data Collection

- Data Collection is first and important step in Data Analysis.
- We are collecting data for Falcon9 first stage landings from two different ways:
  - SpaceX APIs
  - Web Scraping
- The APIs are available on SpaceX website to fetch the Timely data of Rocket launches for different space projects
- Wikipedia keeps all the information about Space missions with reliable data thus these pages could be Scraped to get data.

Data Collection – SpaceX API

- Using Requests library we made a .get() request to the API.
- It then sent response 200 along with the JSON data.
- The data is then normalized using Pandas and converted in DataFrame.
- GitHub URL of the completed SpaceX API calls notebook : [L1_SpaceX_Data_Collection_using_API]

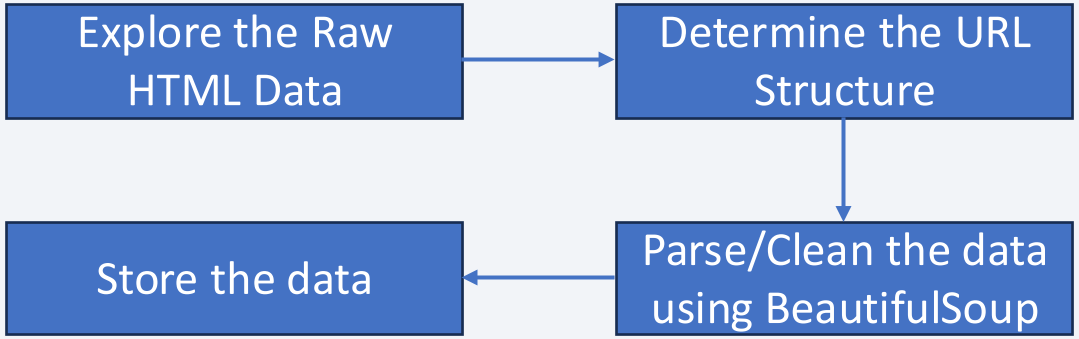


	static_fire_date_utc	static_fire_date_unix	tbd	net	window	rocket	success	details	crew	ships	capsules
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	False	0.0	5e9d0d95eda69955f709d1eb	False	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]
1	None	NaN	False	False	0.0	5e9d0d95eda69955f709d1eb	False	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first stage	[]	[]	[]

Data Collection - Scraping

- Firstly, we sent requests.get() request to the URL of webpage.
- Then html of webpage sent by website is read as text.
- Beautiful Soup is used to read data in organized form.
- GitHub URL of the completed web scraping notebook : [L2_SpaceX_Data_Collection_using_WebScraping]

Web Scrapping Structure :



Web Scraped Data of Falcon9:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.07B0003.18	Failure	4 June 2010	18:45
1	1	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.07B0003.18	Failure	4 June 2010	18:45
2	2	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.07B0004.18	No attempt\n	8 December 2010	15:43
3	3	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.07B0005.18	No attempt	22 May 2012	07:44
4	4	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.07B0006.18	No attempt\n	8 October 2012	00:35

Data Wrangling

- Data collected from API contains data of Falcon1 and Falcon9 , but we want data related to Falcon9 only so we will remove all the rows containing data related to Falcon1 first stage.
- Our data having some missing values denoted by NaN in dataframe of pandas viz 5 missing values for Payload Mass and 26 for Landing Pad. So, we will remove all the rows with missing values for Payload Mass.
- Then we will reset all the Flight numbers starting with 1.
- Finally, we will create new columns “Class” which contains 1 value for successful landing and 0 value for failure in landing using column “Outcome”.
- GitHub URL of completed data wrangling : [L3_SpaceX_Data_Wrangling]

Final DataFrame with 90 rows and 17 columns :

Column names are : FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude.

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	Landir
1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	
2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	
3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	
4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	
5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	

EDA with Data Visualization

- In EDA with Visualization, we find relations between different data features.
- In our project we plotted Scatter Graph plots and Bar Plots to understand correlations between different features.
- We plotted Scatter plots to see :
  - how the FlightNumber and PayloadMass variables would affect the launch outcome.
  - how the FlightNumber and LaunchSite variables would affect the launch outcome.
  - how the PayloadMass and LaunchSite variables would affect the launch outcome.
  - how the FlightNumber and OrbitType variables would affect the launch outcome.
  - how the PayloadMass and OrbitType variables would affect the launch outcome.
- We plotted Bar graph and Line graph to see :
  - Success rate over OrbitType using Bar Plot.
  - Success rate over Year using Line Plot.
- GitHub URL of completed EDA with data visualization notebook :

  L5_SpaceX_EDA_with_Visualization

EDA with SQL

- The Data Analysis is done using SQL Queries and patterns in data are found.
- Data is saved to database using sqllite3 and queries are made using connection and cursor object of sqlite3.
- The Maximum Payload Mass for different boosters and customers are analyzed. Booster Versions with Maximum Payload Mass are queried.
- Mission Outcome are grouped to select their count. Data is queried over time to get insights of success and failure attempts over time and with selected ranged Payload Mass.
- Date of First Successful Landing is searched.
- GitHub URL of completed EDA with SQL notebook : L4_SpaceX_EDA_with_SQL

Build an Interactive Map with Folium

- In this project, we performed three main tasks :
  - Mark all launch sites on a map.
  - Mark the success/failed launches for each site on the map.
  - Calculate the distances between a launch site to its proximities.
- Map is initially created using folium.Map() function with specified co-ordinates and zoom_start, then new elements are added to it using .add_child() with specified child object in parenthesis.
- A Circle and Popup Marker is added to each launch site co-ordinates.
- Marker Clusters are used to map all the success and failures for the launch sites.
- Marker for distance along with lines are added for nearest proximities.
- GitHub URL of completed interactive map with Folium map : [L6_SpaceX_Visual_Analytics_with_Folium]()

Build a Dashboard with Plotly Dash

- We created a Dashboard for SpaceX launch records to present analytics visually.
- We added a Dropdown to select either All sites or Particular site.
- Based on selected dropdown option we display Pie chart for Successful launches for All sites or Selected Site.
- We added a Range Slider to input the range for Payload Mass.
- Based on selected Sites/All Sites and Selected Payload Mass Range we plot the Scatter Plot for Payload mass vs Class over different Booster Versions.
- Dashboard contains one heading below that 4 Divs one by one respectively for Dropdown Menu , Pie Chart , Range Slider and Scatter Plot.
- GitHub URL of completed Plotly Dash lab : L7_SpaceX_App_Dashboard_using_Plotly

Predictive Analysis (Classification)

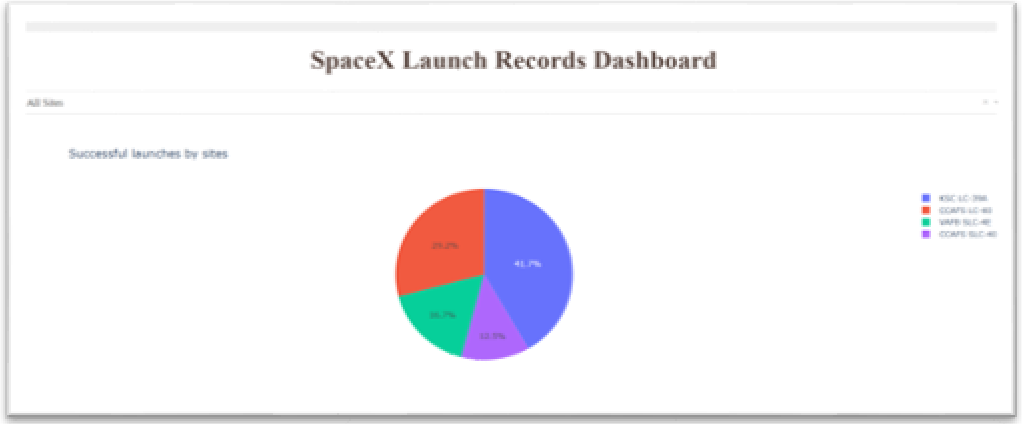
- We have created four different classification models to predict the landing outcome as Success(1) or Failure(0) based on different parameters variables.
  - Logistic Regression Classifier
  - Support Vector Machine Classifier
  - Decision Tree Classifier
  - K – Nearest Neighbor Classifier
- We start with scaling the data using standard scaler then dividing Data into Training and Testing Splits with test size as 20%.
- Then we use GridSearchCV for each algorithm to find best hyperparameters for each algorithm.
- Finally best models are compared over accuracy using confusion matrix and .score() method on test data.
- GitHub URL of completed predictive analysis lab : L8_SpaceX_Prediction_Analysis_(ML)

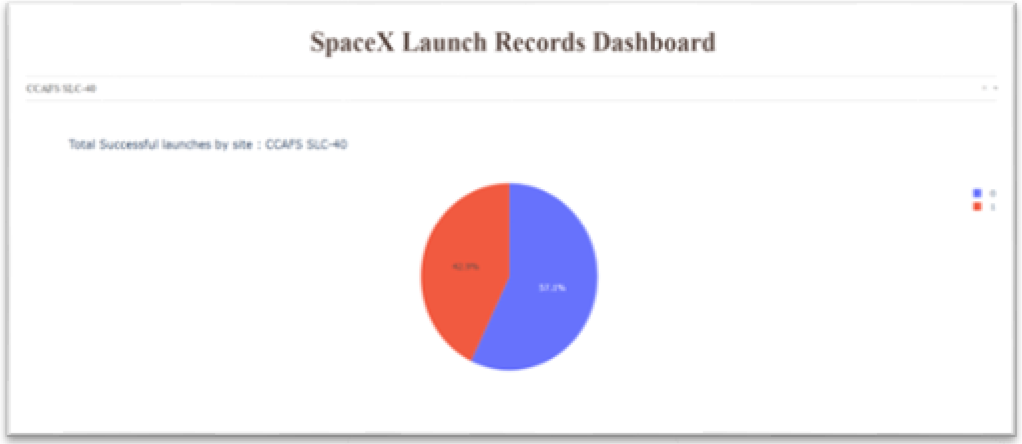
Results

- Exploratory data analysis results :
  - First successful landing is achieved on date : 2015-12-22
  - Boosters with success in drone ship and have payload mass between 4000 and 6000 are F9 FT B1022 , F9 FT B1026 , F9 FT B1021.2 , F9 FT B1031.2 .
  - Booster versions F9 B5 series have carried maximum payload mass.
  - Avg Payload Mass carried by F9 v1.1 Booster is around 2535 kg.
- Predictive analysis results :
  - All the models are giving same test accuracy as 83.33% also same True Positive Rate , Recall and Precision for True labels.
  - Decision Tree Classifier is providing highest train accuracy as 88.75% and it could be because of overfitting while other models are giving train accuracy around 85%.
  - Although there is not too much difference between train and test accuracy so , Decision Tree Classifier could be better model for classification.

Results

- Interactive analytics demo :



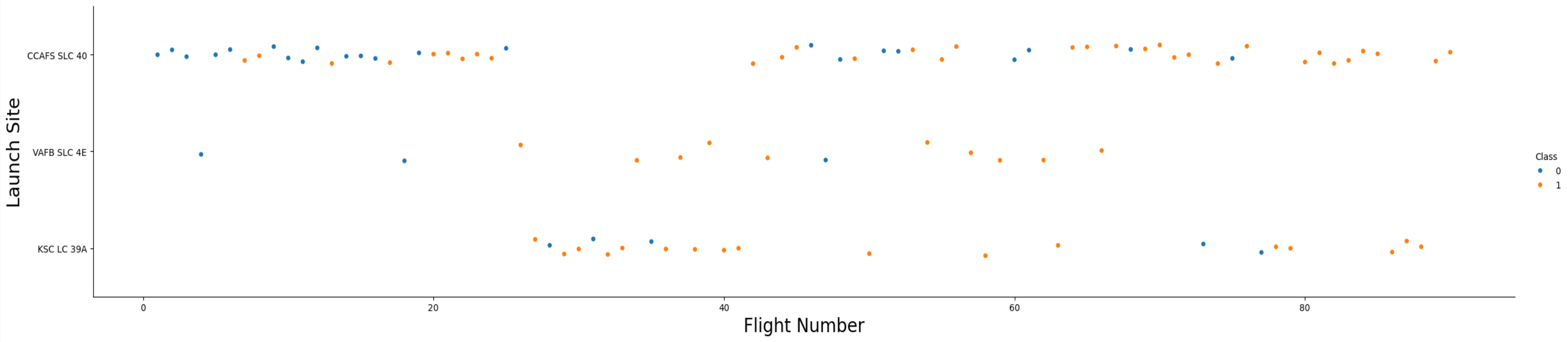


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

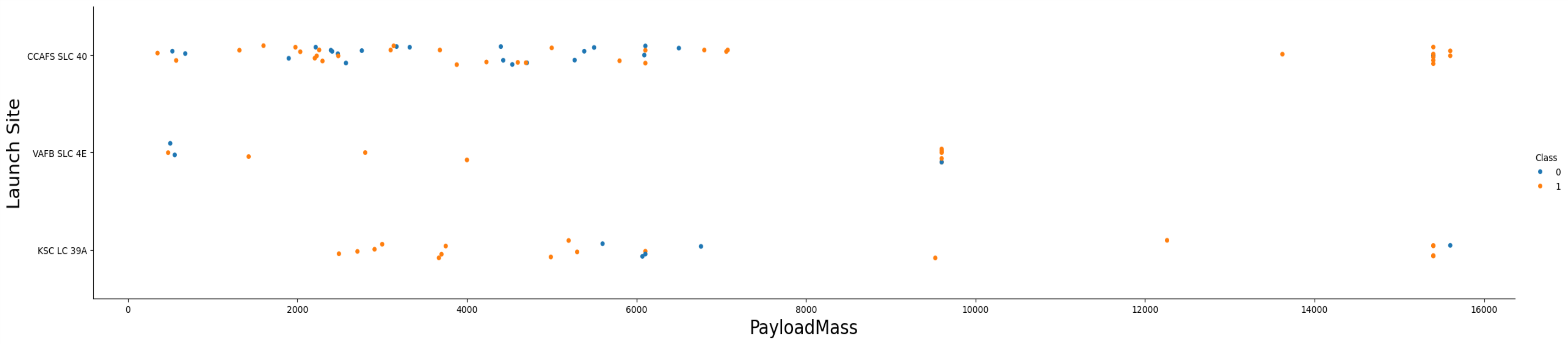
- For each of Launch site as Flight Number increases, the first stage is more likely to land successfully.
- VAFB SLC 4E showing good results in less number of flights than others.



Scatter plot of Flight Number vs. Launch Site

Payload vs. Launch Site

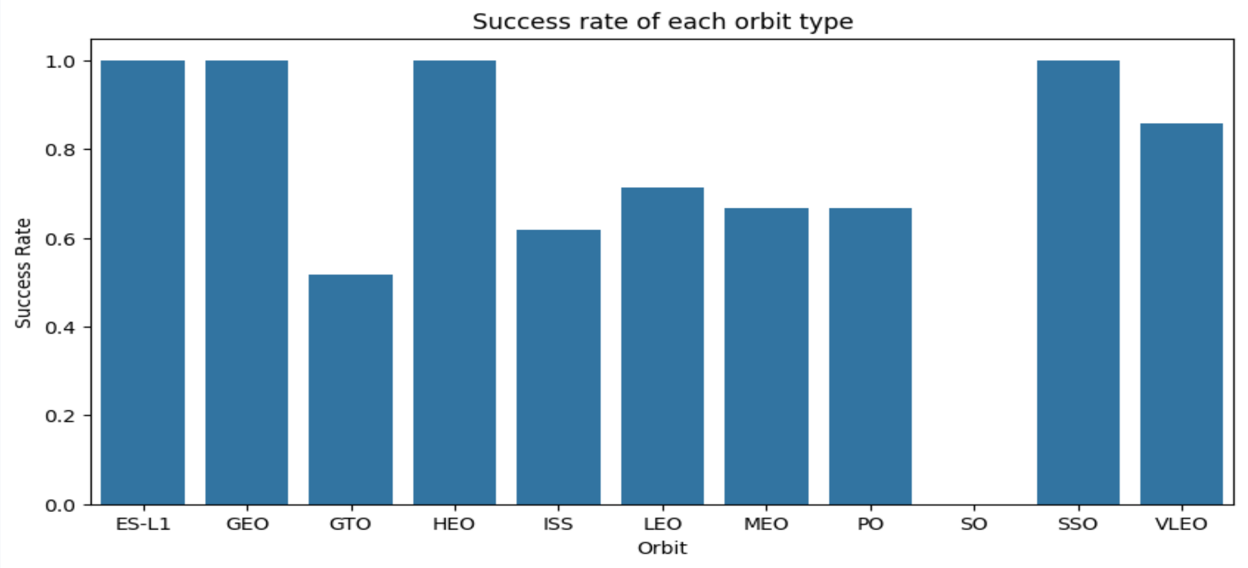
- For Launch Site CCAFS SLC 40 as Payload Mass increases, the first stage is more likely to land successfully.
- But for Launch Sites VAFB SLC 4E and KSC LC 39A there is no significant relation for landing outcome with increasing Payload mass.



Scatter plot of Payload Mass vs. Launch Site

Success Rate vs. Orbit Type

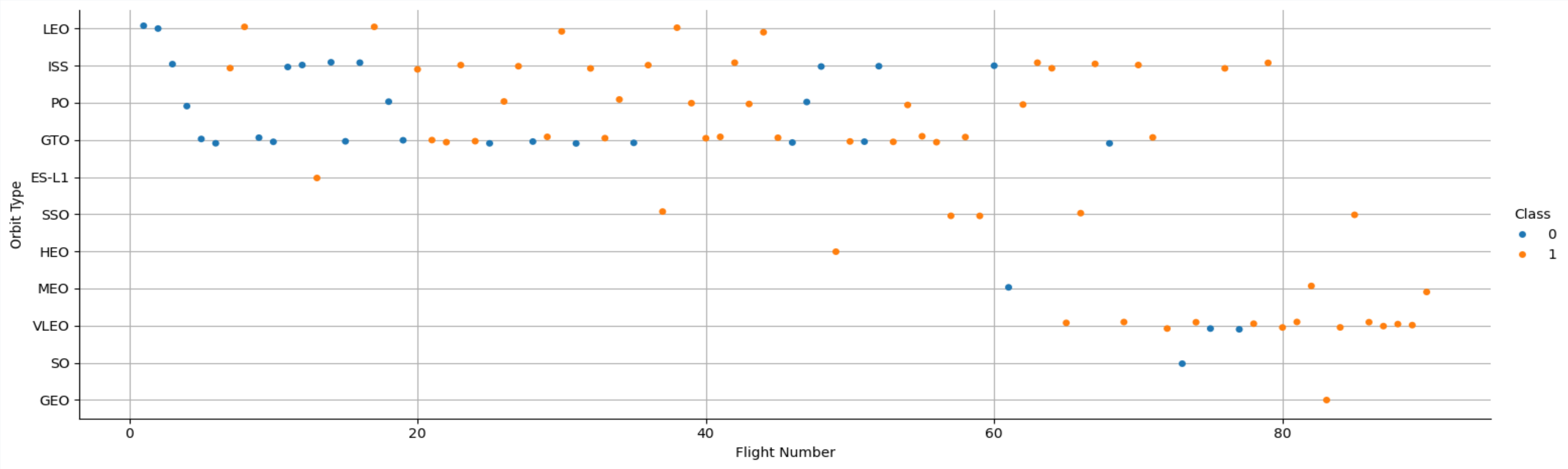
- Flights to orbits ESL1 , GEO , HEO and SSO shows all successful landing of first stage of Falcon9 , while flight to orbit SO shows no success in landing.
- For Flights to other orbits shows success rate equal or more than 0.5.



Bar plot of Success rate vs. Orbit type

Flight Number vs. Orbit Type

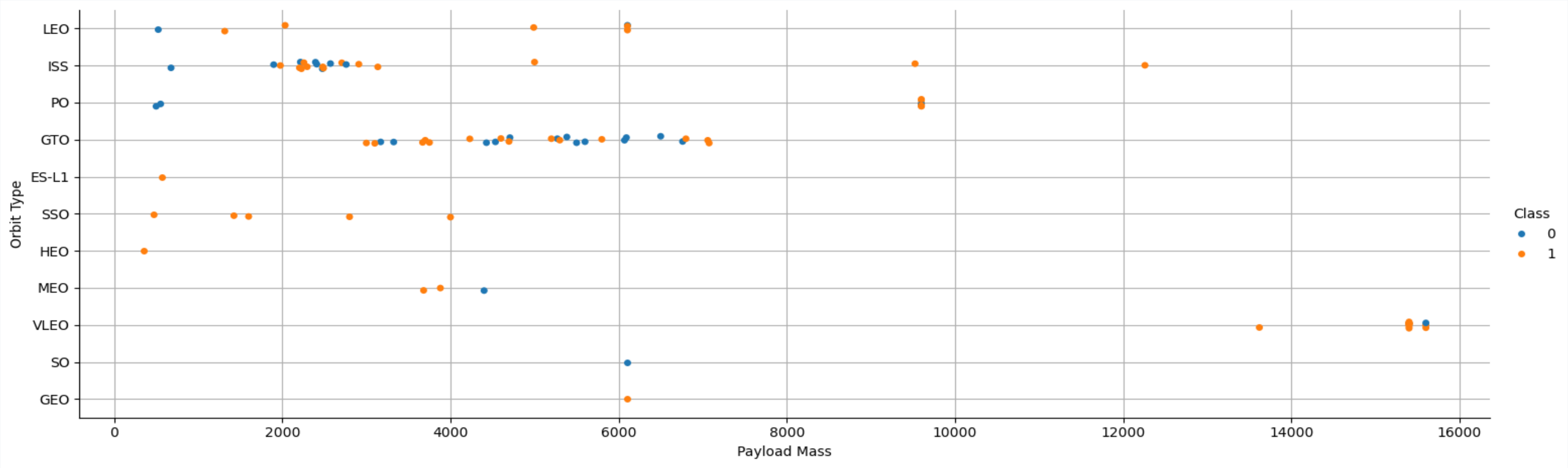
- Flights to orbits LEO , VLEO and MEO shows relation between number of flights for successful landing of first stage of Falcon9.
- Flights to other orbits do not show any or much relation for successful landing with number of flights.



Scatter plot of Flight Number vs. Orbit type

Payload vs. Orbit Type

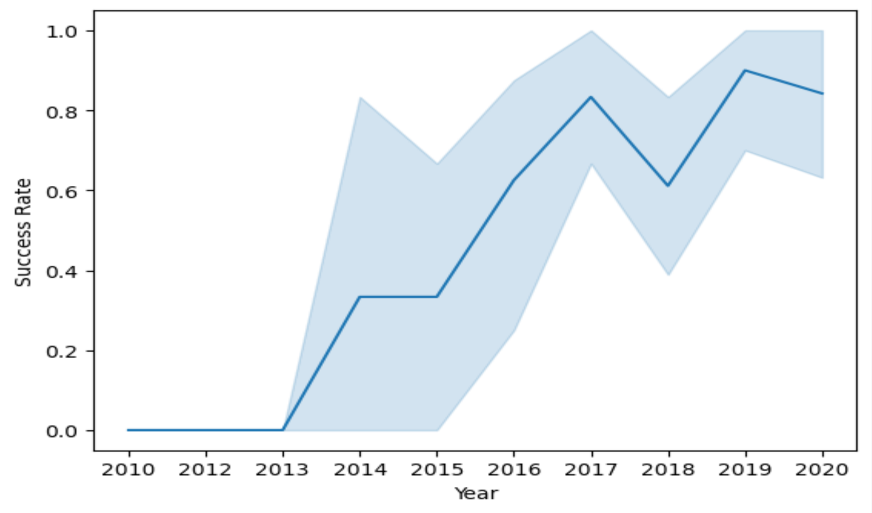
- Flights to orbits LEO , ISS and PO shows increase successful landing of first stage with increasing Payload Mass.
- For Flights to other orbits do not show any relation for payload mass and orbit type for successful landings.



Scatter plot of Payload Mass vs. Orbit type

Launch Success Yearly Trend

- Graph shows that Success rate is increasing continuously from year 2013.
- It also shows that decrease in success rate from year 2017 to 2018.
- After 2018 there is again increase in success rate till 2020.



Line plot of Year vs. Success Rate

All Launch Site Names

- We used Distinct method to find unique Launch Site names from a SPACEXTABLE table using SQL query as :

  %sql select distinct "Launch_Site" from SPACEXTABLE

```
In [12]: %sql select distinct "Launch_Site" from SPACEXTABLE

 * sqlite:///my_data1.db
Done.
Out[12]:
```

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

Launch Site Names Begin with 'CCA'

- Launch Sites with names begin “CCA” are queried using “like” clause on launch site column by passing value as “CCA%”.
- Limit 5 clause is used to get only 5 records.

%sql select * from SPACEXTABLE where "Launch_Site" like "CCA%" limit 5

```
In [77]: %sql select * from SPACEXTABLE where "Launch_Site" like "CCA%" limit 5

 * sqlite:///my_data1.db
Done.
Out[77]:
```

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

Total Payload Mass by NASA (CRS)

- Total payload mass is calculated by using SUM() method on “Payload_Mass__kg_” columns and selecting only “NASA (CRS)” customer using WHERE clause.
- Total Payload Mass for NASA (CRS) is 45596.

%sql select SUM("PAYLOAD_MASS__KG_") as "Total Payload Mass by NASA (CRS) in kg" from SPACEXTABLE where "Customer" = "NASA (CRS)"

```
In [20]: %sql select SUM("PAYLOAD_MASS__KG_") as "Total Payload Mass by NASA (CRS) in kg" from SPACEXTABLE where "Customer" = "NASA (

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

Out[20]:

Total Payload Mass by NASA (CRS) in kg
45596

Average Payload Mass by F9 v1.1

- Average Payload Mass by F9 v1.1 is 2535 kg
- The Average Payload Mass is calculated using AVG() method.
- WHERE clause and like clause are used to select only booster levels with name starting with F9 v1.1.

%sql select ROUND(AVG("PAYLOAD_MASS__KG_"),2) as "Avg Payload Mass carried by F9 v1.1 Booster" from SPACEXTABLE where "Booster_Version" like "F9 v1.1%"

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

In [23]:
```
%sql select ROUND(AVG("PAYLOAD_MASS__KG_"),2) as "Avg Payload Mass carried by F9 v1.1 Booster" from SPACEXTABLE where "Boost
```

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

Out[23]:

Avg Payload Mass carried by F9 v1.1 Booster
2534.67

First Successful Ground Landing Date

- First Successful Ground Landing was achieved on 22^{nd} Dec 2015.
- The MIN() method is used to find earliest date for Successful landing.
- WHERE clause is used to select only “Success (ground pad)” as a Landing Outcome.

%sql select MIN("DATE") as "Date of First successful landing on Ground pad." from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)“.

```
In [35]: %sql select MIN("DATE") as "Date of First successful landing on Ground pad." from SPACEXTABLE where "Landing_Outcome" = "Suc

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

Out[35]:

Date of First successful landing on Ground pad.
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Successful Drone Ship Landings with Payload between 4000 and 6000 are : F9 FT B1022 , F9 FT B1036 , F9 FT B1021.2 , F9 FT B1031.2 .
- WHERE clause is used with and clause to select the Landing Outcome as “Success (drone ship)” and Payload Range as 4000 to 6000 kg. For Range “between .. and ..” clause is used.

%sql select "Booster_Version","Payload_Mass__KG_" from SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)" and "Payload_mass__Kg_" between 4000 and 6000

In [48]:

```
%sql select "Booster_Version","Payload_Mass__KG_" from SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)" and "Pay
```

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

Out[48]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

- There are total 100 successful missions and 1 Failure in flight mission outcomes in data.
- Group by clause is used to group “Mission Outcome” and COUNT() method is used to get count of Outcomes.

%sql select "Mission_Outcome",COUNT("Mission_Outcome") as "Outcome_Count" from SPACEXTABLE Group by "Mission_outcome"

```
In [73]: %sql select "Mission_Outcome",COUNT("Mission_Outcome") as "Outcome_Count" from SPACEXTABLE Group by "Mission_outcome"

 * sqlite:///my_data1.db
Done.
Out[73]:
```

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

Boosters Carried Maximum Payload

- Booster carried Maximum Payload are of F9 B5 series.
- We used sub-query to select maximum payload mass value for “Payload_Mass_kg_”.
- And Selected “Booster_Version” to get list of boosters.

%sql select "Booster_Version" from SPACEXTABLE where "Payload_Mass__kg_" in (select Max("Payload_Mass__kg_") from SPACEXTABLE)

```
In [74]: %sql select "Booster_Version" from SPACEXTABLE where "Payload_Mass__kg_" in (select Max("Payload_Mass__kg_") from SPACEXTABL
 * sqlite:///my_data1.db
Done.
Out[74]:
```

Booster_Version
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 used substr() method to select Month from column Date and used like clause to select year as “2015%”.
- Landing Outcome is selected as “Failure (drone ship)” using WHERE clause.

%sql select substr(Date, 6,2) as Month,"Booster_Version","Landing_Outcome","Launch_Site" from SPACEXTABLE where "Landing_Outcome"="Failure (drone ship)" and Date like "2015%"

In [78]:

```
%sql select substr(Date, 6,2) as Month,"Booster_Version","Landing_Outcome","Launch_Site" from SPACEXTABLE where "Landing_Out
```

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

Out[78]:

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

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

- Date is selected between 2010-06-04 and 2017-03-20 using WHERE clause.
- We used Group by clause to group "Landing Outcomes" and Count() method is used to get number of respective outcomes.
- We used Order by clause on count("Landing Outcomes") along with desc clause to order then in descending order.

%sql select "Landing_Outcome",count("Landing_Outcome") as "Landing Outcome Count" from SPACEXTABLE where "Date" between "2010-06-04" and "2017-03-20" group by "Landing_Outcome" order by count("Landing_Outcome") desc

```
In [92]: %sql select "Landing_Outcome",count("Landing_Outcome") as "Landing Outcome Count" from SPACEXTABLE where "Date" between "201

 * sqlite:///my_data1.db
Done.
Out[92]:
```

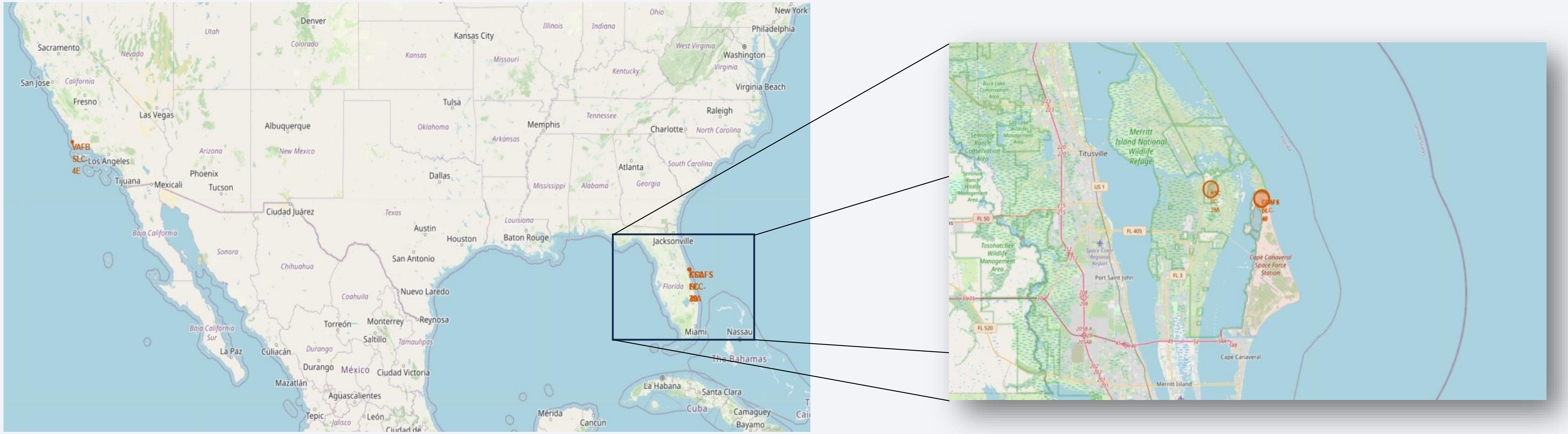
Landing_Outcome	Landing Outcome Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

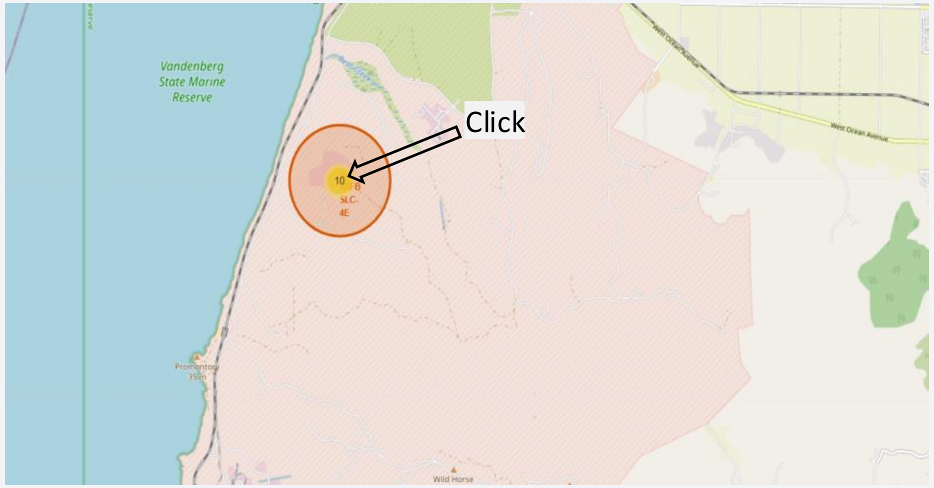
Locations of all Launch Sites on World Map

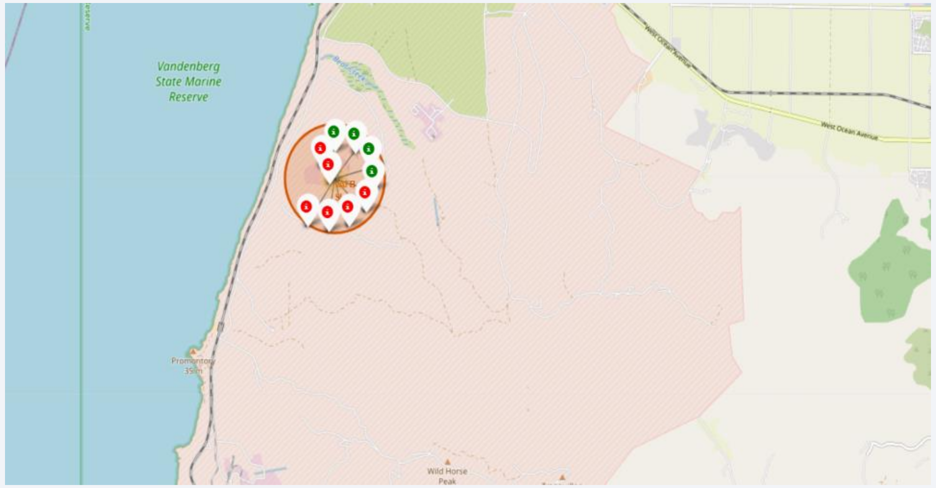
- There are 4 Launch Sites , Three of them are in east coast of US while one is on west coast.
- We added a Circle around each Launch Site with radius of 1km and Marker with Popup as name of Launch Site.



Success and Failed Launches for each site on Map

- For Successful Launch, green colored markers and for failed launch, red colored markers are added to map for each launch site.
- Marker Cluster is created for markers as many markers pointing to same location.
- By clicking on Cluster, We can see red (failure) and green (success) icons.



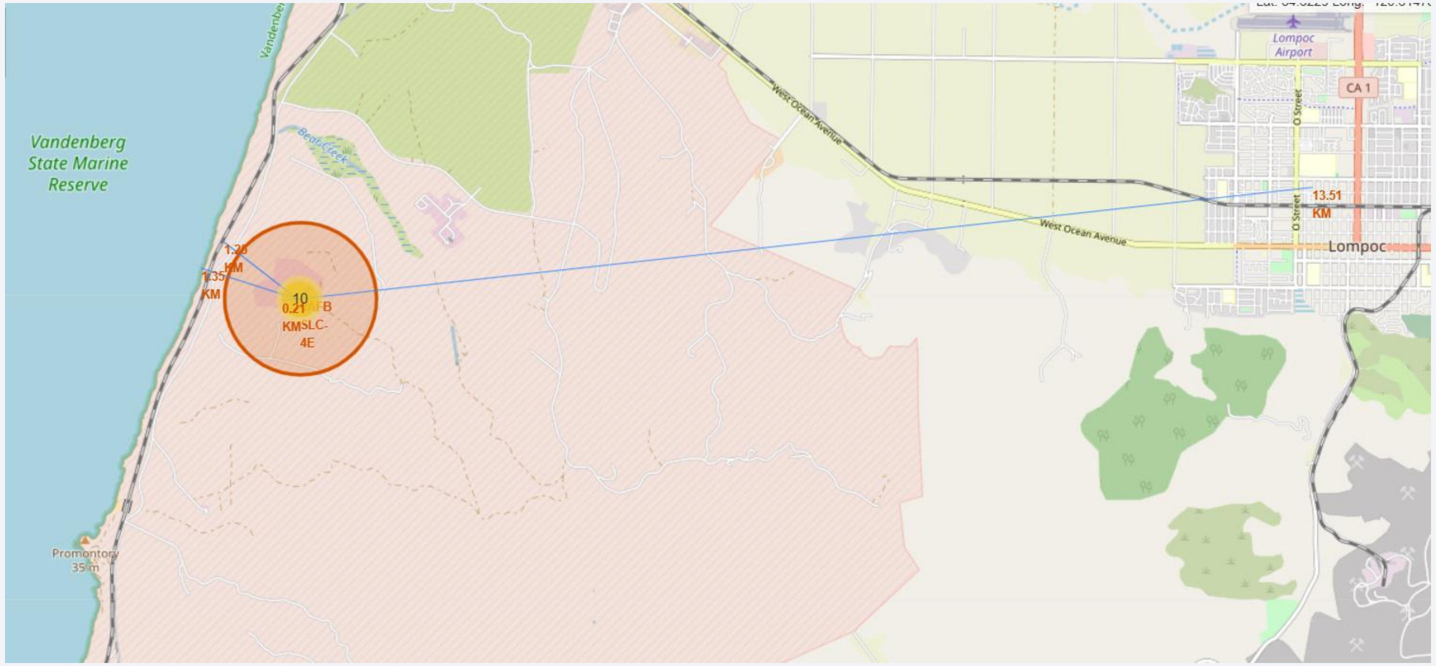


Mapping proximities from Launch Sites

- Distance for closest proximities like coastline, railway, highway and city is calculated using their coordinates and Launch Site coordinates.
- Distance Marker is added to respective points showing distance from launch site.
- Line is drawn from Launch site to the respective Proximity's Point.

<u>Distance from Launch Site:</u>

- Coast line : 1.35 km
- Railway line : 1.25 km
- Highway road : 0.21 km
- City : 13.51 km

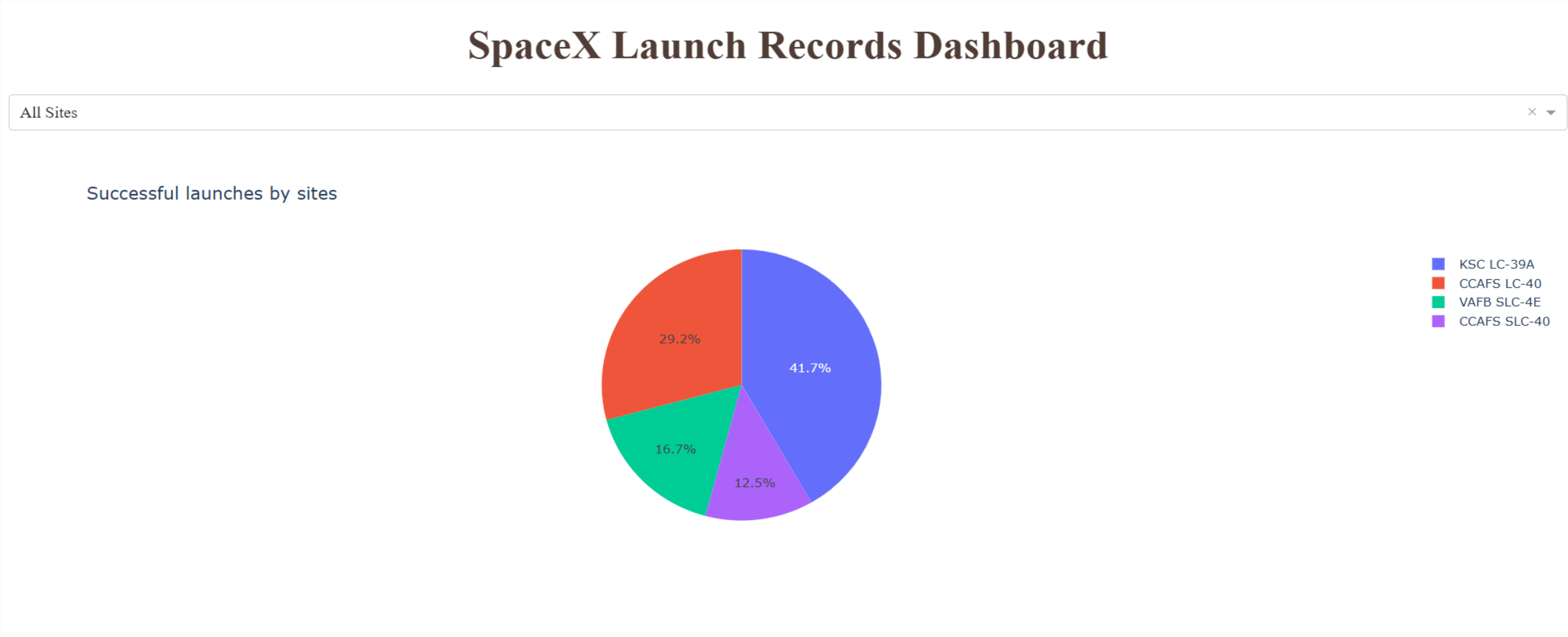


Section 4

Build a Dashboard with Plotly Dash

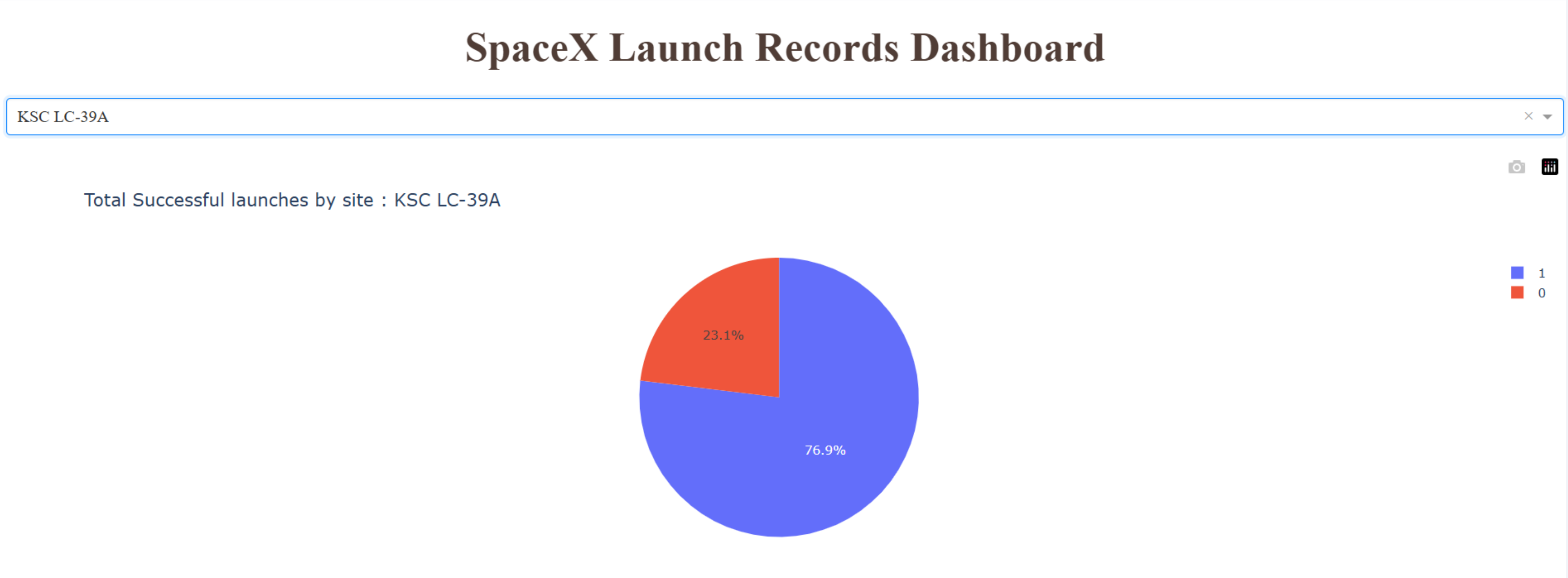
Pie Chart of Launch Success Count for All Sites

- KSC LC -39A Launch Site has most successful Launches while CCAFS SLC-40 has least successful launches of Falcon9.



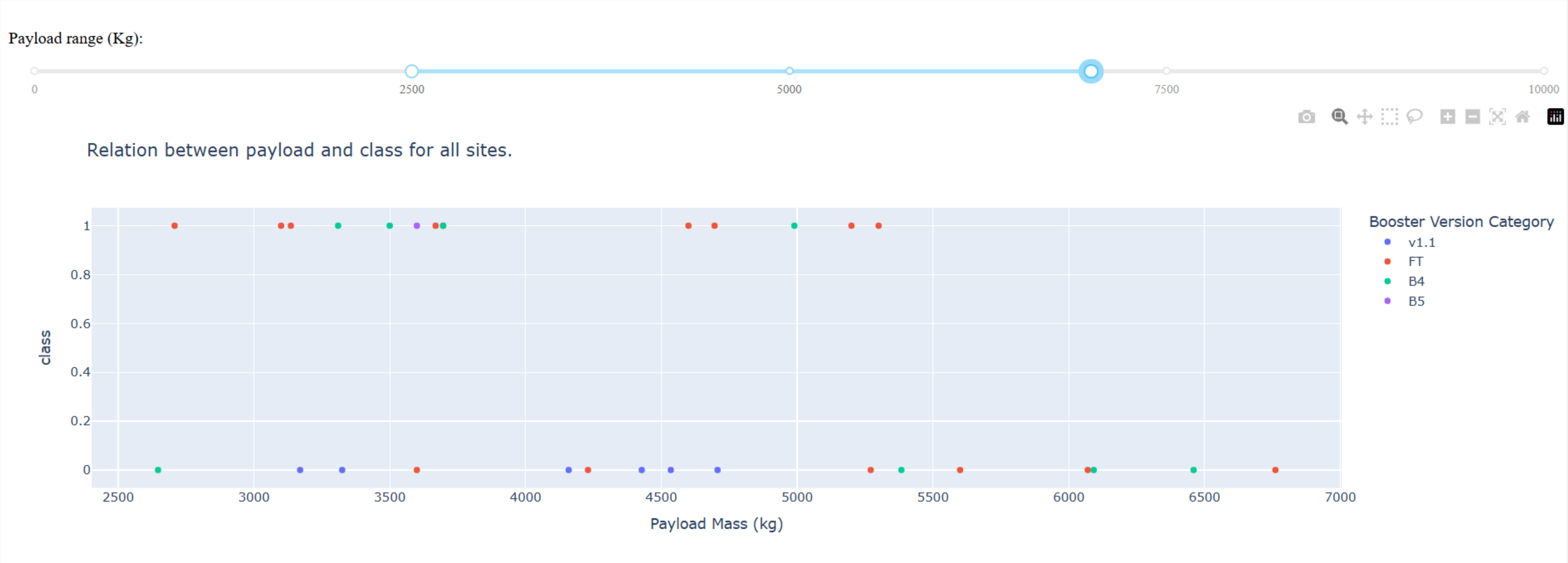
Pie Chart of Lauch Site with Highest Lauch Success Rate

- KSC LC-39A has highest launch success rate with 76.9% of Success in Launching flights.



Scatter Plot of Success rate in Payload Mass range 2500kg to 7000kg

- There is no significant relation between Payload Mass and Success Rate in Payload Mass range 2500 to 7000 kgs.

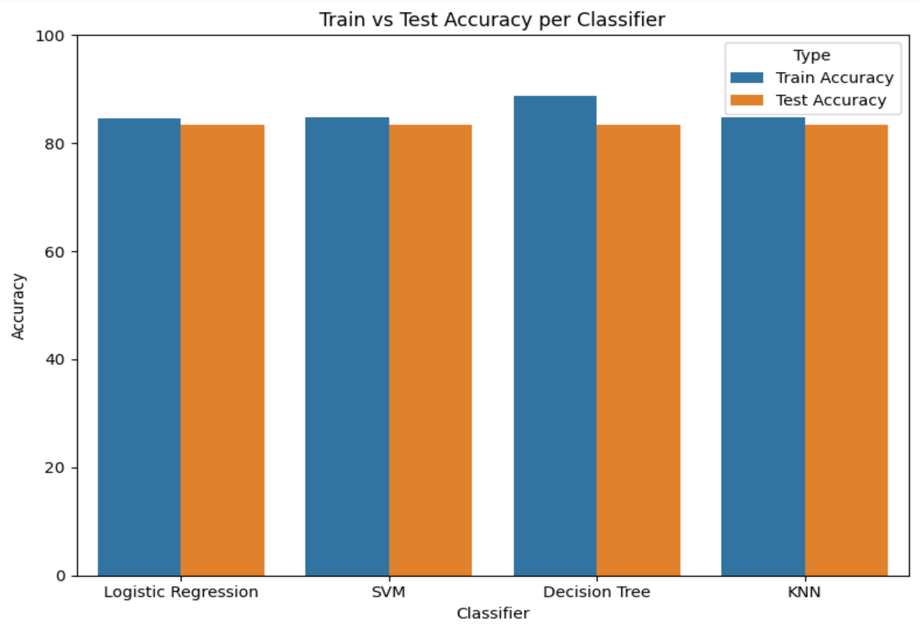


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Test Accuracy of all four models of classification is same equal to 83.33.%
- Decision Tree Classifier is providing highest train accuracy as 88.75% and it could be because of overfitting while other models are giving train accuracy around 85%.
- Although there is not too much difference between train and test accuracy so , Decision Tree Classifier could be better model for classification.

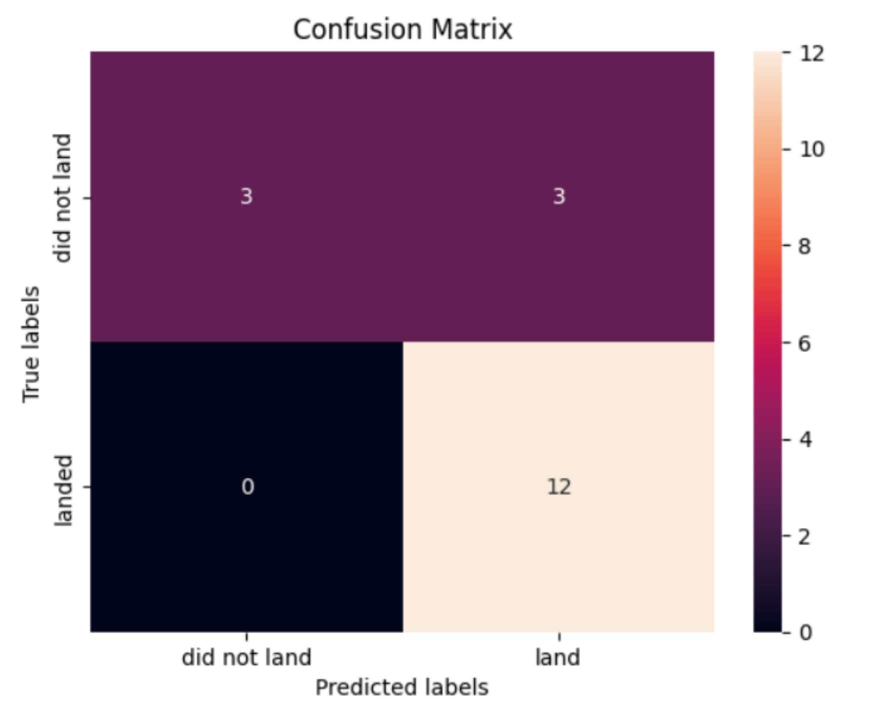



```
Train and Test Accuracies of differnet models :

Logistic Regression Classifier :
        Train Accuracy : 84.64 % | Test Accracy : 83.33 %
SVM Classifier :
        Train Accuracy : 84.82 % | Test Accracy : 83.33 %
Decision Tree Classifier :
        Train Accuracy : 88.75 % | Test Accracy : 83.33 %
KNN Classifier :
        Train Accuracy : 84.82 % | Test Accracy : 83.33 %
```

Confusion Matrix

- The Decision Tree Classifier is better model for classification for landing of first stage of falcon9.
- Decision Tree has 3 False Positives which are not good for model but still it a better model for prediction.
- It has 100% True Positive Rate.
- Decision Tree Classifier Model has Recall of 100% and Precision of 80% for predicting Successful Landing.
- Model's Train Accuracy is Highest equal to 88.75% with Test Accuracy of 83.33%.



Conclusions

- In this project, we try to predict if the first stage of a given Falcon 9 launch will land in order to determine the cost of a launch.
- Each feature of a Falcon 9 launch, such as its payload mass or orbit type, may affect the mission outcome in a certain way.
- Several machine learning algorithms are employed to learn the patterns of past Falcon 9 launch data to produce predictive models that can be used to predict the outcome of a Falcon 9 launch.
- The predictive model produced by decision tree algorithm performed the best among the 4 machine learning algorithms employed.

Appendix

- GitHub Repository of Applied Capstone Project
- L1_SpaceX_Data_Collection_using_API
- L2_SpaceX_Data_Collection_using_WebScraping
- L3_SpaceX_Data_Wrangling
- L4_SpaceX_EDA_with_SQL
- L5_SpaceX_EDA_with_Visualization
- L6_SpaceX_Visual_Analytics_with_Folium
- L7_SpaceX_App_Dashboard_using_Plotly
- L8_SpaceX_Prediction_Analysis_(ML)
- Related Images , Graphs and Charts

Thank you!