IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Beatriz Jimenez Franco

03-07-2023

https://github.com/singulritarian7/Capstone-**Spacex**

Outline

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

Executive Summary

Summary of methodologies
- Data Collection through API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction
Summary of all results
- Exploratory Data Analysis rand predictive analytics result
- Interactive analytics in screenshots

Introduction

- Project background and context
- We will predict if the Falcon 9 first stage will land succesfully..**SpaceX 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 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.**
- Problems you want to find answers
- - What factors determine if the rocket will land successfully?
- - The interaction amongst various features that determine the success rate of a successful landing.
- - What operating conditions needs to be in place to ensure a successful landing program.

Section 1
Methodology

Methodology

Executive Summary		
Data collection methodology:	• Data was collected using SpaceX API (Rest API) and web scraping from Wikipedia.	
Perform data wrangling	One-hot encoding was applied to categorical features	
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	

Describe how data sets were collected.

- Data collection was done using get request to the SpaceX API.

- Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().

- We then cleaned the data, checked for missing values and fill in missing values where necessary.

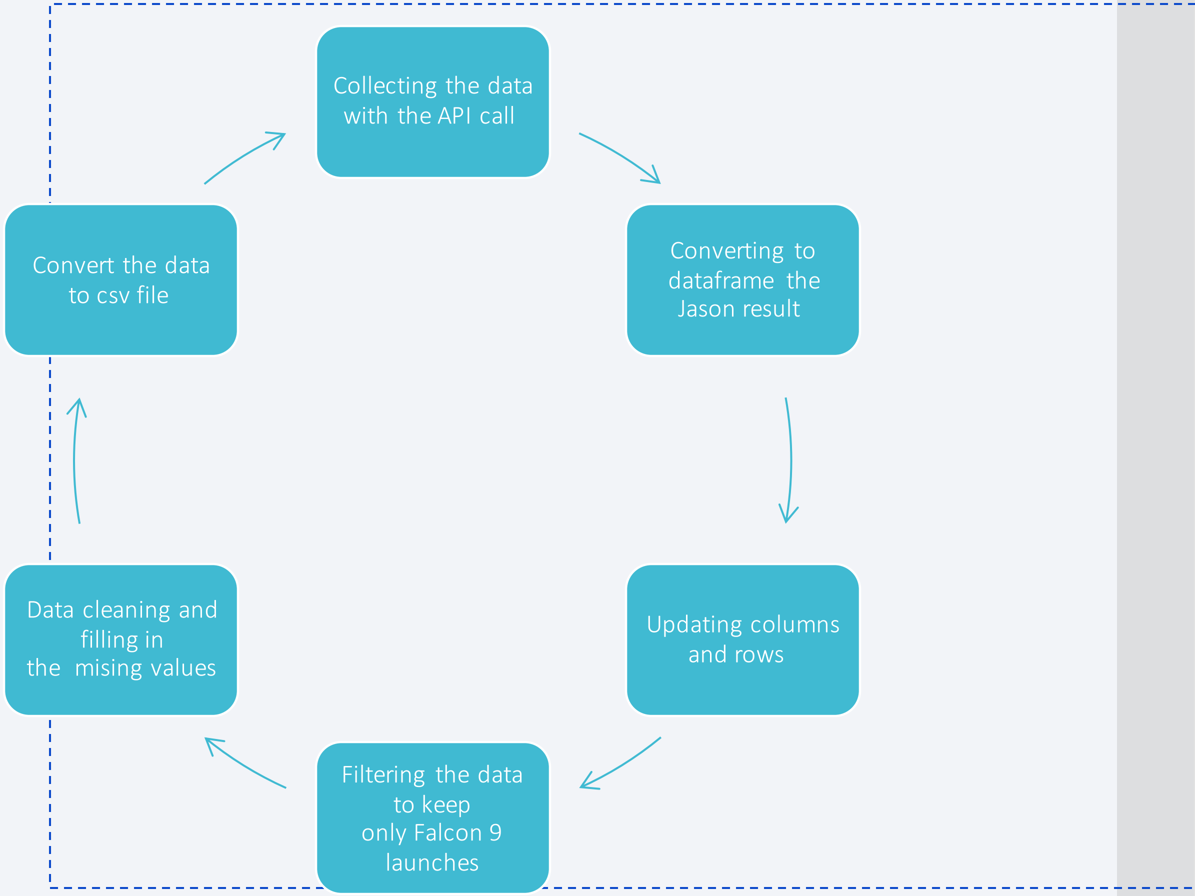
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.

- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis

Data Collection – SpaceX API I

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is

https://github.com/singulritarian7/Capstone-Spacex/blob/master/jupyter-labs-spacex-data-collection-api%20(9).ipynb

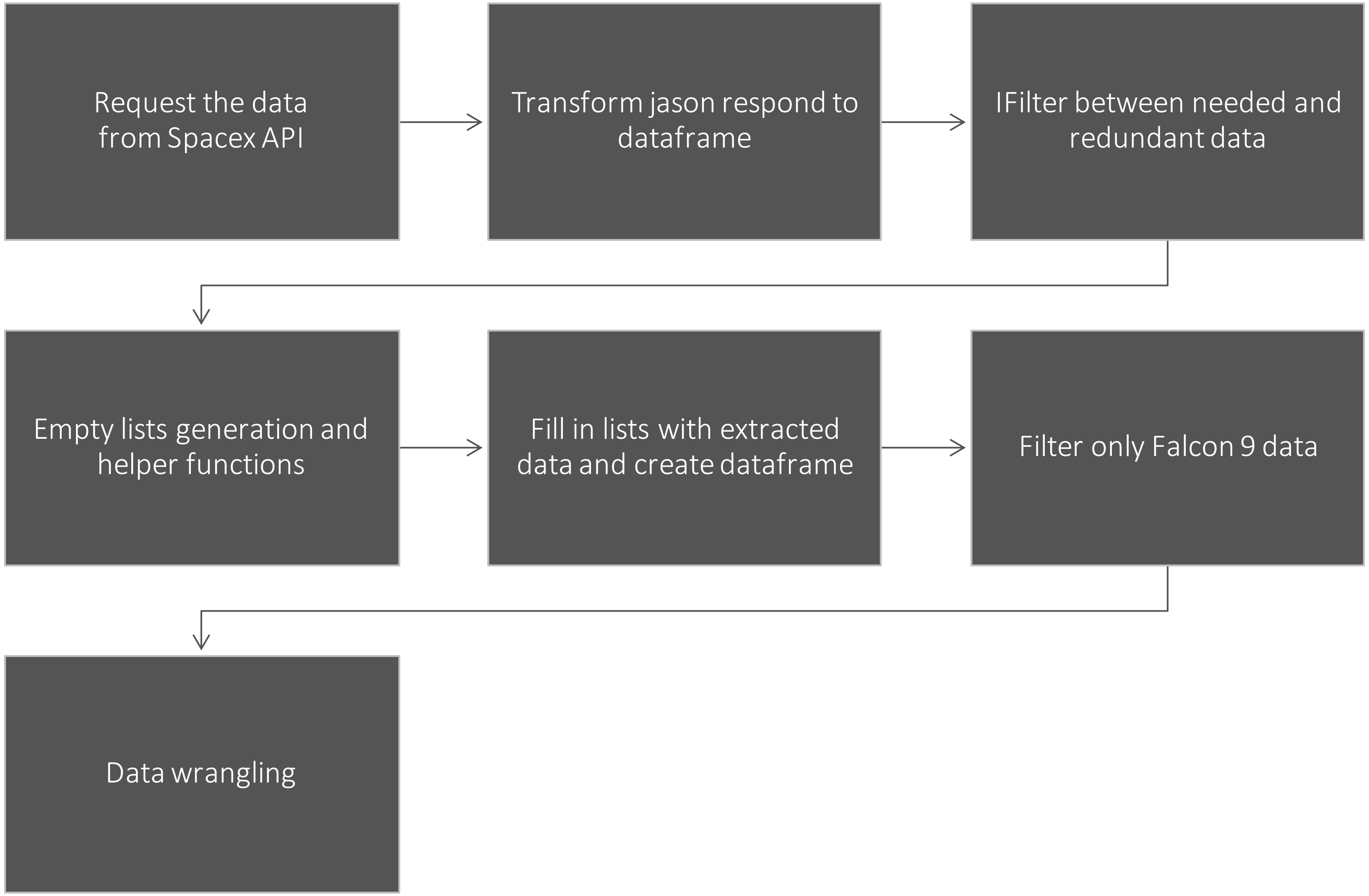


Data Collection – SpaceX API II

Steps for data collection from API:

1. Request data from Spacex API using the url.
2. Convert the Jason result into a dataframe
3. Filter between needed and redundant data
4. Empy lists generation and helper function to extract data
5. Populate empty list with extracted columns data
6. Create a dictionary combining the columns data
7. Create a new dataframe and filter it with only Falcon 9 launches
8. Data wranling (dealing with missing values)

Data collection-Spacex API III



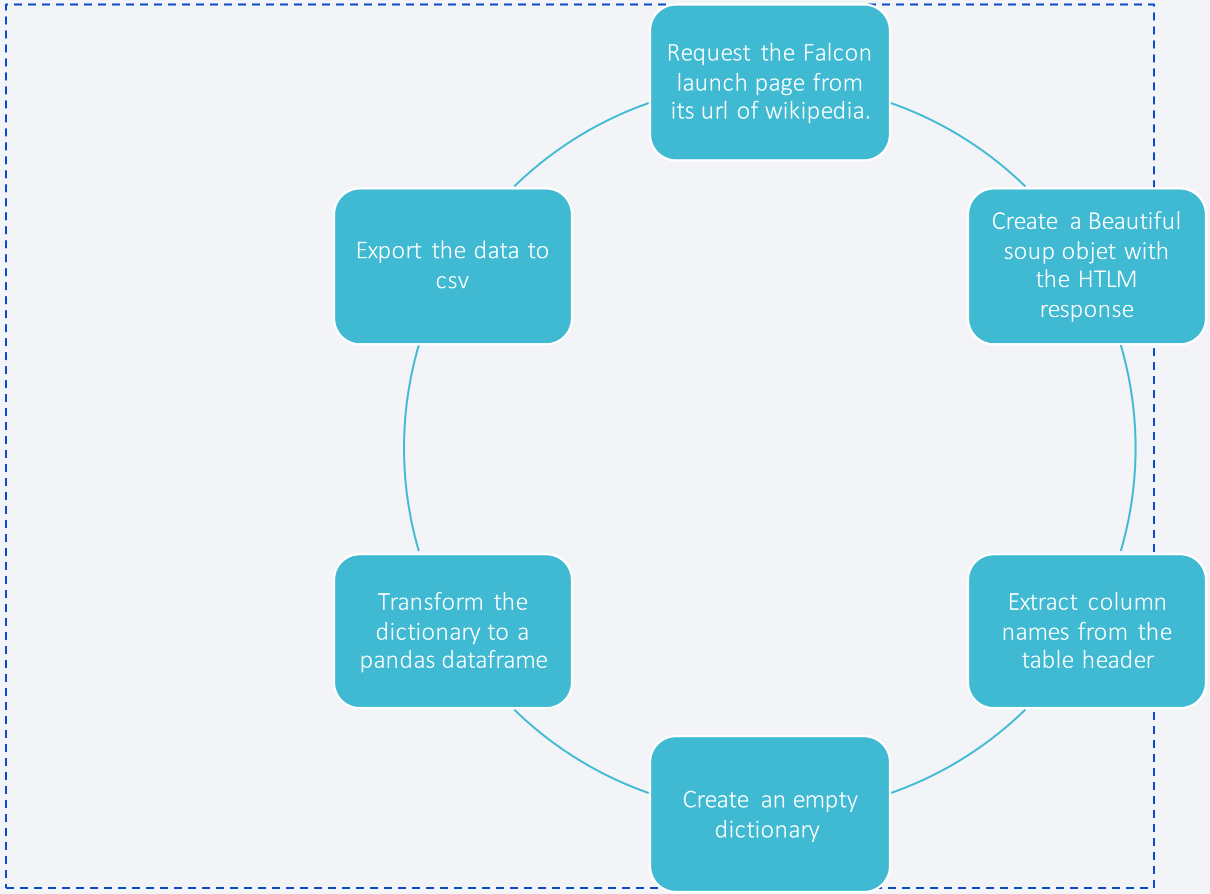
Data Collection – Scraping I

We applied web scrapping to Falcon 9 launch records with BeautifulSoup

We parsed the table and converted it into a pandas dataframe.

The link to the notebook

https://github.com/singulritarian7/Capstone-Spacex/blob/master/jupyter-labs-webscraping%20(4).ipynb



Data Colle

THE STEPS FOR WEB SCRAPING ARE:

1- REQUEST THE WIKI PAGE OF SPACEX FALCON 9 LAUNCH

2- FIND TABLES WITH DATA OBJET OF STUDY

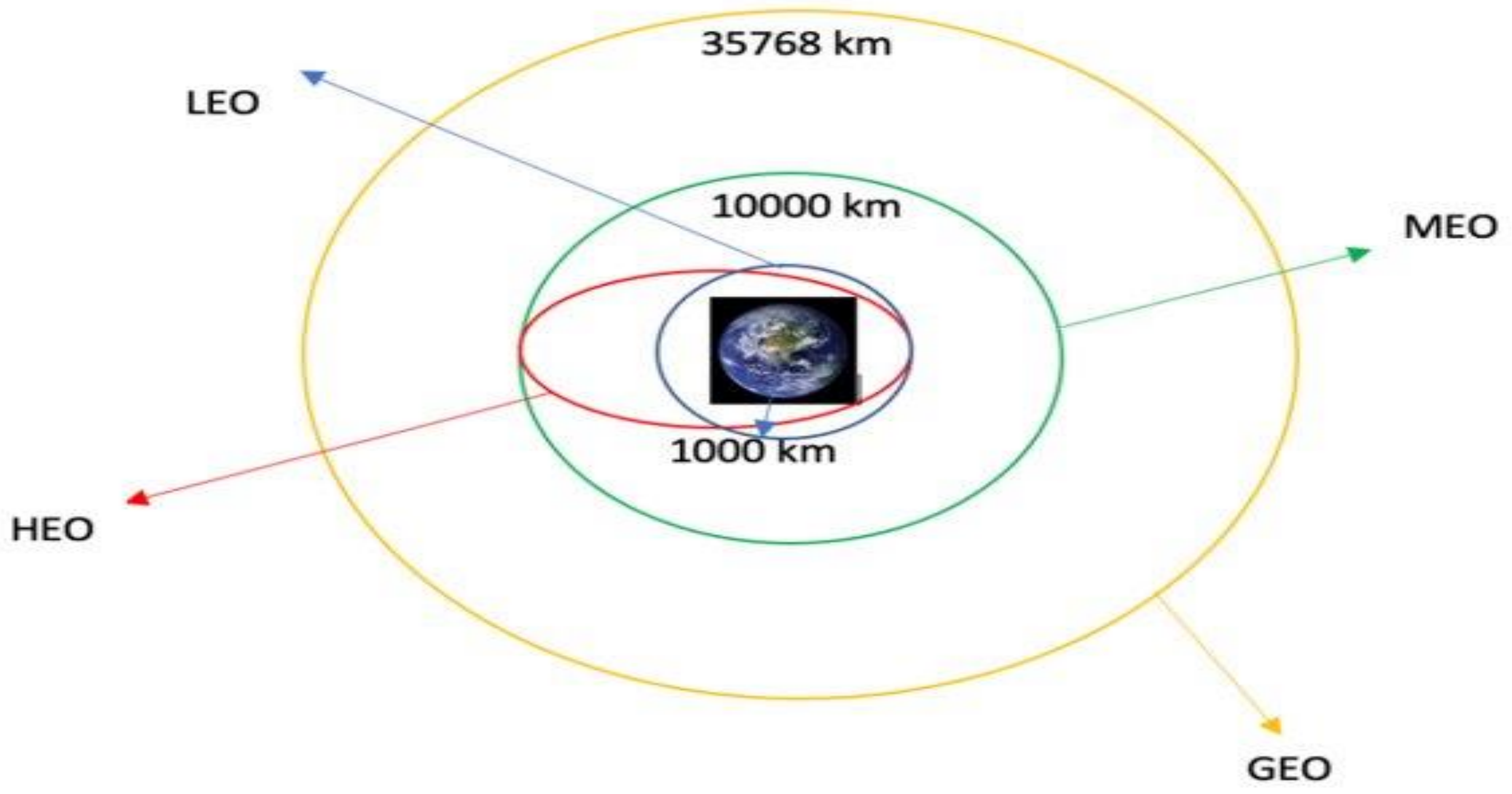
3- ITERATE THROUGH THE TABLES TO EXTRACT THE DATA

4- CREATE AN EMPTY DICTIONARY TO FILL IN

5- APPEND EXTRACTED DATA TO DICTIONARY

6- CREATE DATAFRAME FROM DICTIONARY

Some types of orbits of each launch



Data Wrangling II

We will perform EDA and determine training labels

- Calculate :

  Number of launches for each site

  Number and ocurrent of each orbit

  Number and ocurrence of mission per orbit type

- Create binary landing outcome as the dependent variable
- Calculate the success rate
- Export data to csv

Data Wrangling III

Loading the dataset and perform EDA

↓

Calculate the outcomes or training labels

↓

Finding the set of bad outcomes (Falses and Nones)

↓

Presenting outcomes as 0 or 1

↓

Determinate the success rate

↓

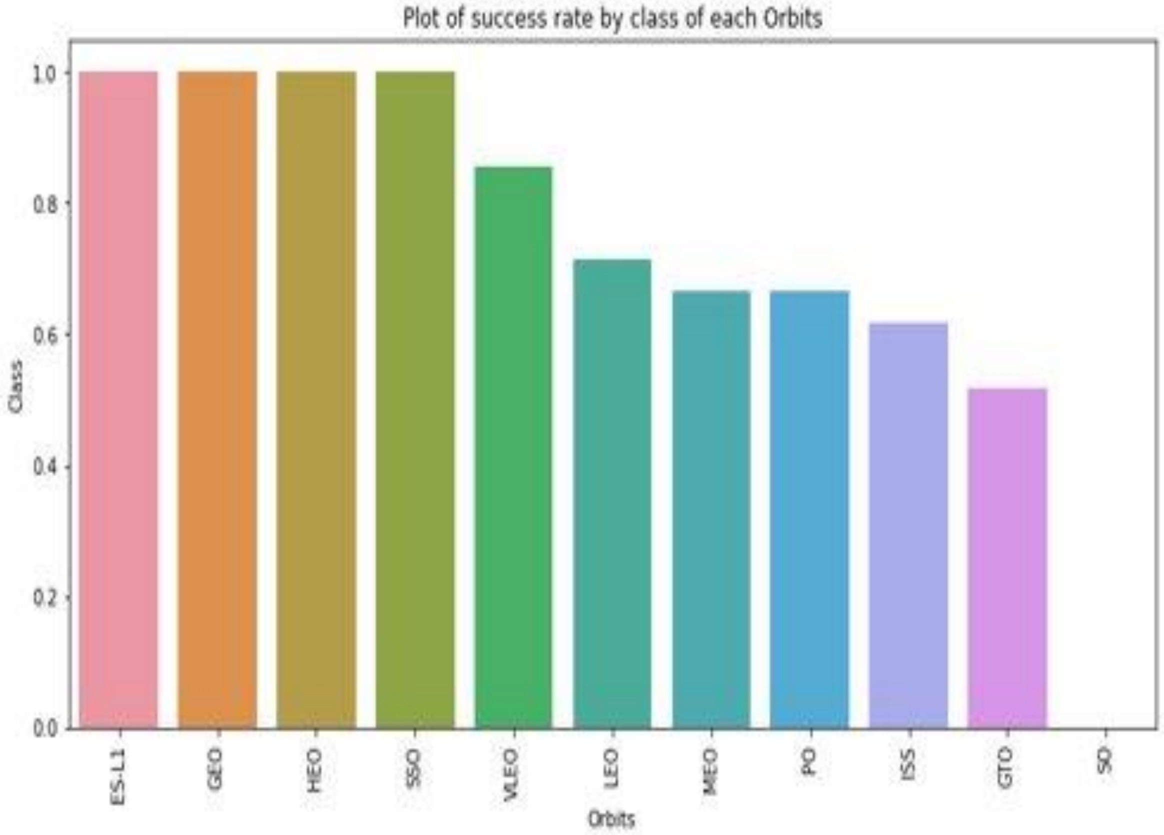
Export the dataset to csv file

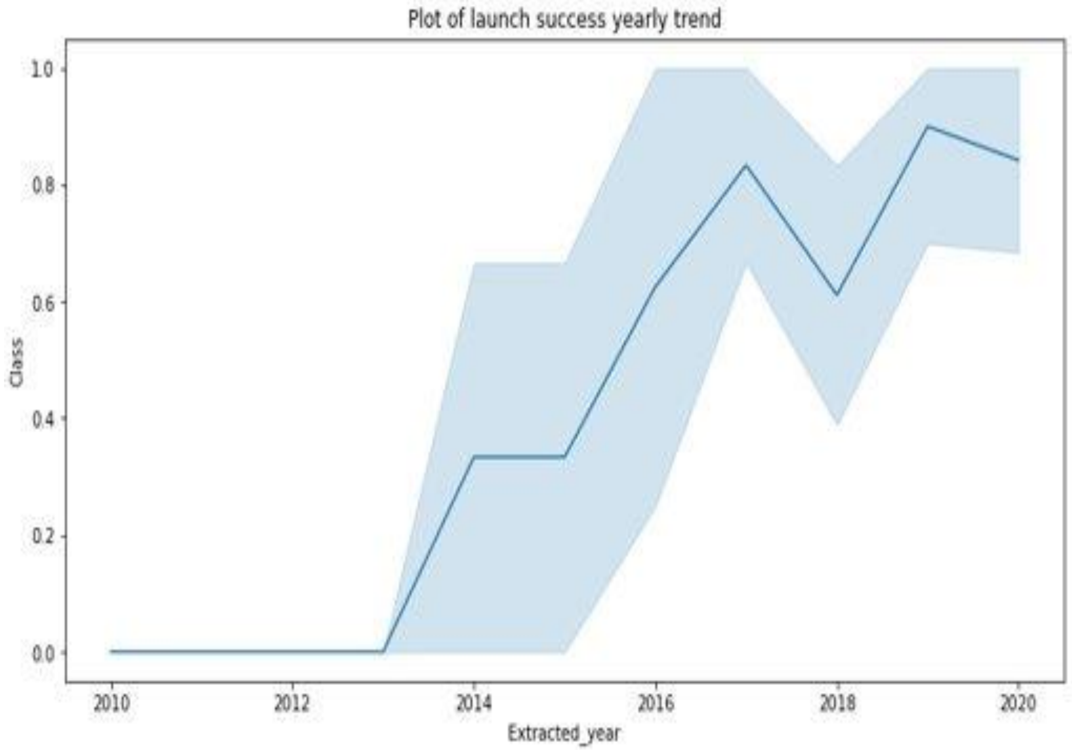
EDA with Data Visualization I

- **We want to make visualizations to select important variables that we will use to predict the success in landing.**
- Scatter plots (cat plots) were used to view the relationships of categorical independent variables like Launch Site, Flight Number, Payload Mass and orbit type. We overlay the class label in each graph.
- • A bar chart was used to visualize the success rate of each orbit type.
- • A line chart was used to visualize the launch success yearly trend.
- **The Github url is :**
- https://github.com/singulritarian7/Capstone-Spacex/blob/master/jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(1).ipynb

EDA with Data Visualization II

We will use the bar chart to determine which orbits have the highest probability of success and the line graph to show the launch success yearly trend.





EDA with Data Visualization III

We make scatter plots to see the relationships between the following attributes:

1- Payload and flight number

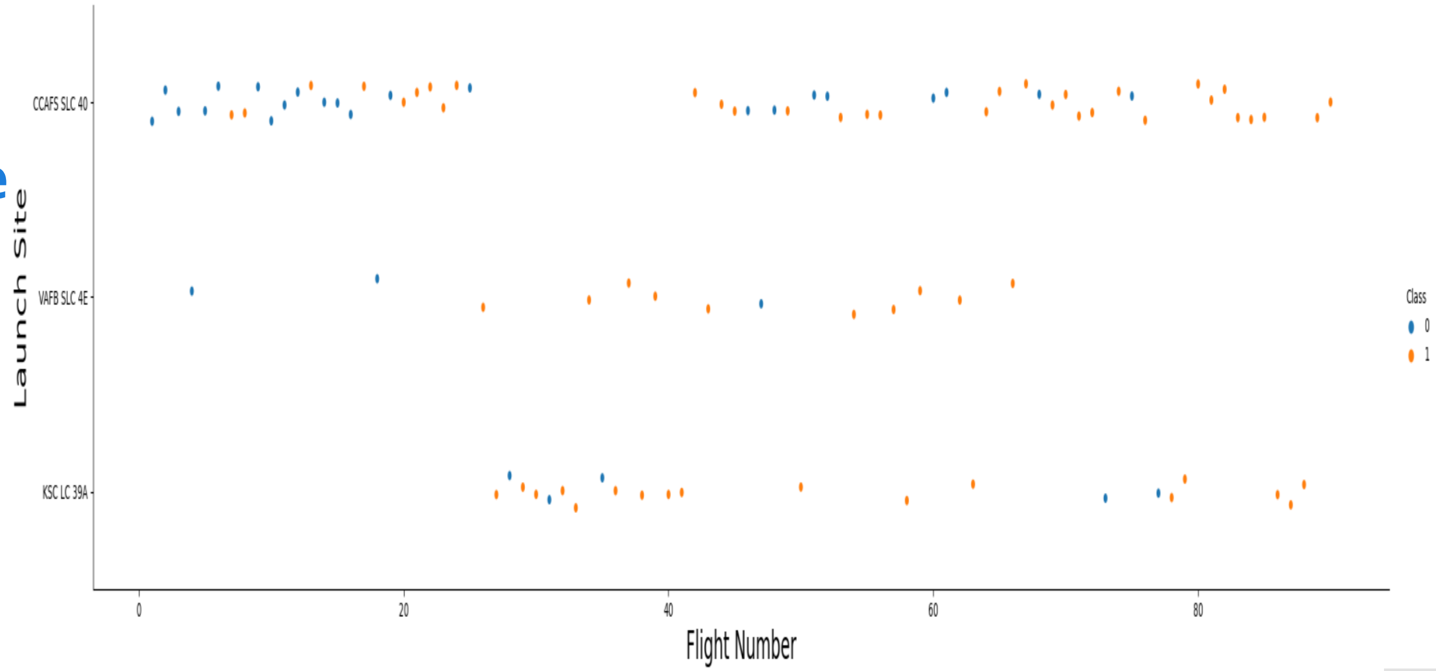
2-Flight number and launch site

3-Payload and launch site

4- Flight number and orbit type

5- Payload and orbit type

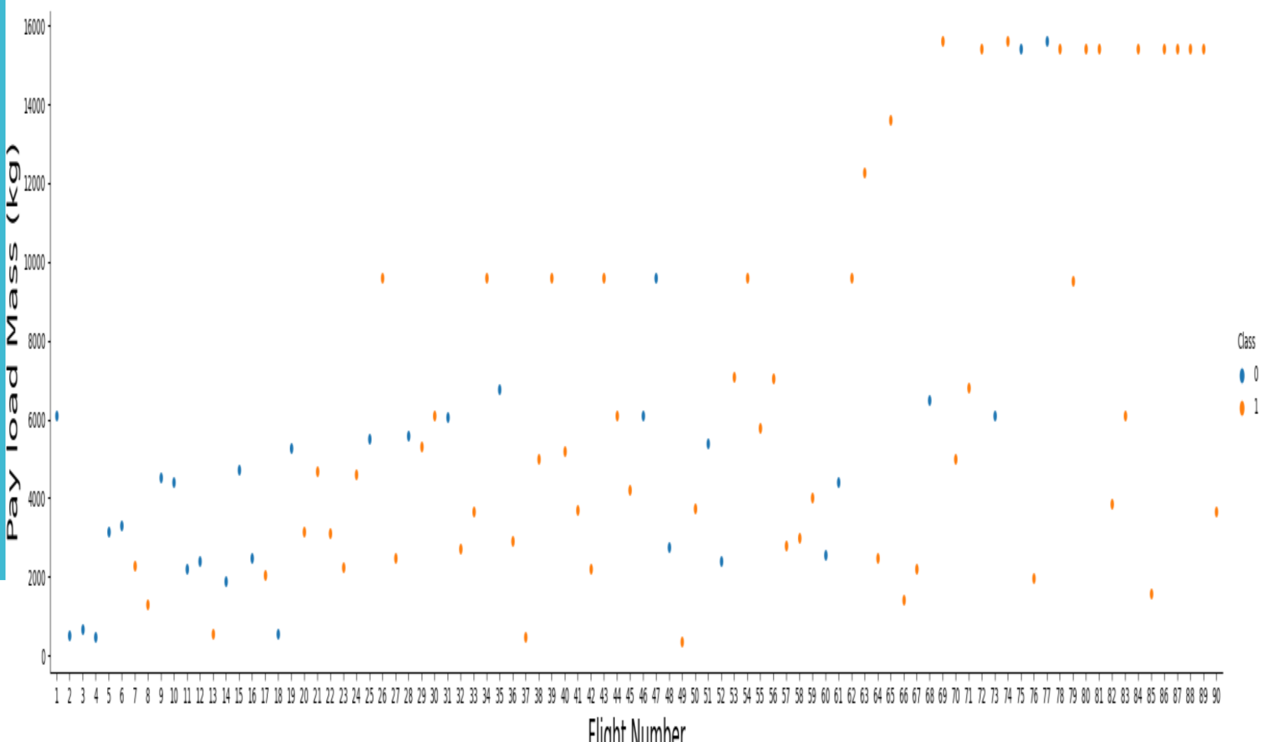
Launch site versus flight number for each class scatter plot

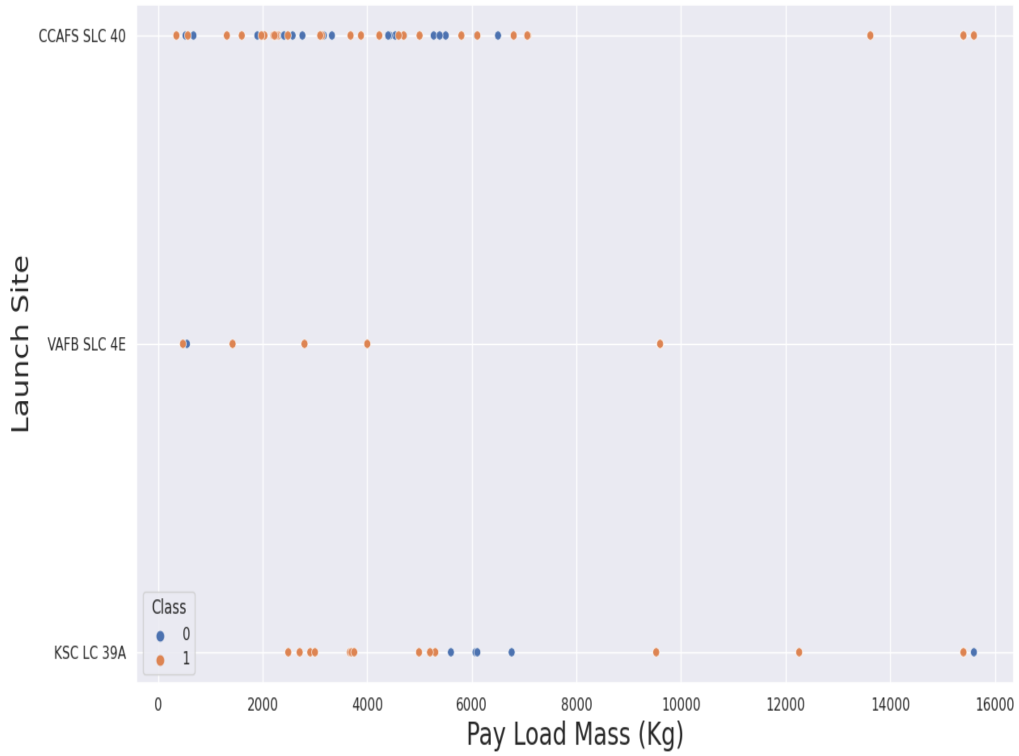


EDA with data visualization IV

We also want to observe if there is any relationship between flight number and their payload mass for wach class.

Scatter plots show depemdency of attributes on each other





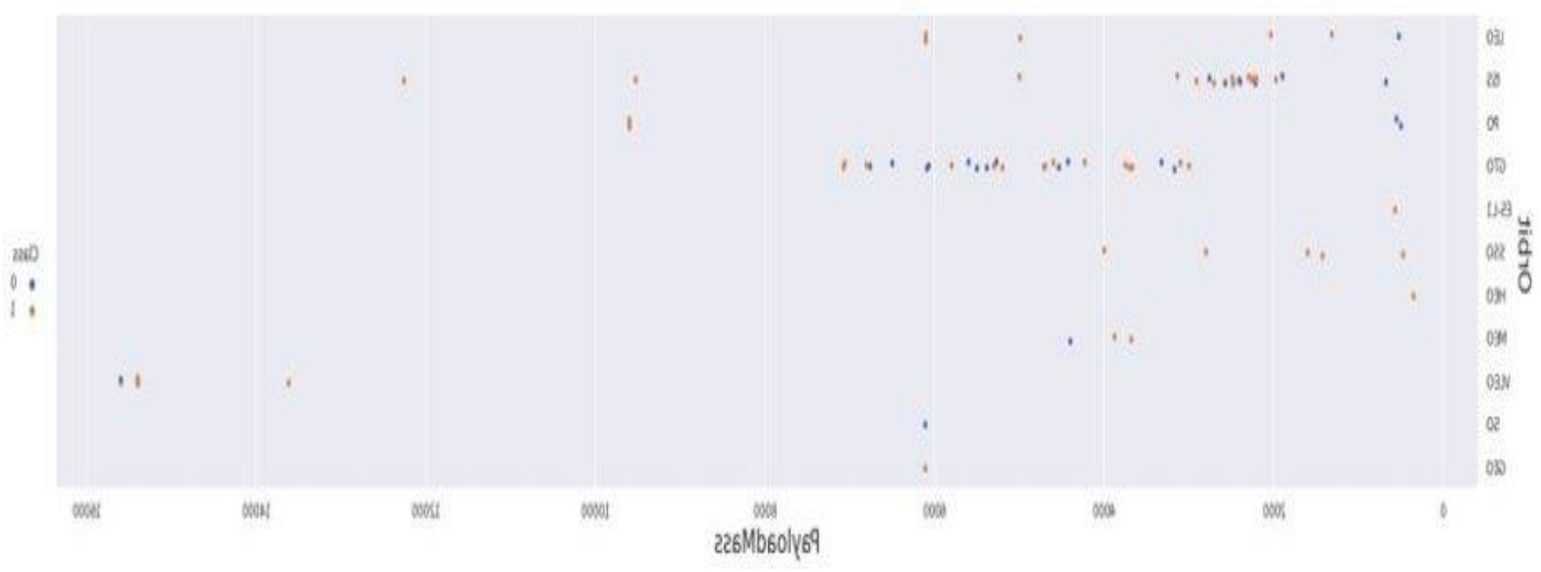
EDA with data visualization V

Visualize the relationship between Flight Number and Orbit Type for each class

EDA with data visualization VI

Plot of Payload mass versus Orbit for each class

- Visualize the relationship between Payload and Orbit type for each class , this time with an example of code.
- `sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)`
- `plt.xlabel("PayloadMass",fontsize=20)`
- `plt.ylabel("Orbit",fontsize=20)`
- `plt.show()`



EDA with SQL

- **We used the SpaceX dataset into a PostgreSQL database in the cloud using a jupyter notebook.**
- We wrote queries to find out :
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- **Add the GitHub URL**
- https://github.com/singulritarian7/Capstone-Spacex/blob/master/jupyter-labs-eda-sql-coursera_sqllite%20(6).ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
    - Are launch sites near railways, highways and coastlines.
    - Do launch sites keep certain distance away from cities.
- https://github.com/singulritarian7/Capstone-Spacex/blob/master/06_SpaceX_Interactive_Visual_Analytics_Folium%20(1).ipynb

Predictive Analysis (Classification)

We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.

We built different machine learning models and tune different hyperparameters using GridSearchCV.

We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.

We found the best performing classification model.

https://github.com/singulritarian7/Capstone-Spacex/blob/master/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1).ipynb

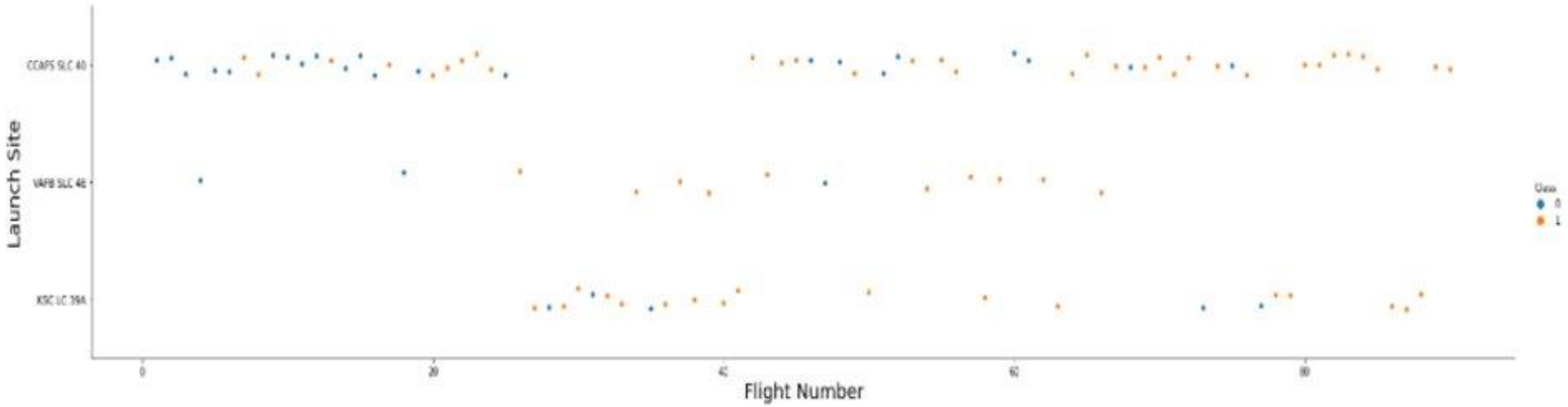
Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- The link to the notebook is
- https://github.com/singulritarian7/Capstone-Spacex/blob/master/spacex_dash_app.py

Results

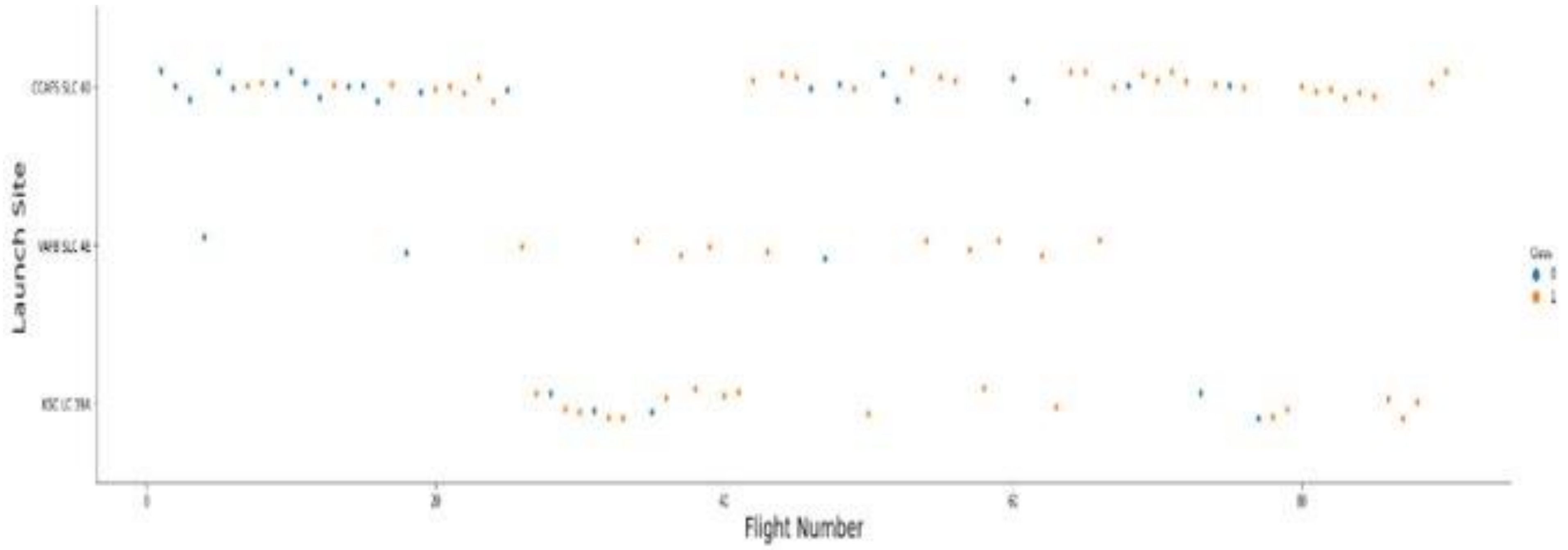
- **Exploratory data analysis results**
- **Interactive analytics demo in screenshots**
- **Predictive analysis results**

Insights drawn from EDA



Flight Number vs. Launch Site

From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.

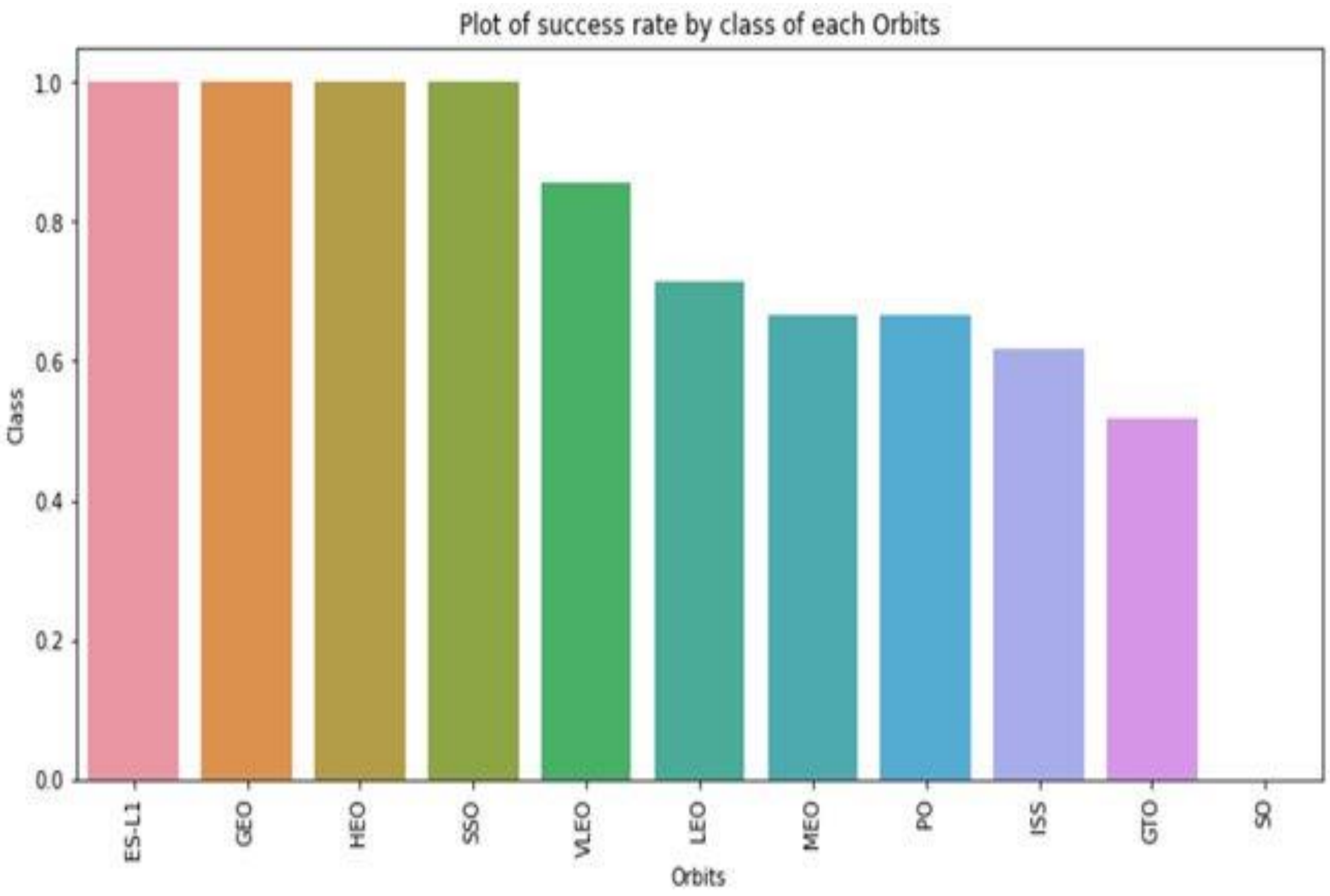


Payload vs. Launch Site

The greater the payload mass for launch site, the higher the success rate

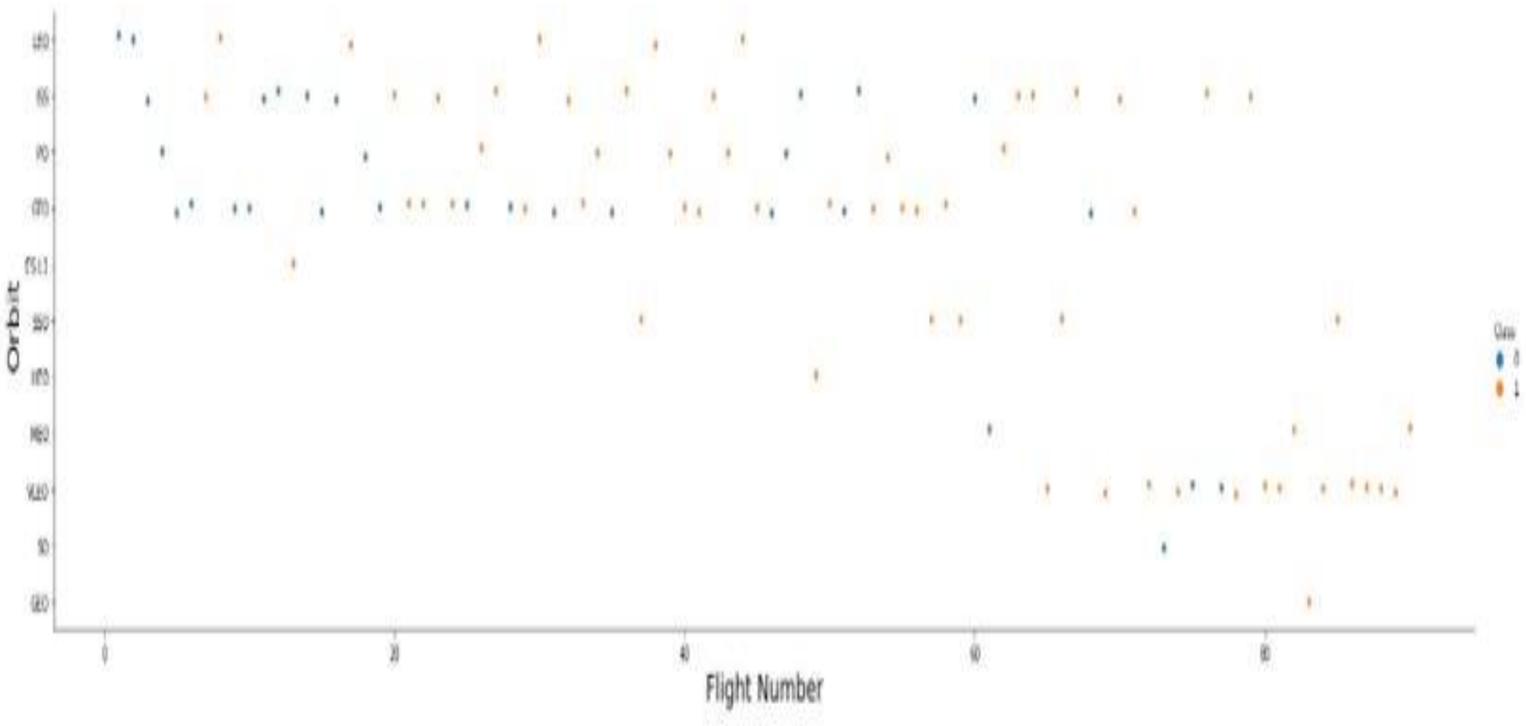
Success Rate vs. Orbit Type

From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate



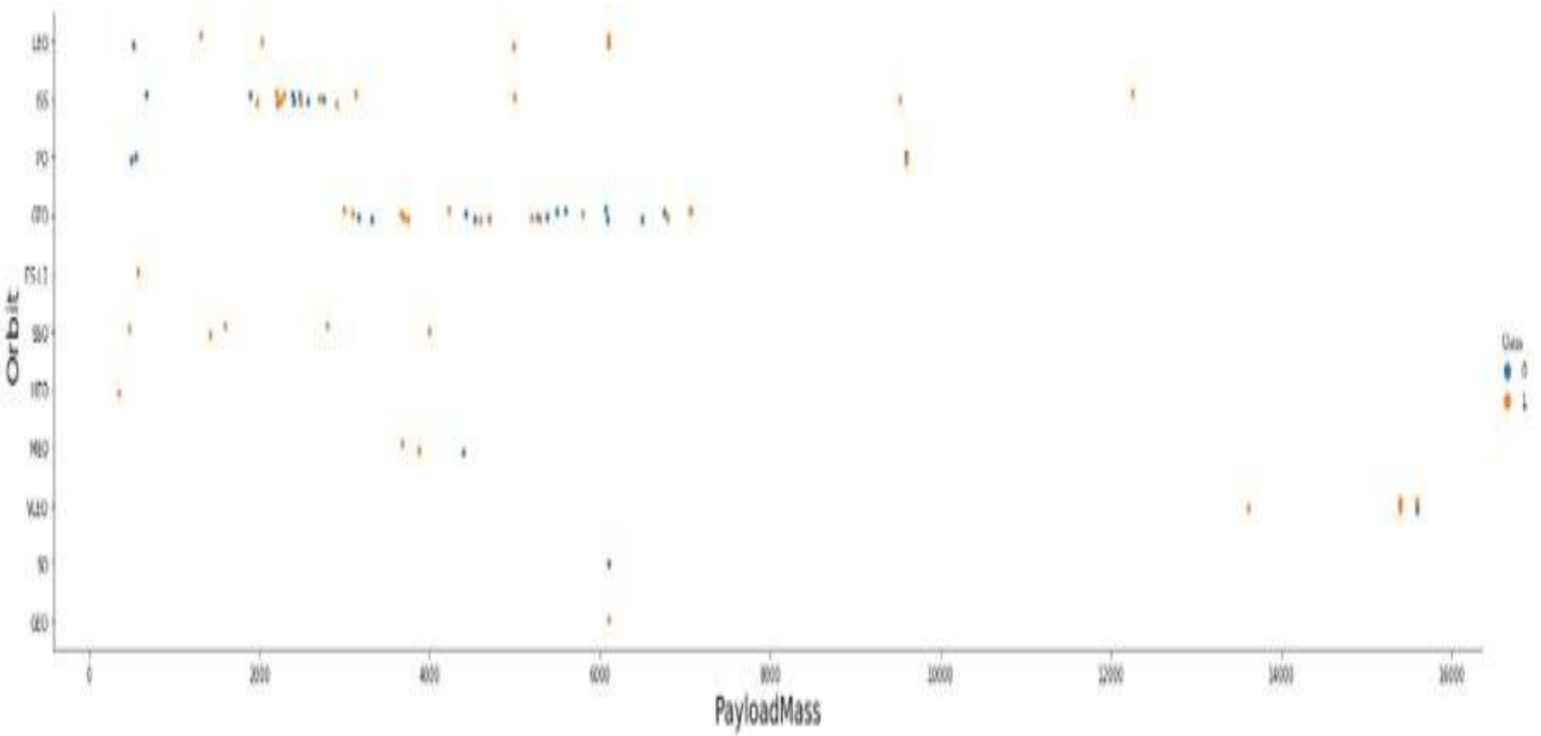
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



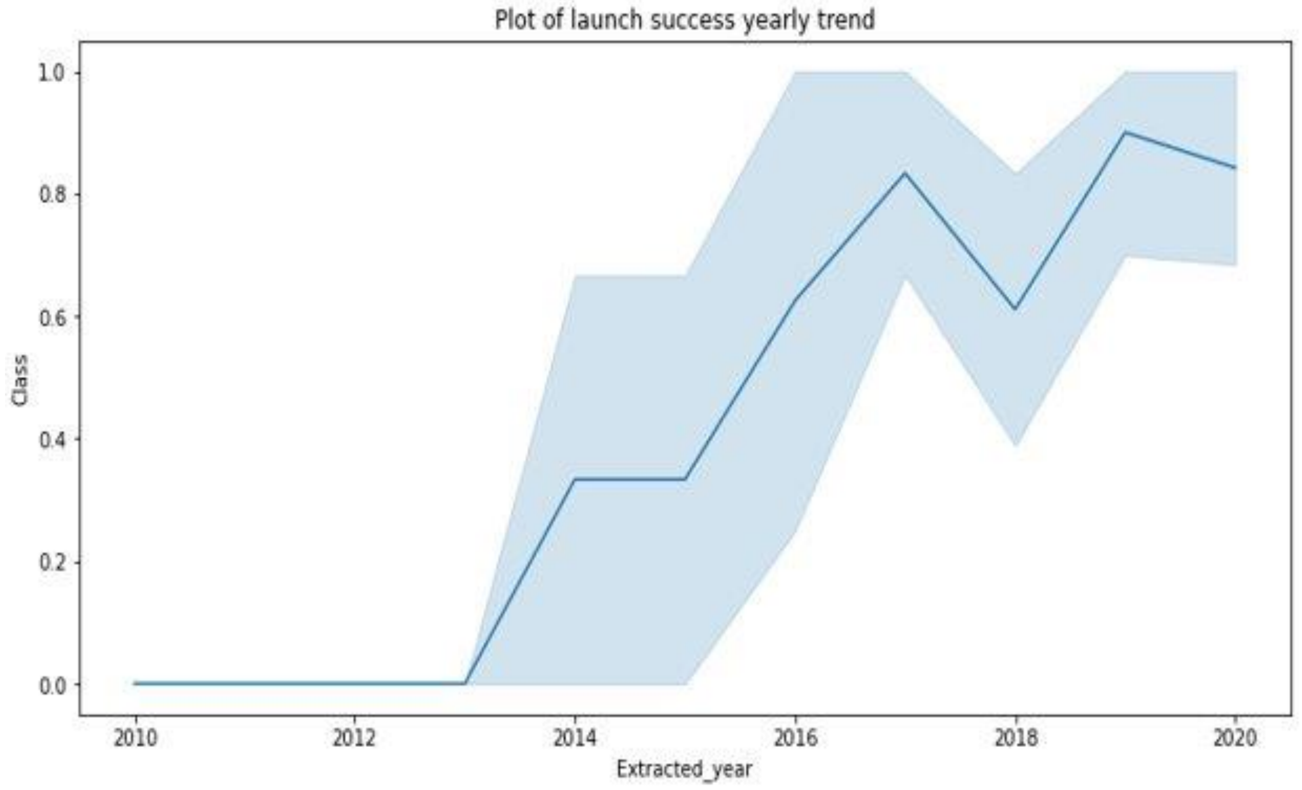
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data

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

Task 1

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

In [8]:

```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
```

Launch Site Names Begin with 'CCA'

- We used the next query to display 5 records where launch sites begin with `CCA`

- Task 2
- Display 5 records where launch sites begin with the string 'CCA'
- In [9]:
- **%sql** SELECT * \
-     FROM SPACEXTBL \
-     WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;
- See the results of the query in the next slide

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0 *	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

- **We calculated the total payload carried by boosters from NASA using the next query**
- Task 3
- Display the total payload mass carried by boosters launched by NASA (CRS)
- In [10]:
- **%sql** SELECT SUM(PAYLOAD_MASS__KG_) \
-    FROM SPACEXTBL \
-    WHERE CUSTOMER = 'NASA (CRS)';
- SUM(PAYLOAD_MASS__KG_)
- 45596.0

Average Payload Mass by F9 v1.1

- **We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4**
- Task 4
- Display average payload mass carried by booster version F9 v1.1
- In [11]:
- **%sql** SELECT AVG(PAYLOAD_MASS__KG_) \
-   FROM SPACEXTBL \
-   WHERE BOOSTER_VERSION = 'F9 v1.1';
- AVG(PAYLOAD_MASS__KG_)
- 2928.4

First Successful Ground Landing Date

- **We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015**
- Task 5
- List the date when the first succesful landing outcome in ground pad was acheived.
- *Hint:Use min function*
- In [12]:
- **%sql** SELECT min(date) from SPACEXTBL where "Landing_Outcome" = 'Success (ground pad)
- min(date)
- 01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- **We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000**
- Task 6
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- **%sql** SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)';
- booster_version
- F9 FT B1022
- F9 FT B1026
- F9 FT B1021.2
- F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- **We used wildcard like '%' to filter for WHERE MissionOutcome was a success or a failure. The number of success missionoutcomes are 101.**
- Task 7
- List the total number of successful and failure mission outcomes
- In [ ]:
- **%sql** SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
- COUNT(*)
- 101

Boosters Carried Maximum Payload I

- **We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.**
- Task 8
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- In [14]:
- **%sql** SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
- **See the results in the next slide**

Boosters Carried Maximum Payload II

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 I

- **We used the WHERE clause to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015**
- Task 9
- **List the records which will display the month names, failure lan ding_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.**
- **Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for yea r.**
- In [38]:
-
```
%sql SELECT substr(Date,4,2) as month,
DATE,BOOSTER_VERSION, LAUNCH_SITE,
'Landing _Outcome' \
FROM SPACEXTBL \
where 'Landing _Outcome' = 'Failure (drone ship)'
 and substr(Date,7,4)='2015';
```
- See the results in the next slide

month	Date	Booster_Version	**Launch_Site**	Landing _Outcome
01	10-01-2015	F9 v1.1B1012	CCAFS LC-40	Failure (drone ship)
04	14-04-2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

2015 Launch Records II Results

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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

- Task 10
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- In [37]:
- `%sql SELECT [Landing _Outcome], count(*) as count_outcomes \`
- `FROM SPACEXTBL \`
- `WHERE DATE between '04-06-2010' and '20-03-2017' group by [Landing _Outcome] order by count_outcomes DESC;`
- See the results in the next slide

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

Landing _Outcome	count_outcomes
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

Section 4

Launch Sites Proximities Analysis

Folium Map Summary

- • Folium Markers were used to show the SpaceX launch sites and their nearest important landmarks like railways, highways, cities and coastlines.
- • Polylines were used to connect the launch sites to their nearest land marks.
- • Furthermore, Folium Circles were used to highlight circle area of launch sites.
- • In order to mark the success/failed launches for each site, marker clusters were used on the map. Whereby Red represents rocket launch failures while Green represents the successes.
- Url of Github notebook, you can open it in Colab too to see the images.
- https://github.com/singulritarian7/Capstone-Spacex/blob/master/Week_3_(A)Interactive_Visual_Analytics_with_Folium_(1).ipynb

Launch sites global map markers

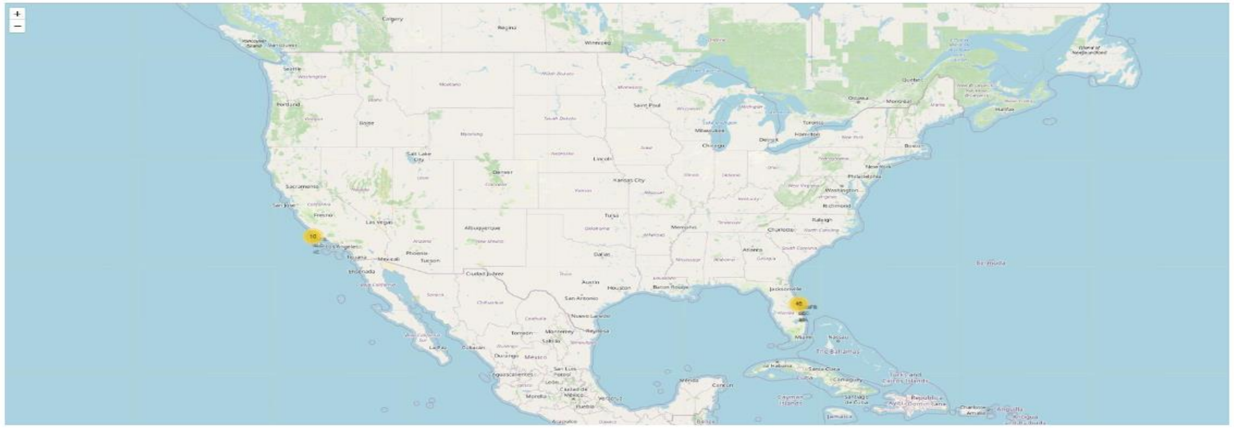
Now, you can explore the map by zoom-in/out the marked areas , and try to answer the following questions:

- Are all launch sites in proximity to the Equator line?
- Are all launch sites in very close proximity to the coast?

Launch Sites

With Markers

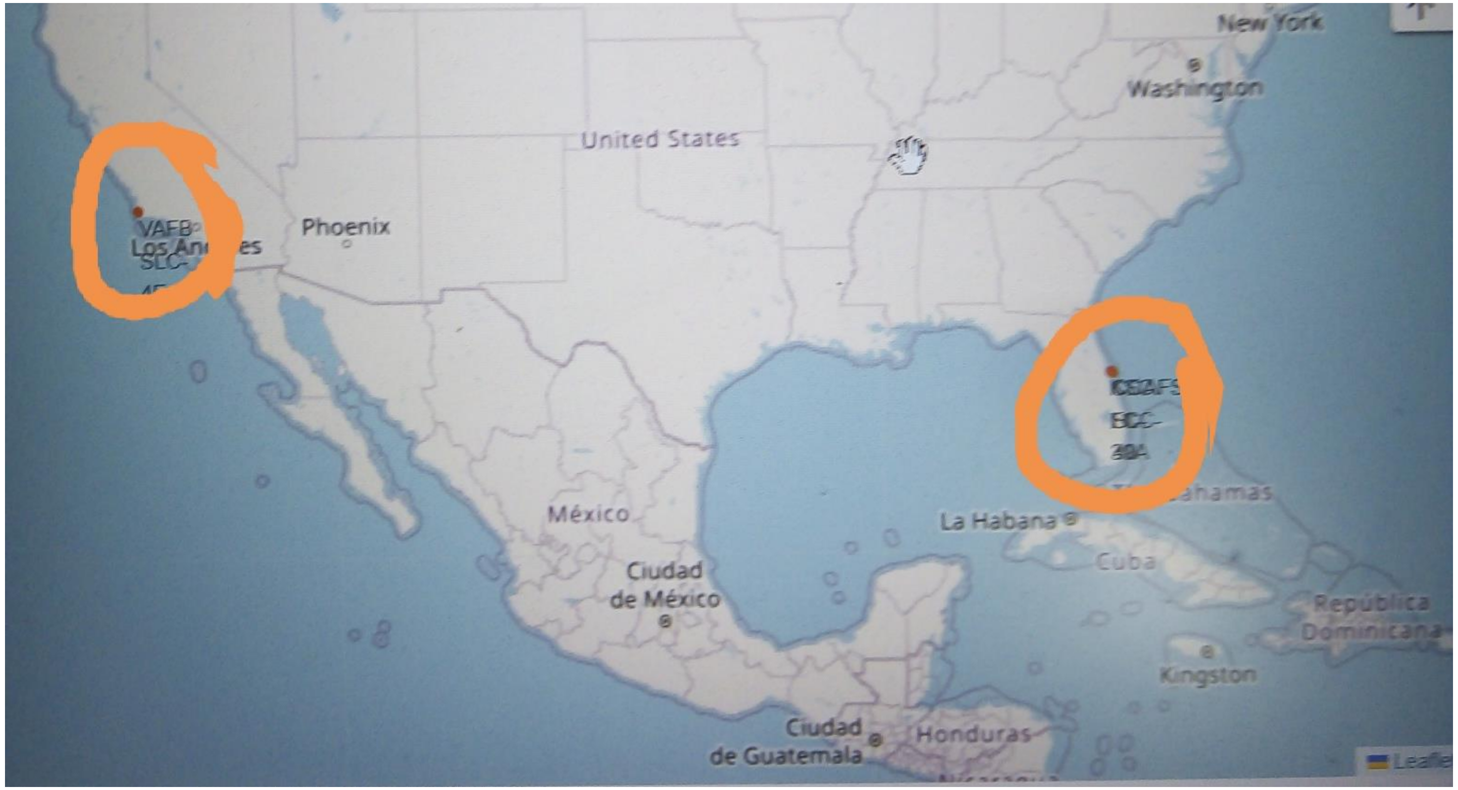
- **Near Equator**: the closer the launch site to the equator, the **easier** it is **to launch** to equatorial orbit, and the more help you get from Earth's rotation for a prograde orbit. Rockets launched from sites near the equator get an **additional natural boost** - due to the rotational speed of earth - that **helps save the cost** of putting in extra fuel and boosters.



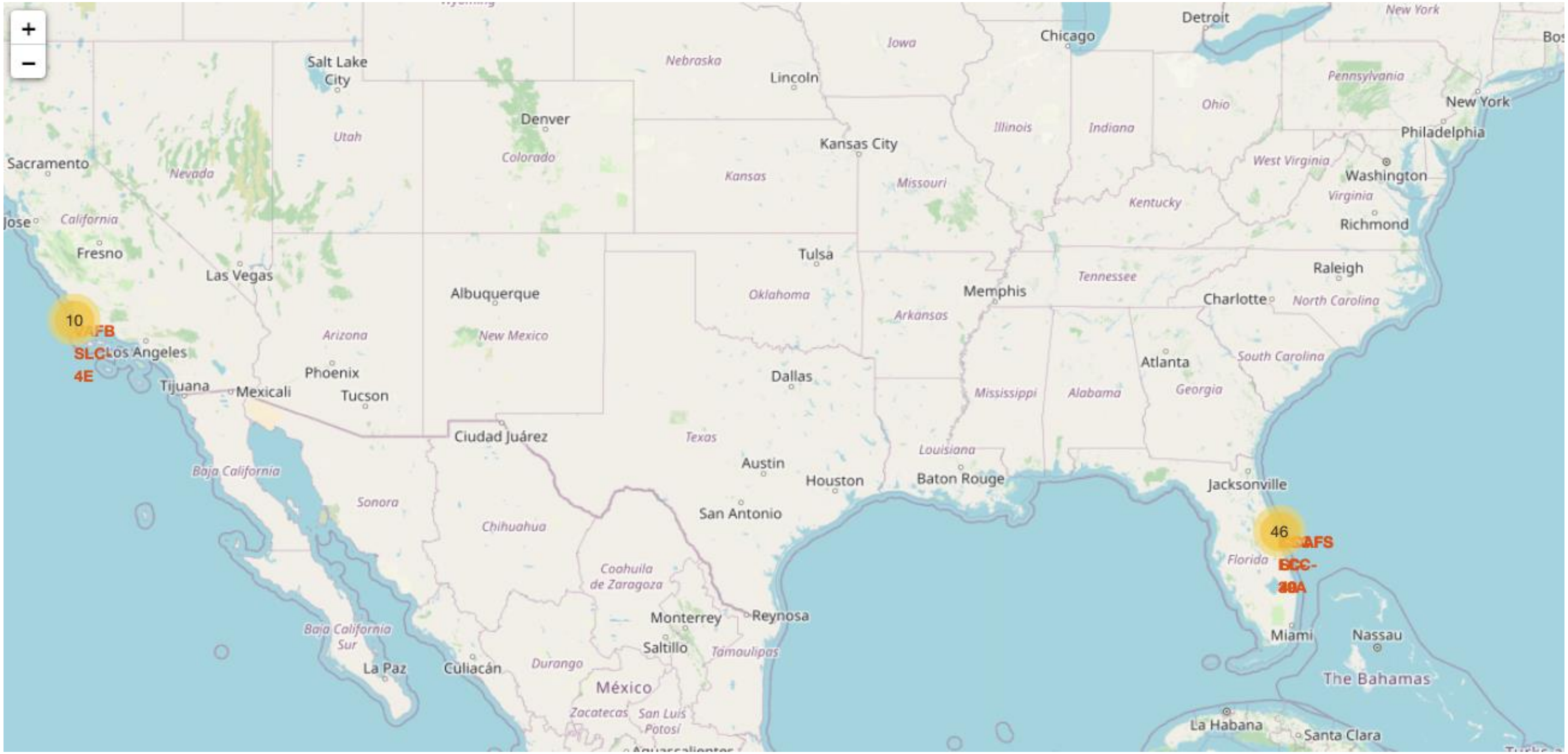
2023

Launch Sites Locations

- 1- We can see that all launch sites are near the coast line.
- 2- Also launch sites are near to the ecuator line, his makes sense as it takes less fuel to get into space from the **equator**



For each launch result in **spacex** data frame, we added a **folium marker** to **marker cluster**



Markers showing launch sites with color labels

Sucess rate of Spacex launches

- The successful launches are represented by a green marker while the red marker represents failed rocket launches.
- It appears that **KSC LC 39A** has the highest success rate of rocket launches

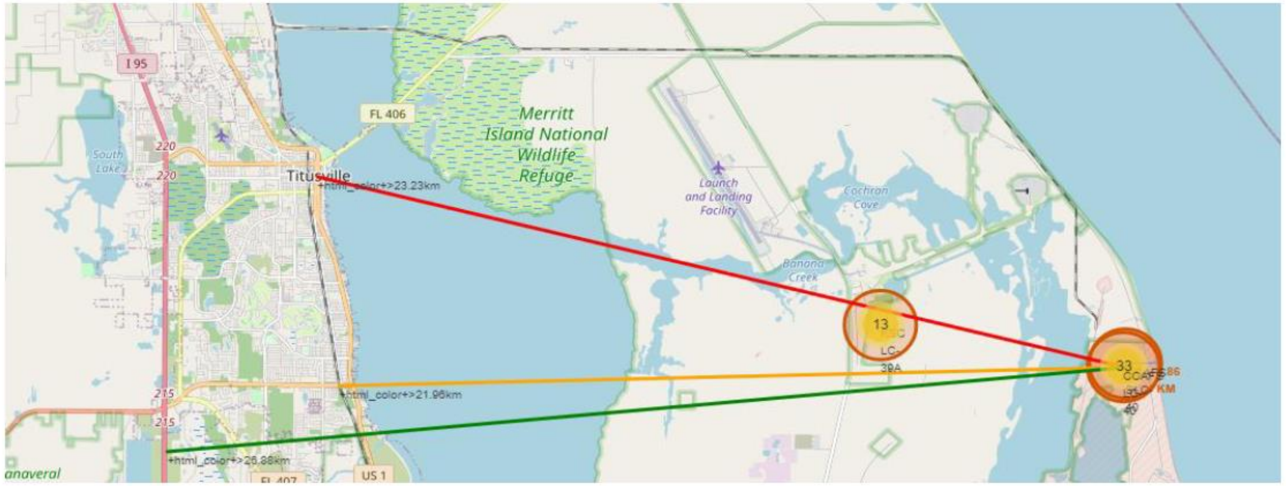
Calculate the distances between a launch site to its proximities

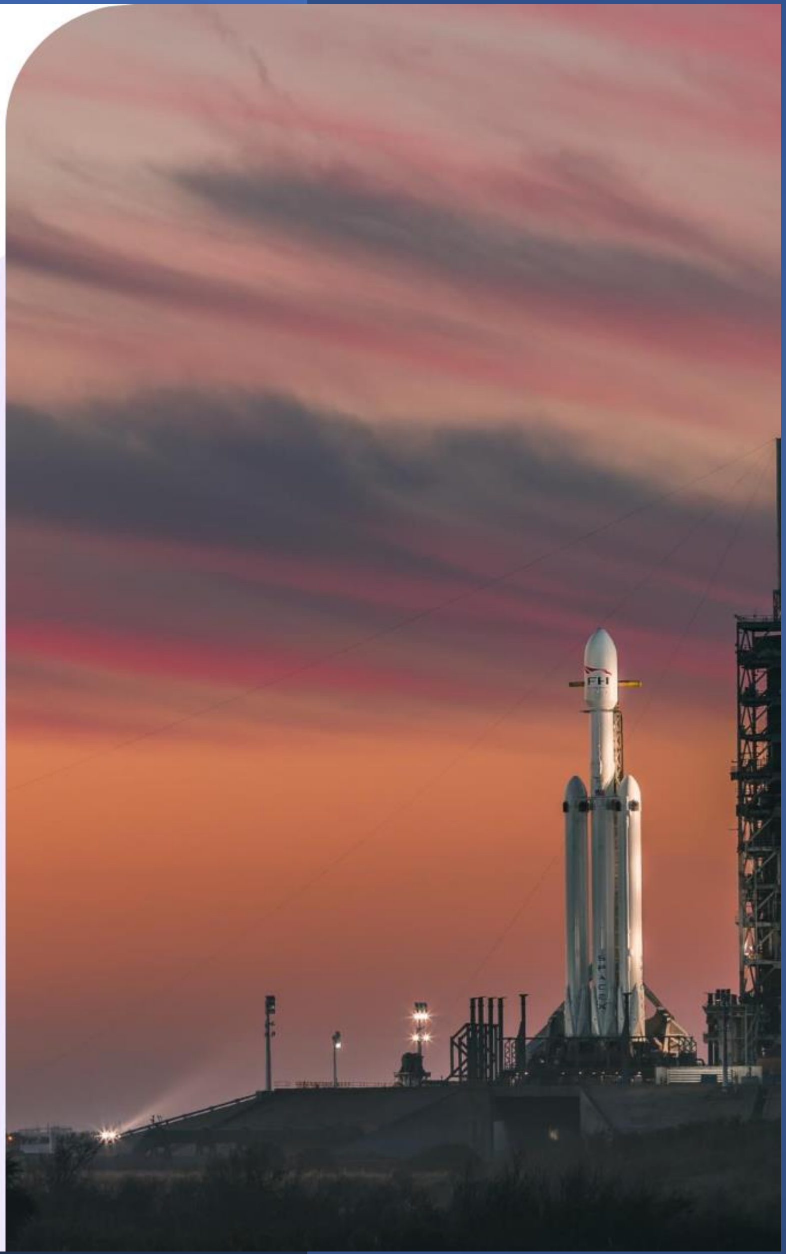
```
] print("City Distance", city_distance)

  print("Railway Distance", railway_distan

  print("Highway Distance", highway_distan

  print("Coastline Distance", distance_coa

  City Distance 23.234752126023245
  Railway Distance 21.961465676043673
  Highway Distance 26.88038569681492
  Coastline Distance 0.8627671182499878
```

Distance to Proximities

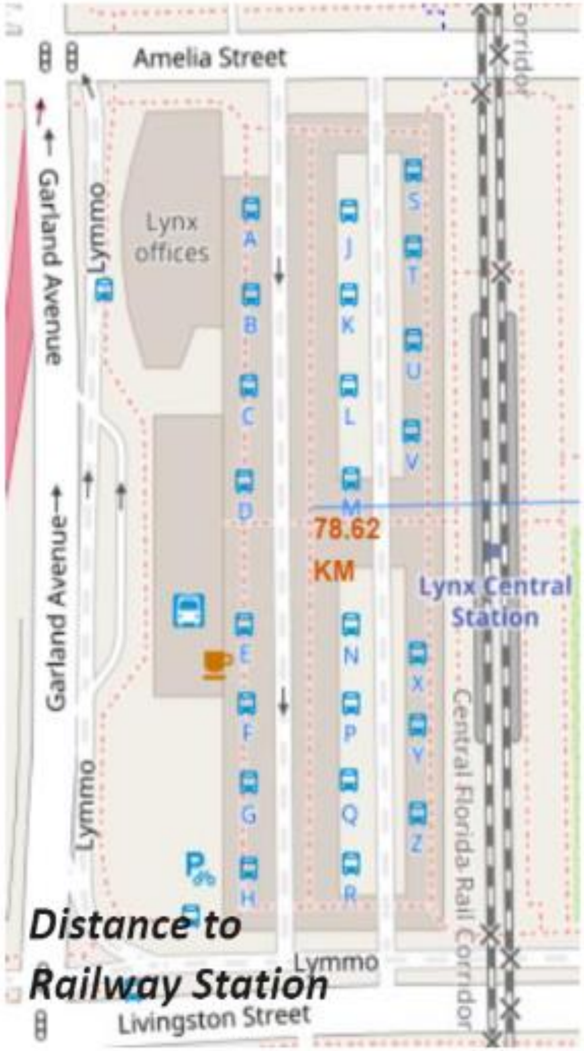
CCAFS SLC-40

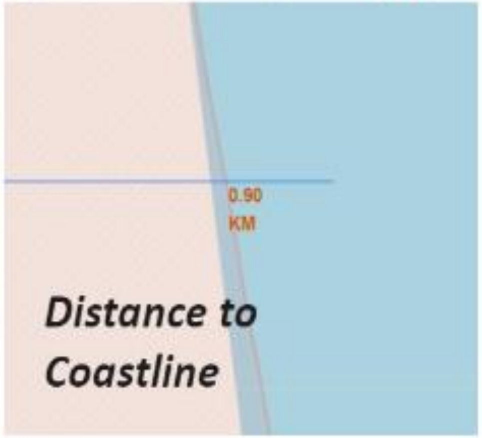
- **.86 km** from nearest coastline
- **21.96 km** from nearest railway
- **23.23 km** from nearest city
- **26.88 km** from nearest highway





Launch Site distance to landmarks





- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Pollylines: Launch Sites to Landmarks

- Launch sites are set up at about 20 km from cities, and one reason can be to prevent any crashes near populated areas.-
- Launch sites are in very close proximity to railways and highways, maybe because it's necessary transportation requirements for rocket parts.
- The sites are close to the coastline possibly due to the many rocket landing tests on the ocean .

Section 5

Build a Dashboard with Plotly Dash

Build a Dashboard with Plotly Dash Summary

- Pie charts and scatter charts were used to visualize the launch records of SpaceX.
- These charts displayed the rocket launch success rate per launch site. The factors that may have been influencing mostly the success rate at each site are the the payload mass and booster versions.
- Successful launches were represented by 1 while failures were represented by 0.

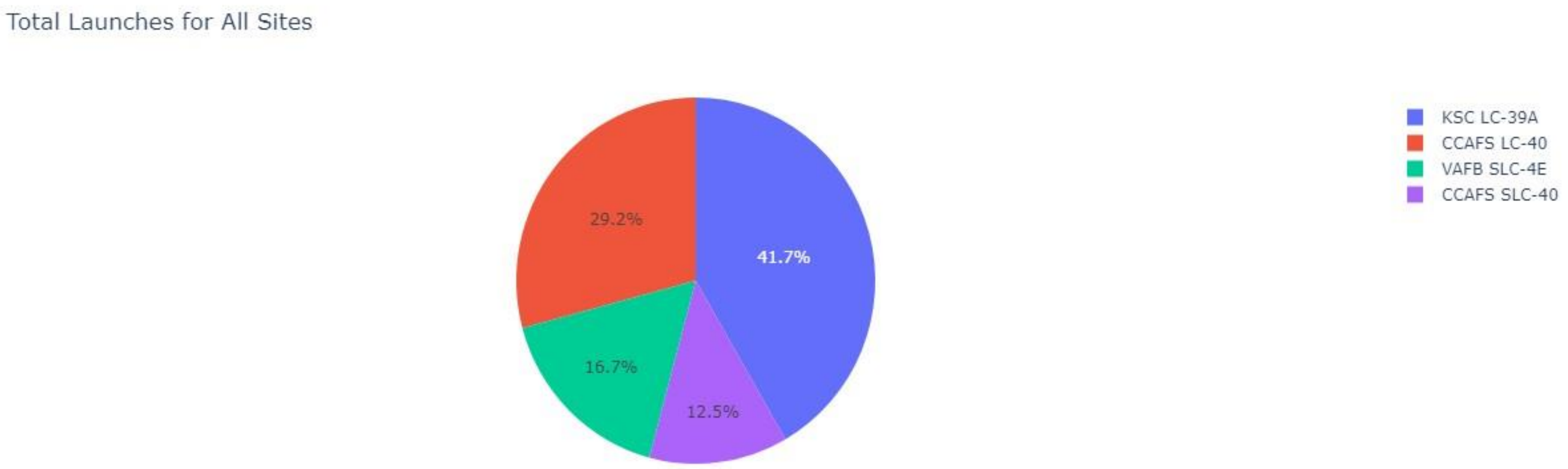
https://github.com/singulritarian7/Capstone-Spacex/blob/master/Piechart_allsites.jpg

https://github.com/singulritarian7/Capstone-Spacex/blob/master/Slider%20dash.jpg

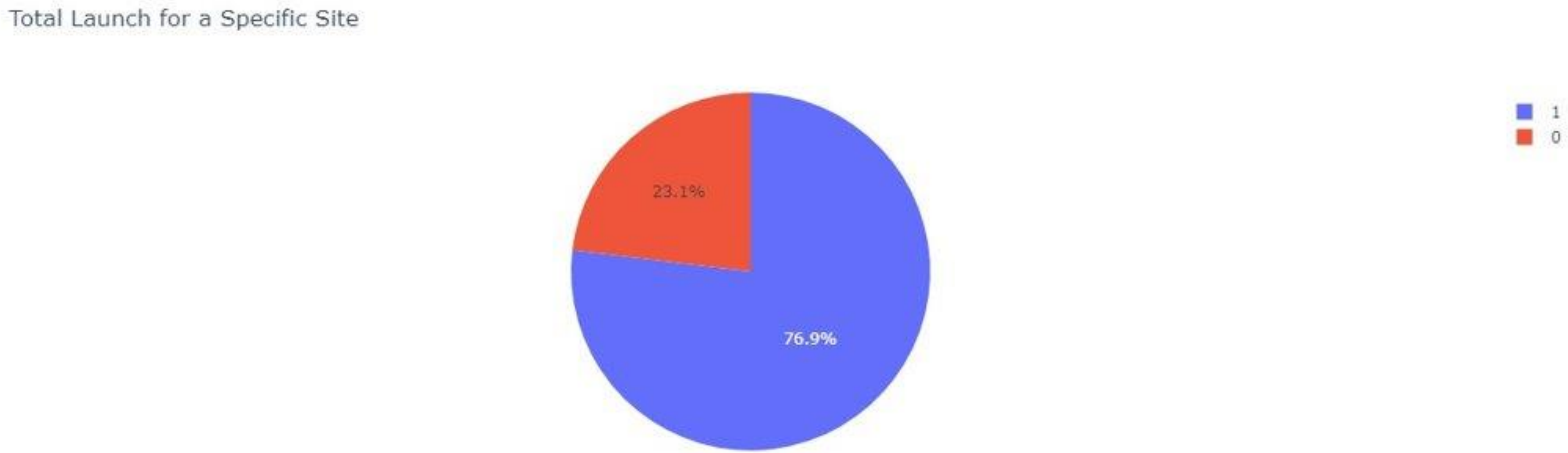
https://github.com/singulritarian7/Capstone-Spacex/blob/master/spacex_dash_app.py

Success launches by all sites

- We have the success rates of each launch site
- Site KSC LC-39A has the most successful launches of all the sites



Total successful Lunches for site KSC LC-39A
As we can see the success ratio is 97 %



Payload Mass vs. Launch Success for All Sites

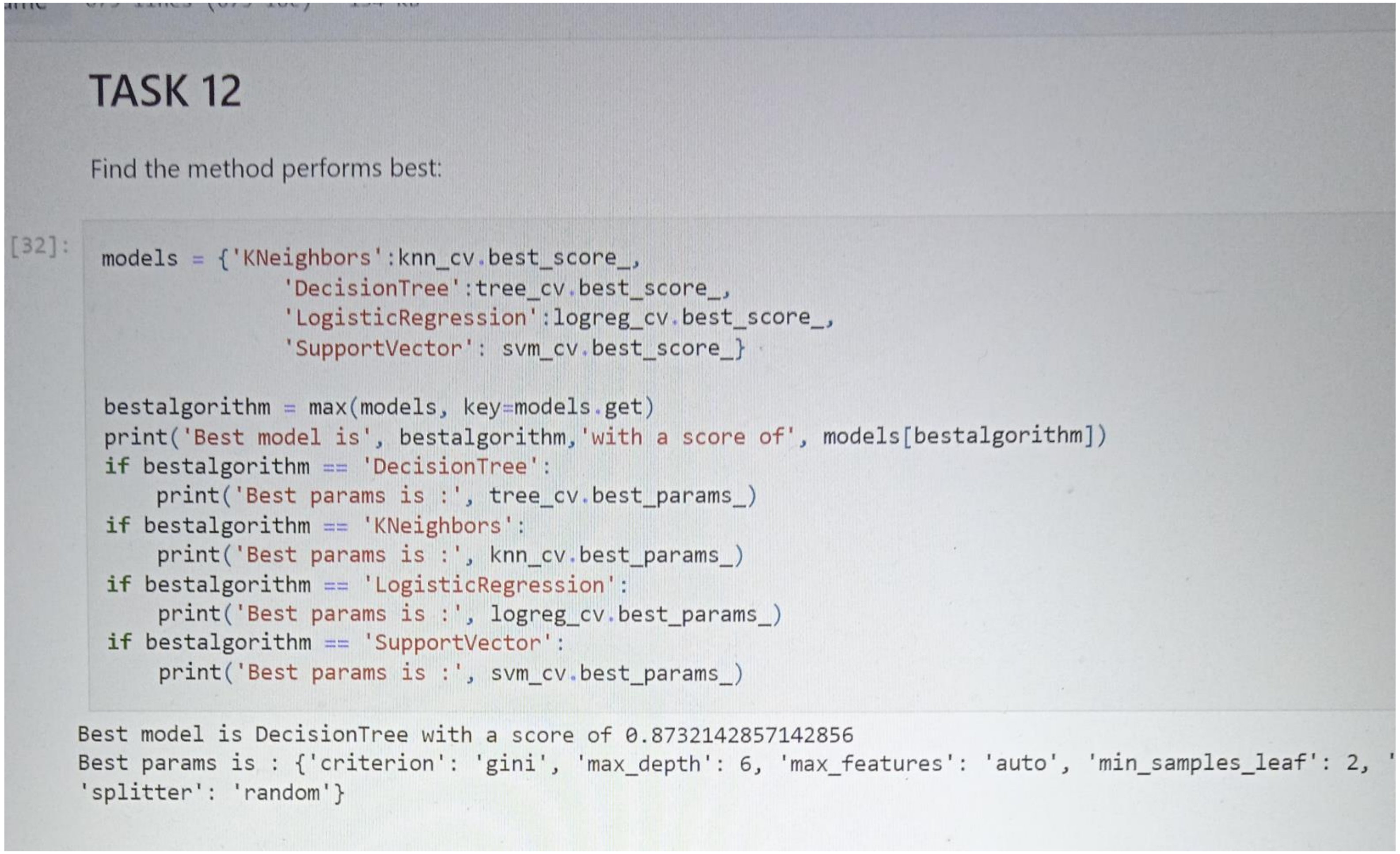
- Payload range between 0-2500 kg has very low success rate for every booster version
- Payload range between 2500 –5000kg , we see the booster version FT has the highest success rate and v1.1 the lowest

Section 5

Predictive Analysis (Classification)

Classification Accuracy

The Decision Tree classifier has the higher accuracy , 0.87



TASK 12

Find the method performs best:

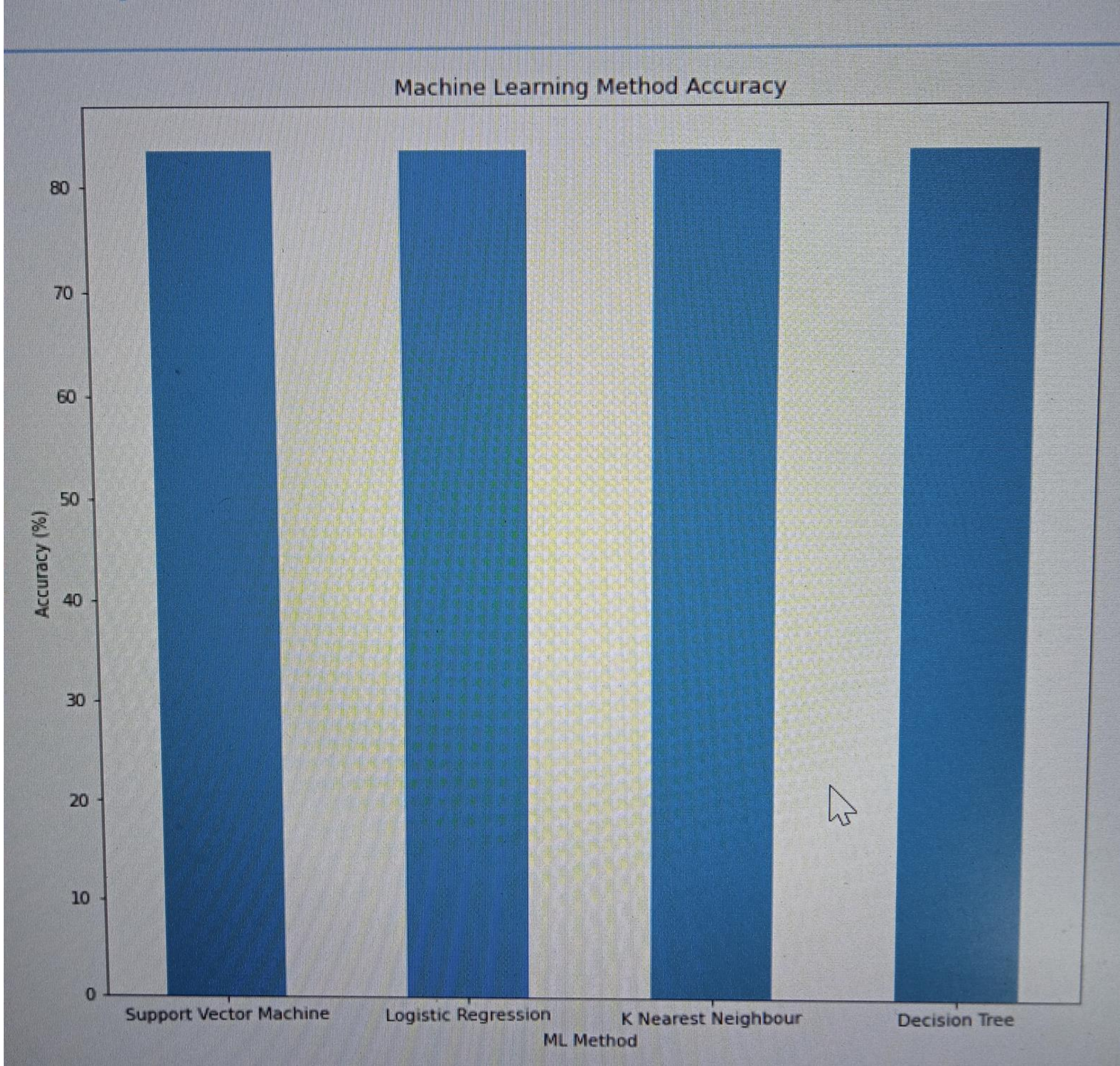
[32]:

```
models = {'KNeighbors':knn_cv.best_score_,
          'DecisionTree':tree_cv.best_score_,
          'LogisticRegression':logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, '
'splitter': 'random'}
```

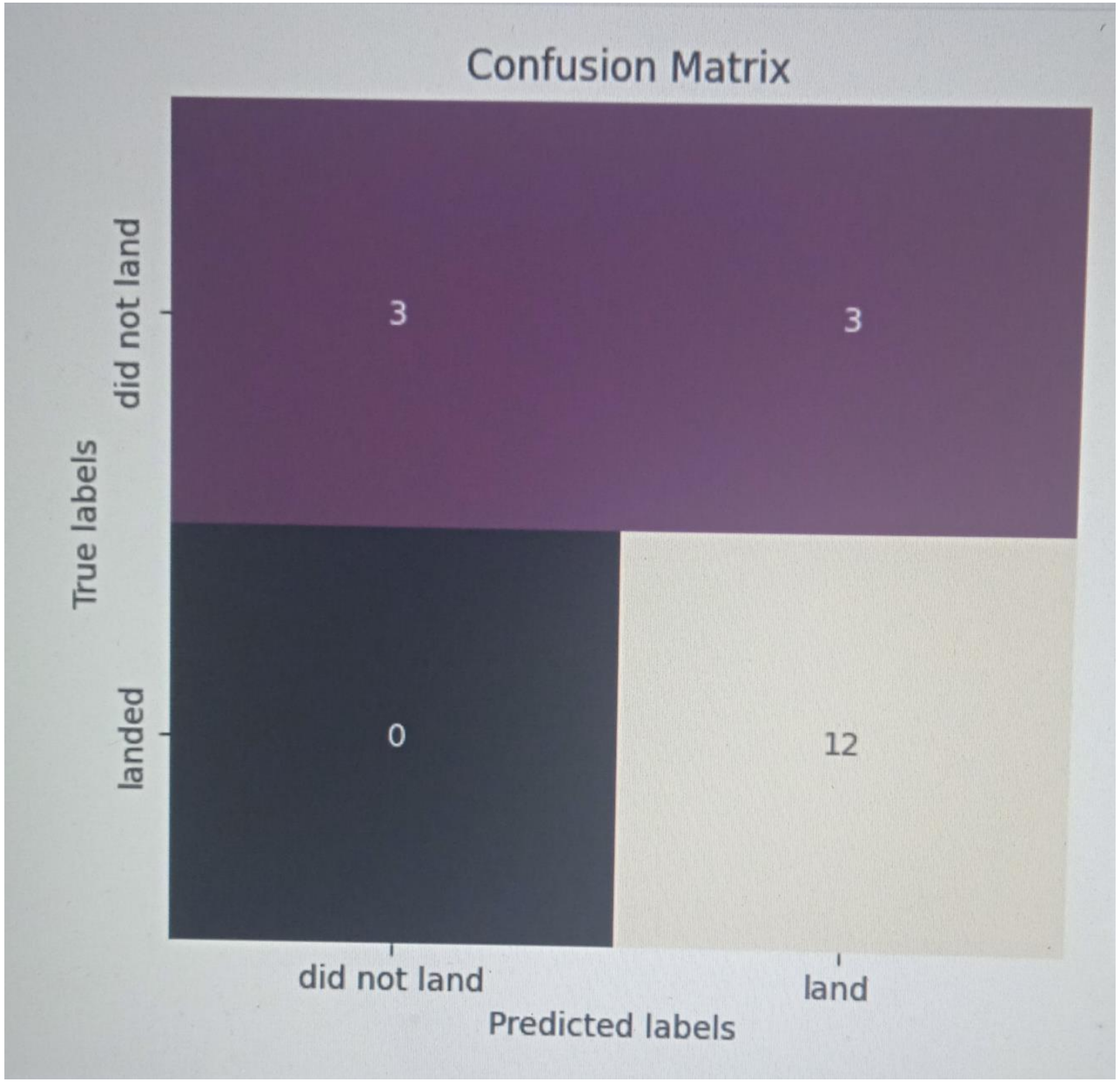
Classification Accuracy

Bar chart of the accuracy of the different methods



Confusion Matrix

- The chart shows the confusion matrix of the decision tree classifier
.- The model only failed to predict 3 labels



Conclusions

- All the launch sites are located near the coast, away from nearby cities
- Site KSC LC-39A had the highest launch success rate of all the launch sites
- Since 2015 the success rate of launches increased with flight number
- Payload range between 0-2500 kg has very low success rate for every booster version
- Payload range between 2500 –5000kg , we see the booster version FT has the highest success rate and v1.1 the lowest
- This data was used to train a ML model, decision tree classifier, to predict the landing outcomes of rocket launches with 87 % of accuracy

Appendix

- SQL query of launch sites beginning with "CCA"
- Github repo: https://github.com/singulritarian7/Capstone-Spacex

Task 2

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

```
In [ ]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://kcq64325:***@dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[ ]:

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	None	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	None	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	None	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	None	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	None	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Thank you!