Winning Space Race with Data Science

Fayed Emad

22 December 2022

Table Of Contents

Executive Summary

The commercial space age is here,companies are making space travels affordable for everyone.
And perhaps the most successful one is SpaceX.
SpaceX launch Falcon 9 at almost a third of a price of others.

The Problem

Competing with spaceX, to do that we need to know why SpaceX sends travels to space for an inexpensive cost compared to others?

Key Figures

- SpaceX reusing the first stage.
- Different launch sites have different success rate.
- There is a relationship between payload mass and success rate

General Solution

- Give the appropriate factors for the success of the launch.
- Predict if the first stage will land or not for reusing, that will reduce the cost.

Taken steps to solve it?

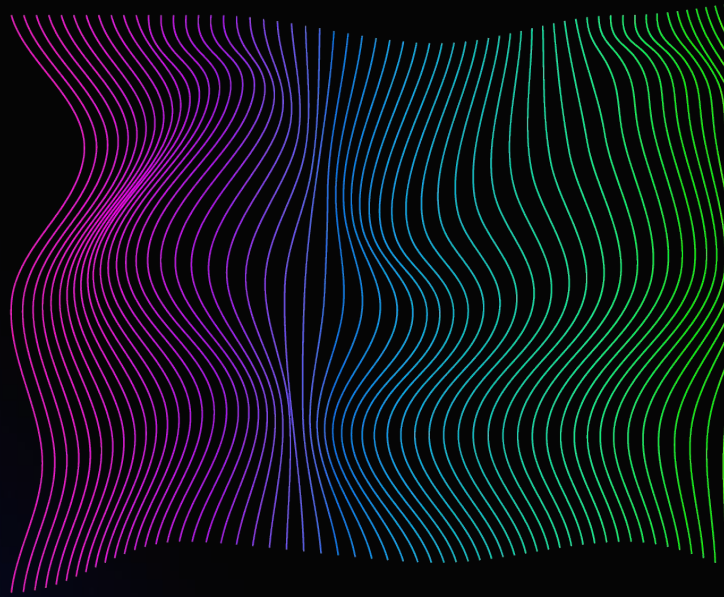
1 - Collected some data about spaceX F9 past launches. From it's API and scraping the Wikipedia.

2 - Did some EDA to gain insights about the data.

3- Trained and evaluated a ML model and obtained a good ML model accuracy.

OUR COMPANY

SpaceY a company that trying to make space travels affordable for everyone.



Introduction

The commercial space age is here,companies are making space travels affordable for everyone. And perhaps the most successful one is SpaceX, Which launches rockets for a relatively low cost compared to other competitors.

Our goals:

- Gather some informations about spaceX launches.
- Reduce the cost of the launches.

Section 1

Methodology

Executive Summary

Data collection methodology

- Fetch spaceX API
- web scraping the wikipedia

Perform data wrangling

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

Perform Exploratory Data Analysis (EDA) Using Visualisations and SQL

Perform interactive visual analytics using Folium and Plotly Dash

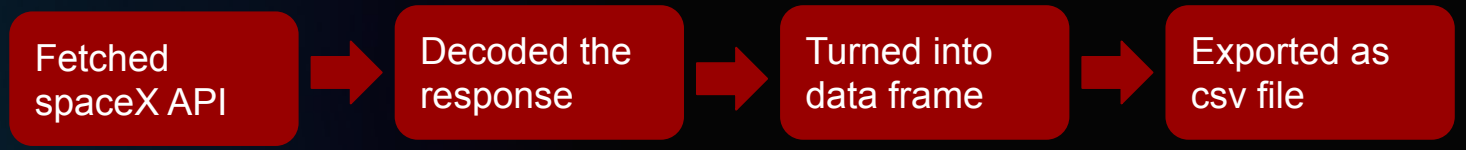
- We used folium to draw circles, markers, lines etc on the map.
- Plotly Dash for building a dashboard.

Perform predictive analysis using classification model.

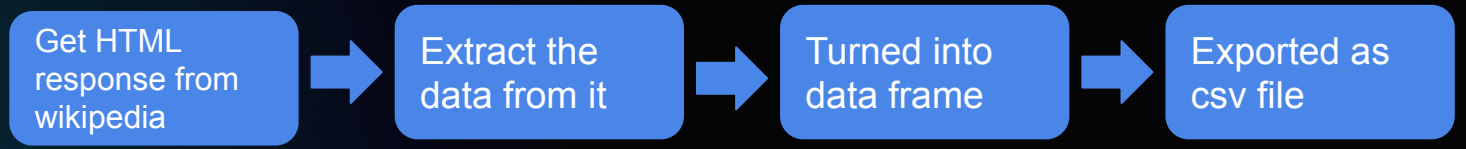
* Build some models , tuned it with gridsearch, evaluated it with accuracy, precision, ROC, AUC, recall, F1-score using scikit-learn.

Data Collection

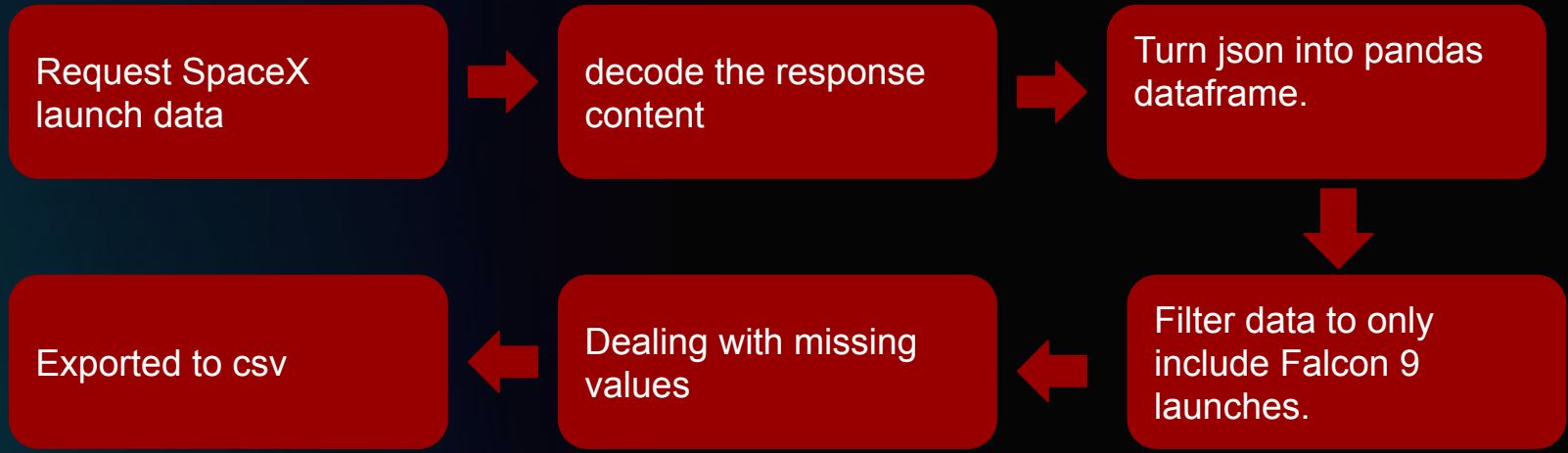
- By fetching spaceX API



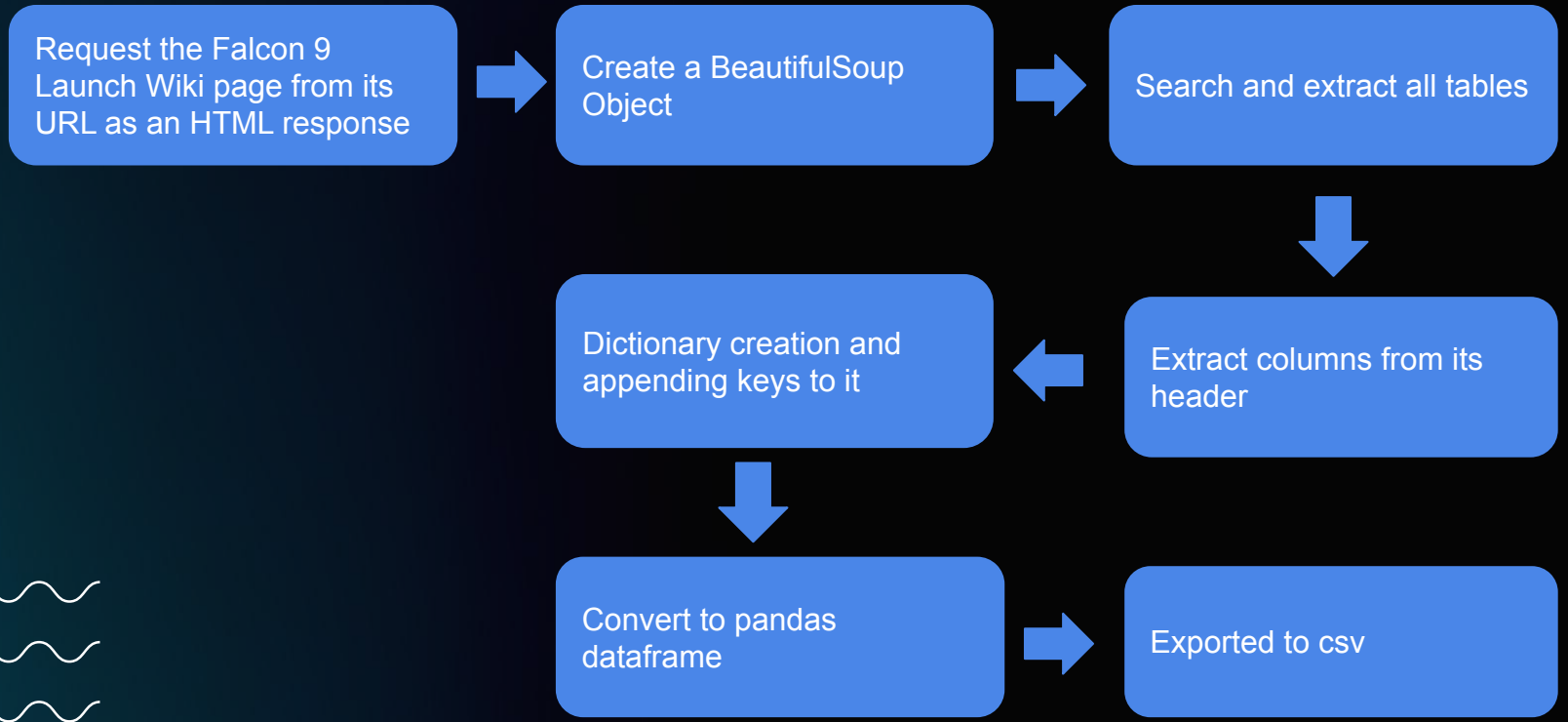
- Web scraping wiki



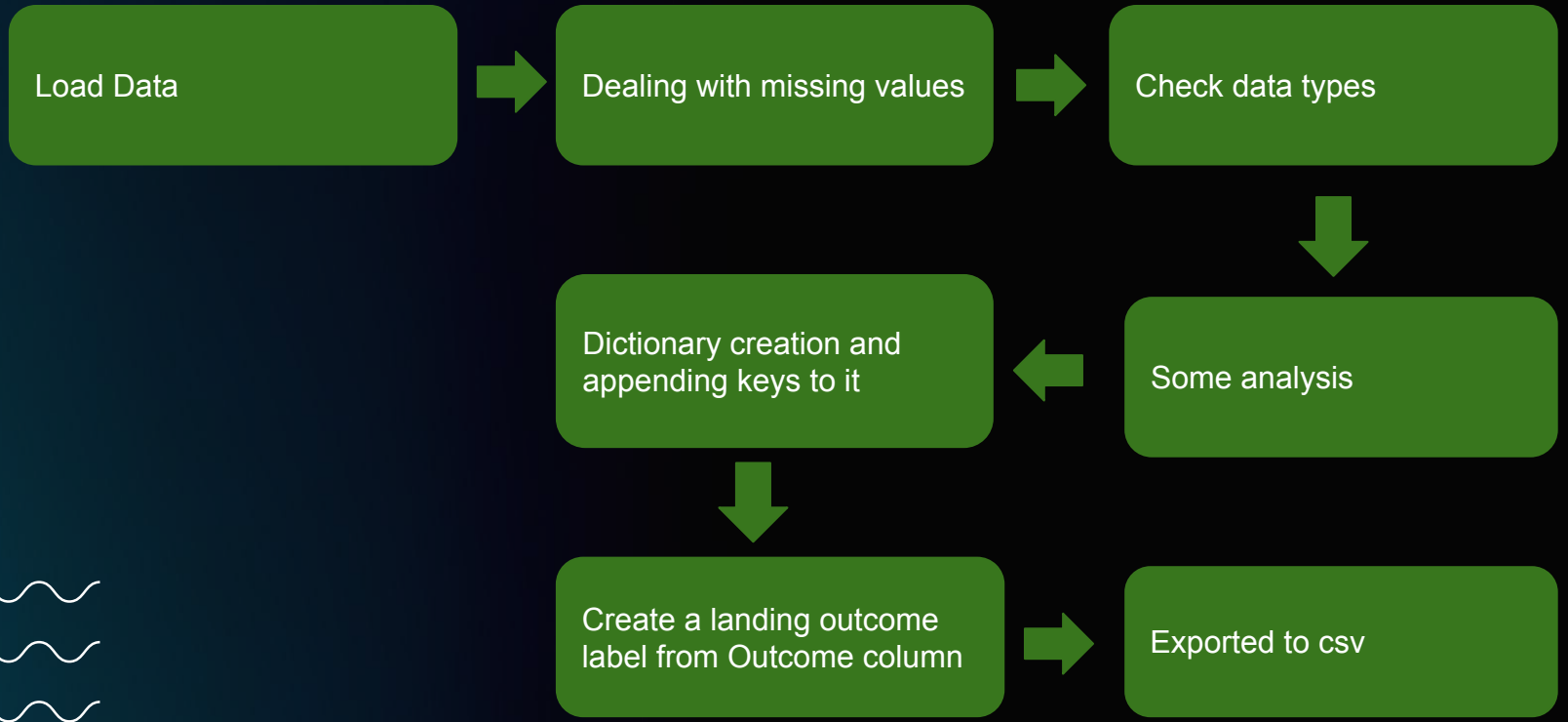
Data Collection - SpaceX API



Data Collection - Web scraping



Data Wrangling



EDA with Data Visualization

We used about three types of charts like:

- **Scatter plot** to find the relationship between features

- **Line chart** for Time Series to find the development of a variable over time

- **Bar plot** to find relationship between categorical data

EDA with SQL

After loading data into DB, we queried some data like:

- Small queries

- Displayed the names of the unique launch sites
- Displayed 5 records where launch sites begin with the string 'CCA'
- Displayed the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.

- Big queries

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

Build an Interactive Map with Folium

We used some map objects such as markers, circles, lines, etc, and added to a folium map:

- We used **Markers:** to mark all launch sites on a map, success/failed launches for each site on the map and some distances.
- A **MarkerCluster: To mark all record in every launch site.**
- **Circles:** to highlight the launch area
- **MousePosition:** to get coordinate for a mouse over a point on the map.
- **PolyLine:** to draw a line between a launch site to the selected coastline point, railway, highway and city.

Build a Dashboard With Plotly Dash

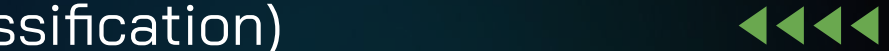
We used two types of plots:

- **Pie Chart:** to show the Percentage of successful launches for each site
- **Scatter Plot:** to show the effect of the payload mass on the success rate

We added some interactive to the dashboard like:

- **Dropdown** to choose what launch site you need to show its charts
- **Slider** to select a specific range for payload mass to get its impact on the success rate.

Predictive Analysis (Classification)



Preprocessing

- Standardize the data using StandardScaler().
- Did some feature engineering.
- Split the data into Train and Test with 80:20 scale.

Build
(using train data)

- trained four types of models using GridSearchCV() to hypertune the models.

Evaluate
(using test data)

- Used many evaluation metrics like accuracy, precision, recall, f1, AUC.
- Plotted the confusion matrix for the models.
- Plotted the ROC curve for every model.

Compare and choose and save

- Compared each model and choosed the best one based on its:
  1 - precision-recall and f1
  2 - ROC-AUC
  3 - Accuracy
  4 - Train time, predict time
- Saved the best model using pickle module.

Results

- Exploratory data analysis results.
- Interactive analysis results.
- Predictive analysis results.

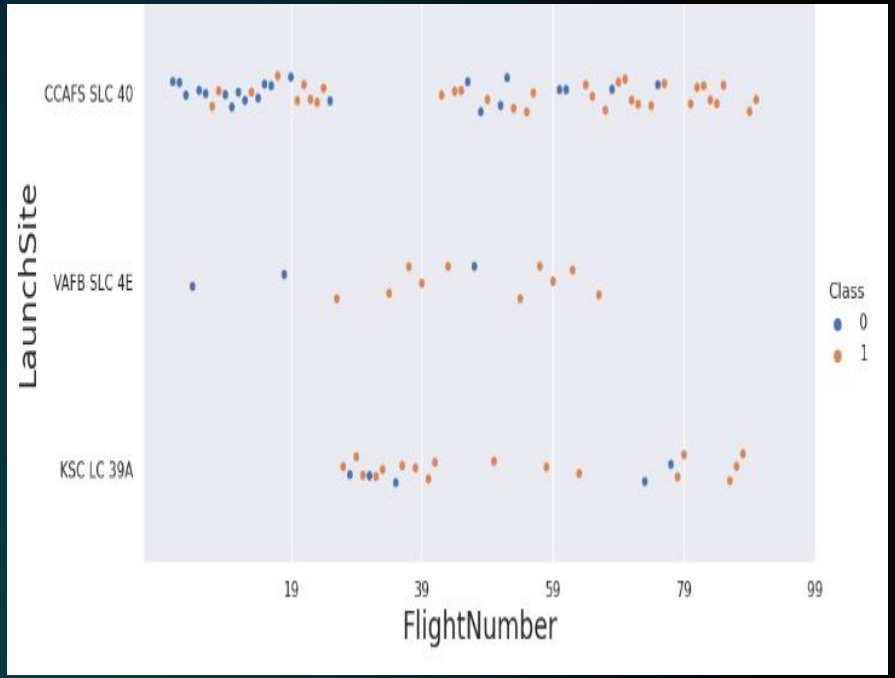
Section 2

Insights drawn from EDA

Flight Number Vs Launch Site

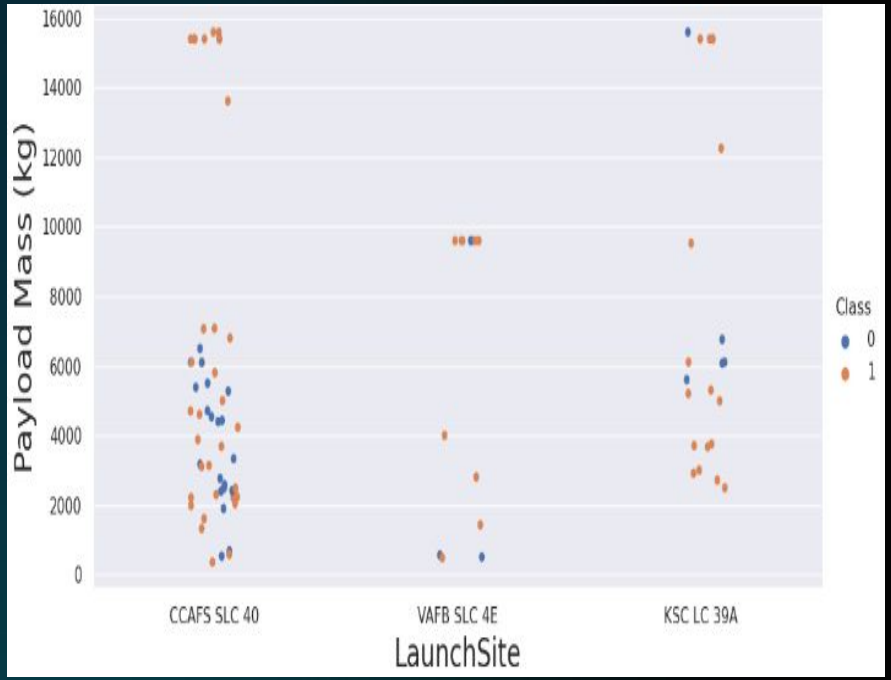
- As we can see the most flights launched from **CCAFS SLC 40** site.
- **KSC LC 39A** have a high success rate
- **VAFB SLC 4E** has the least Number of flights

The most interesting thing is the success rate increased after 30 flights and keeped increasing



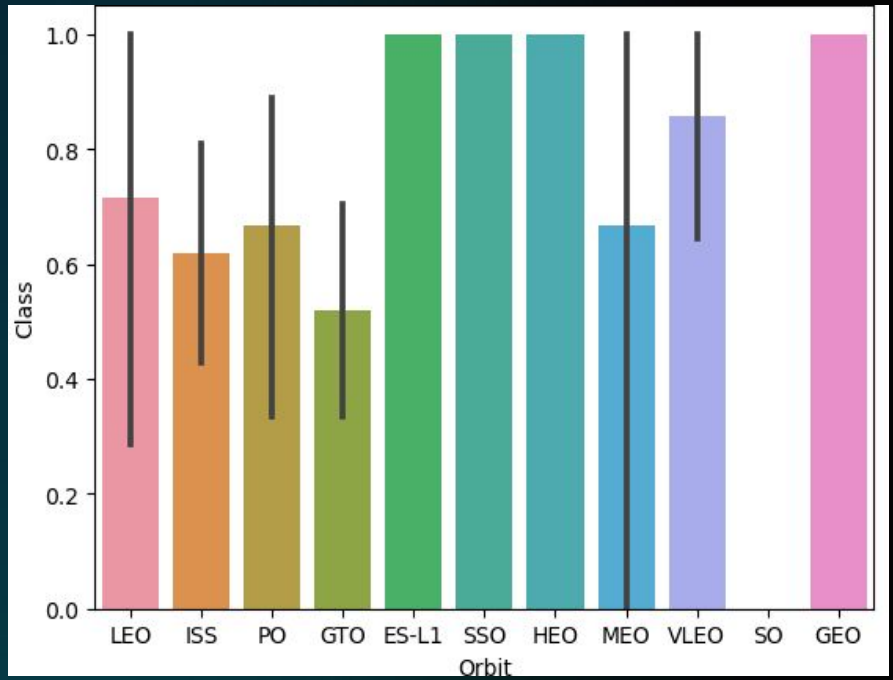
Payload Vs Launch Site

- There is no heavy payload mass rockets (greater than 10000) launched in **VAFB SLC 4E site**
- Most launches about **80%** were between (2000-6200) payload mass.



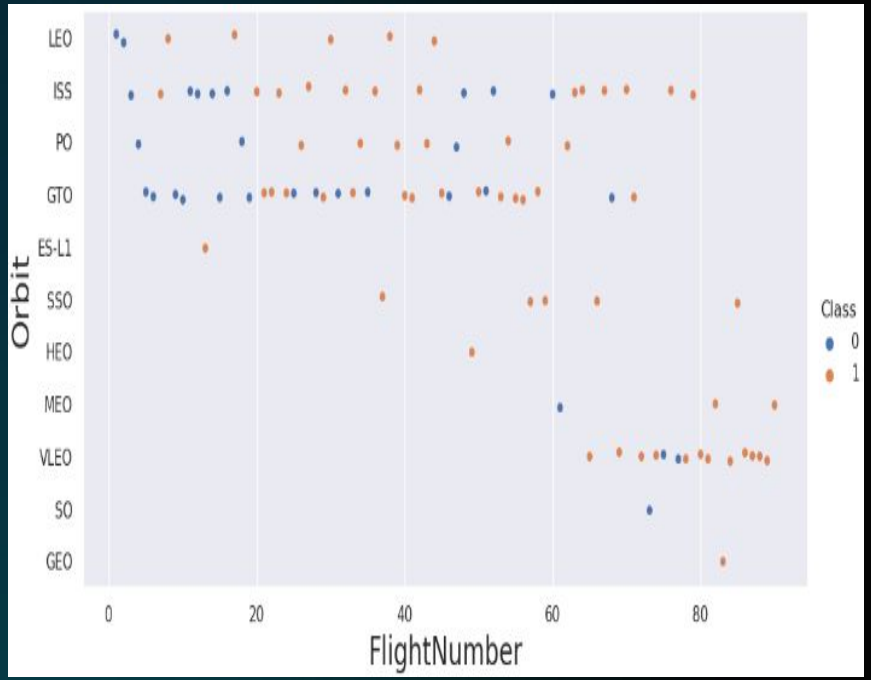
Success Rate Vs Orbit Type

- **GEO**, **SSO**, **HEO** and **ES-L1** has the best success rate.



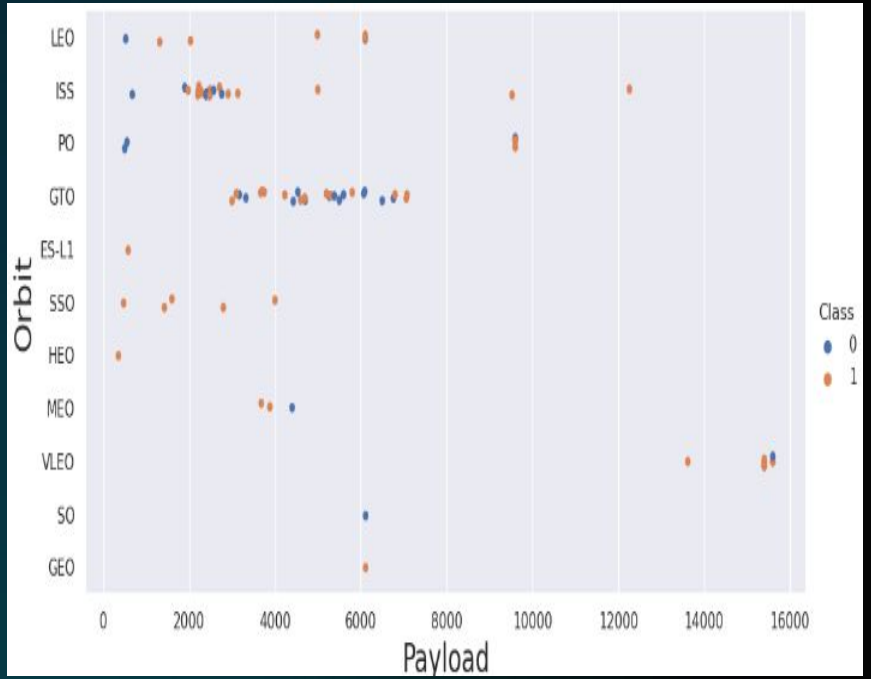
Flight Number Vs Orbit Type

- in the **LEO** orbit the Success appears related to the number of flights
- **t**here seems to be no relationship between flight number when in **GTO** orbit.
- The rockets started to appear in **GEO, SO VLEO, MEO, HEO, SSO** after 36-37 flights.



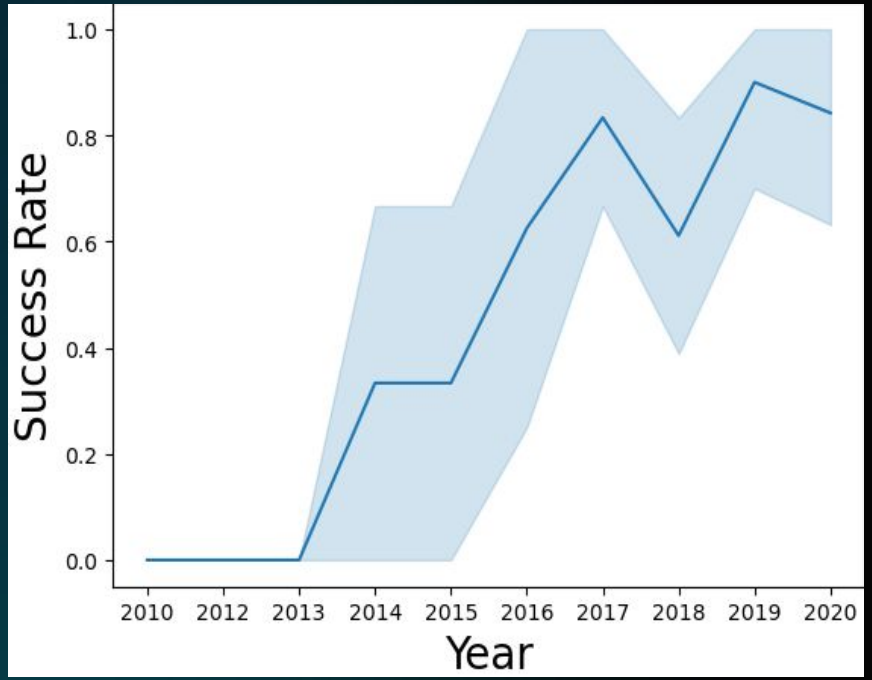
Payload Vs Orbit Type

- 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

- We can observe that the success rate since **2013** kept increasing till **2020**



All Site Names

- Selecting all launch site names by using **UNIQUE()**

```
1  SELECT
2      UNIQUE(LAUNCH_SITE)
3  FROM
4      SPACEX2
```

History | **Results**

Result set 1 | Details

Filter table

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

Launch Sites Begin With ‘CCA’

- Selecting all launch sites begin with ‘CCA’ by using **LIKE.**
- Getting the top 5 by using **LIMIT.**

```
1  SELECT
2      LAUNCH_SITE
3  FROM
4      SPACEX2
5  WHERE
6      LAUNCH_SITE LIKE 'CCA%'
7  LIMIT 5;
```

History | **Results**

Result set 1 | Details

Filter table

LAUNCH_SITE
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Getting the total payload mass using **SUM**
- Setting a condition in **WHERE** to only include NASA (CRS) customer.

```
1  SELECT
2      SUM(PAYLOAD_MASS__KG_)
3  FROM
4      SPACEX2
5  WHERE
6      CUSTOMER = 'NASA (CRS)':
```

History | **Results**

Result set 1 | Details

Filter table

1
45596

Average Payload Mass By F9 v1.1

- Using **AVG** to get the average payload mass.
- Setting a condition in **WHERE** to only get the average for ‘F9 v1.1’

```
1  SELECT
2      AVG(PAYLOAD_MASS__KG_)
3  FROM
4      SPACEX2
5  WHERE
6      BOOSTER_VERSION = 'F9 v1.1';
```

History | **Results**

Result set 1 | Details

Filter table

1
2928

First Successful Ground Landing Date

- Using **MIN** to get the minimum date to get the first one.
- Set a condition in **WHERE** to only include Success outcome.

```
1  SELECT MIN(DATE)
2  FROM
3      SPACEX2
4  WHERE
5      MISSION_OUTCOME = 'Success';
6
```

History | **Results**

Result set 1 | Details

Filter table

1
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- Set 3 conditions using **AND** to only include booster version with payload mass:

  1 - Greater than 4000 kg
  2 - Less than 6000 kg
  3 - Only successful drone ship

```
1  SELECT
2      booster_version
3  FROM
4      SPACEX2
5  WHERE
6      PAYLOAD_MASS__KG_ > 4000
7      AND PAYLOAD_MASS__KG_ < 6000
8      AND LANDING__OUTCOME = 'Success (drone ship)';
```

History | **Results**

Result set 1 | Details

Filter table

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

Total Number Of Succeeded Mission Outcomes

- Using **COUNT** to count the mission outcomes.
- Set a condition to only include Success outcomes.
- Using **LIKE** to get any outcome begin with Success.

```
1  SELECT
2      COUNT(MISSION_OUTCOME) AS COUNT
3  FROM
4      SPACEX2
5  WHERE
6      MISSION_OUTCOME LIKE 'Success%';
7
```

History | **Results**

Result set 1 | Details

Filter table

COUNT
100

Total Number Of Failed Mission Outcomes

- Using **COUNT** to count the mission outcomes.
- Set a condition to only include Failed outcomes.
- Using **LIKE** to get any outcome begin with Fail.

```
1  SELECT
2      COUNT(MISSION_OUTCOME) AS COUNT
3  FROM
4      SPACEX2
5  WHERE
6      MISSION_OUTCOME LIKE 'Fail%';
7
```

History | **Results**

Result set 1 | Details

Filter table

COUNT
1

Booster Carried Maximum Payload

- Using **MAX** to get the maximum payload mass.
- Set condition to only get the maximum payload mass in the every version.
- Subquery because you can't say column = MAX(column) it will throw an error.
- **ORDER BY** to order the outcomes and **DESC** to order it in descending order.

```
1  SELECT
2      BOOSTER_VERSION
3  FROM
4      SPACEX2
5  WHERE
6      PAYLOAD_MASS__KG_ = (
7      SELECT
8          MAX(PAYLOAD_MASS__KG_)
9      FROM
10         SPACEX2)
11 ORDER BY
12     booster_version
13 DESC;
```

History | **Results**

Result set 1 | Details

Filter table

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

2015 Launch Records

- **YEAR** to extract the year from DATE
- Set a condition to only include
  1 - failed drone ship outcome
  2 - year 2015

```
1  SELECT
2      BOOSTER_VERSION, LAUNCH_SITE, DATE
3  FROM
4      SPACEX2
5  WHERE
6      LANDING__OUTCOME = 'Failure (drone ship)'
7      AND YEAR(DATE) = 2015;
```

History | **Results**

Result set 1 | Details

Filter table — Total:2

BOOSTER_VERSION	LAUNCH_SITE	DATE
F9 v1.1 B1012	CCAFS LC-40	2015-01-10
F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

- **COUNT** for counting the outcomes
- Condition to only include:
  1 - Date Between ‘2010-06-04’ and ‘2017-03-20’
- **AND** is a logical operator for multiple conditions
- **GROUP BY** to group the outcomes
- **ORDER BY** and **DESC** for ordering (ranking) the outcomes in descending order

```
SELECT
    LANDING__OUTCOME,  COUNT(LANDING__OUTCOME) AS Count
FROM
    SPACEX2
WHERE
    DATE BETWEEN '2010-06-04'
    AND '2017-03-20'
GROUP BY
    LANDING__OUTCOME
ORDER BY
    COUNT(LANDING__OUTCOME)
DESC;
```

History | **Results**

Result set 1 | Details

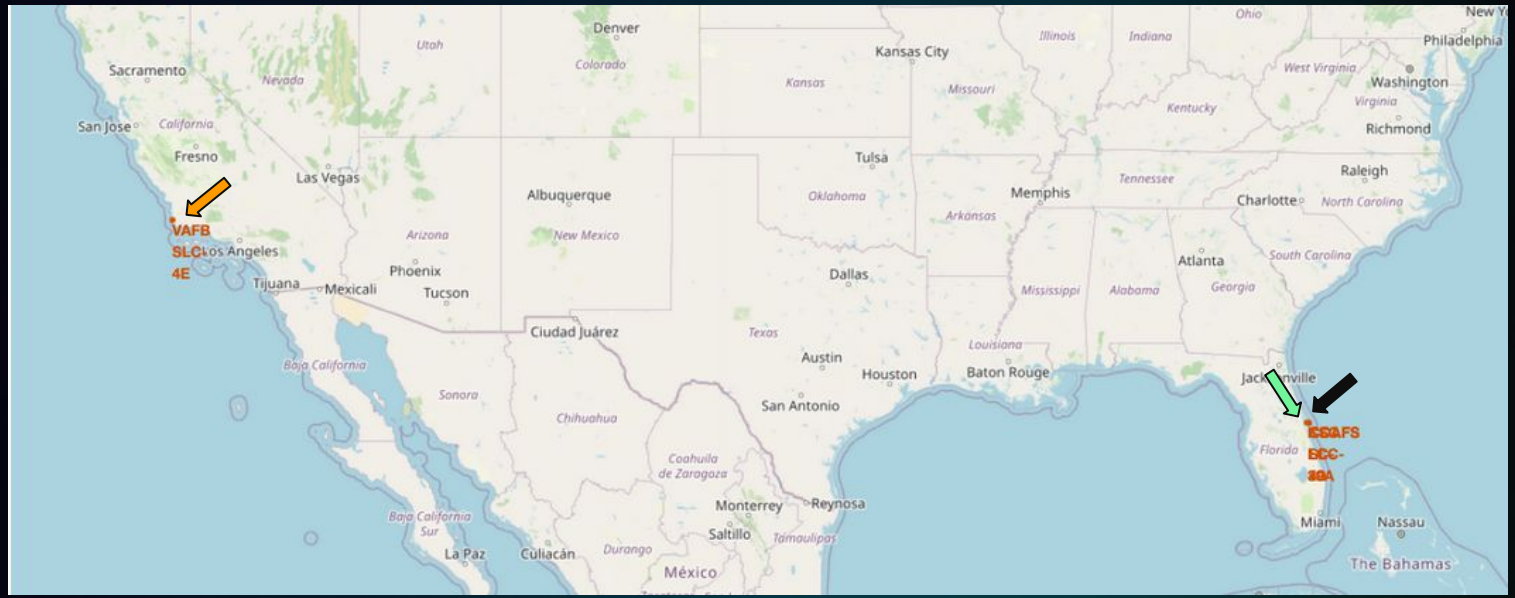
Filter table

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

Section 3

Launch Sites Proximities Analysis

Launch Sites Locations



Despite the multiple launch sites, they are all located at US Coasts, in Florida and California.

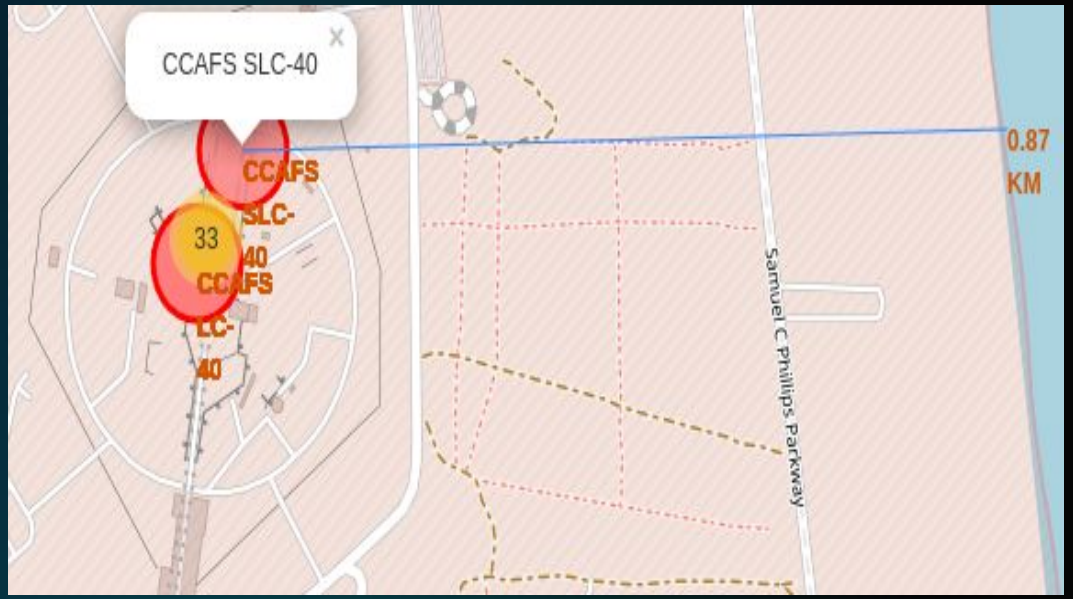
Succeeded and failed launches locations

You can easily identify succeeded and failed launches.

- **Green** for succeeded.
- **Red** for failed.

Visualize Distance Between A Selected Launch Site and Coastline

CCAFS SLC-40 is in close proximity to coastline with almost 1 km away

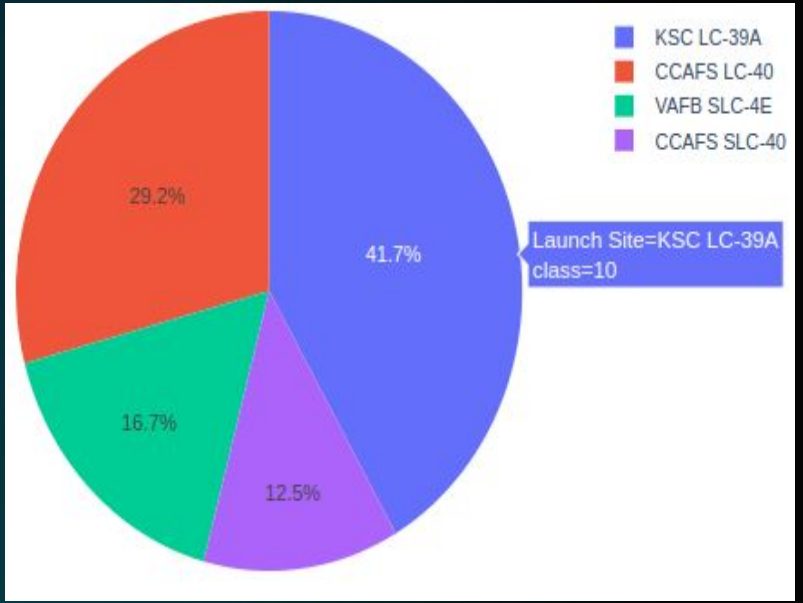


Section 4

Build a Dashboard with Plotly Dash

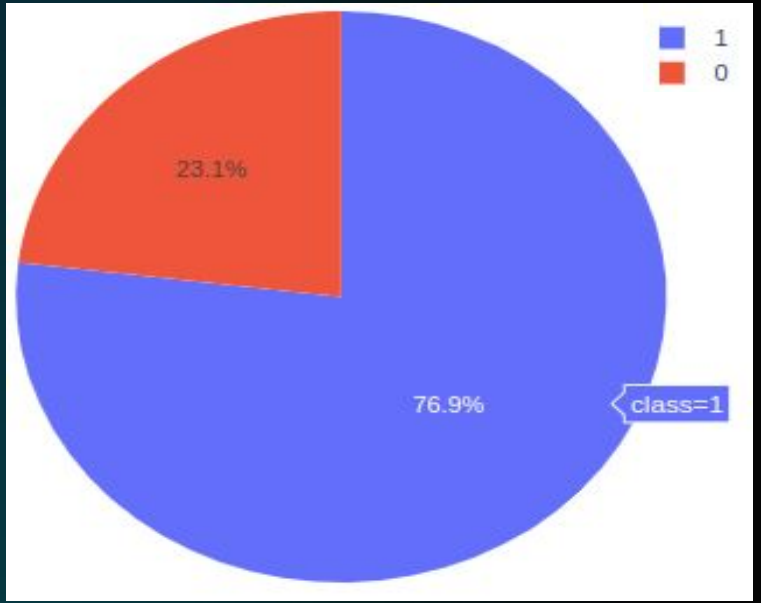
Which Site has the largest success count?

- We see that **KSC LC-39A** has **41.7%** success rate and it is the highest.



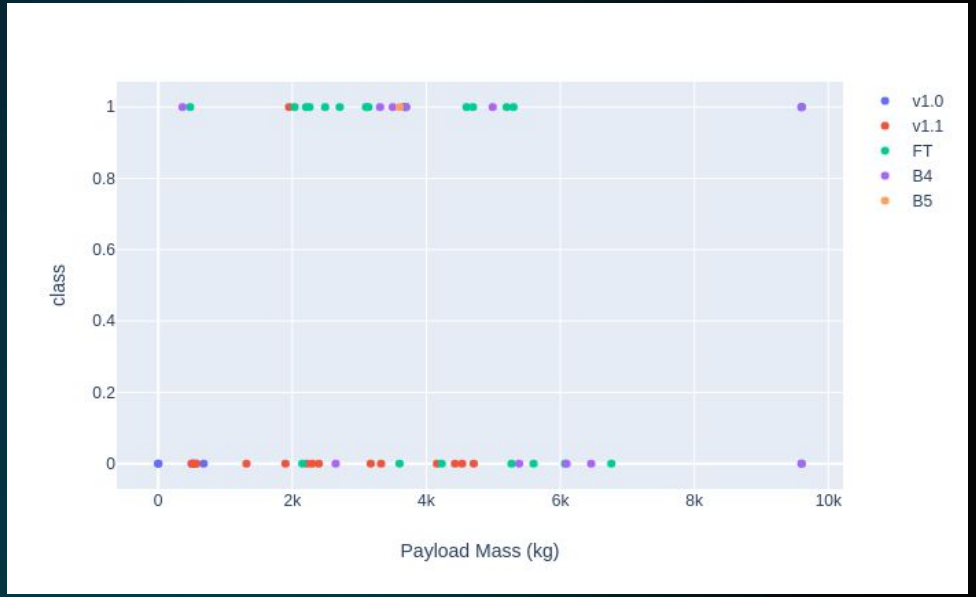
KSC LC-39A Success Count

- **1** means succeeded **0** means failed
- It came with **76.9%** success ratio, which is good.



Which Booster Version has the highest success count?

- We see that **FT** version has the highest succeeded counts.
- **V1.1** have the highest failed counts.

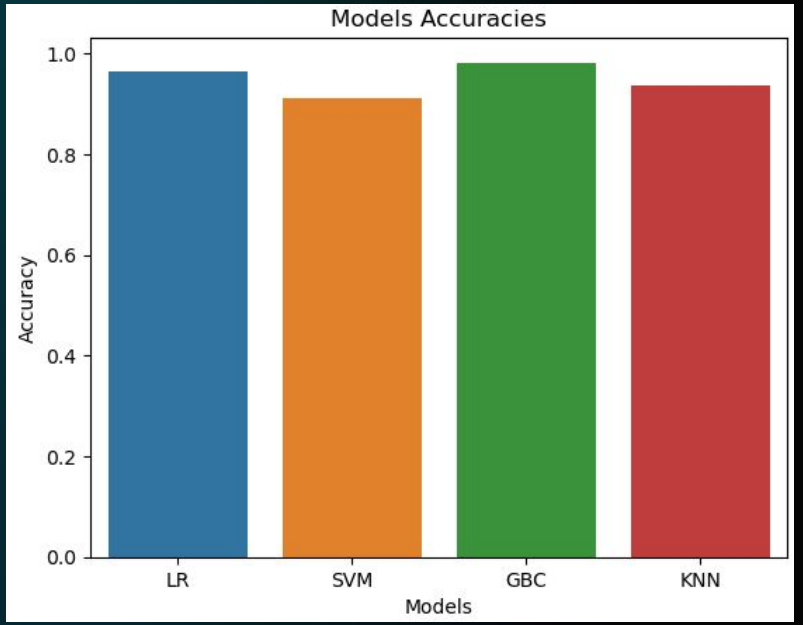


Section 5

Predictive Analysis (Classification)

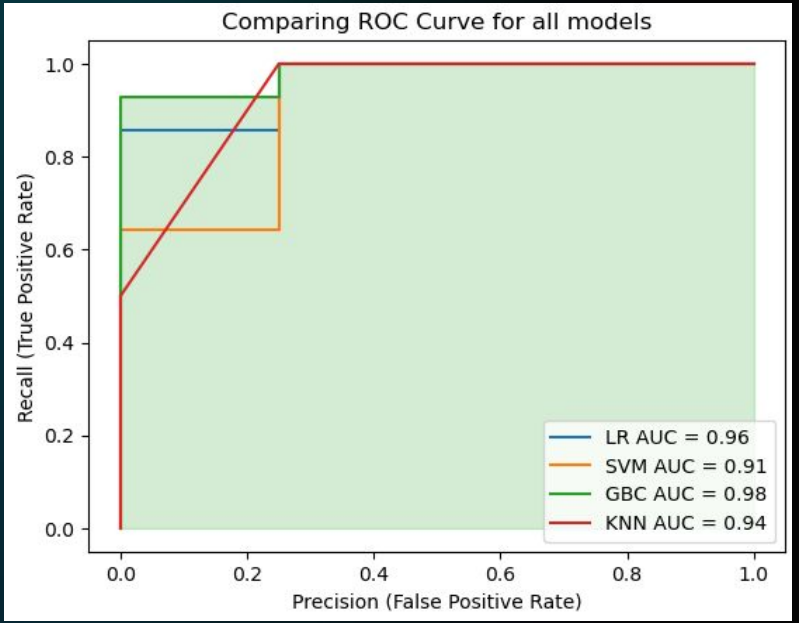
Models Accuracies

- We see that **Gradient Boosting** has the best accuracy with 98% but we can’t consider it as the best yet.
- The difference between **Logistic Regression** and **Gradient Boosting** is 2%



Models ROC Curve and AUC

- We see that **Gradient Boosting** has the best **AUC** score with 0.98 but we can't consider it as the best yet.
- The difference between **Logistic Regression** and **Gradient Boosting** is 2%

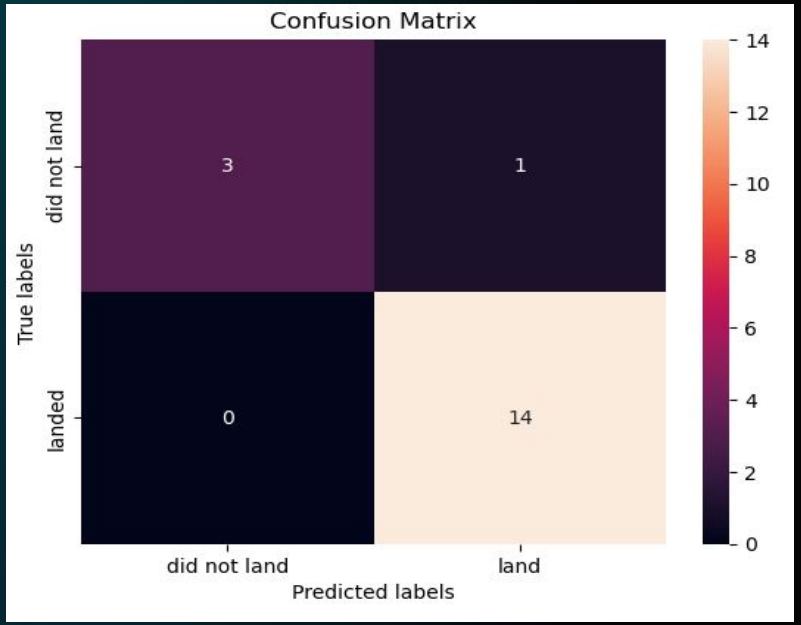


Models Training Time

- **Logistic Regression** took 253ms to train.
- **Support Vector** took 2.43s,
- **K-nearest Neighbors** took 41.3s.
- **Gradient boosting** took 5m 46s, Which is too long comparing to the models above.

Logistic Regression Confusion Matrix

- We see that it successfully labeled all of the landed ones.



Conclusions

- **KSC LC-39A** has the most successful launches.
- **GEO, HEO, SSO and ES-L1** Orbits have the best success rate.
- Low weighted payloads perform better than haviers ones.
- Success rate improved over years
- **Logistic Regression** performs better in all tests.

Project Progress

Task	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Data							
Analysis							
Modeling							
Wrapped Up							
Presentation							