IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alex CODERCH
8th of March 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through SpaceX API
  - Data Collection with Web Scraping of Wikipedia
  - Data Wrangling and first analysis
  - EDA with SQL
  - EDA with Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  The objective of this project is to predict the successful landing of the Falcon 9 first stage.

  SpaceX promotes its Falcon 9 rocket launches on its official website, offering these launches at a cost of $62 million, significantly lower than competitors who charge upwards of $165 million.

  A major factor contributing to SpaceX's cost efficiency is the ability to reuse the first stage of the rocket.

  By accurately forecasting whether the first stage will land successfully, we can gain insights into the overall cost-effectiveness of a launch.

  This information can be valuable for alternative companies seeking to compete with SpaceX in the rocket launch market.

- Problems you want to find answers
  - What factors influence the success of the first stage landing?
  - What is the rate of successful landings?

Section 1

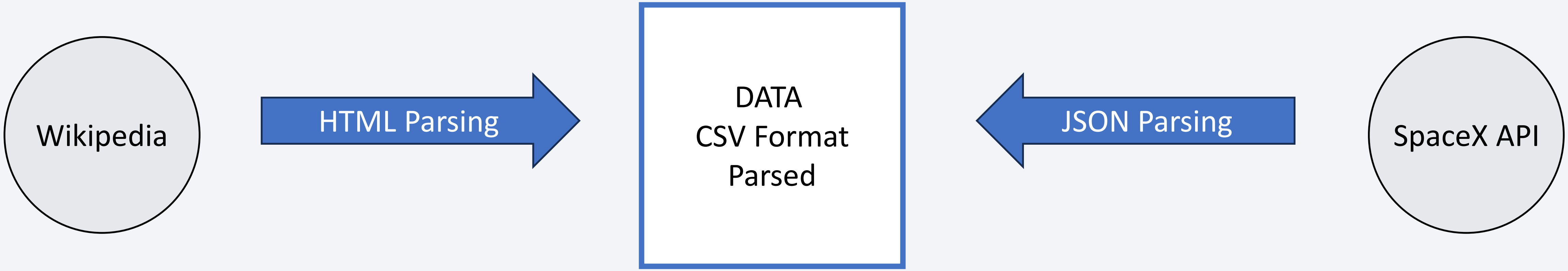
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Web Sraping & SpaceX API collection
- Data wrangling
- Exploratory data analysis (EDA) using
  - Visualization with Python
  - SQL
- Interactive visual analytics using
  - Folium and Plotly Dash
- Machine Learning predictive analysis

Data Collection



Data was collected using the SpaceX API and through web scraping from Wikipedia. The gathered data was then extracted, decoded, cleaned, and subsequently exported to a CSV file.

Data Collection – SpaceX API

- Initially, we retrieve and parse the SpaceX launch data using a GET request, which is subsequently converted into a Pandas DataFrame.
- We then filter this DataFrame to include only Falcon 9 launches.
- Following that, we conduct data wrangling to clean and prepare the data.

- GitHub URL : https://github.com/AlexCod96/IBM_Capstone/blob/14bb7e147b68b64b6b5c7b1121a3f7accfb2c2e0/spacex-data-collection-api.ipynb

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
static_json = response.json()
data= pd.json_normalize(static_json)
```

Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
data.head()
```

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_data[launch_data['BoosterVersion'] != 'Falcon 1']
data_falcon9
```

```
# Calculate the mean value of PayloadMass column
# Replace the np.nan values with its mean value
PayloadMass = pd.DataFrame(data_falcon9['PayloadMass'].values.tolist()).mean(1)
PayloadMass
```

Data Collection - Scraping

- First we request the Falcon9 Launch Wiki page from its URL.
- We extracted all column/variable names from the HTML table header.
- Then we create a data frame by parsing the launch HTML tables.
- GitHub URL :https://github.com/AlexCod96/IBM_Capstone/blob/14bb7e147b68b64b6b5c7b1121a3f7accfb2c2e0/webscraping.ipynb

```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url)
data.status_code
```

```
200
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Data Wrangling

- First analysis : Number of launches on each site.
- Second : Number and occurrence of each orbit.
- Third : Occurence of mission outcome of each orbit.
- Last :Processing Landing outcome
- GitHub URL :https://github.com/AlexCod96/IBM_Capstone/blob/main/spacex-data-wrangling.ipynb

```python
# Apply value_counts on Orbit column
df["Orbit"].value_counts()
```

```
Orbit
GTO       27
ISS       21
VLEO      14
PO         9
LEO        7
SSO        5
MEO        3
ES-L1      1
HEO        1
SO         1
GEO        1
Name: count, dtype: int64
```

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

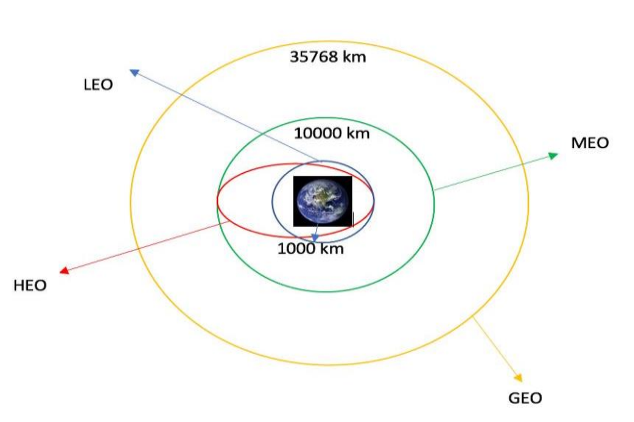
```python
# landing_outcomes = values on Outcome column
landing_outcomes = df["Outcome"].value_counts()
landing_outcomes
```

```
Outcome
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: count, dtype: int64
```

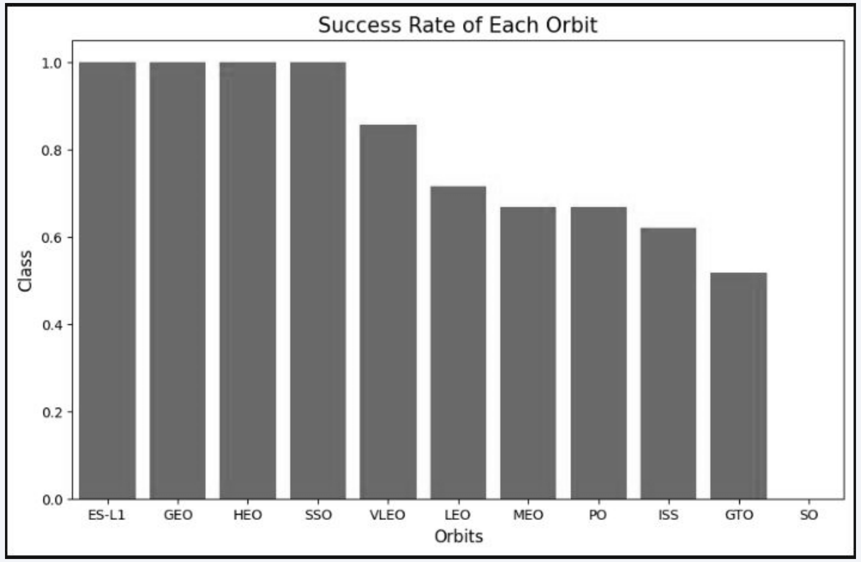
```python
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []

for i, j in df["Outcome"].items():
    if j in bad_outcomes:
        outcome = 0
        landing_class.append(outcome)
    else:
        outcome = 1
        landing_class.append(outcome)
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

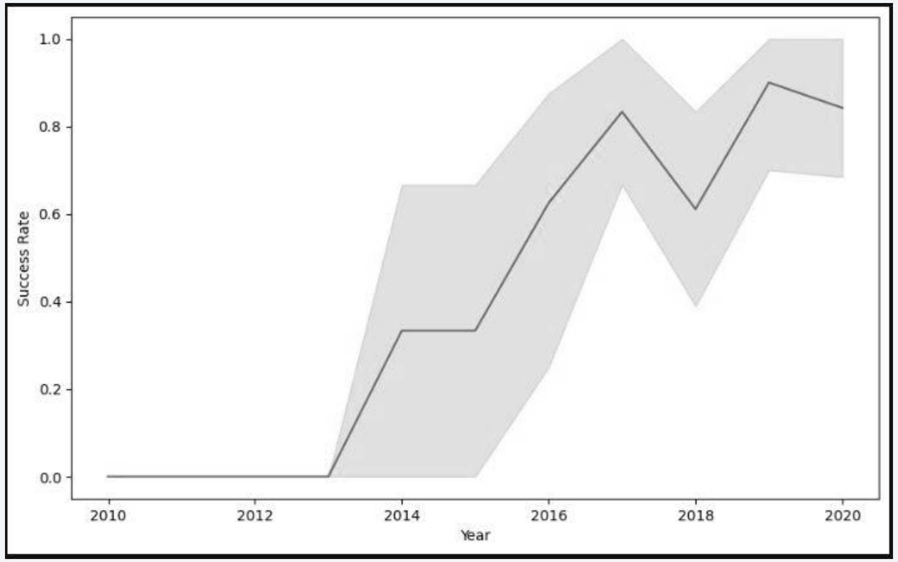
```python
df['Class']=landing_class
df[['Class']].head(8)
```



EDA with Data Visualization



Visualisation of the successrate for each orbit



Evolution of the success rate of SpaceX mission through the years

GitHub URL of the completed task :
https://github.com/AlexCod96/IBM_Capstone/blob/14bb7e147b68b64b6b5c7b1121a3f7accfb2c2e0/eda-dataviz.ipynb

EDA with SQL

Here are the main key points of interest obtained through SQL analysis

- Get the names of all unique launch sites

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

- Get 5 records of launch site starting with “CCA

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

- Get the average payload mass gone with the Falcon 9 v1

```
%sql SELECT AVG(payload_mass__kg_) AS Average_Payload_Mass_KG FROM SPACEXTBL WHERE booster_version = 'F9 v1.1';
```

- Get the total number of mission outcomes, both successes and failures

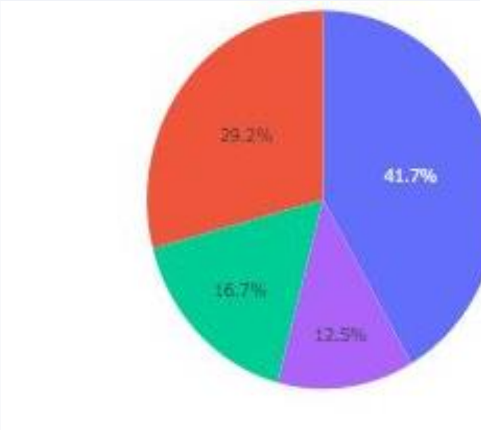
```
%sql SELECT mission_outcome, COUNT(mission_outcome) AS Count \
    FROM SPACEXTBL GROUP BY mission_outcome;
```

- GitHub URL :https://github.com/AlexCod96/IBM_Capstone/blob/14bb7e147b68b64b6b5c7b1121a3f7accfb2c2e0/eda-sql.ipynb

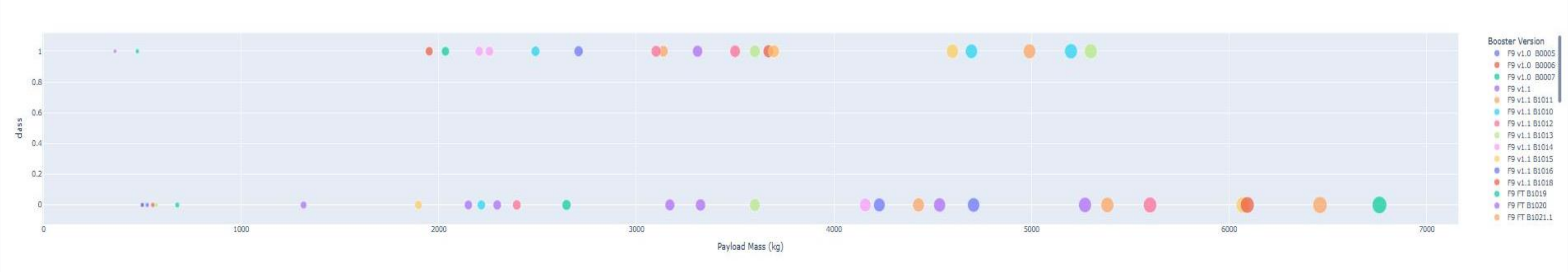
Build an Interactive Map with Folium

- Here is the main done
  - Marking all launch sites on a map using site's latitude and longitude coordinates.
  - Marking the success/failed launches for each site on the map.
  - Calculating the distances between a launch site to its proximities.
- Thanks to the insights we have gained about launch sites, we can conclude that:
  - They are near railways, which is good because it allows transporting the heavy components needed to assemble the rockets.
  - They are near highways, which allows proper communication so that personnel can get there easily.
  - They are near coastline to ensure that no debris falls on populated areas in case of failures.
  - They keep certain distance away from cities to mitigate risks in the event of failure.

Build a Dashboard with Plotly Dash



- We developed a dashboard using Plotly Dash, featuring :
  - pie charts that display the total launches for each launch site,
  - a scatter plot illustrating the relationship between the launch outcome and the payload mass across different versions of the booster.
- GitHub URL : https://github.com/AlexCod96/IBM_Capstone/blob/14bb7e147b68b64b6b5c7b1121a3f7accfb2c2e0/spacex_dash_app.py



Predictive Analysis (Classification)

- We used numpy and pandas, transformed the data and then split it for training and testing.

```
Y = data['Class'].to_numpy()
Y
```

```
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
print (f"Train set: {X_train.shape}  {Y_train.shape}")
print (f"Test set: {X_test.shape}  {Y_test.shape}")
```

```
Train set: (72, 83)  (72,)
Test set: (18, 83)  (18,)
```

- We have created different machine learning models using GridSearchCV.

```
GridSearchCV
best_estimator_: LogisticRegression
LogisticRegression
```

- Then we concluded which launch method is the best :

```
Best model is DecisionTree with a score of 0.8714285714285713
Best params is: {'criterion': 'gini', 'max_depth': 8, 'max_features': None, 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'best'}
```

- GitHub URL : https://github.com/AlexCod96/IBM_Capstone/blob/14bb7e147b68b64b6b5c7b1121a3f7accfb2c2e0/spacex-machine-learning-prediction.ipynb

Results

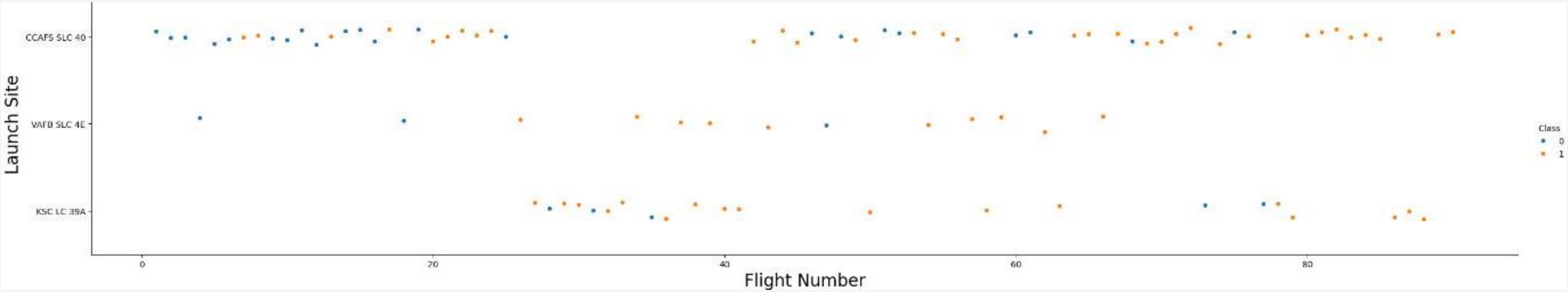
- The exploratory data analysis has shown us that successful landing outcomes are correlated with flight number, as the company gets more experienced and learn from its error. This is mainly apparent in 2015 onward
- All launch sites are located near the coast, to avoid crash in populated areas, Sites are also located near highways and railways to facilitate transportation of equipment.
- The machine learning were able to predict the landing success of rockets with an accuracy score of 87,14%.

Section 2

Insights drawn from EDA

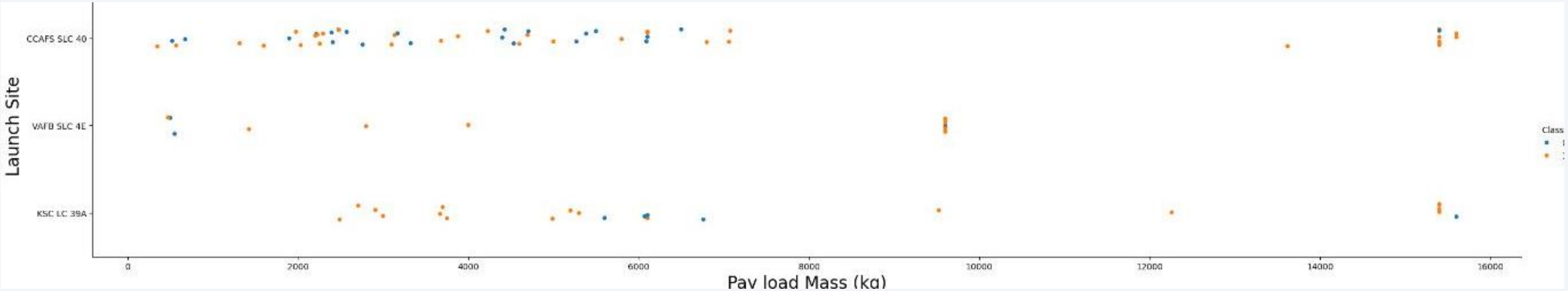
Flight Number vs. Launch Site

- CCAFS SLC 40 and KSC LC 39A have a higher number of flights compared to VAFB SLC 4E.
- The higher the number of flights, the higher the success rate.



Payload vs. Launch Site

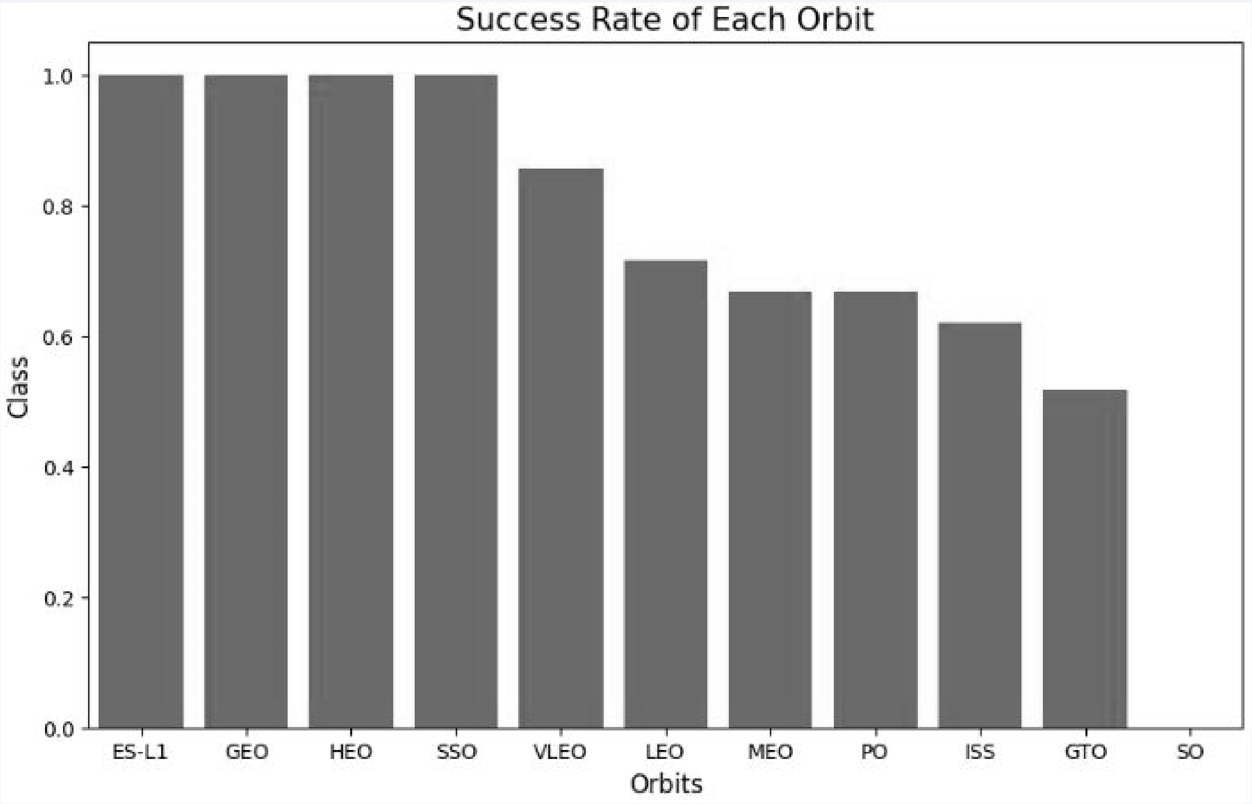
- The highest failure rate is recorded for payload masses between 4,000 kg and 6,000 kg.
- Payload masses over 8,000 kg have a higher success rate.
- In general, the higher the payload, the higher the success rate.



Success Rate vs. Orbit Type

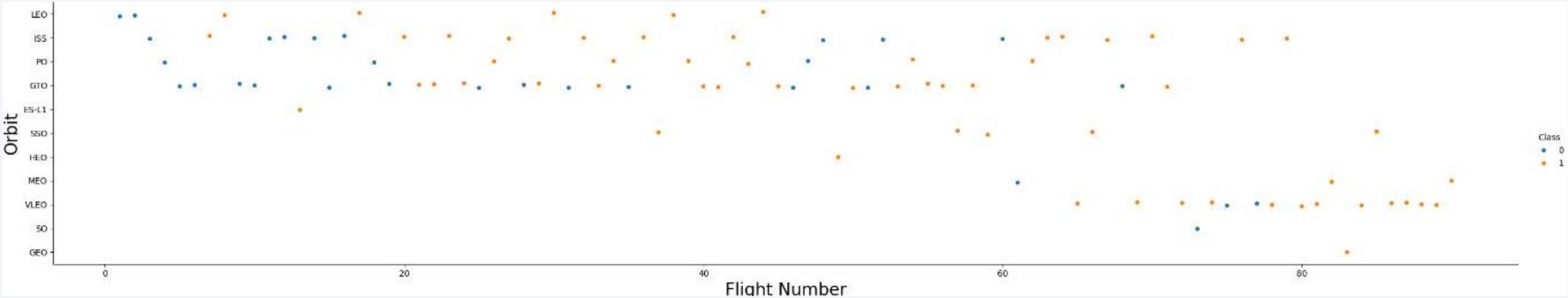
- Show a bar chart for the success rate of each orbit type

- Show the screenshot of the scatter plot with explanations



Flight Number vs. Orbit Type

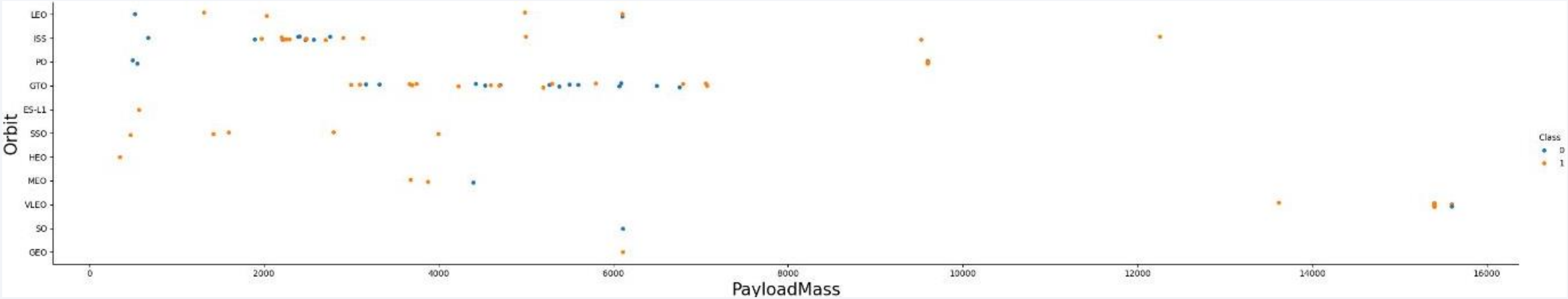
- Here is the number of flight vs the type of orbit scatter plot



- We can see a change over the time of the type of orbit used. SpaceX started with LEO and ISS orbit, and then switch to VLEO after 65 flights

Payload vs. Orbit Type

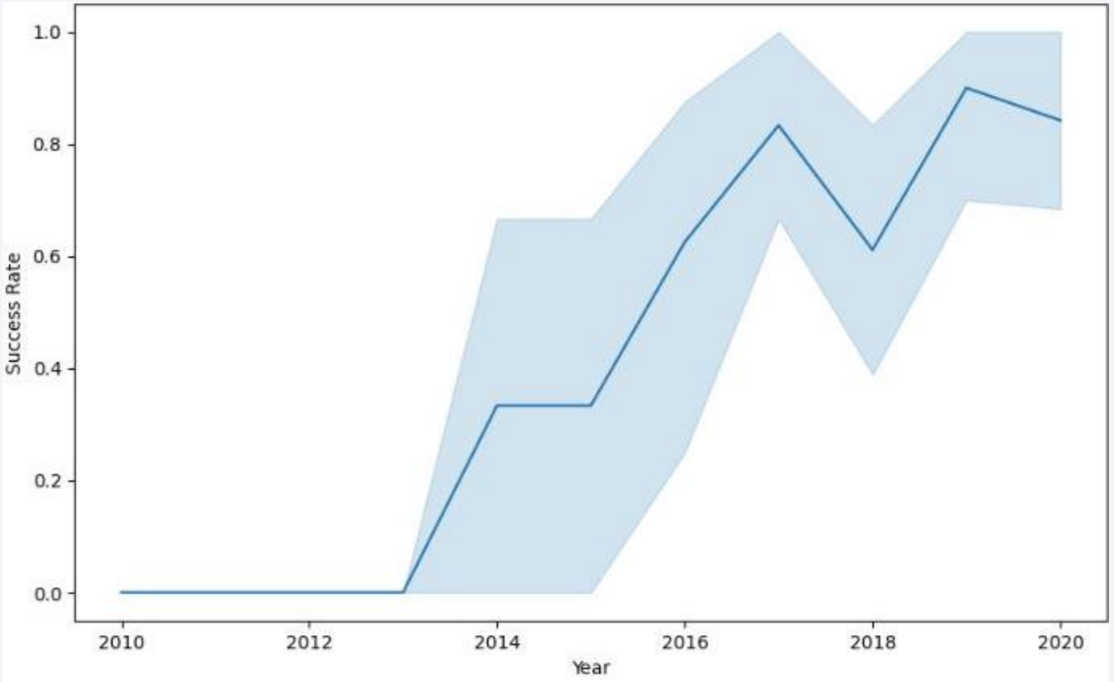
- Here is the scatter plot of the payload vs Orbit type used



- We can see that VLEO is used for larger payload, while GTO whas used the most for lower payload

Launch Success Yearly Trend

- The success rate trend is an improvement over time, with a dip in 2018



All Launch Site Names

- Here are the names of the sites used :

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

- The SQL command to get them is :

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

Launch Site Names Begin with 'CCA'

- The 5 launch sites with their names starting with 'CCA' are the following :

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

- The SQL command to get them are :

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

Total Payload Mass

- The total payload is 45 596kg, and has been obtained with this SQL command

```
%sql SELECT SUM(payload_mass__kg_) AS Total_Payload_Mass_KG \
    FROM SPACEXTBL WHERE customer = 'NASA (CRS)';
```

Average Payload Mass by F9 v1.1

- Using the “AVG” command, we got the average payload of 2928,4kg

```
%sql SELECT AVG(payload_mass__kg_) AS Average_Payload_Mass_KG FROM SPACEXTBL WHERE booster_version = 'F9 v1.1';
```

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

Average_Payload_Mass_KG
2928.4

First Successful Ground Landing Date

- Getting the first successful landing has been done with the filter “WHERE” on the list of the landing and a minimum on the date :

```
%sql SELECT MIN(DATE) AS First_Successful_Landing \
     FROM SPACEXTBL WHERE landing_outcome = 'Success (ground pad)';

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

First_Successful_Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

- The query to get the list is the following. It uses a "WHERE" to filter the landing outcome

```
%sql SELECT PAYLOAD \
     FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' \
     AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

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

- Present your query result with a short explanation here

```
%sql SELECT mission_outcome, COUNT(mission_outcome) AS Count \
    FROM SPACEXTBL GROUP BY mission_outcome;
```

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

- The query used is the following, with a select

```
%sql SELECT booster_version, payload_mass__kg_ \
    FROM SPACEXTBL \
    WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL);
```

2015 Launch Records

- List the failed landingoutcomes in drone ship, their booster versions, and launch site names for in year 2015

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

- There is the query used for listing the two failed launch in 2015

```
%sql SELECT SUBSTR(Date,6,2) AS Month, DATE,BOOSTER_VERSION, LAUNCH_SITE, [Landing_Outcome] \
    FROM SPACEXTBL \
    WHERE [Landing_Outcome] = 'Failure (drone ship)' AND SUBSTR(Date,0,5) = '2015';
```

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

- 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

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

- Here is the query used to list and order the landing outcome in the date range :

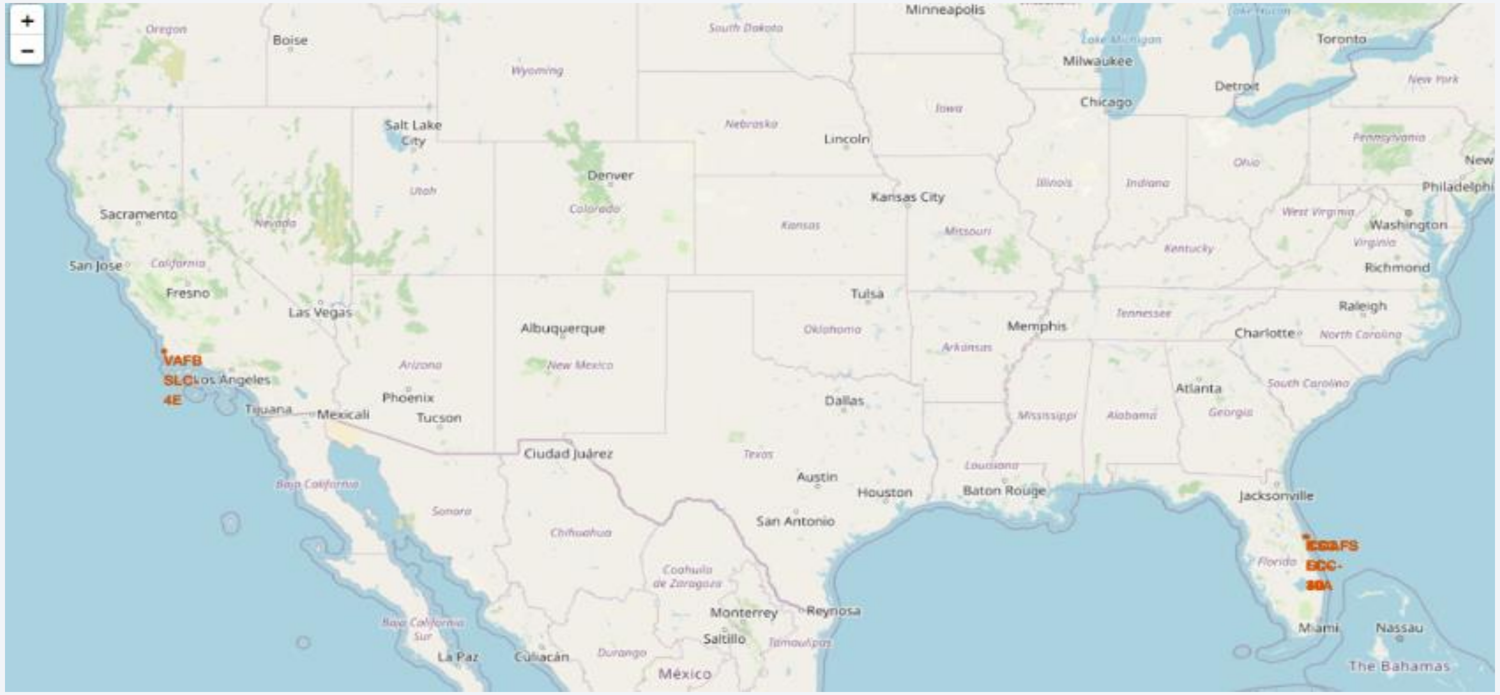
```
%sql SELECT [Landing_Outcome], COUNT(*) AS Count \
     FROM SPACEXTBL \
     WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY [Landing_Outcome] \
     ORDER BY Count DESC;
```

Section 3

Launch Sites Proximities Analysis

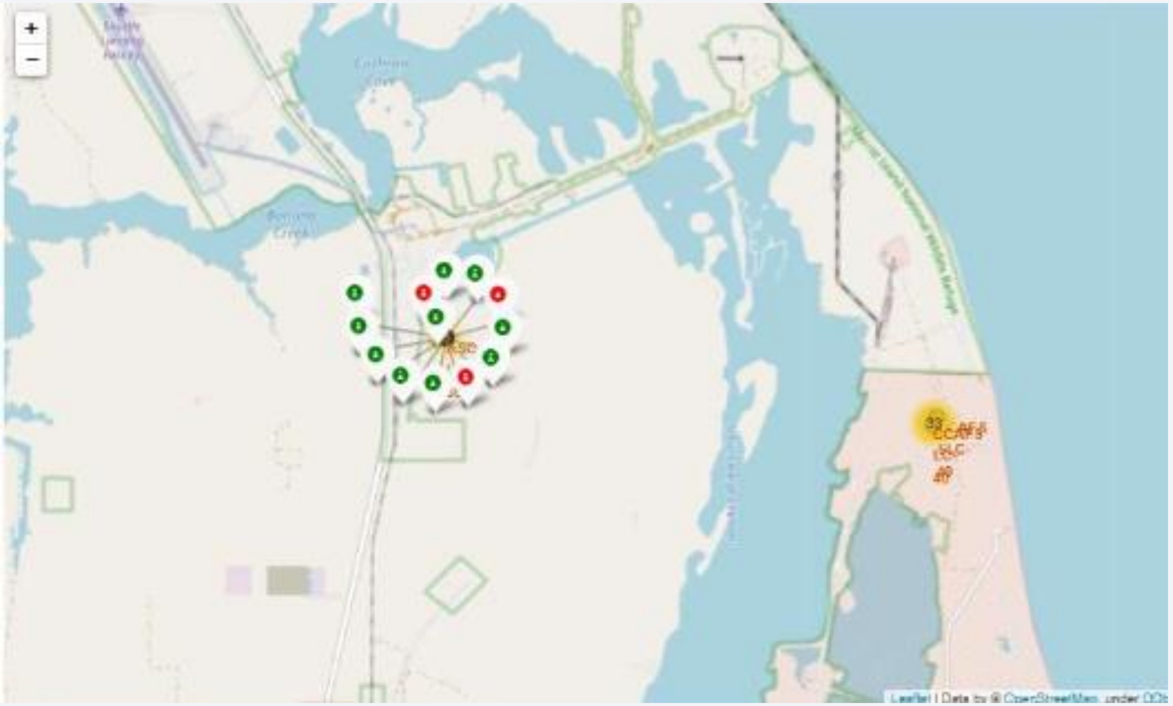
Location of the Launch Sites

- Launch sites are on both west and east coast of the US.
- Both are southern, as a closer to the equater, the easier the launch



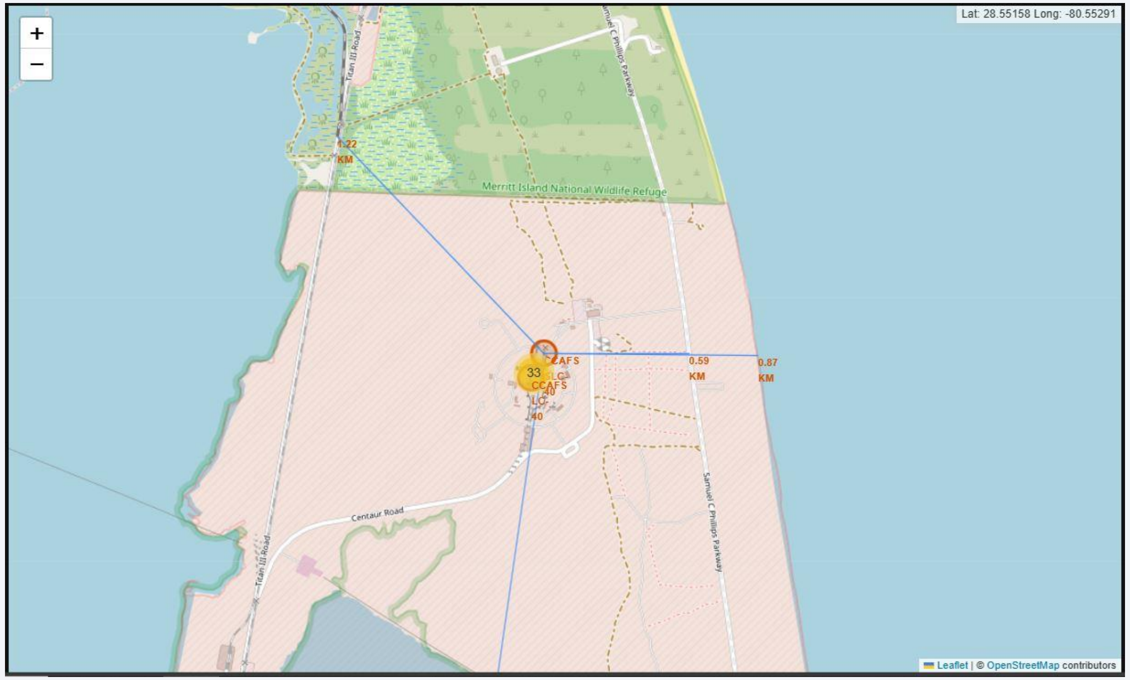
Success Rate of Launches in Florida base

- Launch failure are marked in RED
- Launch success are marked in GREEN



Launch Sites surrounding landmark

- The launch side are close to seashore (less than 2km away)
- Railway are within close proximity
- Cities are also close but further away for safety

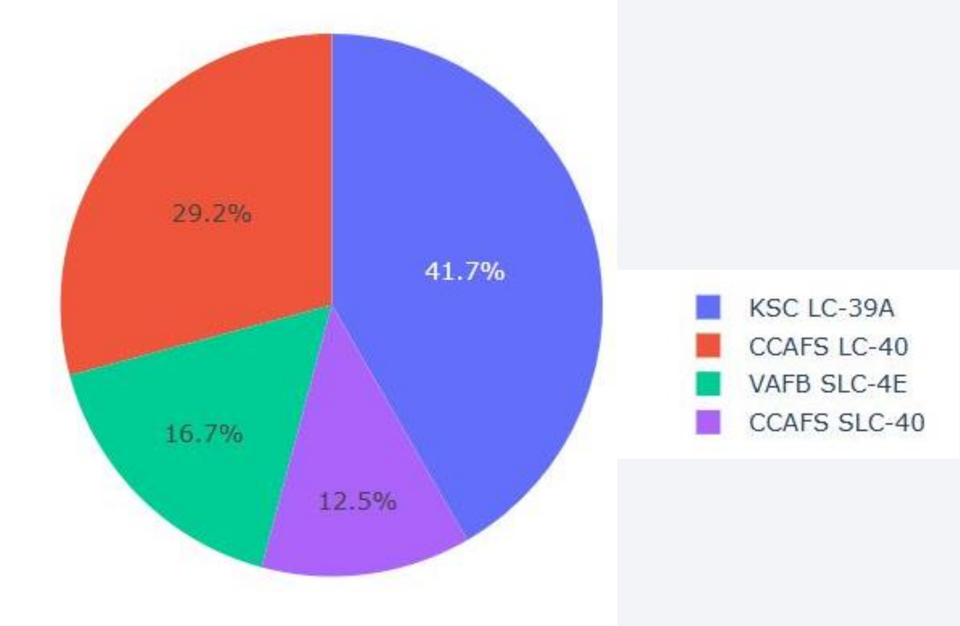


Section 4

Build a Dashboard with Plotly Dash

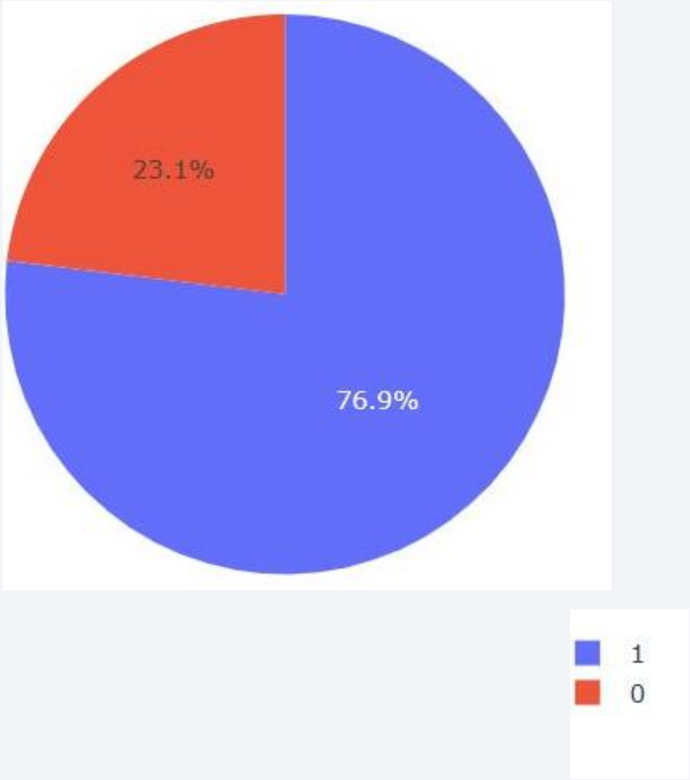
Launch site success rate repartition

- This pie chart shows tha KSC LCA-39A is the main launch success site
- The CCFS LC-40, despite being the first site used, is still second in term of success number proportion



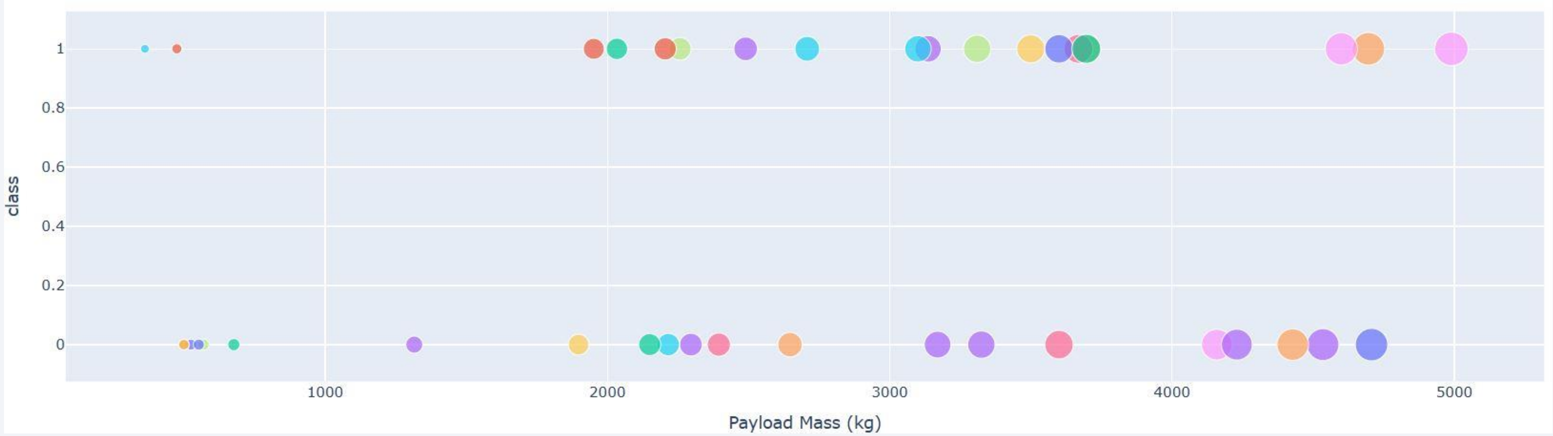
KSC LC-39A success rate plot

- 76,9% of the launch were successful on this launch site
- This is the best launch site recorded
- Success is in blue

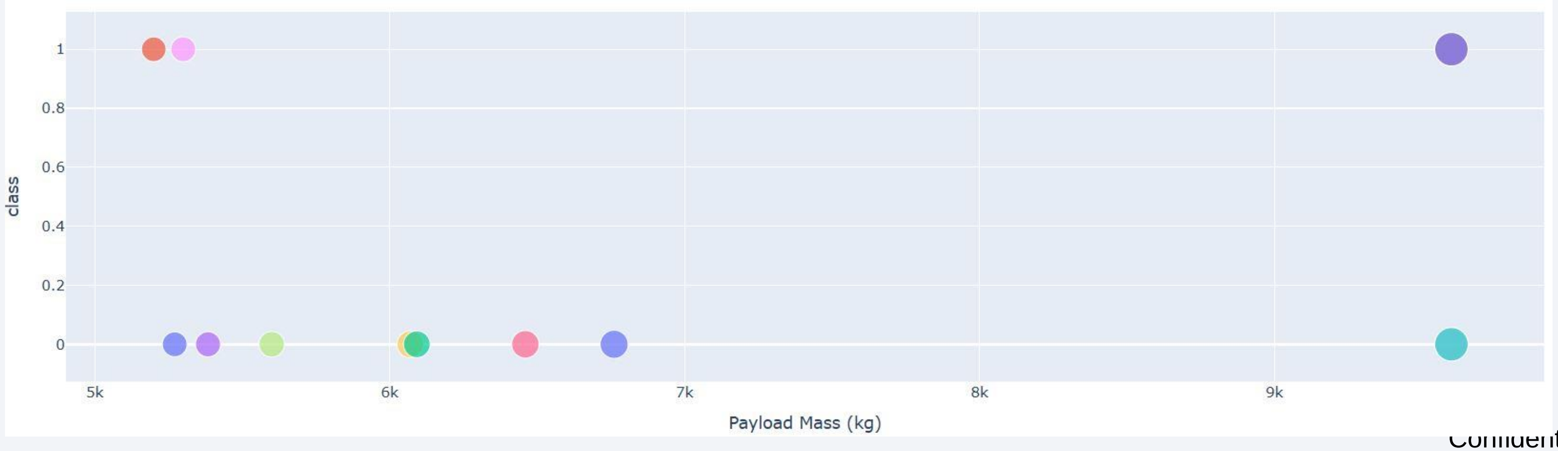


Payload vs Outcome scatter plot

- Lower Payload seems to have a higher success rate



- High Payload statistic is skewed by the low number of launch, only 11

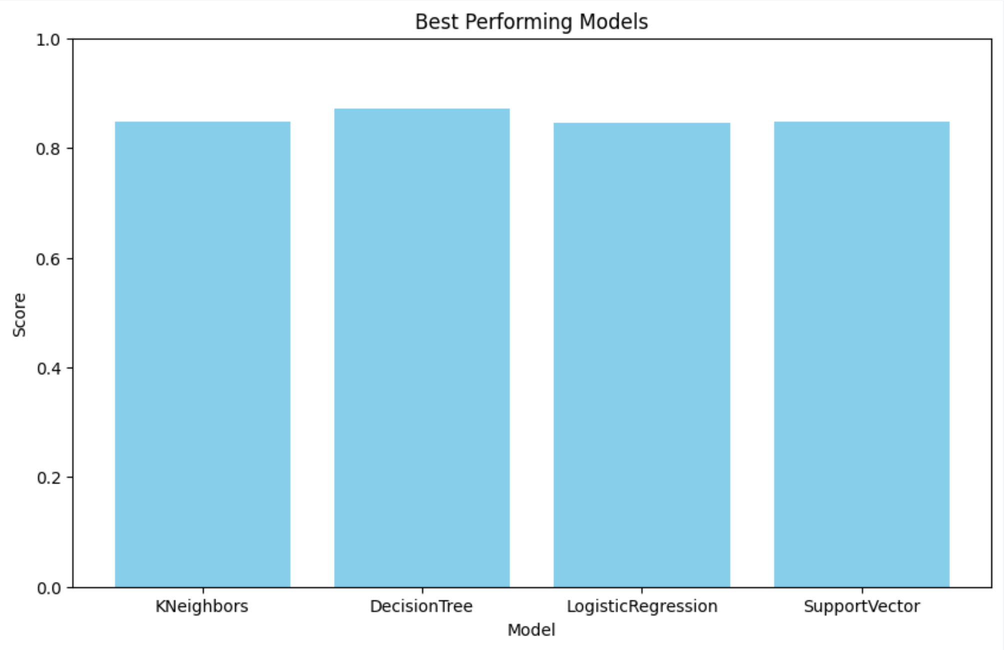


Section 5

Predictive Analysis (Classification)

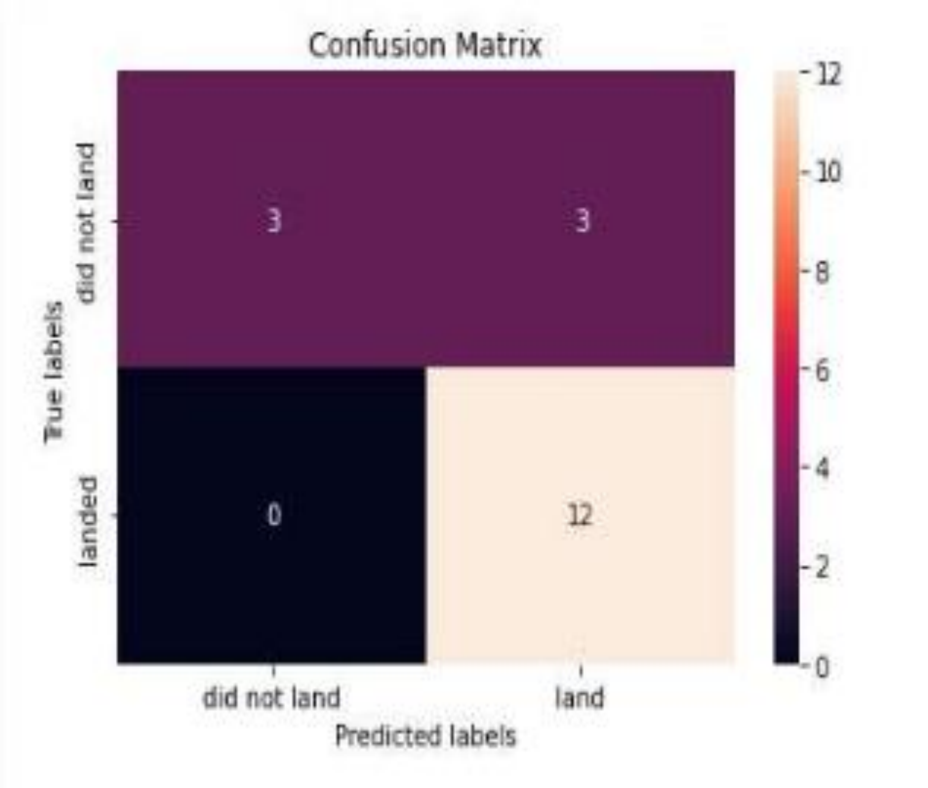
Classification Accuracy

- The best model is the Decision-Tree with the 87,14%



Confusion Matrix

- The model successfully predicted all tests outcomes but 6 with Logistic Regression Model



Conclusions

- To compete with Space X Through this process, a general picture of their success methods are
  - All their launch sites are located near the coast, away from nearby cities. This enabled to them to tes their rocket landings without much interference.
  - Site KSC LC-39A had the highest launch success rate out of all the launch sites.
  - From 2015 onwards, the success rate of rocket landings significantly increased. It was also apparent that landing success increased with flight number
- All this data was used to train a machine learning model that can predict the landing outcome of rocket launches with 83.33% accuracy

Thank you!