IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Data Science Capstone Project

Ing. Ondrej Pekarcik MSc.
April 18, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection and data cleansing
  - Data wrangling
  - Exploratory data analysis in Python with data visualization
  - EDA with SQL
  - Building an interactive visual analytics map with Folium
  - Building an interactive dashbord using Plotly Dash
  - Predictive analysis
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics figures
  - Predictive analytics results

Introduction

- **Project background and context**

Space Exploration Technologies Corp. (SpaceX) is an American aerospace manufacturer, a provider of space transportation services, and a communications corporation headquartered in Hawthorne, California. SpaceX was founded in 2002 by Elon Musk with the goal of reducing space transportation costs to enable the colonization of Mars. SpaceX manufactures the Falcon 9 and Falcon Heavy launch vehicles, several rocket engines, Cargo Dragon, crew spacecraft, and Starlink communications satellites.

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.

Thus, we will predict if the first stage of Falcon 9 will land, knowing the odds of successful landing, we can specifically determine the cost of a launch. This information can be used if a competing rocket company wants to bid against SpaceX for a rocket launch.

- Key research problems

- Correlation analysis of each rocket types variables and its respective landing rate
- Finding the best conditions to achieve highest successful landing rate

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web Scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia
- Perform data wrangling
  - Conversion of outcomes into training labels with booster landing outcomes
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Finding best Hyperparemeter for Classification Trees, Logistic Regression and SVM

Data Collection

- Data collection process consisted of a combination of **API requests** from SpaceX API and **web scraping** data from Wikipedia *SpaceX, Falcon 9 and Falcon Heavy Launches Records* page.
- SpaceX API data columns: `FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude`
- Wikipedia web scrape data columns: `Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time`

SpaceX API

Data Scraping

Data Collection – SpaceX API

1. Data request from SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Decoding the response as .JSON as Pandas dataframe

```
data = pd.json_normalize(response.json())
```

3. Cleaning and filtering the data using DF subset

```
data = data[['rocket', 'payloads', 'launchpad', 'cores',
'flight_number', 'date_utc‘]]
```

4. Assigning global variables for requests as lists

```
BoosterVersion = []
...
Latitude = []
```

5. Calling SpaceX API using the defined functions

```
getBoosterVersion(data)
...
getCoreData(data)
```

6. Constructing the dataset using obtained data

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
...
'Latitude': Latitude}
launch_df= pd.DataFrame({ key:pd.Series(value) for key, value in
launch_dict.items() })
```

7. Filtering the DF to only include Falcon 9

```
data_falcon9 = launch_df[launch_df["BoosterVersion"] == "Falcon 9" ]
```

8. Data Wrangling (dealing with missing values)

```
# Calculate the mean value of PayloadMass column
PayloadMass_mean = data_falcon9.PayloadMass.mean()

# Replace the np.nan values with its mean value
data_falcon9["PayloadMass"] =
data_falcon9["PayloadMass"].replace(np.nan, PayloadMass_mean)
```

9. Exporting dataframe to a CSV file

```
data_falcon9.to_csv(r"C:\Data Science Capstone\data_falcon9.csv")
```

- *GitHub URL*

Data Collection – Web Scraping

1. Getting HTML response

```
html_data = requests.get(static_url).text
```

2. BeautifulSoup object creation

```
soup = BeautifulSoup(html_data, "html5lib")
```

3. Extracting column names from the HTML table header

```
html_tables = soup.find_all("table")
column_names = []
for row in first_launch_table.find_all("th"):
    name = extract_column_from_header(row)
    if(name != None and len(name) > 0):
        column_names.append(name)
```

4. Creating empty dictionary with extracted column names

```
launch_dict= dict.fromkeys(column_names)
launch_dict['Flight No.'] = []
...
launch_dict['Time‘]=[]
```

5. Filling up the launch_dict with launch records

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            print(flight_number)
            launch_dict["Flight No."].append(flight_number)
            datatimelist=date_time(row[0])
...
(Refer to the notebook for the entire code)
```

6. Exporting dataframe to a CSV file

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
df.to_csv(r"C:\Data Science Capstone\spacex_web_scraped.csv")
```

- *GitHub URL*

Data Wrangling (1/3)

There are three different scenarios based on landing locations under which the booster did not land successfully, it is therefore necessary to distinguish outcomes into specific traning labels for training supervised models as:

```
True ASDS: the mission result has successfully landed on the drone ship
False ASDS: the mission result has not landed on the drone ship
True Ocean: the mission result has successfully landed in a specific area of the ocean
False Ocean: the mission result has not successfully landed in a specific area of the ocean
True RTLS: the mission result successfully landed on the ground pad
False RTLS: the mission result has not successfully landed on the ground pad
```

While successful booster landing labels are assigned value 1 and unsuccessful `0`.

1. Loading dataset prepared using SpaceX API

```
df=pd.read_csv("data_falcon9.csv")
```

2. Calculating the number of launches on each site

```
df["LaunchSite"].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

- *GitHub URL*

3. Calculating the number and occurrence of each orbit

```
df.Orbit.value_counts()
```

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

Data Wrangling (2/3)

4. Calculating the number and occurence of mission outcome per orbit type

```
landing_outcomes = df.Outcome.value_counts()
```

```
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: Outcome, dtype: int64
```

5. Enumerating landing outcomes and creating set where the second stage did not land successfully

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)

bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

- [*GitHub URL*]

Data Wrangling (3/3)

6. Creating a landing outcome label

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for outcome in df.Outcome:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

7. Determining the success rate for every landing in dataset

```
df["Class"].mean()
```

```
0.666666666666
```

8. Exporting dataframe to a CSV file

```
df.to_csv("dataset_part_2.csv", index=False)
```

- *GitHub URL*

EDA with Data Visualization

- Bar Chart:
  - *Orbit Type versus Success Rate of Orbit Types*

A bar chart was used to show comparisons among discrete categories of orbit types.

- Line Chart
  - Year versus Success Rate

A line chart was used to display historical landing success rates.

- Scatter Chart:
  - Flight Number versus Launch Site
  - Payload versus Launch Site
  - Flight Number versus Orbit Type
  - Payload versus Orbit Type

Scatter chart was used to visualize relationship between flight number and launch sites as well as to show relationship between payload mass and launch site to recognize historical trends and evaluate possible outcomes.

- *GitHub URL*

EDA with SQL

- Importing the dataset into SQL table in a database and executing folowing SQL queries to:

- Display the names of the unique launch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'.
- Display 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 succesful landing outcome in ground pad was acheived.
- 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 names of the booster_versions which have carried the maximum payload mass using subquery.
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

- *GitHub URL*

Build an Interactive Map with Folium

- Performing interactive visual analytics using Folium to:
- 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.
- Find geographical patterns about launch sites.

- Following geographical patterns about launch sites were found:
- Launch sites are in close proximity to railways, highways and coastline.
- Launch sites keep certain distance away from cities.

- *GitHub URL*

Build a Dashboard with Plotly Dash

- Dashboard application was written in Python using Plotly Dash libraries to construct interactive:

  - Pie chart
    - To show total success launches displayed by all, by combination of selection of sites or by individual launch sites. Pie chart enables to immediately see the success rate of individual launch sites.
  - Scatter plot
    - To interactively display the relationship between landing outcomes and payload mass by different types of boostets.
    - Contains two inputs; individual or any combination of launch sites and its respective payload mass.
    - Scatter plot largely assists with determining dependency of success on the launch site, payload mass and booster version type.

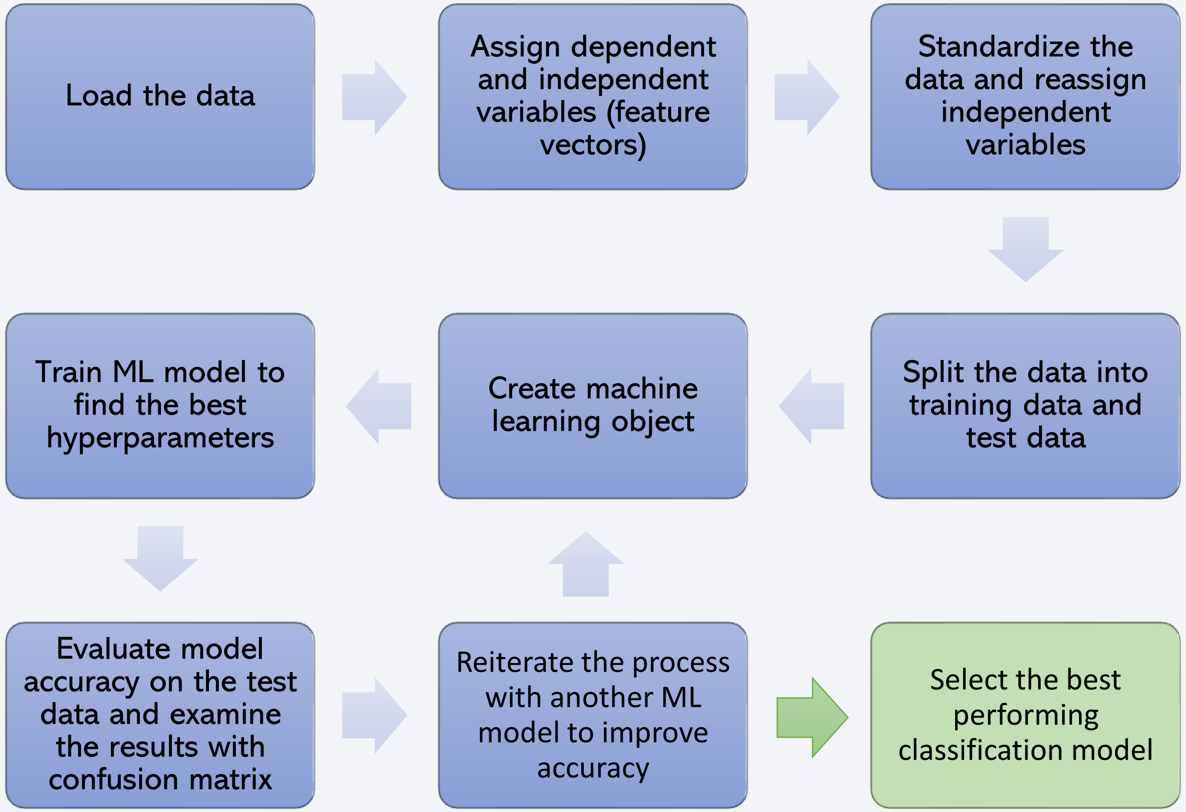
- *GitHub URL (Python code)*

Predictive Analysis (Classification)

- Performing exploratory data analysis and determining training labels
  - Create a column for the class.
  - Standardize the data.
  - Split into training data and test data.

- Finding best hyperparameter for KNN, SVM, Classification Trees and Logistic Regression.
  - Find which method performs best using test data.

Supervised Machine Learning Model Selection Process



- [*GitHub URL*]

Results

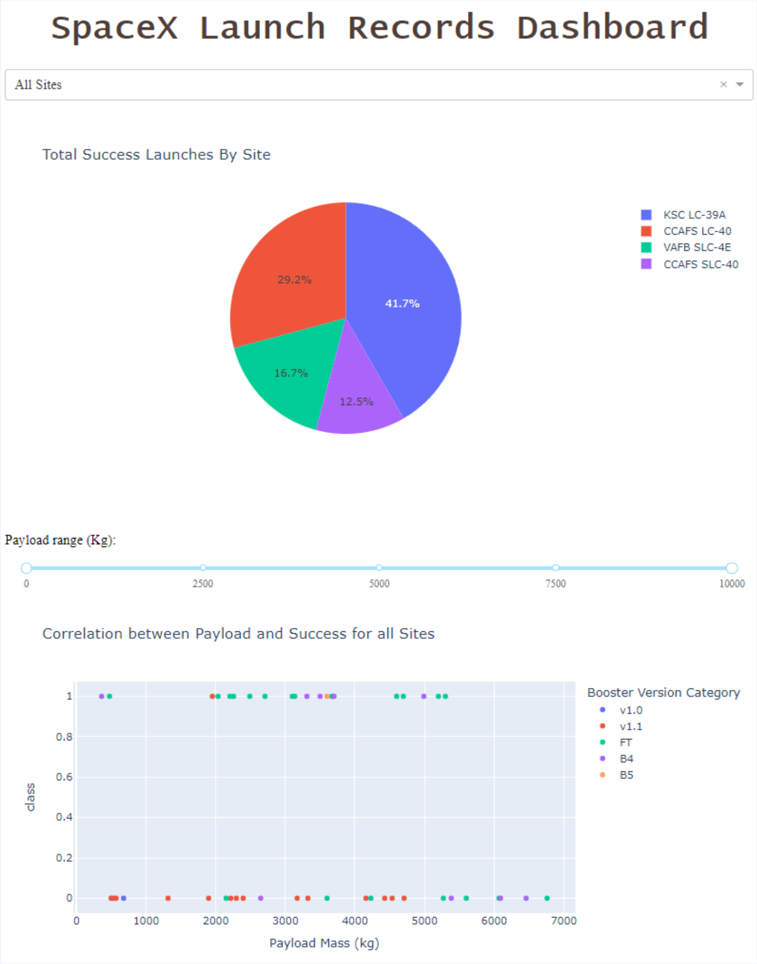
- The EDA results with visualization, EDA with SQL, interactive map with Folium as well as interactive analytics dashboard are shown in the following slides.

- The accuracy of all ML methods has been virtually the same across all of the models. After testing the models on test data, all methods delivered accuracy of ~83.3%.

Accuracy result

Algorithm	Accuracy
KNN	0.8333333
Decision Tree	0.8333333
SVM	0.8333333
LogisticRegression	0.8333333

Demo screenshot: Dashboard with Plotly Dash developed for interactive analytics



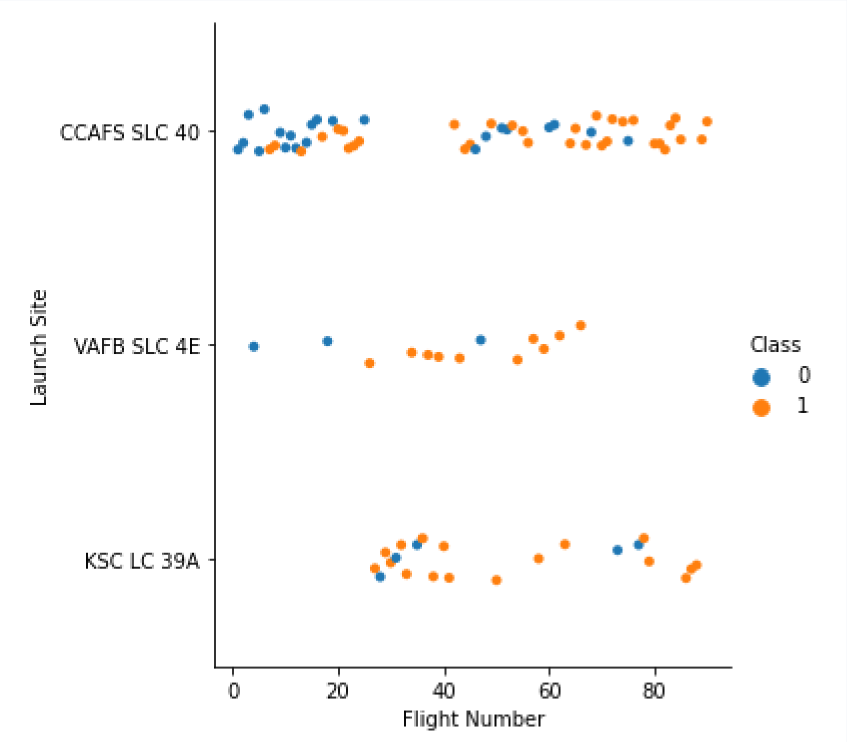
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- **Scatterplot legend description:**

Mission outcomes are divided into two categories, `Class 0` (blue color) repeseenting unsuccessful landing and `Class 1` (orange color) successful landing.



- There is clear trend of achieving a higher landing success rate with an increased number of flights.
- More flights have improved landings across all launch sites.

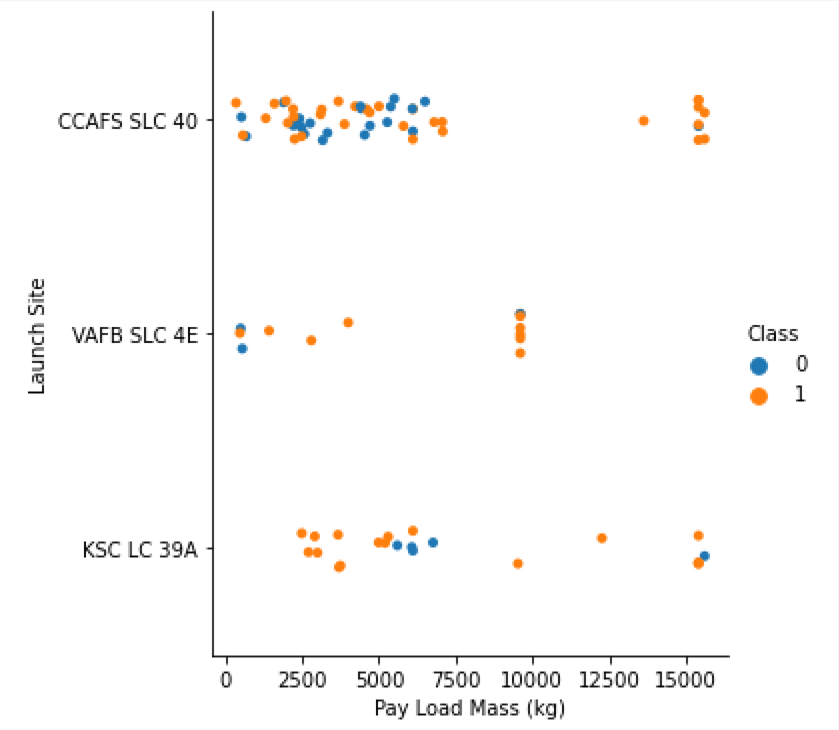
Payload vs. Launch Site

- **Scatterplot legend description:**

Mission outcomes are divided into two categories, `Class 0` (blue color) repesenting unsuccessful landing and `Class 1` (orange color) successful landing.

- Heavier payload has historically resulted in a higher success rate, however there is no clear pattern visible to derive such conclusions based only on respective launch sites as it can be also attributed to previous conclusion that low number of flights generally led to lower success rate just as in case of CCAFS SLC-40 launch site.

- There have been no rockets launched from VAFB-SLC 4E launch site with payload greater than 10 metric tons.

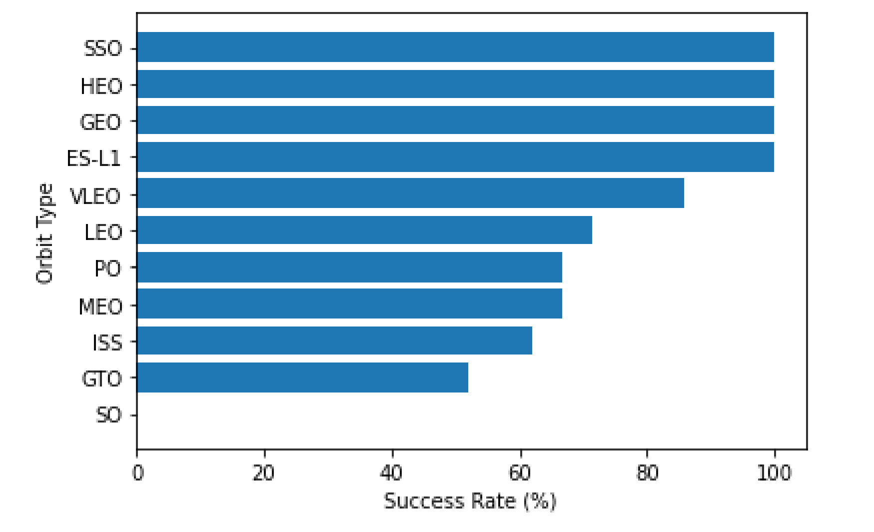


Success Rate vs. Orbit Type

- Missions with SSO, HEO, GEO and ES-L1 orbits have all achieved 100% landing success rate, however HEO, GEO and ES-L1 orbits have been used only once, which makes previous general conclusion statistically insignificant, while SSO orbit has been used five times.

- While being the most common type of orbital destination, both GTO and ISS orbits have ac hieved lowest landing success rates between 53% and 62%, while LEO orbit; being the third most common orbit with a total of 14 flights; achieved success of roughly 85%.

- SO orbit was used only once and ended in a landing failure.



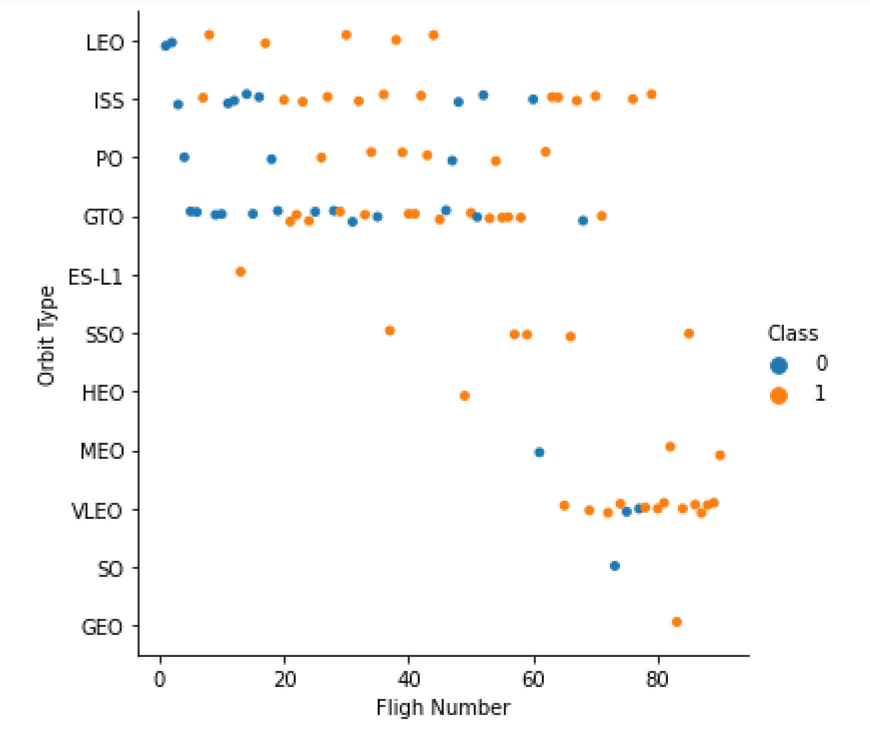
Flight Number vs. Orbit Type

- **Scatterplot legend description:**

Mission outcomes are divided into two categories, `Class 0` (blue color) repeseting unsuccessful landing and `Class 1` (orange color) successful landing.

- As in the first figure (Flight Number vs. Launch Site), the successful landing outcome has been historically postively correlated with number of flights, rate of success in LEO, ISS, and PO orbits has historically increased with number of flights as well, however there seems to be no relationship between flight number when in GTO orbit.

- Most recently used VLEO orbit has considerably higher average landing success rate similar to its precedessor LEO orbit.



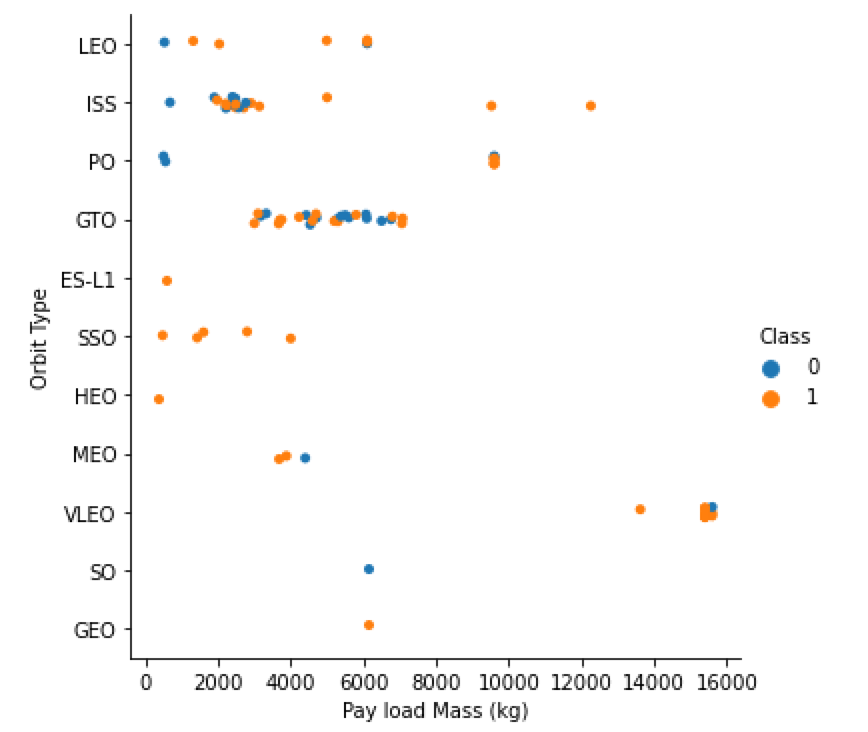
Payload vs. Orbit Type

- **Scatterplot legend description:**

Mission outcomes are divided into two categories, `Class 0` (blue color) repesenting unsuccessful landing and `Class 1` (orange color) successful landing.

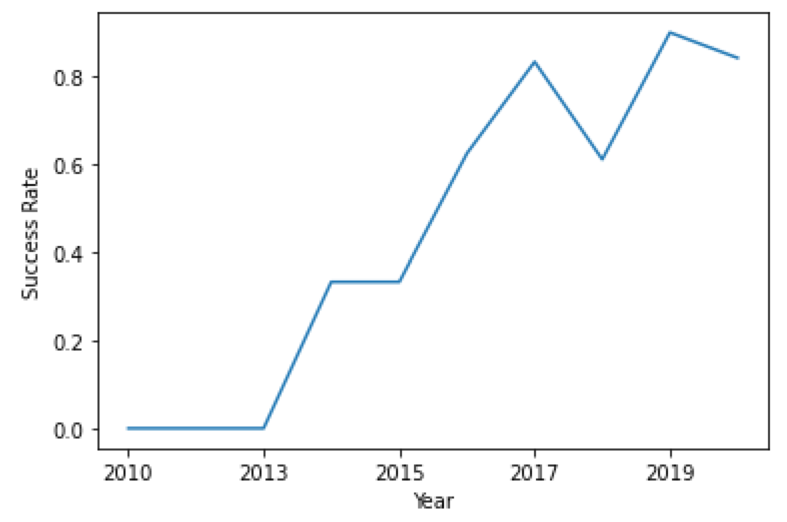
- With heavier payloads the successful landing or positive landing rates are higher for PO, LEO and ISS.

- The same is inconclusive for GTO orbit type. In case of GTO orbit, both positive landing rates and negative landing rates do not form clear trend in relation to payload mass.



Launch Success Yearly Trend

- Historically, there has been an increasing number of successful missions between 2013 and 2017.
- Rate of success declined slightly in 2018.
- Most recently the average success rate increased above 80%.



All Launch Site Names

QUERY:

```
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

RESULT:

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

- SQL DISTINCT clause enables to only include unique values in a result from the Launch_Site column from the SPACEXTBL table in the database.
- There are 4 launch sites: CCAFS LC-40, VAFB SLC-4E, KSC LC-39A and CCAFS SLC-40.

Launch Site Names Begin with 'CCA'

QUERY:

```
SELECT * FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5
```

- By using LIMIT 5 clause in the query, only 5 records of the SPACEXTBL were displayed.
- To call specific string beggining with CCA, LIKE operator and the percentage sign (%) added after the specific string were used.

RESULT:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

QUERY:

```
SELECT SUM(PAYLOAD_MASS__KG_) AS
total_payload_mass_kg
FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)'
```

RESULT:

total_payload_mass_kg
45596

- SUM( ) function was used to calculate the sum of column PAYLOAD_MASS__KG_.
- To perform calculations only on the NASA (CRS) customer, WHERE clause was further specified by the string containing customer name.

Average Payload Mass by F9 v1.1

QUERY:

```
SELECT AVG(PAYLOAD_MASS__KG_) AS
avg_payload_mass_kg
FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

RESULT:

avg_payload_mass_kg
2928.4

- AVG() function was used to calculate simple arithmetic average of column PAYLOAD_MASS__KG_. as a avg_payload_mass_kg.
- To perform calculations only on the F9 v1.1 booster, WHERE clause was further specified by the string containing booster version id.

First Successful Ground Landing Date

QUERY:

```
SELECT MIN(DATE) AS first_successful_landing_date
FROM SPACEXTBL
WHERE `Landing _Outcome` = 'Success (ground pad)'
```

RESULT:

first_successful_landing_date
01-05-2017

- MIN(DATE) function was used to find the first successful landing date from the column DATE.
- To find only the first successful landing on the ground pad, WHERE clause was further specified by a string containing appropriate Landing _Outcome.

Successful Drone Ship Landing with Payload between 4000 and 6000

QUERY:

```
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE `Landing _Outcome` = 'Success (drone ship)'
    AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

RESULT:

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

- On top of the specific string in the where clause (`BETWEEN 4000 AND 6000`) was added to the WHERE clause with AND `operator` to display only the records with specific payload.

- To return only successful landings on a drone ship, WHERE clause was further specified by string containing appropriate `Landing _Outcome`.

Total Number of Successful and Failure Mission Outcomes

QUERY:

```
SELECT MISSION_OUTCOME, COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

RESULT:

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

- `GROUP BY` statement was used to list the total number of successful and failure mission outcomes.
- `COUNT()` function was used to count the number of outcomes as `total_number` in each `Mission_Outcome`.
- SpaceX has historically achieved mission success rate above 99% in the analyzed data series.

Boosters Carried Maximum Payload

QUERY:

```
SELECT DISTINCT BOOSTER_VERSION, PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL);
```

RESULT:

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

- Subquery was used to list the names of the `booster_versions` which have carried the maximum payload mass.
- In order to find only boosters that carried the heaviest payload from `SPACEXTBL` table, `WHERE` clause was further specified by `SELECT MAX()` subquery enclosed by round brackets.
- Boosters version F9 B5 B10XX.X carried the heaviest payload mass.

2015 Launch Records

QUERY:

```
SELECT "Landing _Outcome", BOOSTER_VERSION,
LAUNCH_SITE
FROM SPACEXTBL
WHERE `Landing _Outcome` = 'Failure (drone ship)'
and SUBSTR(Date,7,4)='2015'
```

RESULT:

Landing _Outcome	Booster_Version	Launch_Site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- `SUBSTR(DATE, 4, 2)` was used to compensate for the lack of monthnames of SQLLite.
- To perform search only on the failed landing on a drone ship in 2015, WHERE clause was further specified by the string containing appropriate `Landing _Outcome` and SUBSTR.

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

QUERY:

```
SELECT ("Landing _Outcome"), COUNT("Landing
_Outcome") AS TOTAL_LANDINGS
FROM SPACEXTBL
WHERE Date BETWEEN '04-06-2010' AND '20-03-2017'
GROUP BY "Landing _Outcome"
ORDER BY TOTAL_LANDINGS DESC
```

RESULT:

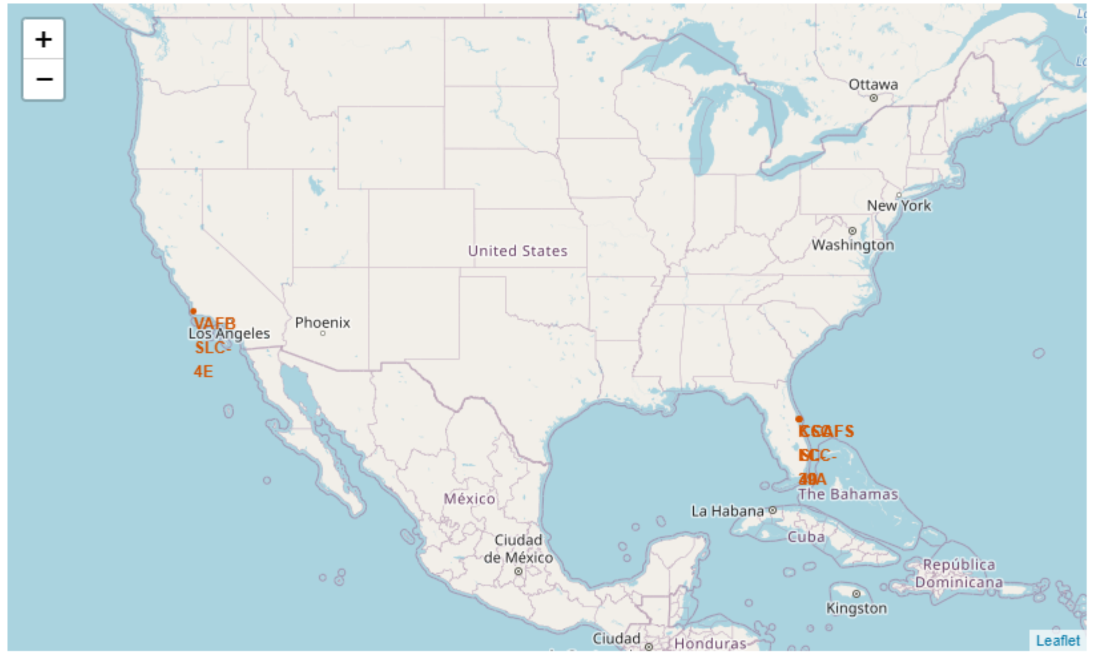
Landing _Outcome	TOTAL_LANDINGS
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

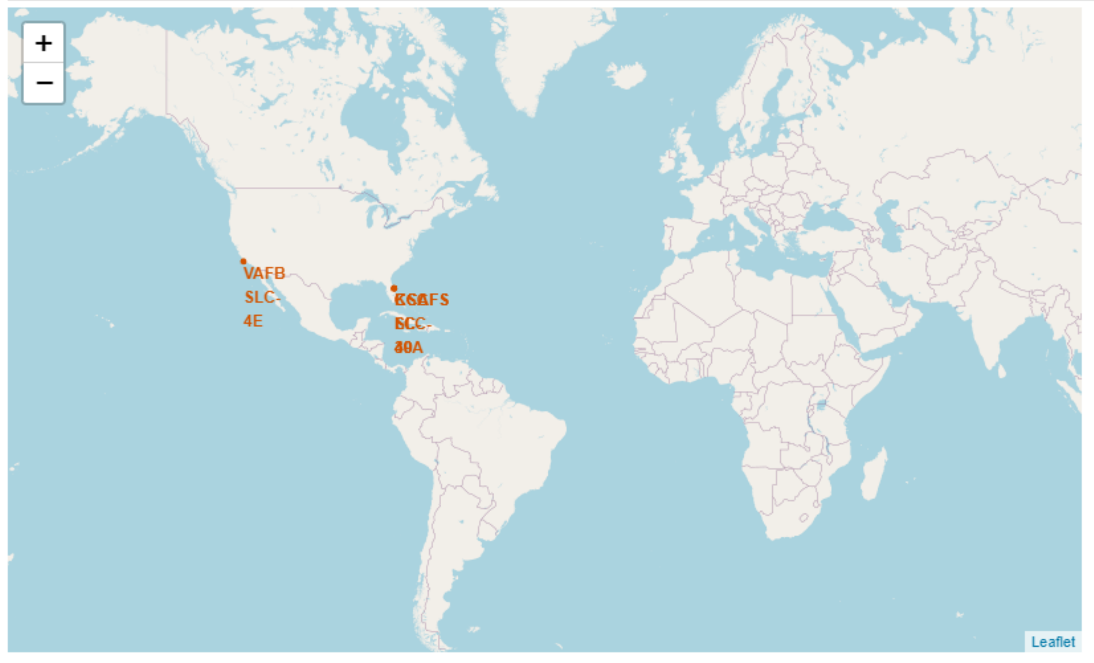
- GROUP BY statement was used to perform counts by unique landing outcomes, ORDER BY statement followed by keyword followed by DESC was used to sort the result in a descending order.
- To perform the query only on the specific time frame BETWEEN along with Date column was used to return the result.

Section 3

Launch Sites Proximities Analysis

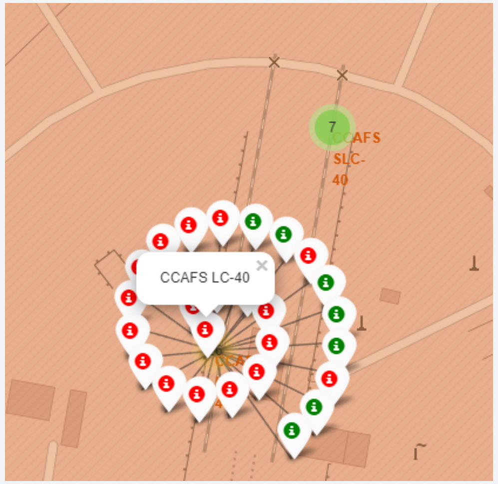
Launch Sites Locations – Folium Visual Analytics

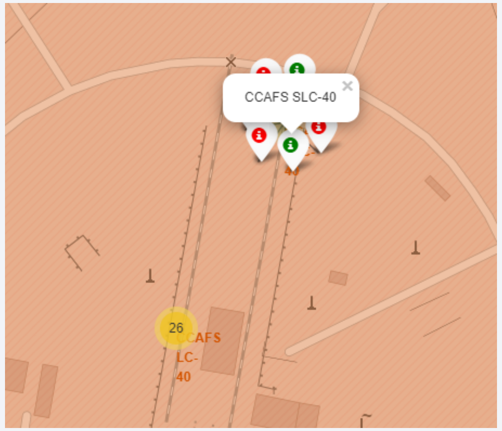


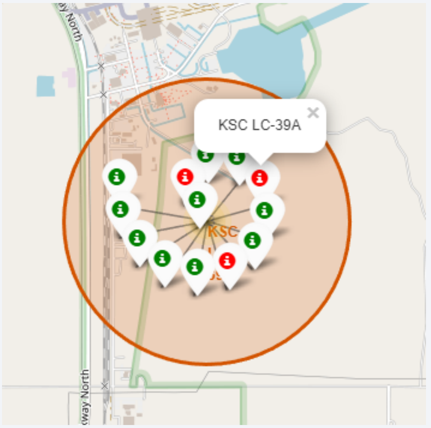


- All SpaceX launch sites are located in the United States, namely in California's Space Launch Complex 4 near Vandenberg Space Force Base and on Merritt Island and on Cape Canaveral Space Station in Florida.
- All launch sites are located near the coast with safe distance from urbanized densely populated areas.
- Launch sites are specified by label, latitude and longitude using the data from spacex_launch_geo dataset. Folium class `.map.marker()` is used to draw markers on the map.

Launch Outcomes – Folium Visual Analytics

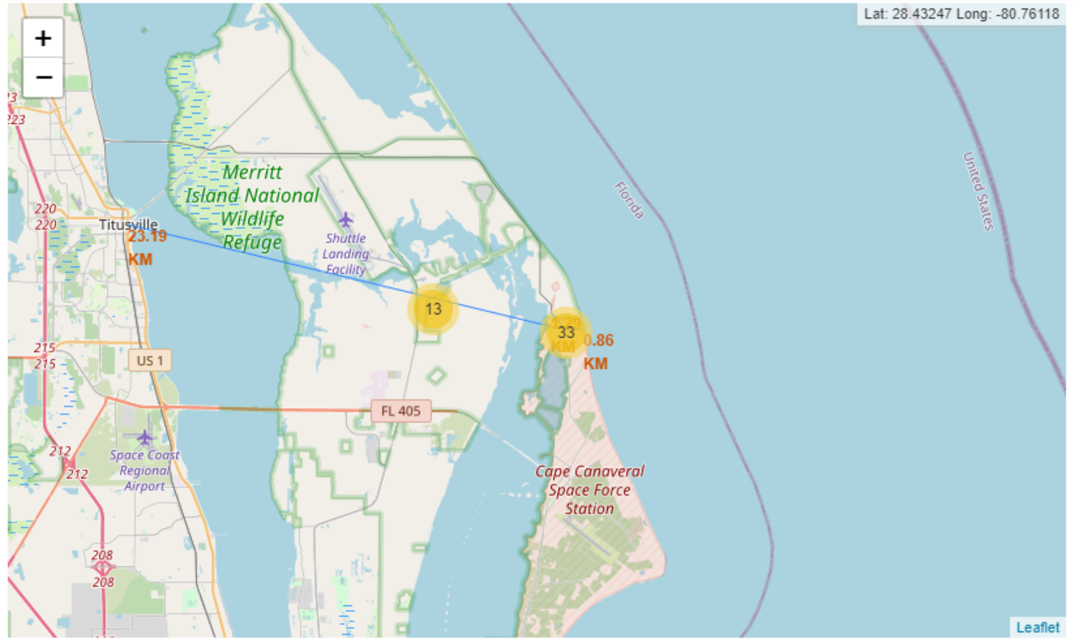


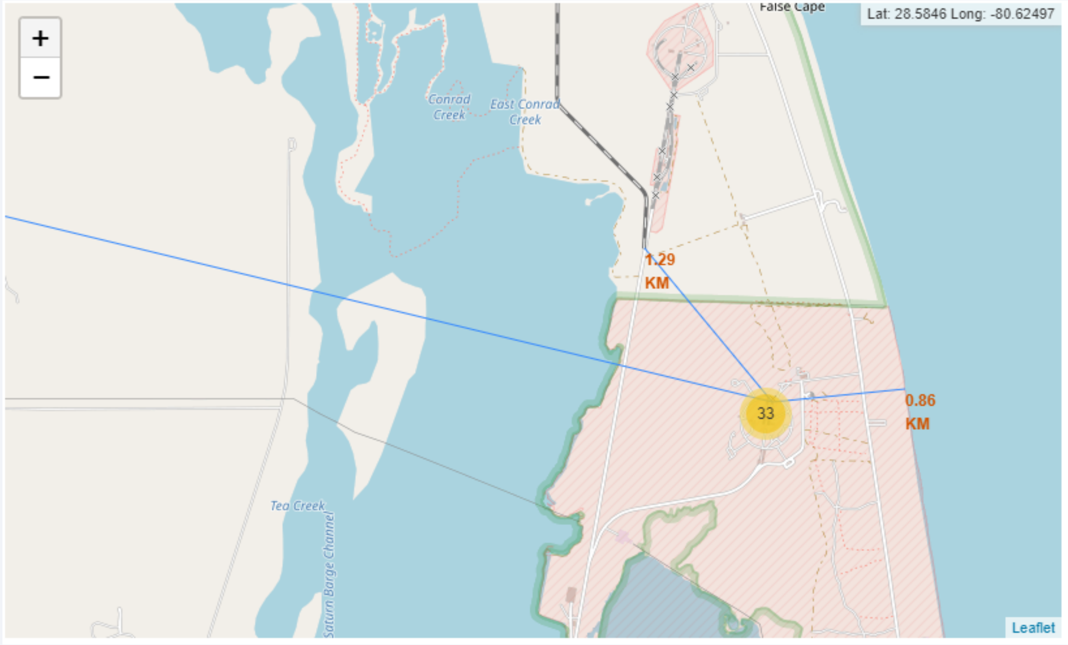




- Visualization is further enhanced by adding markers for all launch records.
- Launch outcomes are labeled with marker clusters, specific landing outcomes are displayed by clicking on the cluster. Failed landings (`class=0`) are labeled red and successful landings (`class=1`) green.

Proximity Analysis – Folium Visual Analytics





- All launch sites are in close proximity to railways, highways and coastlines while keeping a certain distance away from cities to maximize safety in case of launch or landing failure.
- To calculate the distance between two points on a sphere based on longitude and latitude, a new function based on the Haversine formula is used.

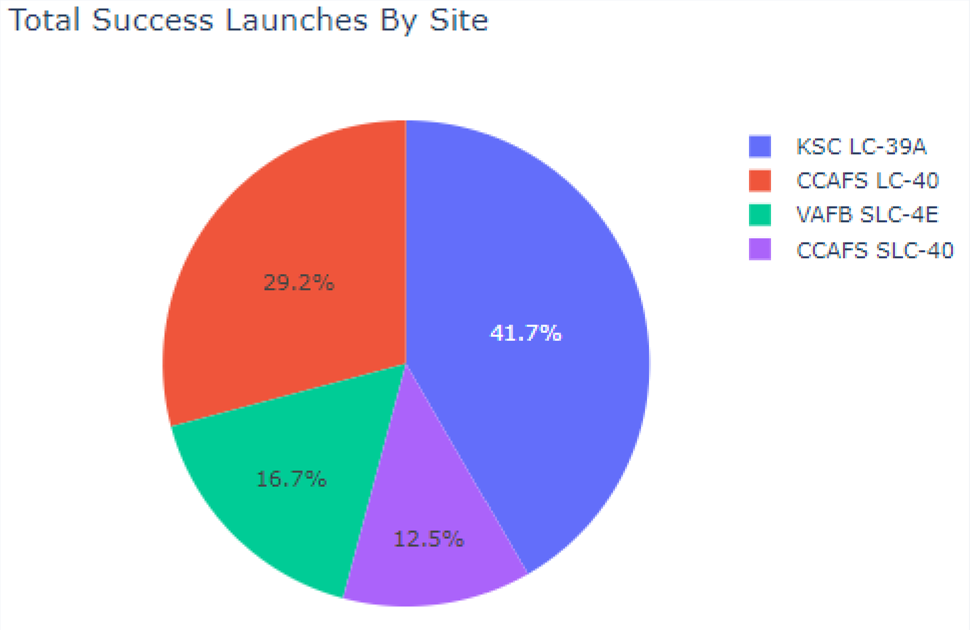
Section 4

Build a Dashboard with Plotly Dash

Total Success Launches By All Sites

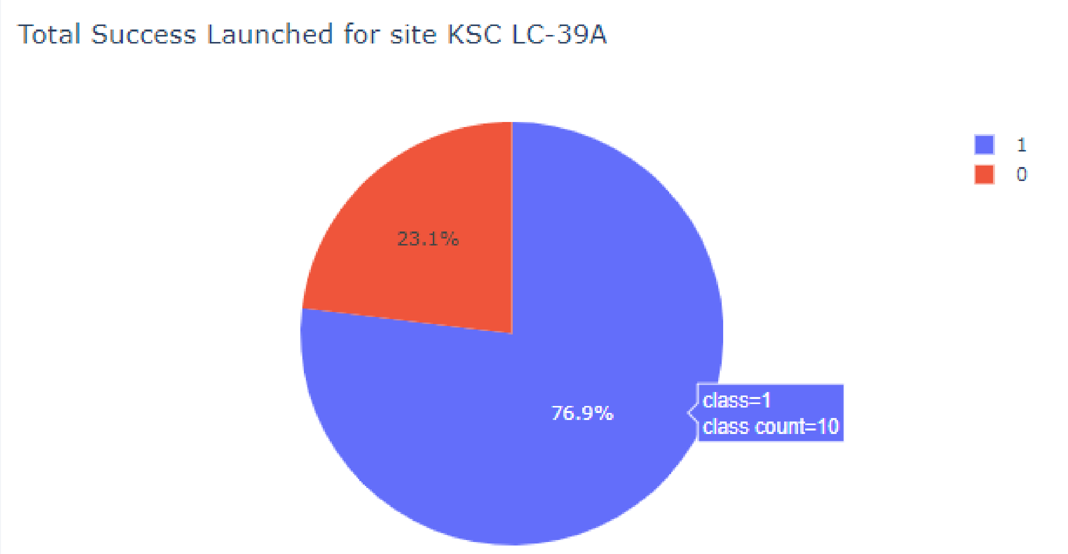
- Most of successful launches have been accomplished at KSC LC-39A launch site located on Meritt Island, Florida.

- California's launch site has been seldom used in comparison to the ones located in Florida hence VAFB-SLC-4E launch site composes only 16.7% of successful launches.

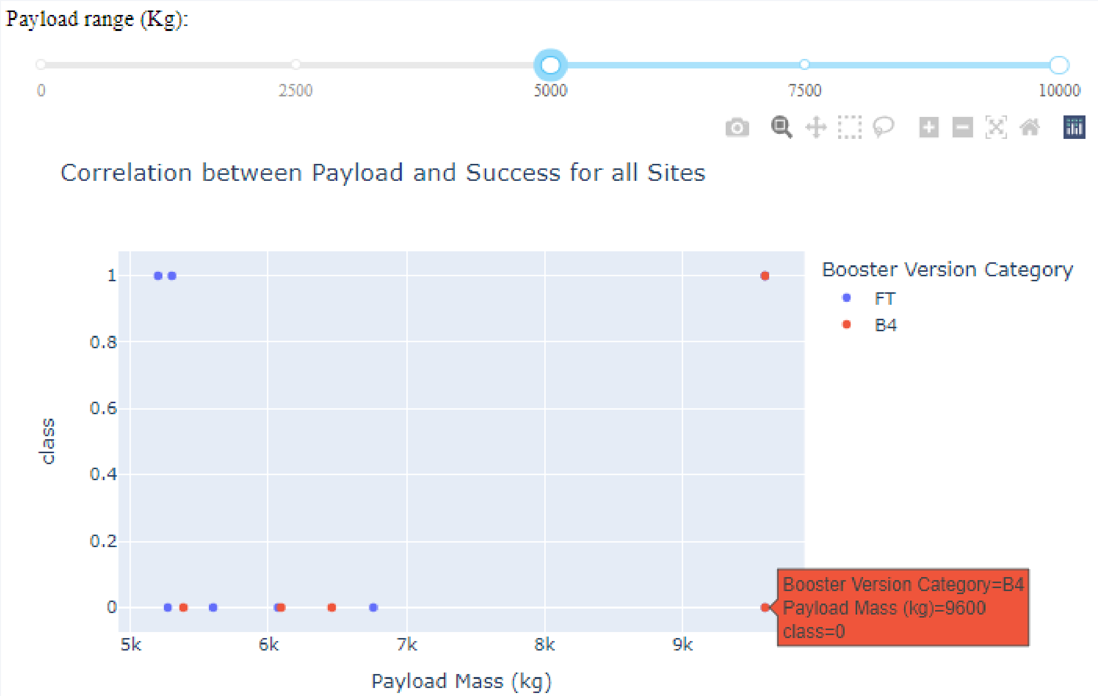


Most Successful Launch Site

- KSC LC-39A launch site; located on Merritt Island, Florida, has the highest success rate of 76.9% achieved with 10 landing successes and 3 landing failures.



Payload Mass & Success Correlation



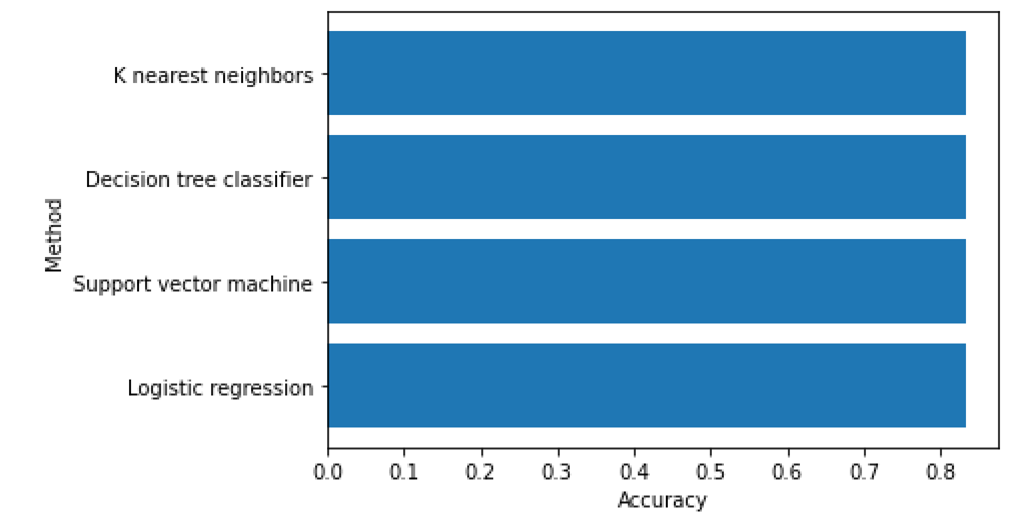
- Based on correlation analysis, missions with payload mass between 0 to 5 tons have higher launch success rate than those with heavier payloads of up to 10 tons.
- Only FT and B4 booster versions have been used to carry loads heavier than 5 tons.

Section 5

Predictive Analysis (Classification)

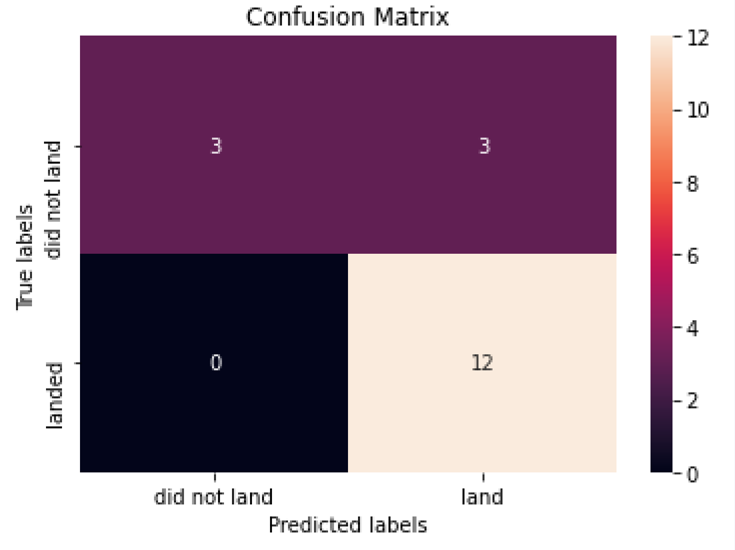
Classification Accuracy

- All built classification models have demonstrated the same accuracy on the test data.
- Reason behind is rather small test size containing only 18 test samples.
- Bigger dataset is needed in order to increase precision of accuracy evaluation.



Confusion Matrix

- Since all models have performed virtually the same on the test data, the confusion matrix is the same for all classification models.
- Out of a test sample consisting of 18 samples, classification models correctly predicted 12 successful landings (true positive) and 3 failed landings (true negative).
- However models have also predicted 3 successful landings, while the actual landing outcome was failure (false positive type 1 error).
- Overall all of the built classification models have demonstrated the ability to predict successful landings.



Conclusions

- The rate of successful landing has improved as the number of flights increased across all launch sites, most recent success rates have climbed above 80%.
- SSO, HEO, GEO and ES-L1 orbits have achieved 100% success rate, however there was only 1 orbit attempt with HEO, GEO and ES-L1 destinations. Successful landing on LEO orbit appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
- The launch success rate is higher with payloads below 5 metric tons in comparison to heavier payloads. KSC LC-39A launch site; located on Merritt Island, Florida, has the highest success rate of 76.9%.
- All supervised classification learning methods (K-Nearest Neighbors, Support Vector Machines, Logistic Regression and Decision Trees algorithm) have demonstrated the same accuracy (83.33%), bigger dataset is needed in order to increase precision of accuracy evaluation.

Appendix

- GitHub URL
- List of Falcon 9 and Falcon Heavy launches
- SpaceX Falcon 9 official webpage
- IBM (Coursera) – Applied Data Science Capstone URL

Thank you!