IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- In order to achieve our goals, we have collected sample data from the SpaceX Launch data through APIs, we have analyzed the data and performed many statistical analysis and data visualization techniques, finally we have applied some Machine Learning algorithms in order to Predict the desired results.

- Based on our analysis we were able to identify the best way in order to determine that the first stage of the launching can land successfully so it can be reused and thus save multi-millions of dollars for the company

Introduction

- The commercial space age is here, companies are making space travel affordable for everyone. However, some companies are making it more affordable than others by reusing the first stage of the launcher.

- We had to train machine learning models in order to predict the success of the first stage landing so it can be reused and thus make the space travel more affordable.

Section 1

Methodology

Methodology

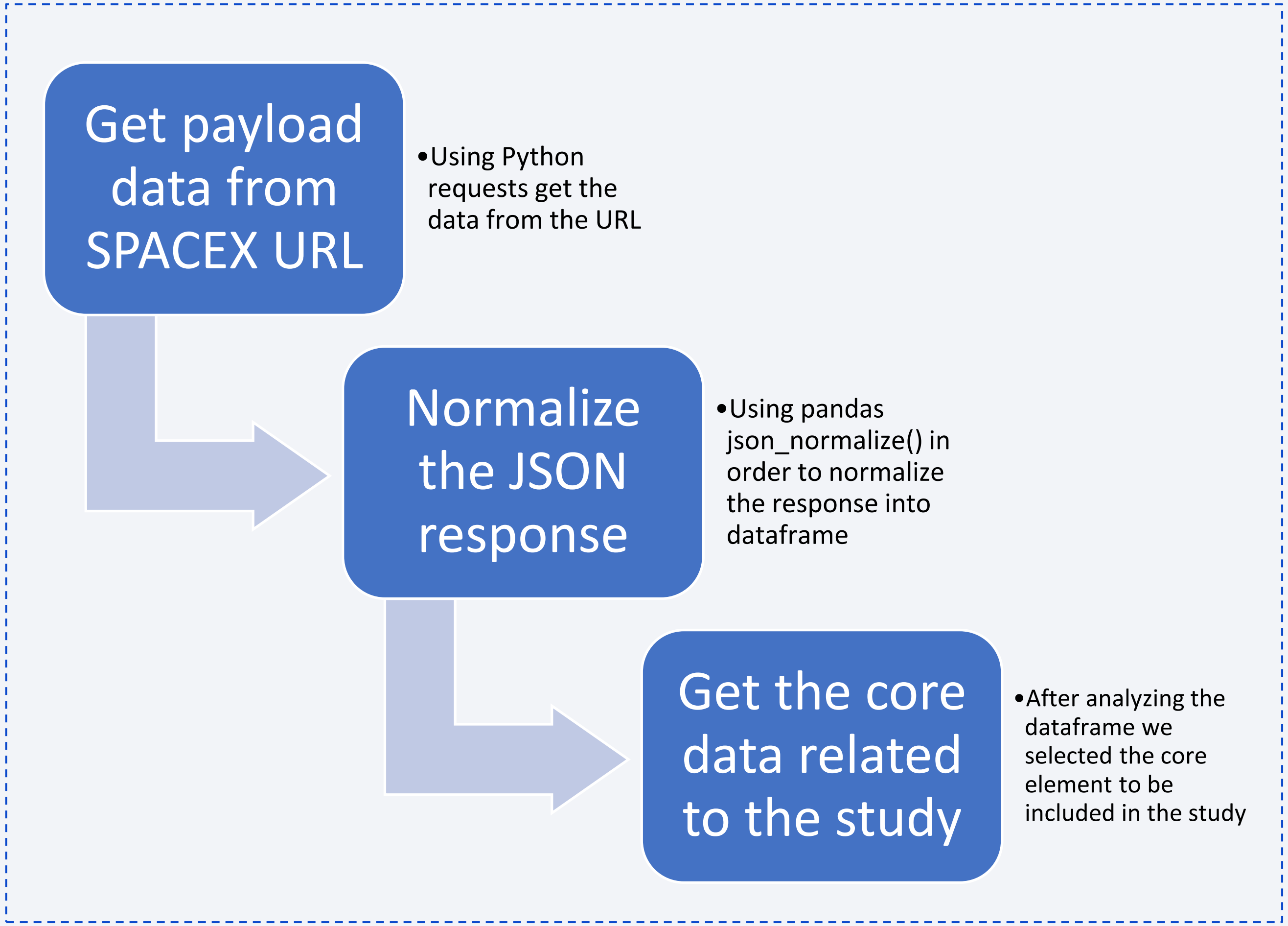
Executive Summary

- Data collection methodology
- Perform data wrangling
- 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 through REST APIs from Space X launch data; we called the APIs and got the response in JSON format.
  - We performed an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.
  - Next, we normalized the JSON response.
  - Next, we want to collect all relevant column names from the HTML table header
  - Next, we filtered the data to get the latest Falcon 9 data only, as old data might cause noises that are irrelevant to our study.

Data Collection – SpaceX API



https://github.com/manhalabboud/presentation/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Perform HTTP Get request to get the page containing the data

- Use python BeautifulSoup on the response in order to be able to navigate through the page

Get the Header Columns

- Navigating through the page using BeautifulSoup library we get all headers in order to mark the data column names

Create dataframe by parsing tables in the page

- Iterate through tables and add rows to the dataframe

https://github.com/manhalabboud/presentation/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Missing values

- Check missing values by executing data.isnull().sum()

Deal with Missing values

- Update the missing values by replacing them with the mean value of the column

https://github.com/manhalabboud/presentation/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

EDA with Data Visualization

- Plotted the Flight Number vs Payload Mass (KG) to see how the continuous launch attempts and payload affects the launch outcome.
- Plotted the Flight Number vs Launch Site to see how the continuous launch attempts from each launch site affects the launch outcome
- Scattered the launch site and the payload to find any relationship
- In a bra chart checked the success rate for each orbit type
- Scattered the flight number, payload and orbit type to find any relationship

https://github.com/manhalabboud/presentation/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Average Payload Mass for Booster Falcon 9
- The year of the first successful landing outcome
- Total success and failure mission outcomes
- Count of landing outcomes between 2010 and 2017

https://github.com/manhalabboud/presentation/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We added the location of each Launch Site to the Map for an easier comparison and review
- Marked the success/failed launches for each launch site so we can easily retrieve the information about the success rate per each launch site.
- Checked the distance between the launch site and nearest coastal area, highway, railway and city, in order to check if there are any distances that needs to be respected between the launch site and other locations

https://github.com/manhalabboud/presentation/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Added drop down list for launch sites, including all sites for better drilling down of dashboards per launch site
- Added a Range Slider to be able to drilldown on the Payload Mass to check the relation with the success rate for each site.
- Pie chart describing the success factor for all sites and can be drilled down for each site
- Scatter plot to see the relation between the success factor and the payload

https://github.com/manhalabboud/presentation/blob/main/Plotly%20dashboards.txt

https://github.com/manhalabboud/presentation/blob/main/plotly%20dashboards%20screen.jpg

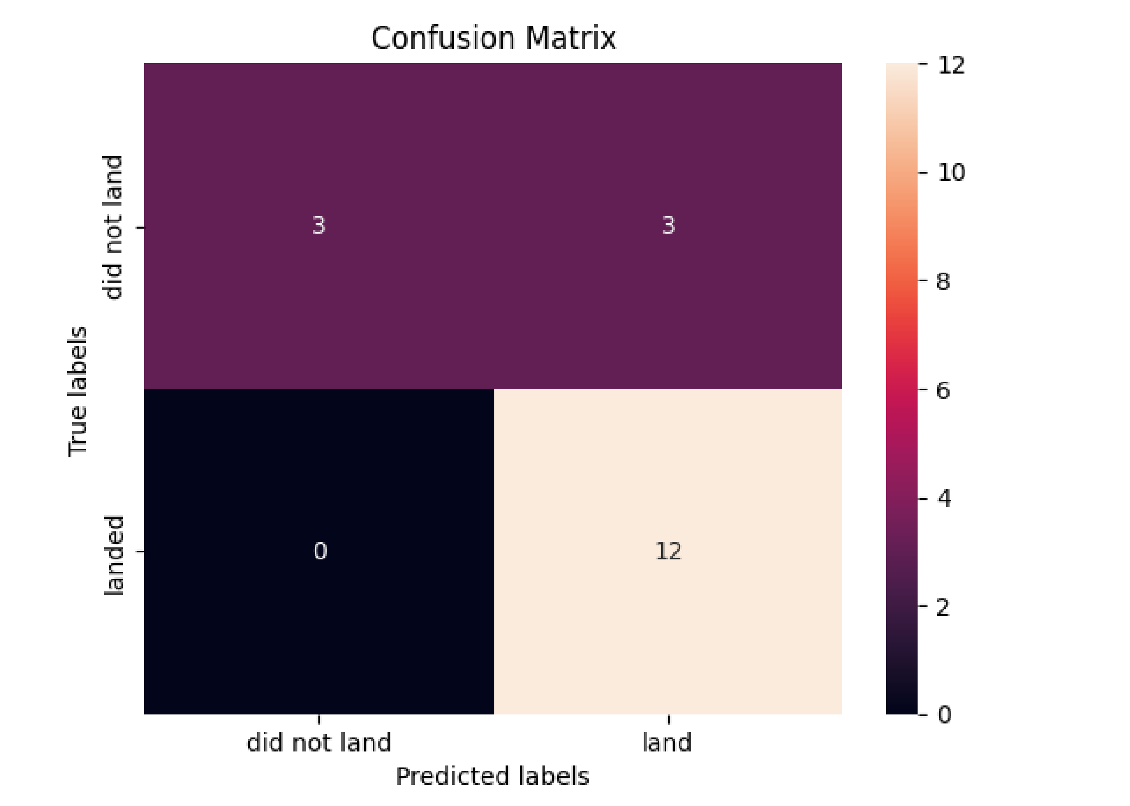
Predictive Analysis (Classification)

- **split**
  - Split the data into testing and training data
- **LR**
  - Fit a Logistic Regression model on the training data, then calculate the score on the testing data.
- **SVM**
  - Fit a Support Vector Machine model on the training data, then calculate the score on the testing data
- **DT**
  - Fit a Decision Tree model on the training data, then calculate the score on the testing data
- **KNN**
  - Fit a K Nearest Neighbor model on the training data, then calculate the score on the testing data
- **Compare**
  - By comparing the confusion matrix between all methods we decide the best method to be used

https://github.com/manhalabboud/presentation/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

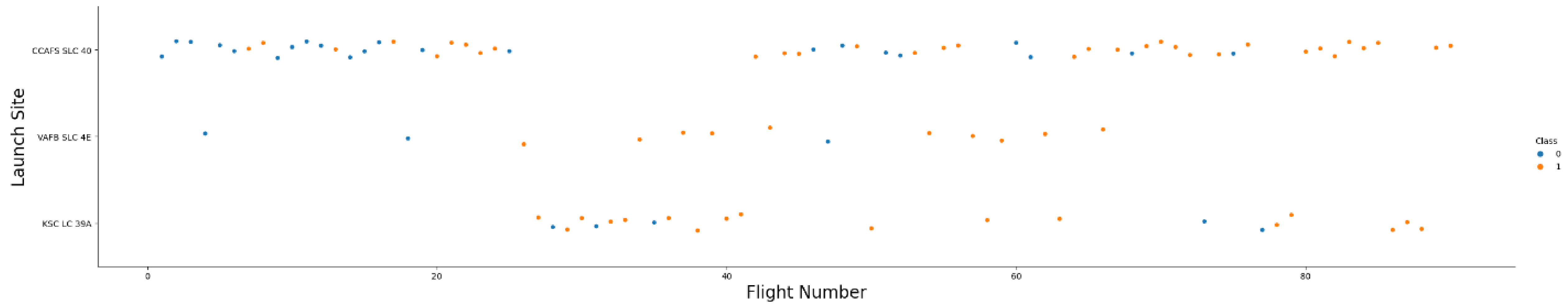
- By comparing confusion matrix we found out the Logistic Regression and Decision Trees are the best methods with a score of 83.33% accuracy



Section 2

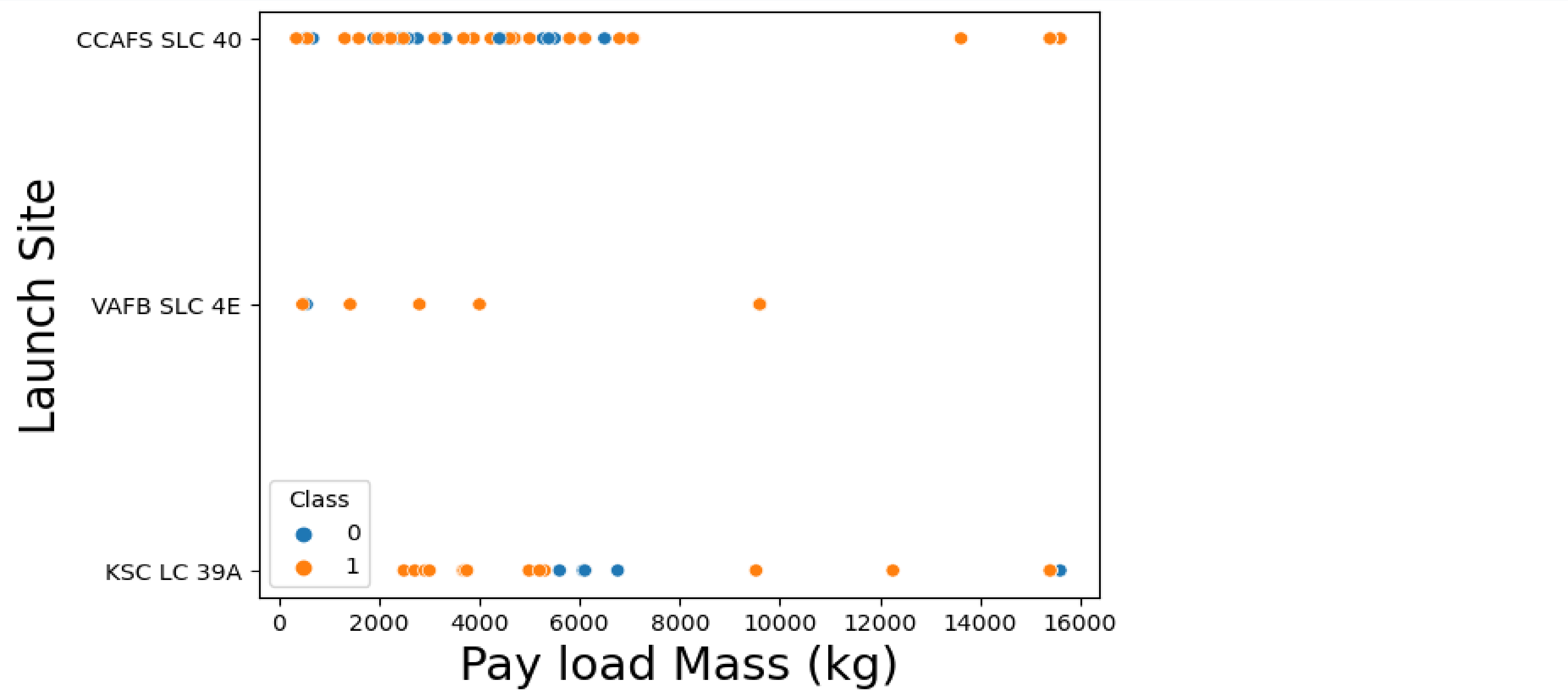
Insights drawn from EDA

Flight Number vs. Launch Site



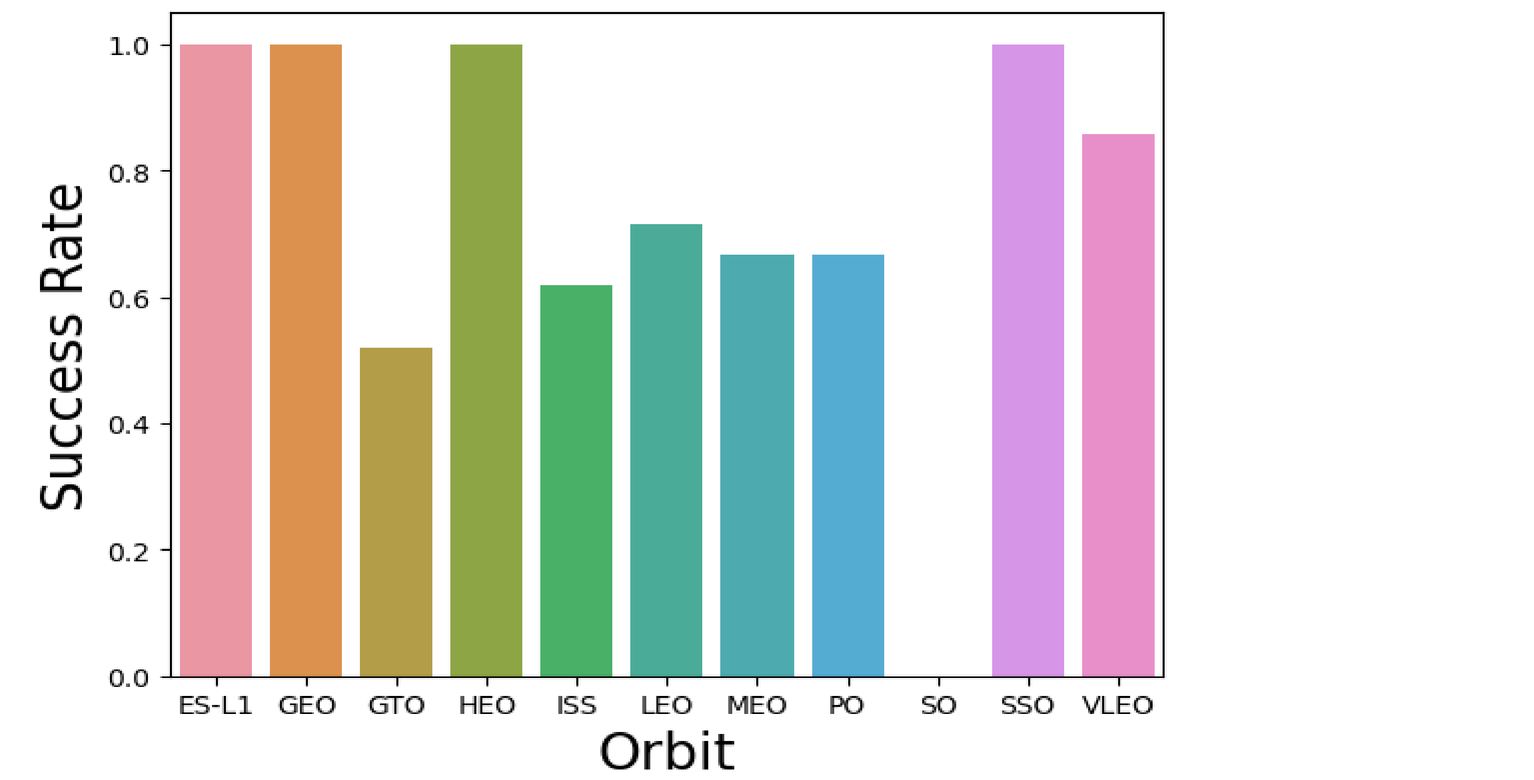
- The success rate is increasing with more flights

Payload vs. Launch Site



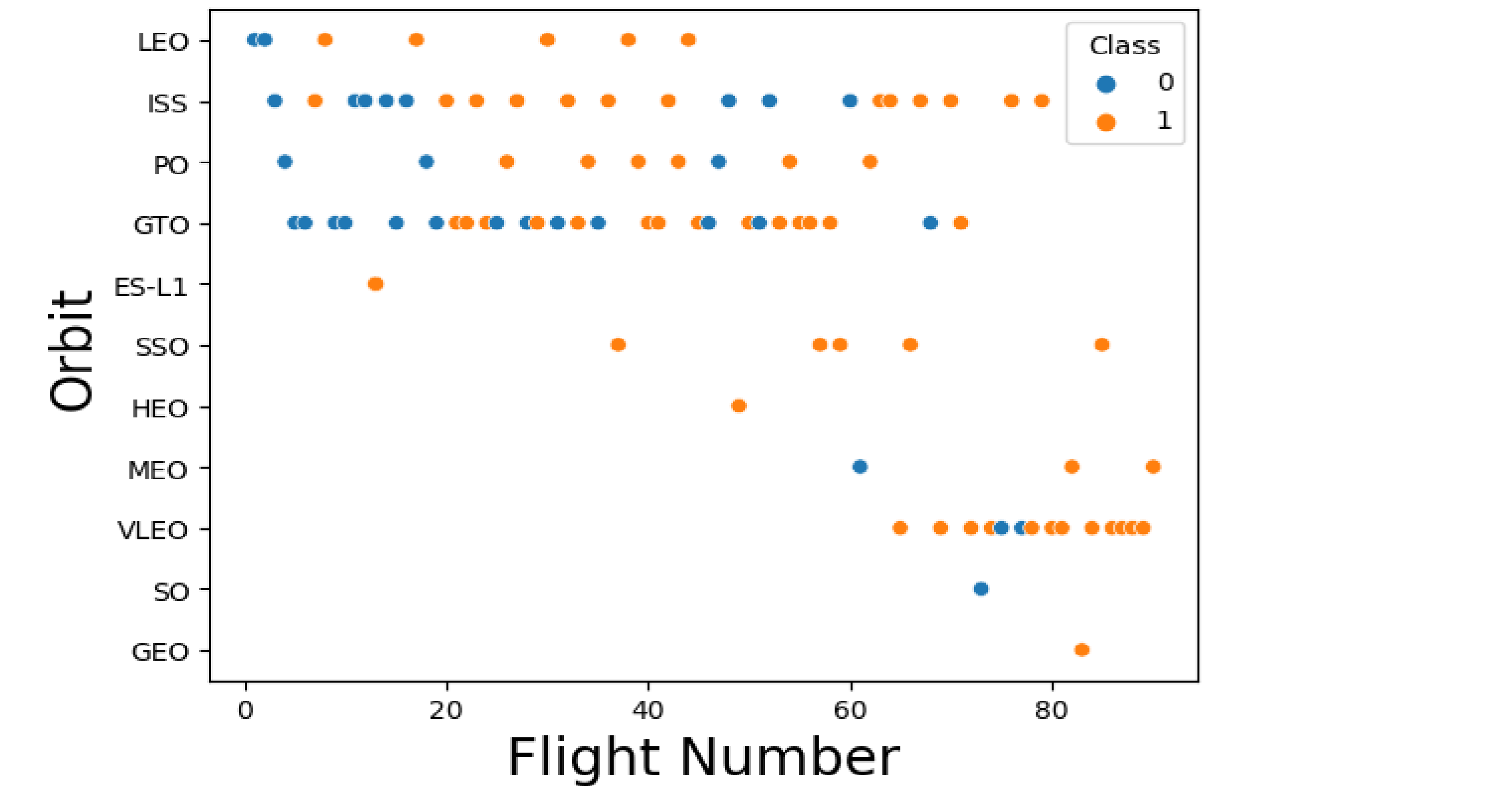
- For CCAFS SLC 40 with heavy load the success is 100%
- For VAFB SLC 4E no heavy loads were launched
- For KSC LC 39A we cannot have any insight.

Success Rate vs. Orbit Type



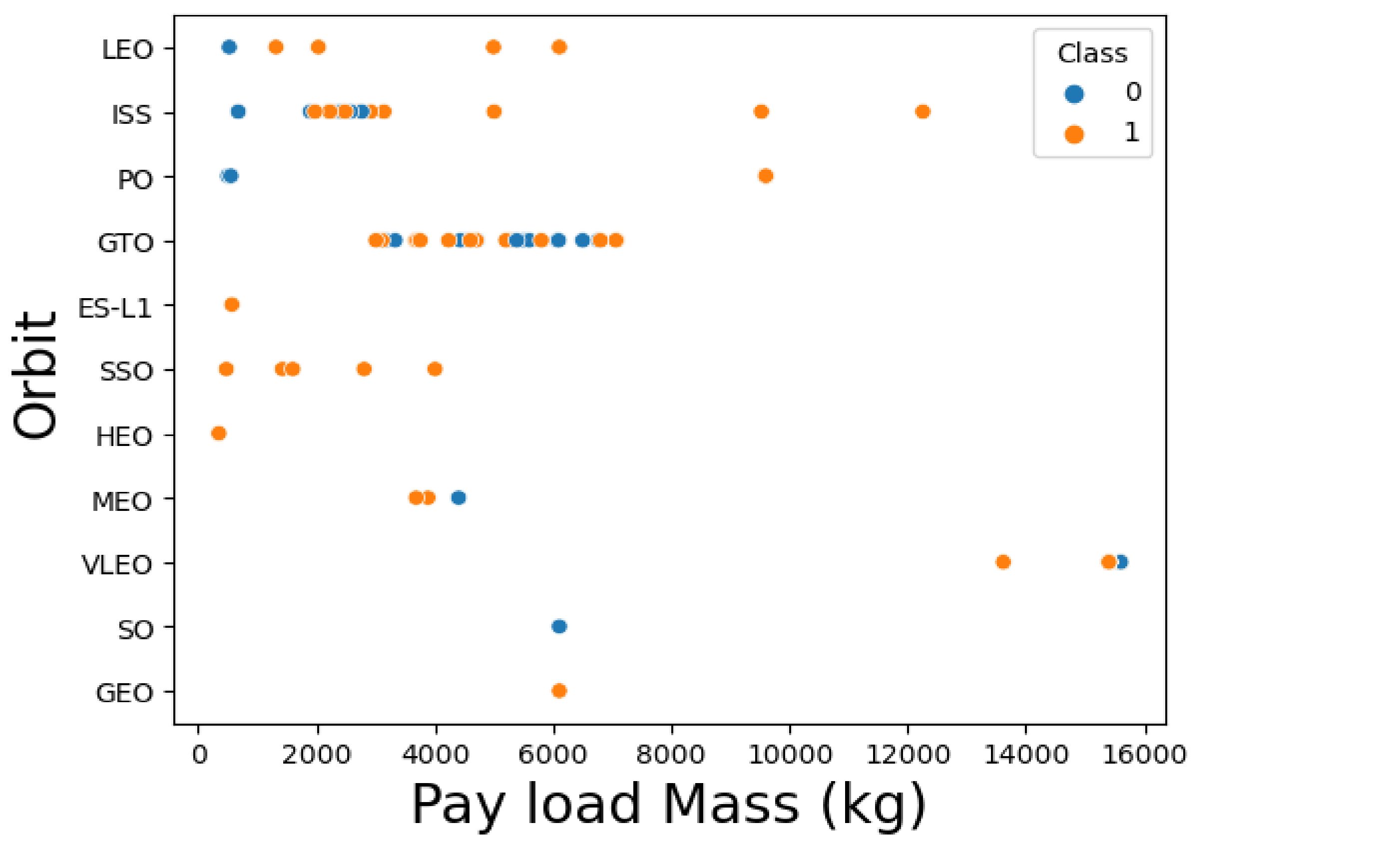
- Orbits ES-L1, GEO, HEO and SSO have the highest success rate

Flight Number vs. Orbit Type



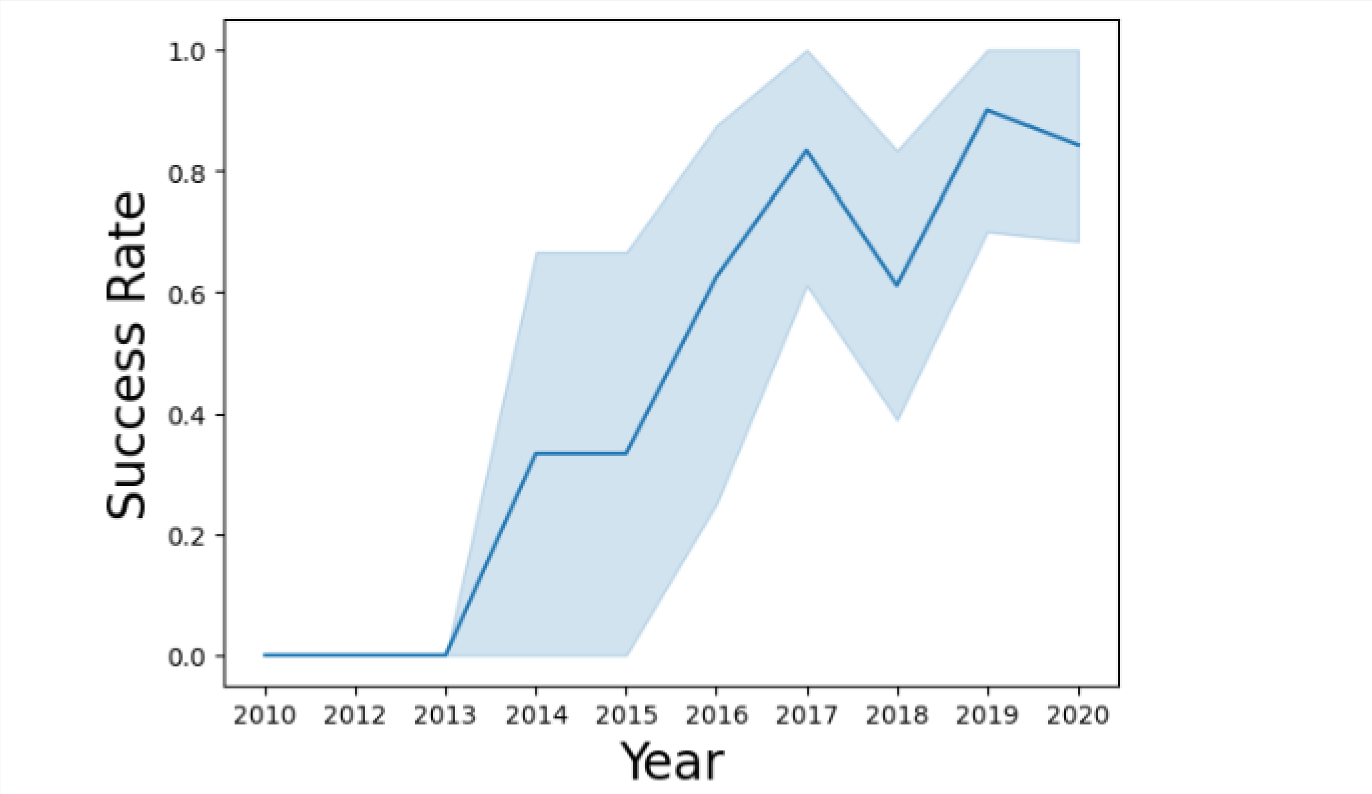
- LEO success is related to number of flights but for GTO we cannot get any insight

Payload vs. Orbit Type



- For heavy pay loads the success rate is higher for ISS, LEO and Polar while for GTO we cannot get any insights

Launch Success Yearly Trend



- Before 2013 there was no success attempt, and after 2013 the success rate got increased

All Launch Site Names

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

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

[9]:

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

- Using distinct we get the unique launch sites

Launch Site Names Begin with 'CCA'

```
[10]: %sql select * from SPACEXTABLE where "Launch_Site" like ('CCA%') limit 5
 * sqlite:///my_data1.db
Done.
```

[10]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Task 3

- Using like operator to get the launch sites with name begins with CCA and using the limit we limit the results to 5 records

Total Payload Mass

Task 3

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

```
[11]: %sql select sum("PAYLOAD_MASS__KG_") from SPACEXTABLE where Customer = 'NASA (CRS)'

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

[11]:

sum("PAYLOAD_MASS__KG_")
45596

- Using the aggregation function sum we can get the total payload mass for a customer.

Average Payload Mass by F9 v1.1

Task 4

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

```
13]: %sql select avg("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Booster_Version" like 'F9 v1.1%'
```

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

13]:

avg("PAYLOAD_MASS__KG_")
2534.6666666666665

- Using the aggregation function avg we can get the average pay load mass for boosters and using the like keyword we determine the booster version like ‘F9 v1.1’

First Successful Ground Landing Date

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

Hint:Use min function

```
[14]: %sql select min(Date) from SPACEXTABLE where "Landing_Outcome" = 'Success (ground pad)'
```

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

[14]:

min(Date)
2015-12-22

- Using the min function on the Date column we get the first date and we filter by landing outcome to be success

Successful Drone Ship Landing with Payload between 4000 and 6000

```
[17]: %sql select "Booster_Version" from SPACEXTABLE where  "Landing_Outcome" = 'Success (drone ship)' and "PAYLOAD_MASS__KG_" between 4000 and 6000

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

[17]:

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

- Query the names of the drone ships and we filter by the landing outcome and payload mass between 4000 and 6000 kg

Total Number of Successful and Failure Mission Outcomes

```
[18]: %sql select "Mission_Outcome", count(*) from SPACEXTABLE group by "Mission_Outcome"

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

[18]:

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

- We use the group by expression in order to group by Mission Outcome and we get the count of records

Boosters Carried Maximum Payload

Task 8

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

[19]:
```
%sql select "Booster_Version" from SPACEXTABLE where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_") from SPACEXTABLE)
```

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

[19]:

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

- Query all the names of boosters and by using a sub query to get the max payload we filter the records

2015 Launch Records

```
[30]: %sql select Date, 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 substr(Date,0,5)='2015'

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

[30]:

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

- Filtering on the year and the failure launch records we get the failures in 2015

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

```
[36]: %sql select "Landing_Outcome", count(*) from SPACEXTABLE where "Landing_Outcome" in ('Failure (drone ship)','Success (ground pad)')
and Date between date('2010-06-04') and date('2017-03-20')  group by "Landing_Outcome"  order by count(*) desc

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

[36]:

Landing_Outcome	count(*)
Success (ground pad)	5
Failure (drone ship)	5

- Filter between the given dates and get the counts grouping by the outcome

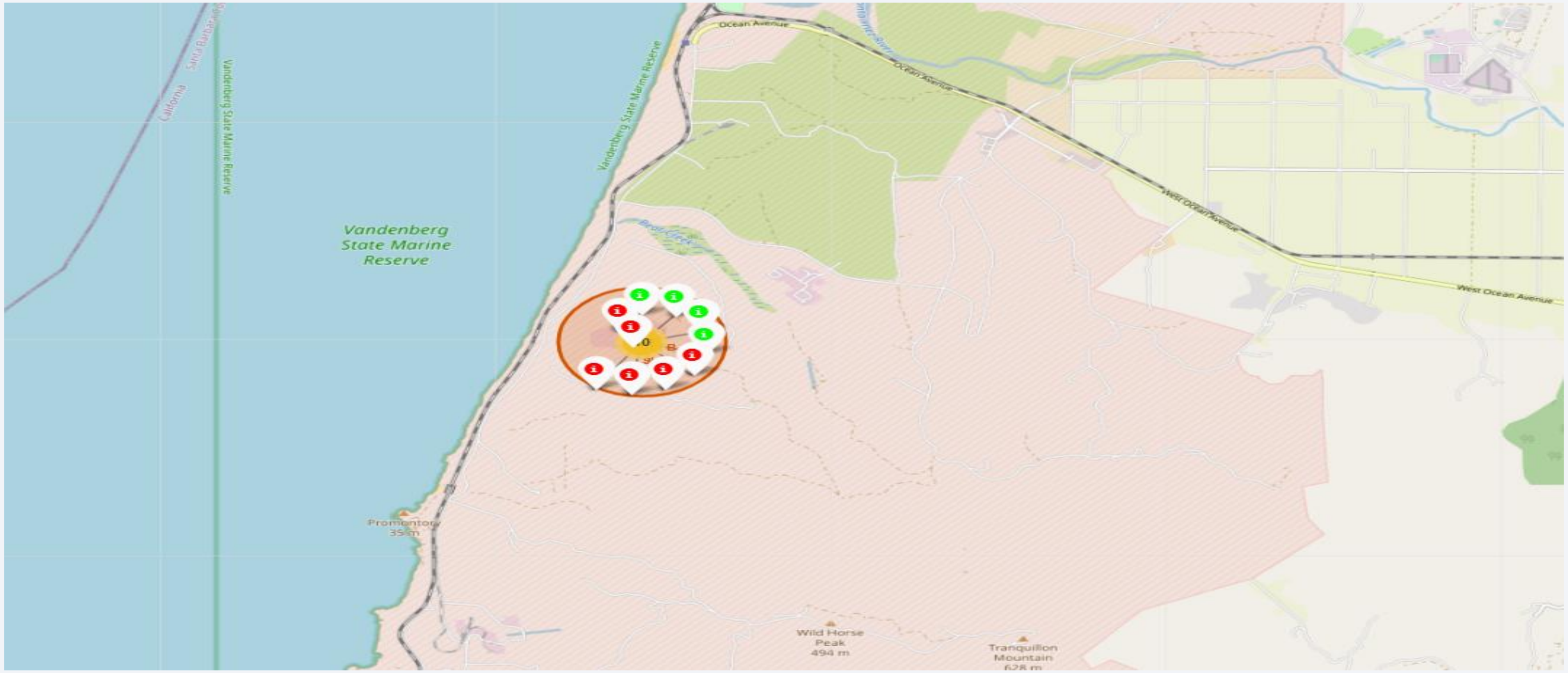
Section 3

Launch Sites Proximities Analysis

Launch Sites locations

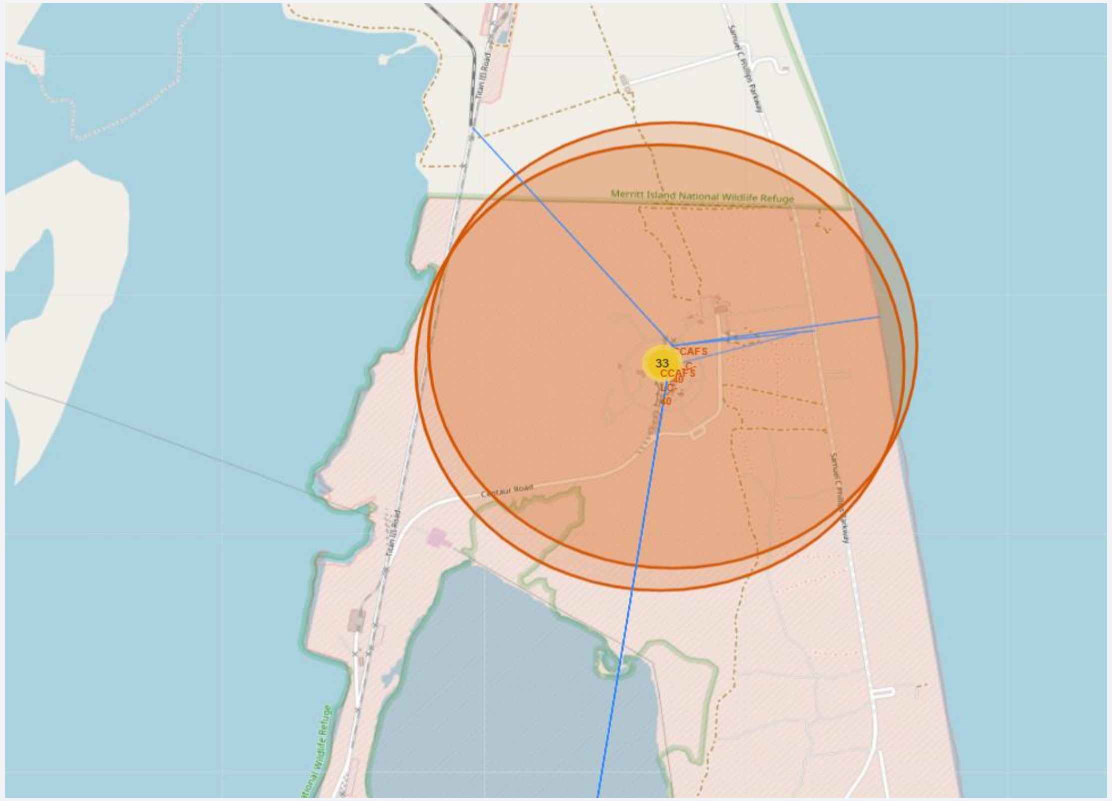
- Checking the launch site locations

Success / Failure attempts markers



- Here we can see the total attempts in each launch site with the success / failure attempts clearly identified by colors

Launch Site distance proximiteis

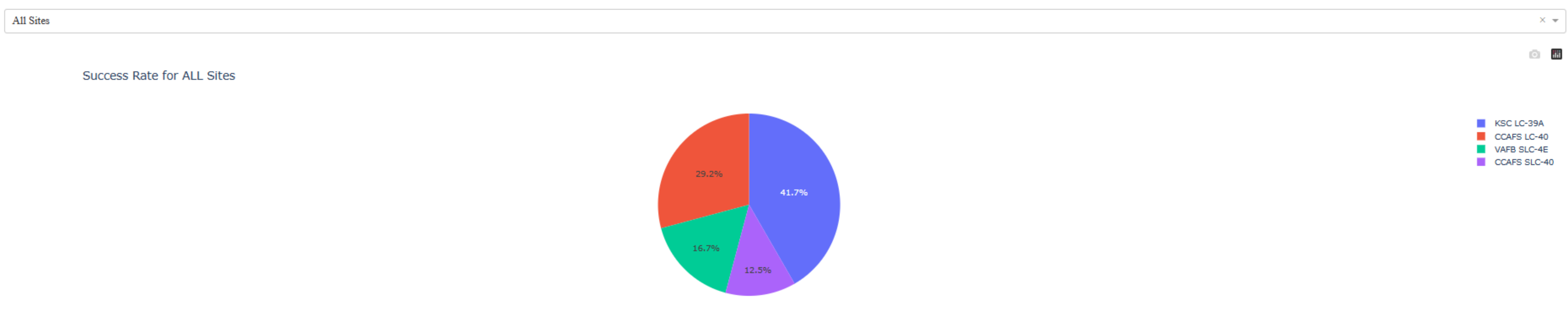


- The distance form launch site to nearest coast, highway and city

Section 4

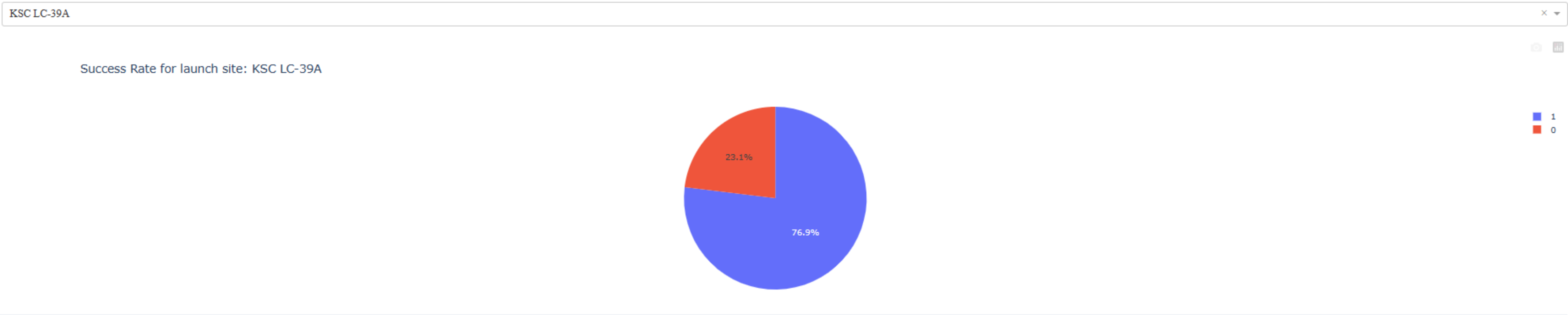
Build a Dashboard with Plotly Dash

Success count per Launch Sites



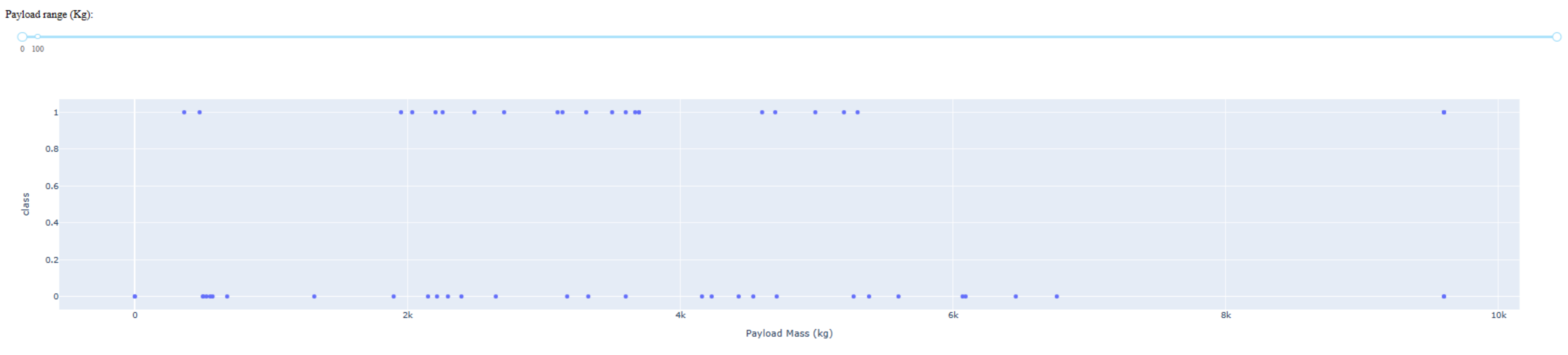
- Here we can see the success rate for KSC site is more than other sites.

Launch Site with highest success ratio



- KSC LC-39A Launch Site has a success ration of 76.9%

Payload Mass vs Success Rate

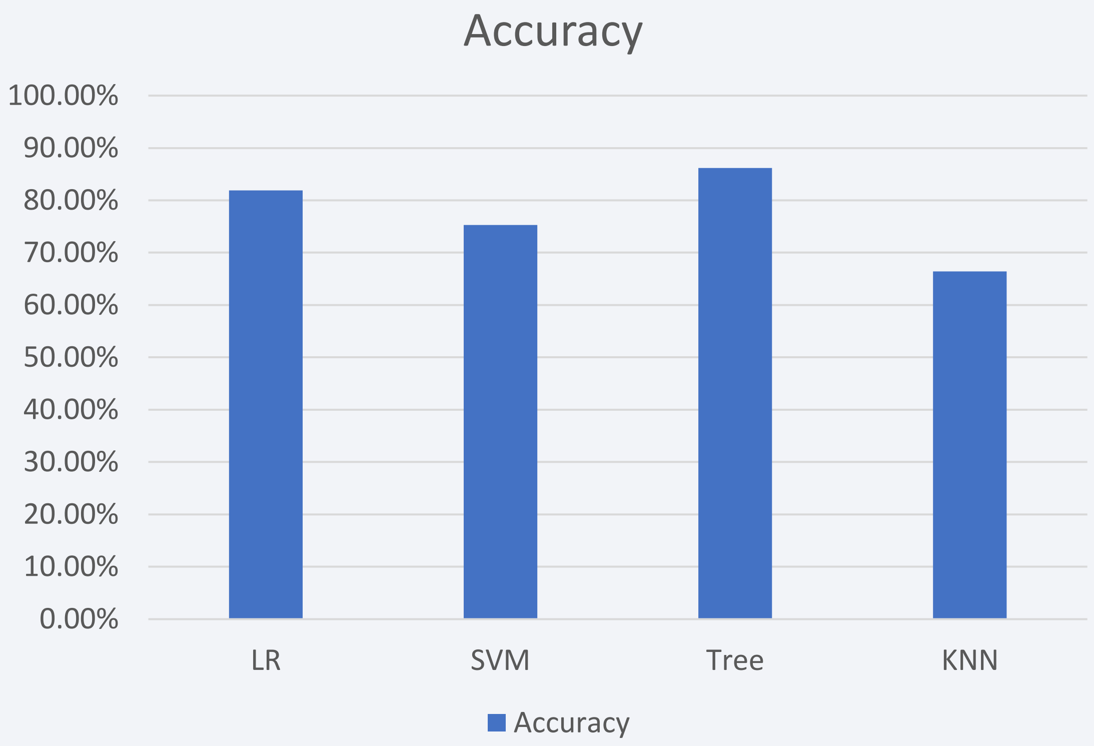


- Here we see how changing the payload mass range will help us determine the best payload for a better success rate

Section 5

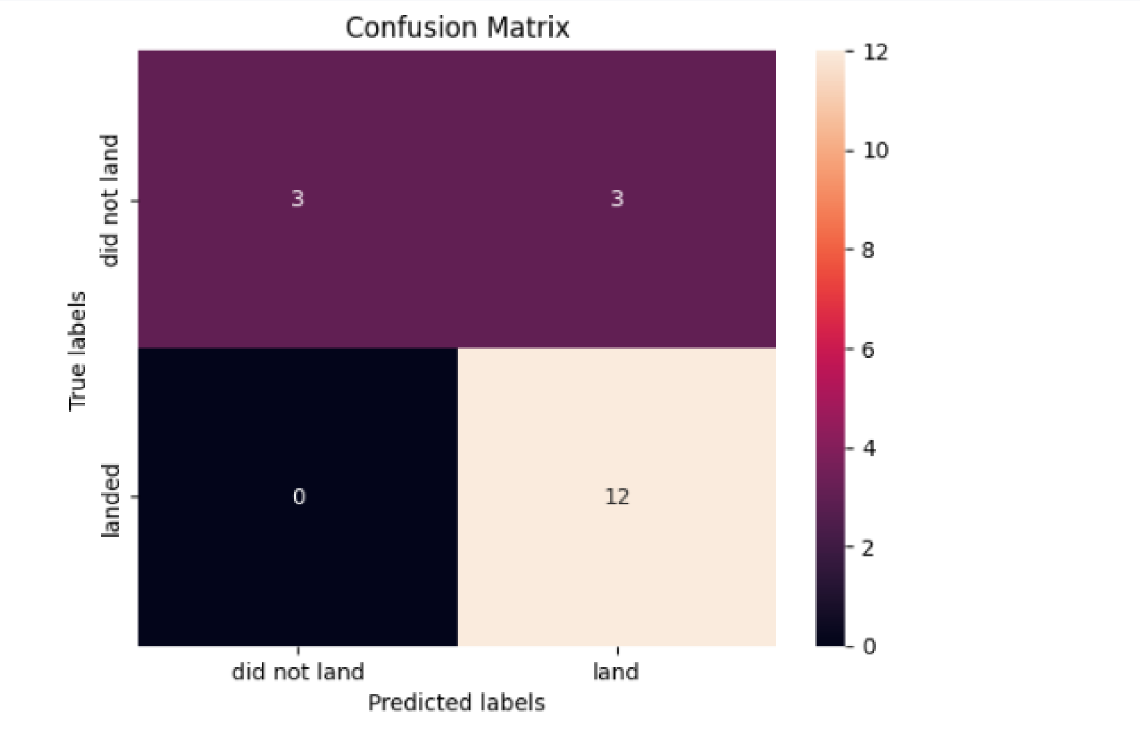
Predictive Analysis (Classification)

Classification Accuracy



- Decision Tree has the best accuracy among other methods

Confusion Matrix



- For successful landing the model prediction is 100% accurate while for unsuccessful landing the predication is 50% accurate

Conclusions

- Using Machine learning model based on decision tree methodology we can have the best accuracy to predict the successful landing of the first stage of the launcher

Appendix

- https://api.spacexdata.com/v4/launches/past
- https://github.com/manhalabboud/presentation/blob/main/jupyter-labs-spacex-data-collection-api.ipynb
- https://github.com/manhalabboud/presentation/blob/main/jupyter-labs-webscraping.ipynb
- https://github.com/manhalabboud/presentation/blob/main/jupyter-labs-spacex-data-collection-api.ipynb
- https://github.com/manhalabboud/presentation/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb
- https://github.com/manhalabboud/presentation/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb
- https://github.com/manhalabboud/presentation/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb
- https://github.com/manhalabboud/presentation/blob/main/plotly%20dashboards%20screen.jpg
- https://github.com/manhalabboud/presentation/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Thank you!