Green Data Academy rekrykoulutus

Hei,

olen hakemassa mukaan tähän rekrykoulutukseen. Tässä esityksessä työnäyte hakemukseni liitteeksi.

Terveisin,
Sami Lähde

samiolavi.lahde@gmail.com
p. 040 82 045 09

IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sami Lähde
September 2023

Table of contents

- Executive Summary
- Introduction of the project
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

Summary of methodologies

- Data Collection was performed using the SpaceX REST API and by Web Scraping the SpaceX Wikipedia page
- Data Wrangling enabled the handling of missing values and formatting the data
- Exploratory Data Analysis (EDA) was performed with SQL queries and Python visualization libraries
- Data Visualization was carried out with Folium maps and using Plotly Dash for comprehensive dashboarding
- Predictive Analysis (Classification) was implemented with the Python “scikit-learn” package

Summary of results

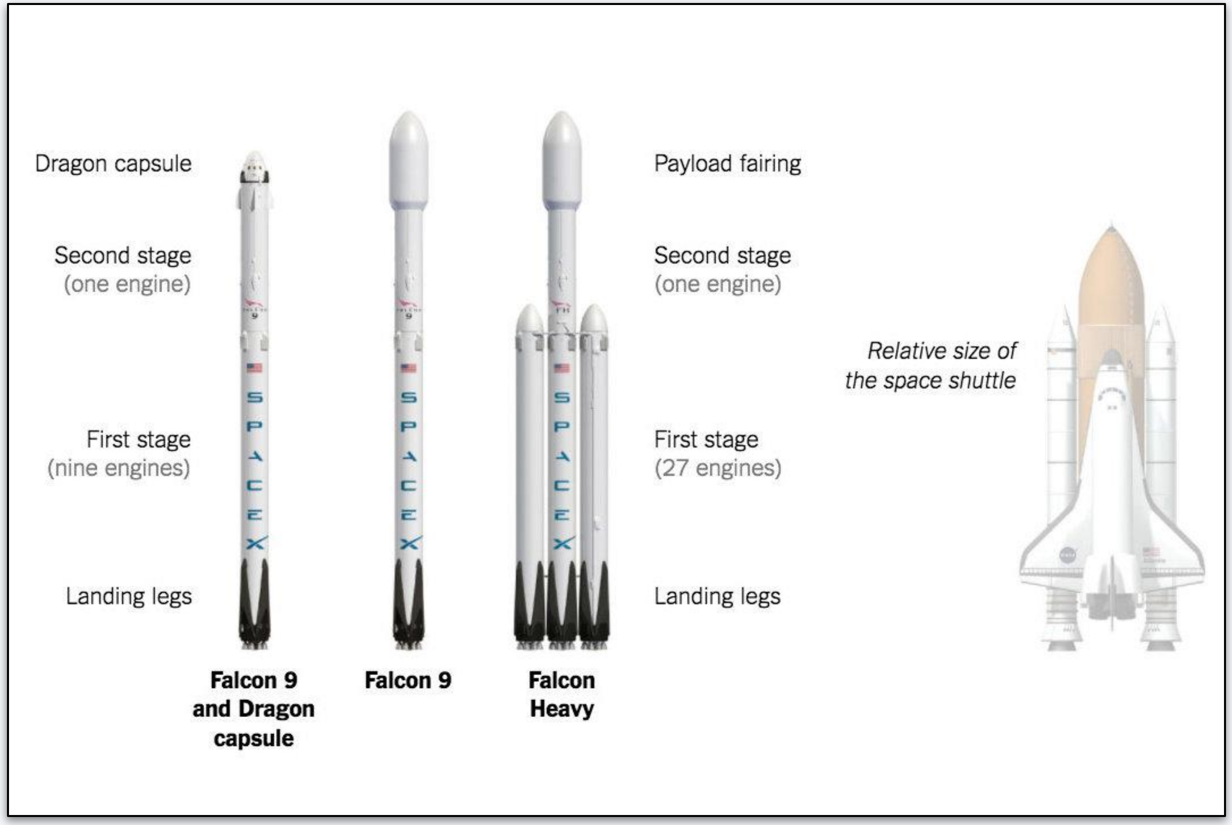
- Data Collection and Data Wrangling allowed for the obtainment of a robust dataset
- EDA identified the features to be used for the predictive analysis stage
- Interactive Analytics Visualizations are presented in screenshots
- Predictive Analysis using Machine Learning techniques indicated the best classification model to predict successful launches in the future

Introduction of the project

Project background and context

SpaceX advertises Falcon 9 rocket launches with a cost of 62 million dollars while other providers cost upward 165 million dollars each. Falcon 9 rocket has two stages: the first stage (booster) carries the second stage and the rockets payload. Much of the saving is because SpaceX can reuse the first stage.

Therefore if we can determine if the first stage will land and can be then reused, we can determine the cost of Falcon 9 launch in the future. This information can be used for example by companies and individuals who are thinking about investing or bidding against SpaceX.



Dragon capsule
Second stage (one engine)
First stage (nine engines)
Landing legs
Payload fairing
Second stage (one engine)
First stage (27 engines)
Landing legs
Relative size of the space shuttle
Falcon 9 and Dragon capsule
Falcon 9
Falcon Heavy

Source of the picture: www.spacex.com

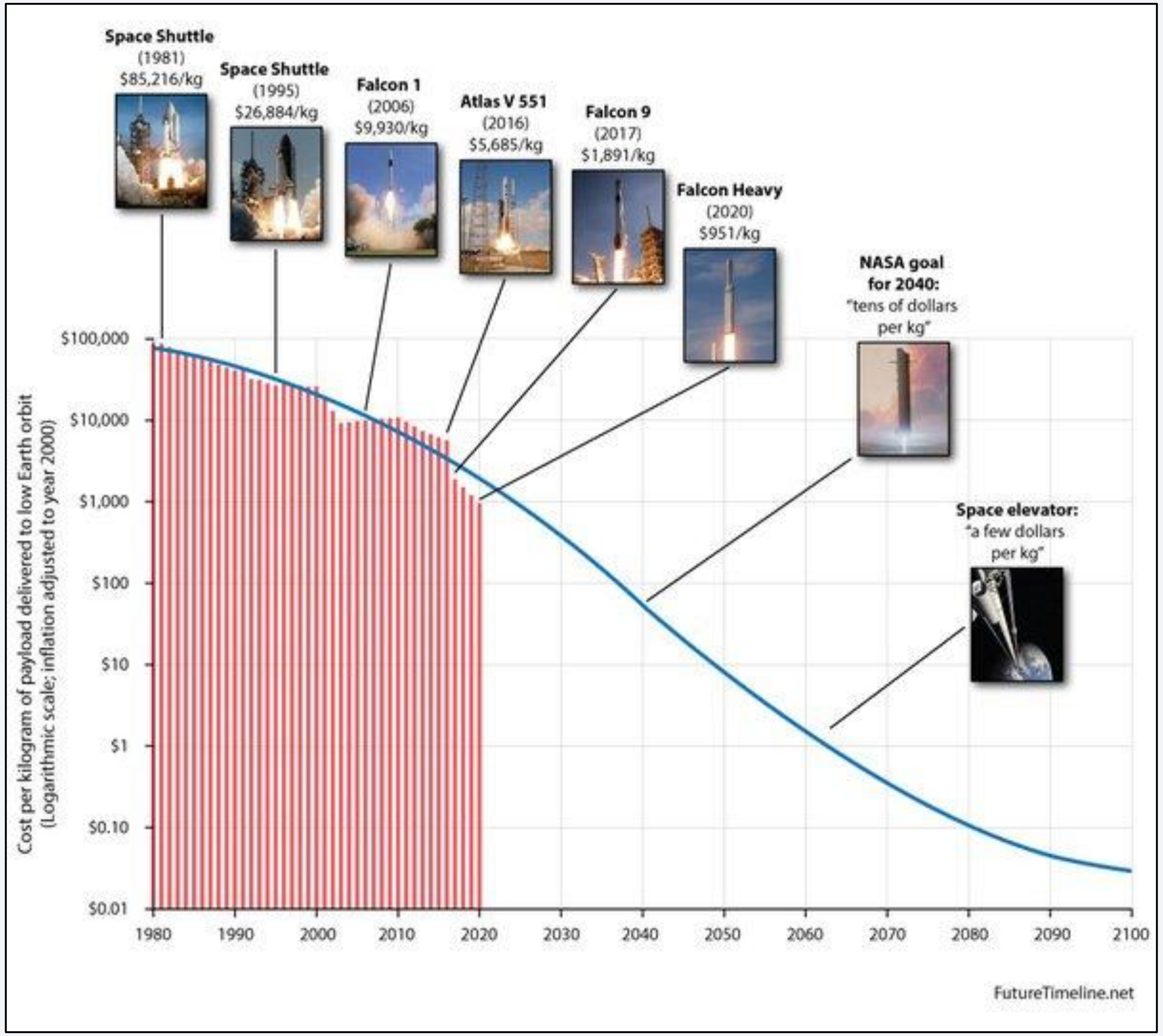
Introduction of the project

Problems we want to find answers

First we want to make an explanatory analysis on SpaceX previous launches to see how well they have done in the past.

We want find factors affecting successful landings of the rockets, the effect of each relationship of rocket variables on outcome and answer has the rate of successful landings increased over time.

After that we want to find a predictive model that can help us predicting the future SpaceX launch success.



Source of the picture: www.futuretimeline.net

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**

  Data collected via SpaceX Rest API & Web Scraping from Wikipedia

- **Perform data wrangling**

  Transforming data for Machine Learning with one hot encoding data fields

- **Perform exploratory data analysis (EDA) using visualization and SQL**

  Show patterns between data using Scatter and bar graphs

- **Perform interactive visual analytics using Folium and Plotly Dash**

  Folium and Plotly Dash Visualizations

- **Perform predictive analysis using classification models**

  Build and evaluate classification models

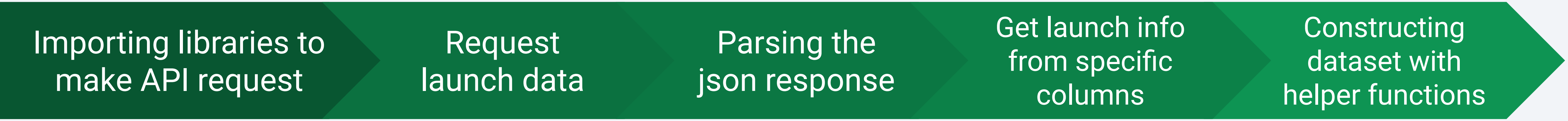
Data Collection

Data for the project was collected from:

1. SpaceX REST API and
2. SpaceX Falcon 9 Wikipedia page

In the next two slides there is flowcharts for both data collection method and one flowchart about data wrangling with links to the original codes stored in GitHub.

Data Collection – SpaceX API



Importing libraries to make API request
Request launch data
Parsing the json response
Get launch info from specific columns
Constructing dataset with helper functions

```
import pandas as pd
import request
import numpy as np
import datetime
```

```
spacex_url = https://api.spacexdata.com/v4/launches/past

response = request.get(spacex_url)
```

response.status_code = 200 (*OK, request was successful*)

```
data_1 = response.json()

df = pd.json_normalize(data_1)
```

```
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

```
def getBoosterVersion(data)

def getLaunchSite(data):

def getPayloadData(data):

def getCoreData(data):
```

Link to GitHub

Data Collection – Scraping from Wikipedia

Importing libraries to make request

Request data from Wikipedia tables

Extracting each table

Constructing dataset with helper functions

Making sure scraping succeed

```
import pandas as pd
import request
import sys
import re
import unicodedata

from bs4 import
BeautifulSoup
```

```
static_url =
https://en.wikipedia.org/w/index
.php?title=List_of_Falcon_9_and_
Falcon_Heavy_launches&oldid=1027
686922

response =
request.get(static_url)

soup =
BeatifulSoup(response.cont
ent, “html.parser”)

all_tables =
soup.find_all(“table”)
```

```
for table_number, table in
enumerate(soup.find_all('t
able', "wikitable
plainrowheaders
collapsible")):
```

```
def
date_time(table_cells):

def
booster_version(table_cells):

def
landing_status(table_cells):

def get_mass(table_cells):

def
extract_column_from_header
(row):
```

```
for key in
launch_dict_2.keys():
    values_length =
len(launch_dict_2[key])
    print(f"Key: {key},
Length: {values_length}")

Key: Flight No., Length:
121 Key:
Version,Booster[b],
Length: 121
.. ETC (OK!)
```

Link to GitHub

Data Wrangling

Read the data to dataframe

Read the data collected from SpaceX REST API to pandas DataFrame

Define the task

There is a need to create a new label to the dataframe which divides landing outcomes to either as “success” or “failure”.

1 = Success
0 = Failure

Sort “failures” and “success”

Values defined to labeled as “failures”:

```
'False ASDS',
'False Ocean',
'False RTLS',
'None ASDS',
'None None'
```

Create a set “bad_outcomes” that helps to create a new column to the dataframe

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

Add new column for the dataframe

Insert new column to the original dataframe. This new column now has the landing outcome values as either “1” or “0”.

```
df_modified["LandingClass"]
= landing_class_1
```

Save the new dataframe for Machine Learning use

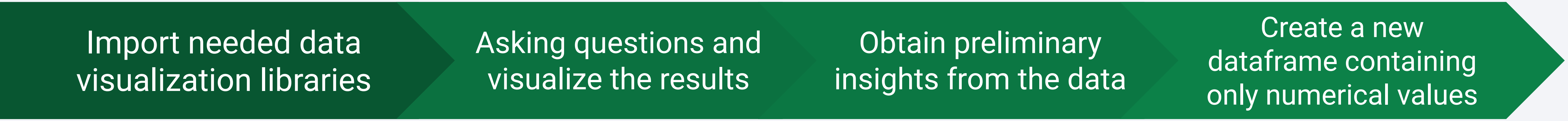
Link to GitHub

EDA with Data Visualization

After the data for the project was collected and wrangled, **E**xploratory **D**ata **A**nalysis (EDA) was done to find patterns in the data and see how well previous SpaceX Falcon 9 launches had succeeded. This information would also help to determine the labels for training supervised model.

In the next slide there is a flowchart describing how data was visualized and how selecting the features for supervised model was done including the link to the original code stored in GitHub.

EDA with Data Visualization



Import needed data visualization libraries

Asking questions and visualize the results

Obtain preliminary insights from the data

Create a new dataframe containing only numerical values

import matplotlib.pyplot as plt

import seaborn as sns

We wanted to know for example how different features affected each other and what had been the success rate of previous Falcon 9 launches, visualization was needed to communicate our findings better

Features that were selected to be used in success prediction in the future module were:

'FlightNumber', '
PayloadMass',
'Orbit',
'LaunchSite',
'Flights',
'GridFins',
'Reused',
'Legs',
'LandingPad',
'Block',
'ReusedCount',
'Serial'

get_dummies() function was used for features that were categorical columns (Orbits, LaunchSite, LandingPad and Serial).

After that all numerical columns were casted to 'float64' type. This new dataframe with only numerical values was then taken to the next module (Predictive Analysis, see page 21)

Link to GitHub

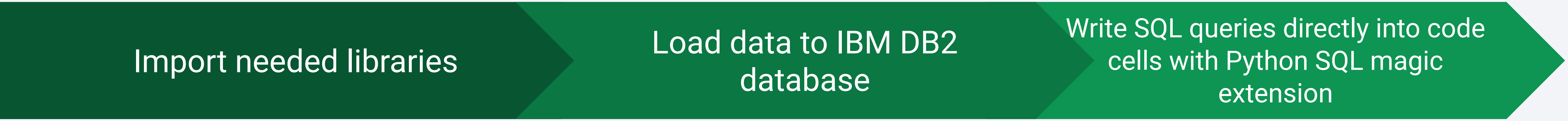
EDA with SQL

After the data for the project was collected and wrangled, **E**xploratory **D**ata **A**nalysis (EDA) was done to find patterns in the data and also see how well previous SpaceX Falcon 9 launches had succeeded.

Besides data visualization, EDA was also done by using SQL.

In the next slide there is a flowchart describing how exploratory data analysis was done with SQL including the link to the original code stored in GitHub.

EDA with SQL



Import needed libraries

Load data to IBM DB2 database

Write SQL queries directly into code cells with Python SQL magic extension

```
import sqlalchemy
import ibm_db_sa
import pandas
import ibm_db
```

101 rows of data was loaded to DB2 database with zero missing values

Connecting to database with magic:

```
%load_ext sql

%sql
ibm_db_sa://hdj24637:icUBSUt0
OKsYdiCE@XXX(hidden)
```

Start writing SQL queries

Link to GitHub

Interactive Visual Analytics

Interactive Visual Analytics enables users to direct data exploration and analytics and compared to static graphs it tells the users more compelling story.

To obtain more insights from gathered data about SpaceX Falcon 9 previous launches, two visual analytic tools were created:

1. Launch Site Geographical Data Map with Folium and
2. Data Dashboard with Plotly Dash

In the next slides there is a descriptions how Geo Map and Dashboard were created and links to the original codes stored in GitHub.

Launch Site Geographical Data Map with Folium

For the map I used circles, markers, marker clustering, polylines, and the mouse position plugin to create an informative and interactive map that visualizes launch sites, their proximity to each other, and provides coordinates information for user interaction.

Circles: I created circles on the map using the folium.Circle function. These circles represent areas around each launch site with a radius of 1 kilometer. Each circle has a popup label that shows the launch site name when clicked. The circles are filled with a yellow color.

Markers: I added markers to the map using the folium.Marker function. These markers represent the exact coordinates of each launch site. They have custom icons with a label displaying the launch site name. The icons are small and white.

MarkerCluster: I used the MarkerCluster plugin to group markers together when they are close to each other on the map. This improves map readability when multiple launch sites are nearby. Both the circles and markers were added to the MarkerCluster for better organization.

PolyLine: I drew a blue polyline on the map connecting the launch site to a selected coastline point. This line represents the path from the launch site to a specific location on the coastline.

Mouse Position: I added the "MousePosition" plugin to the map, which displays the latitude and longitude coordinates (Lat and Long) of the mouse cursor's position in real-time at the top-right corner of the map. This feature allows users to easily identify the coordinates of points on the map.

Link to GitHub

Data Dashboard with Plotly Dash, 1

Plots, graphs and interactions added to Dashboard

Dropdown Menu: I added a dropdown menu (dcc.Dropdown) to allow users to select a specific launch site or view data for all sites. This interaction gives users the flexibility to focus on individual launch sites or view aggregated data for all sites.

RangeSlider: I included a RangeSlider (dcc.RangeSlider) that allows users to filter data based on the payload mass. Users can select a range of payload masses, and the dashboard updates the plots accordingly. This interaction enables users to explore how payload mass correlates with launch success.

Pie Chart: I added a pie chart (dcc.Graph) that displays the distribution of launch results (success and failure) for the selected launch site or all sites combined. This chart provides an overview of the success rate and the number of failures for the chosen site(s).

Scatter Plot: I included a scatter plot (dcc.Graph) to visualize the correlation between payload mass and launch success. The scatter plot also differentiates points by the booster version category. Users can observe how different payload masses and booster versions affect launch outcomes.

Link to GitHub

Data Dashboard with Plotly Dash, 2

Why these elements were selected to the Dashboard

Dropdown Menu: The dropdown menu allows users to select specific launch sites or view data for all sites. This feature is essential for user customization and site-specific analysis. It enables users to explore launch results for different locations.

RangeSlider: The RangeSlider provides an interactive way for users to filter data based on payload mass. This feature helps users investigate the relationship between payload mass and launch success. Users can define custom payload mass ranges to analyze specific scenarios.

Pie Chart: The pie chart summarizes launch outcomes (success and failure) for the selected launch site(s). It offers a quick overview of the distribution of success and failure events. Users can easily compare success rates across different sites or for all sites combined.

Scatter Plot: The scatter plot visualizes the correlation between payload mass and launch success. By including booster version categories as color distinctions, users can identify patterns related to different booster versions. This plot provides insights into how payload mass and booster versions influence launch outcomes.

Predictive Analysis (Classification)

In Machine Learning, **classification** is one supervized learning approach.

In classification target attribute is a categorical variable.

In our case this target is either "0" or "1" representing fail and success for Falcon 9 landing.

By using data from previous Falcon 9 launches that we gathered from SpaceX API and Wikipedia, we can build a classifier and then fill it with data from future launches to predict if the future launch will be a failure or success.

Predictive Analysis (Classification)

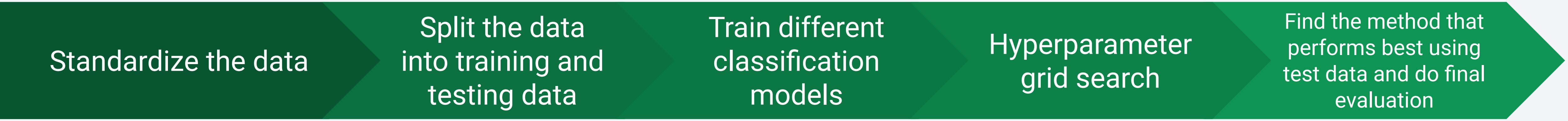
To be able to predict if the first stage of Falcon 9 will land successfully in the future, we build a Machine Learning Pipeline.

There is different kind of classification algorithms in machine learning. Here we are using four kind of them.

1. Logistic Regression
2. Support Vector Machine
3. Decision Tree Classifier and
4. K-Nearest Neighbour

In the next slide there is a flowchart of this pipeline including the link to the original code stored in GitHub.

Predictive Analysis (Classification)



Standardize the data

Split the data into training and testing data

Train different classification models

Hyperparameter grid search

Find the method that performs best using test data and do final evaluation

```
from sklearn.preprocessing import StandardScaler
```

Standardize the dataframe that has all the numerical and boolean values

```
X_train, X_test, Y_train, Y_test = train_test_split(X_standardized, Y, test_size=0.2, random_state=2)
```

Split the data into train and test sets. Into the equation goes the standardized dataframe and one NumPy array (Y) which contains the “0” and “1” values describing previous launch classes

Create Logistic Regression, Support Vector Machine, Decision Tree Classifier and K-Nearest Neighbour objects and then create a GridSearchCV object for each of them.

Fit the objects to find best hyperparameters

Display the best hyperparameters and accuracy of the validation data for each of the four classification model

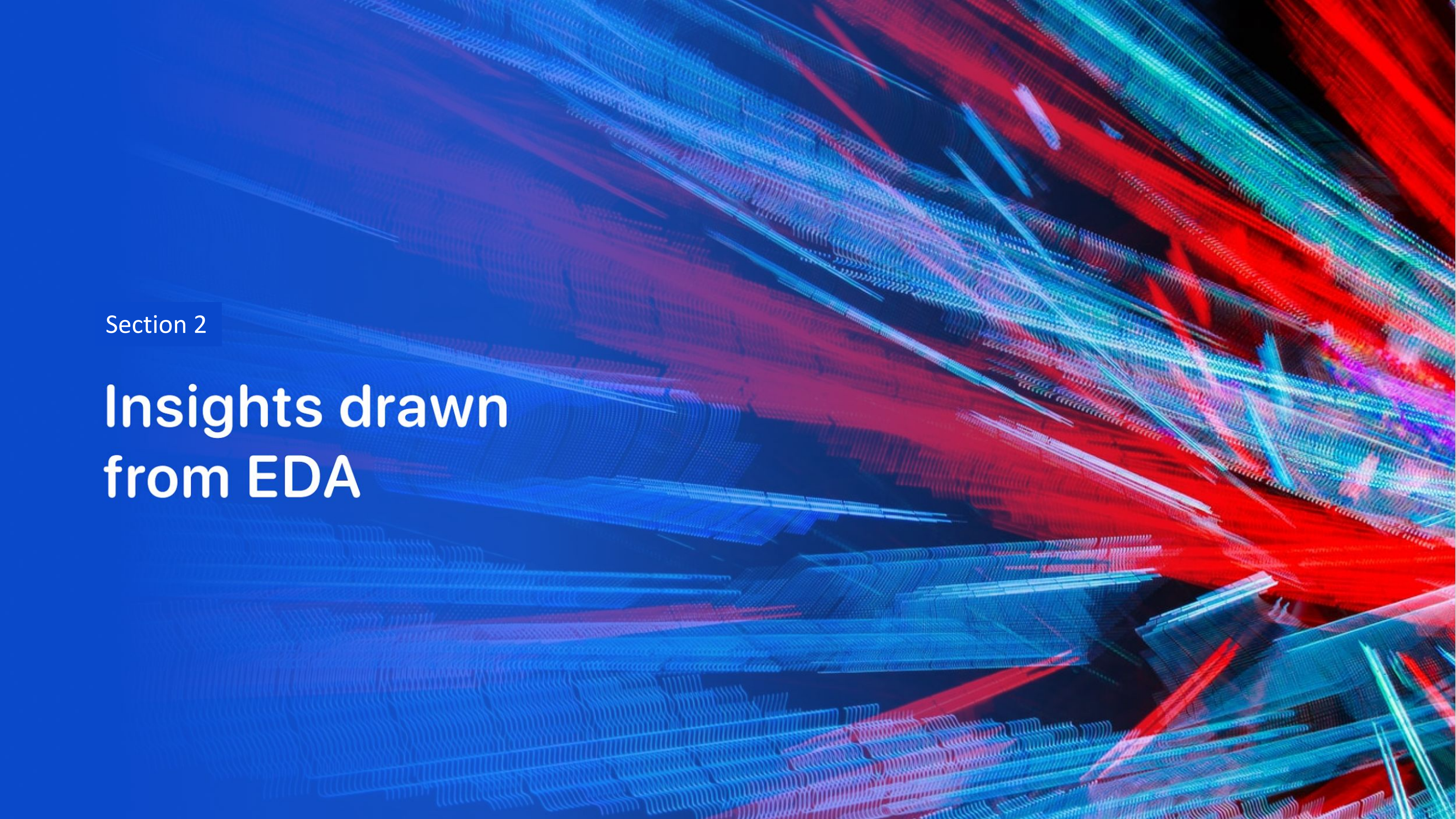
Use confusion matrix plot to see how well each model can distinguish between different classes (true positives, true negatives, false positives, false negatives)

Link to GitHub

Results

The results of the methodology will be presented in the next sections:

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



Section 2

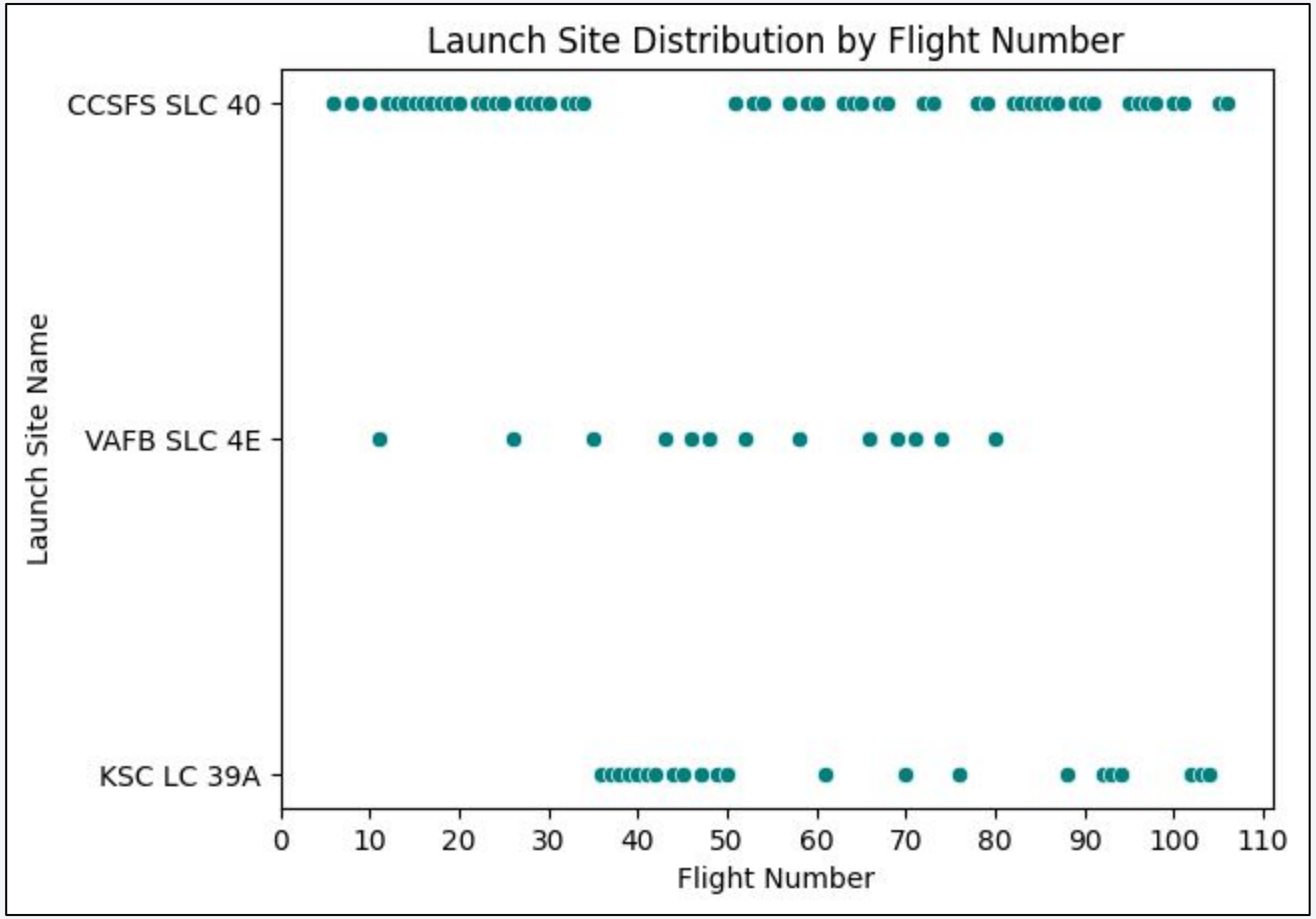
Insights drawn from EDA

Flight Number vs. Launch Site

Most of the launches has been released from CCSFS SLC 40 site even though there has been a small gap on launches from that site.

In the site KSC LC 39A the first launch was launch number 36. Number of launches grew rapidly after that.

First flight number in the dataset is 6 and not 1 explaining the gap in the plot.



CCSFS SLC 40: Vandenberg Air Force Base Space Launch Complex

VAFB SLC 4E: Cape Canaveral Space Launch Complex

KSC LC 39A: Kennedy Space Center Launch Complex

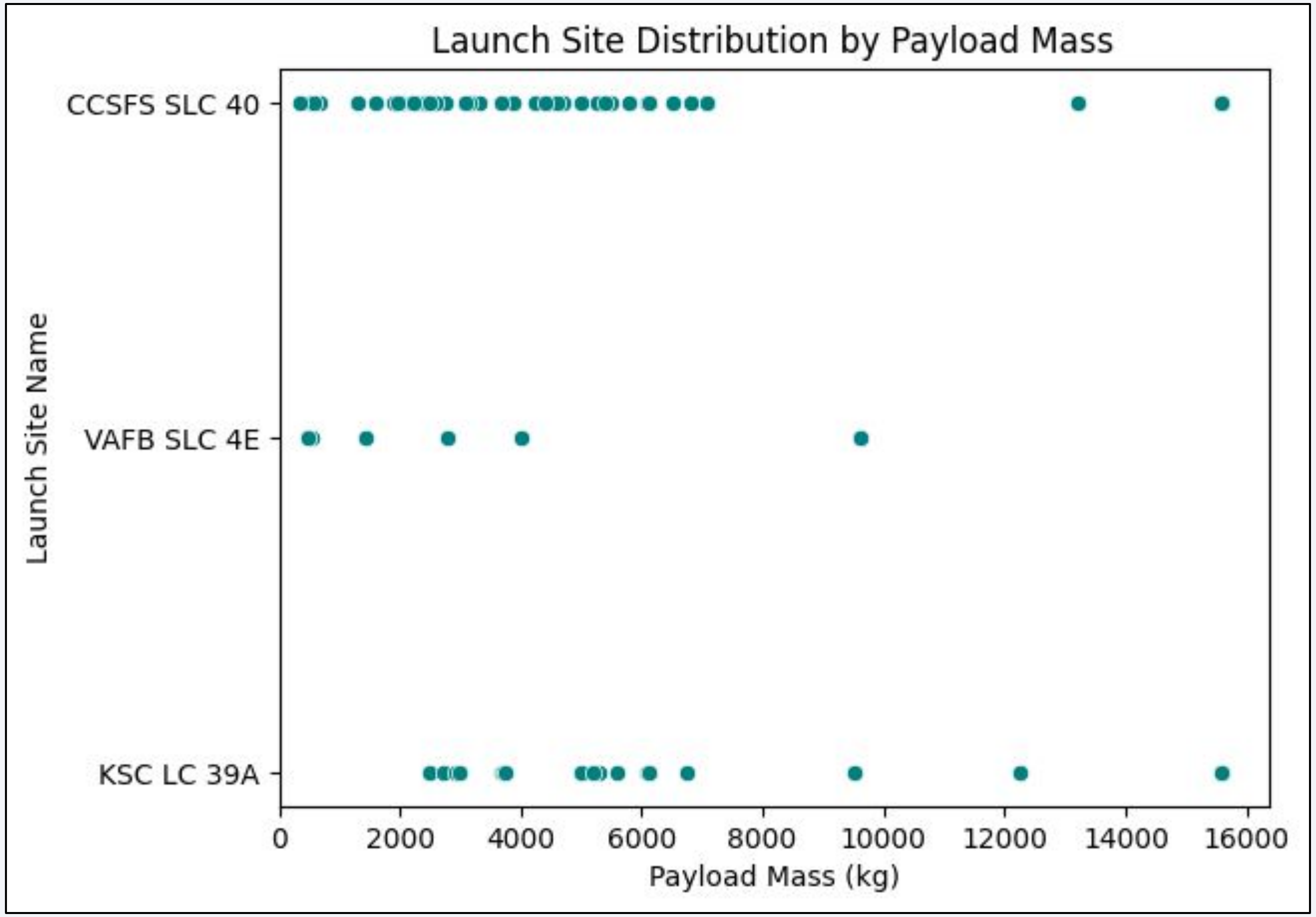
Payload vs. Launch Site

Payload masses from launch site KSC LC 39A have been on average heavier than from other two sites.

- Site KSC LC 39A average 7 644 kg
- Site VAFB SLC 4E average 5 919 kg
- Site CCSFS SLC 40 average 5 563 kg

From site VAFB SLC 4E there has been only 7 unique payload masses launched whereas from CCSF SLC 40 the variation of payload masses is highest (46 different payload masses).

Different launch sites appear to specialize in different payload mass categories.



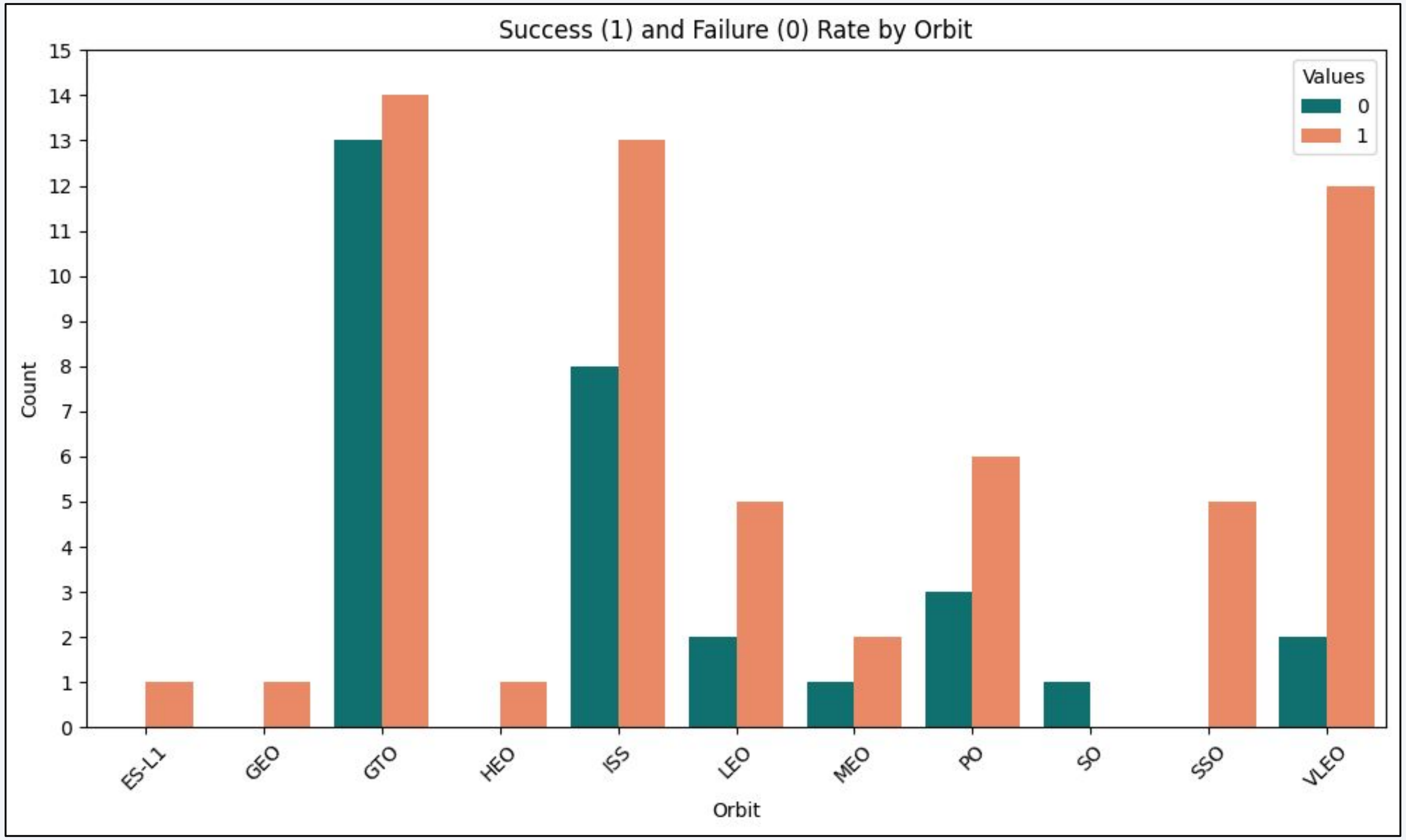
Success Rate vs. Orbit Type

Each Falcon 9 aims to an dedicated orbit.

For example **GTO** (geosynchronous orbit) is a high Earth orbit that is located 35 786 kilometers above Earth's equator. It is a valuable spot for monitoring weather, communications and surveillance.

Top 3 orbit destinations that Falcon 9 has been launched are orbits GTO, ISS and VLEO.

The best success rate so far has been achieved when destination has been orbits ES-L1, GEO, HEO or SSO. All of them has success rate 100 %. There has been only one launch to orbit SO which unfortunately failed.

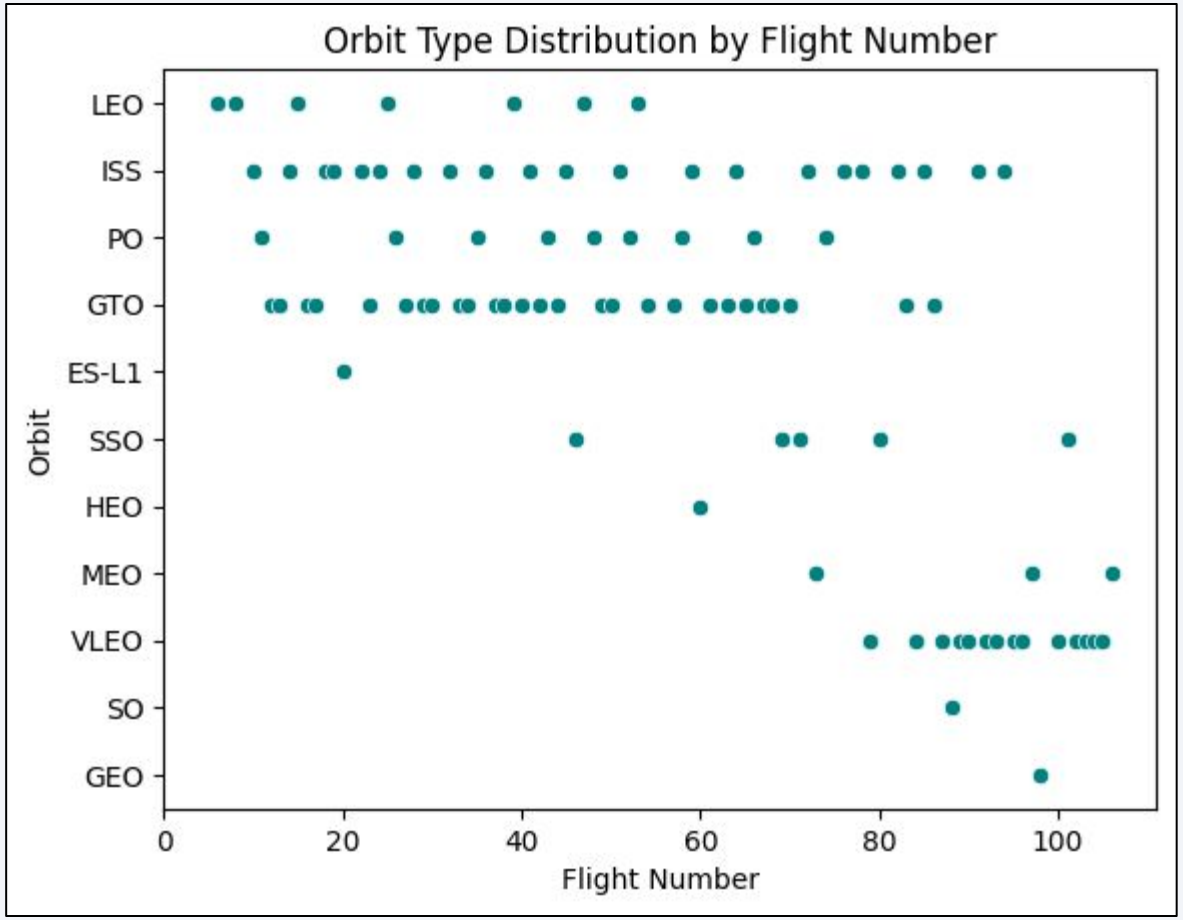


Flight Number vs. Orbit Type

Falcon 9 launches to orbits GTO and ISS have been done evenly, but for example orbit VLEO has been the destination only since flight number 79.

After that orbit VLEO has been the number one orbit destination although orbit MEO has been the latest orbit destination so far.

From the graph we can see that SpaceX has targeted different orbits thru times taking new orbit destinations to its repertoire gradually in time. Some of the orbits has also been dropped out from the roster suchs as orbit LEO.

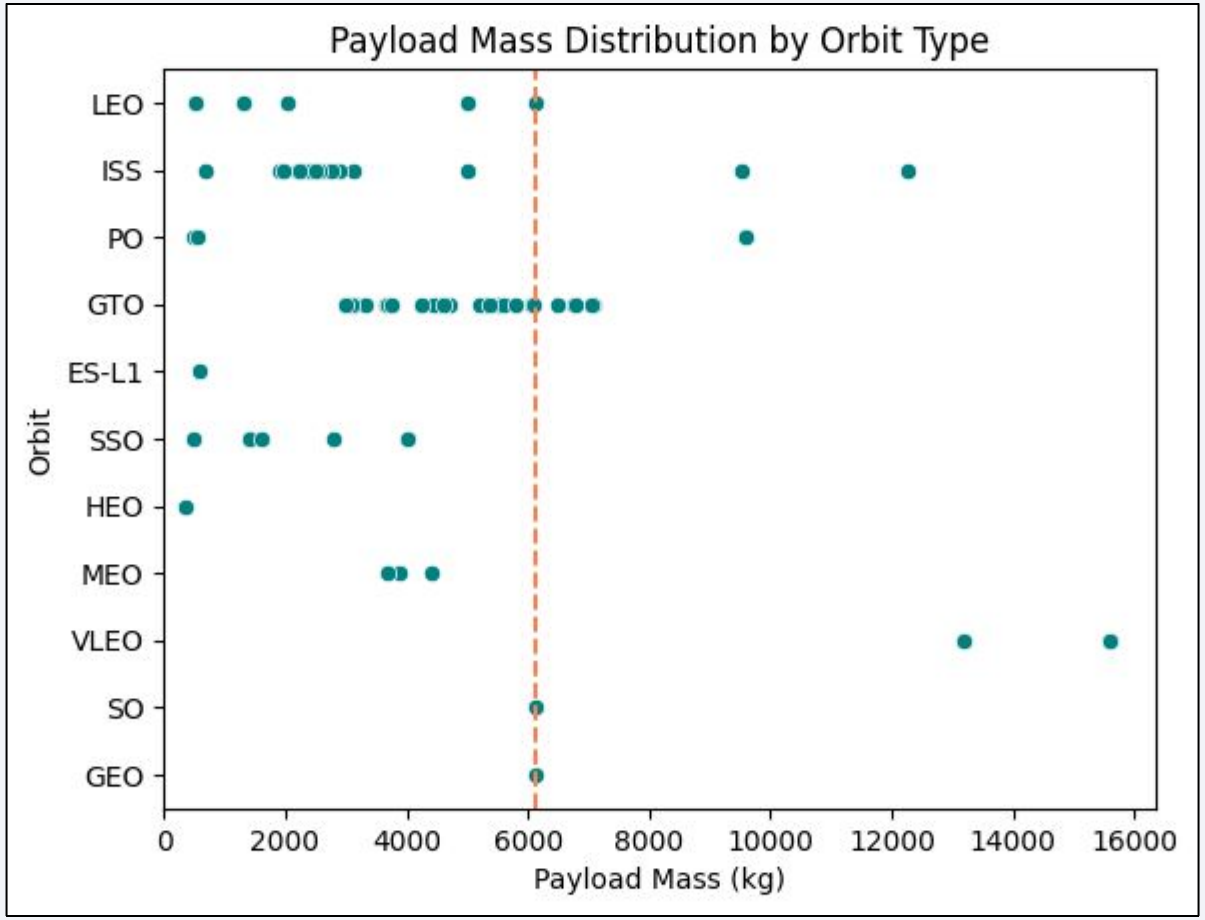


Payload vs. Orbit Type

The average payload mass of Falcon 9 has been 6 123 kilos that is presented in the graph as dotted line.

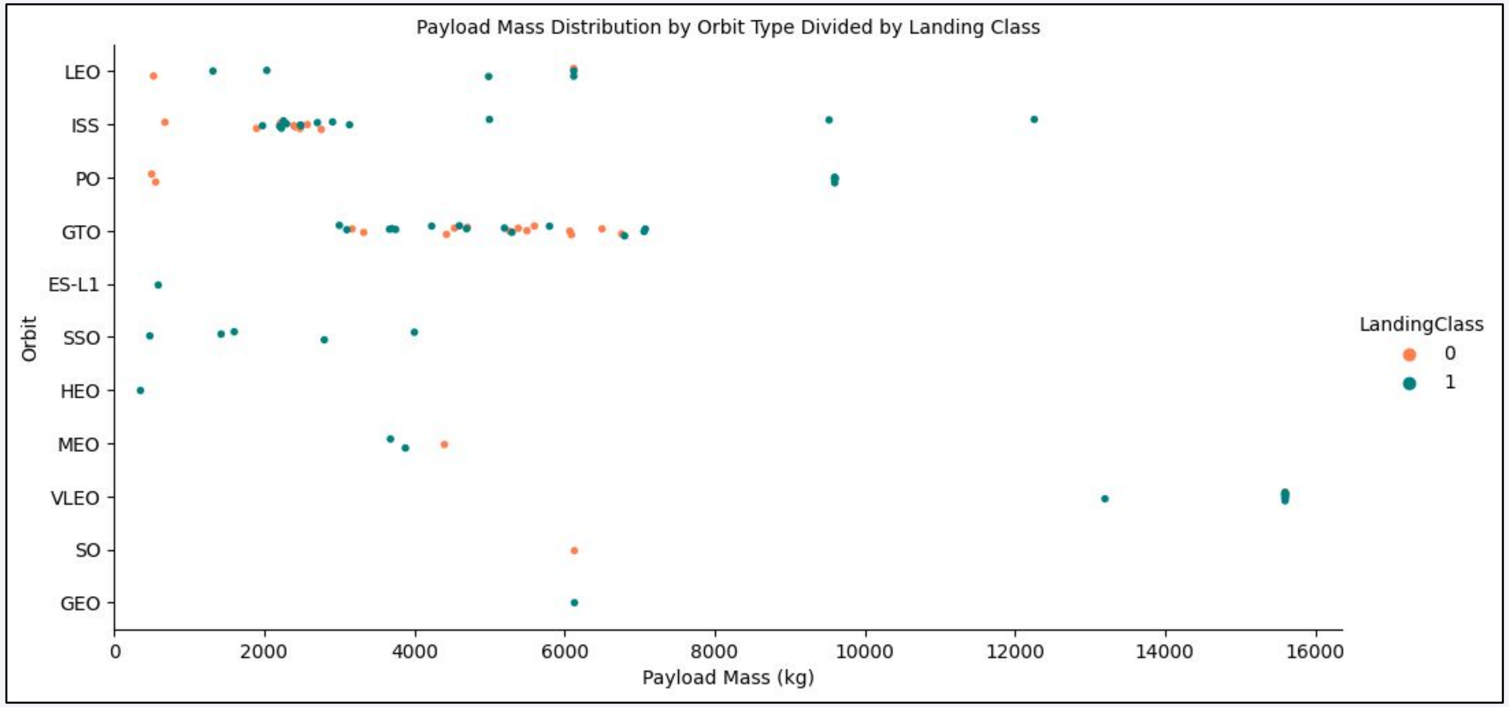
Orbits that has been the target of Falcon 9 launches with heavier than average payload mass are orbits ISS, PO, GTO and VLEO.

Even though scattered, there has been a tendency to launch rockets with lighter payload masses so far.



Payload vs. Orbit Type with Landing Class

When we include the landing class outcome to the plot, we see that the payload mass appears to greatly affect the success of launches in certain orbit types. For the LEO orbit type, heavier payloads positively affect the success rate of missions, while similar results are observed for the PO and ISS orbit types. For the GTO orbit type specifically, lighter payloads appears to have a marginally positive effect on the success rate, but that effect is not particularly strong due to small sample size for low payload weights. SSO and LEO launches seems to require lower payload masses, but are highly successful.

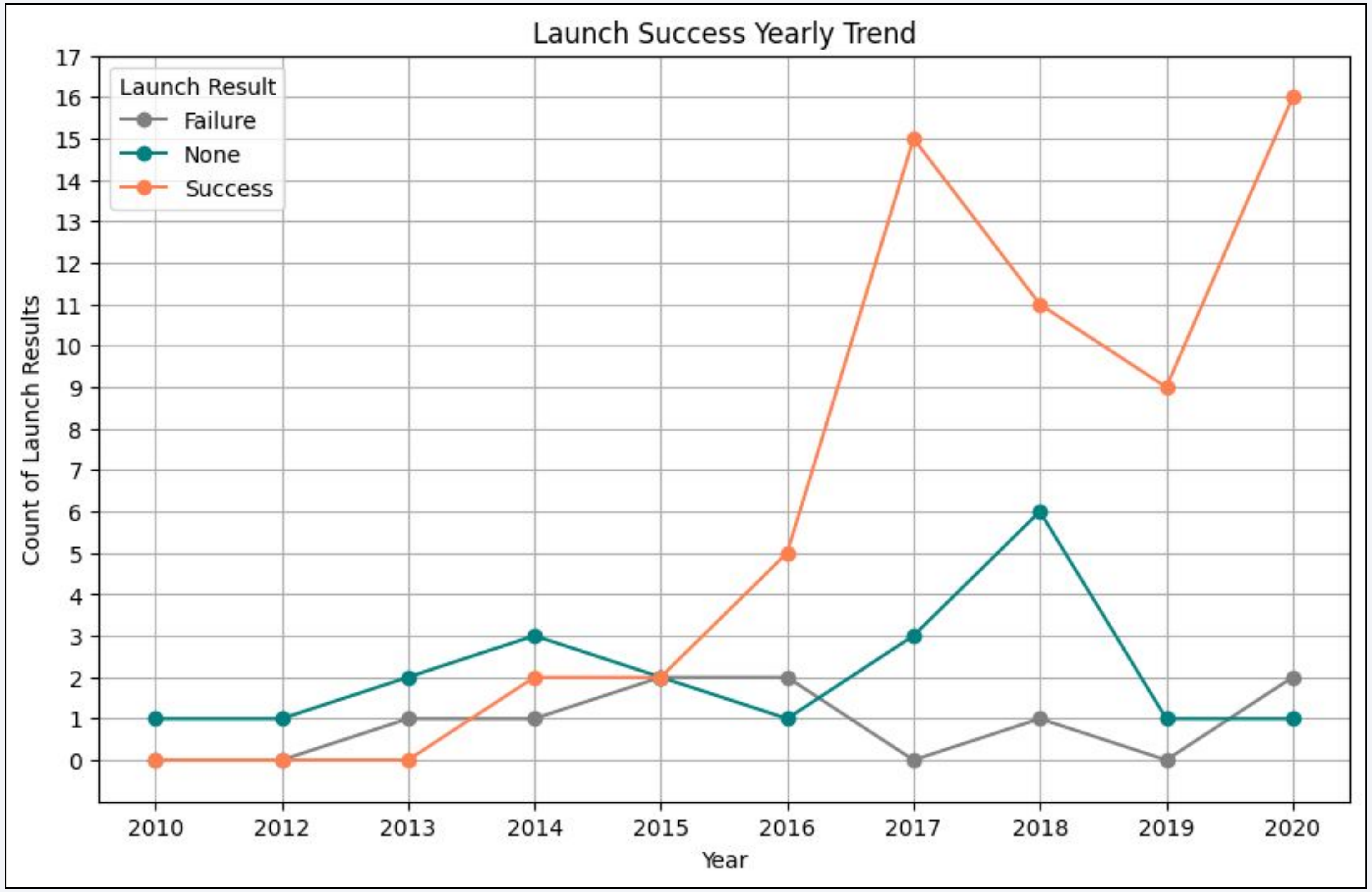


Launch Success Yearly Trend

General trend of Falcon 9 launches has been mostly upwards.

Falcon 9 missions can be divided to three different categories by mission outcome (Success, Failure and None). Falcon 9 have been targeted to land into either ocean, ground pad or drone ship.

If label of outcome is “Success” it means that a specific rocket managed to land to its target. “Failure” means the opposite. “None” types represent a failure to land also even though there is no record were rocket was supposed to land.



All Launch Site Names

SQL query

```
%sql select DISTINCT
LAUNCH_SITE as
"Unique_launch_sites"
from SPACE
```

Result

Falcon 9 has been launched from three different sites. One site has two different launch platforms near to each other.

- **Site 1A**: CCAFS LC-40 (Vandenberg Air Force Base Space Launch Complex)
- **Site 1B**: CCFS SLC-40 (Vandenberg Air Force Base Space Launch Complex)
- **Site 2**: KSC LC-39A (Kennedy Space Center Launch Complex)
- **Site 3**: VAFB SLC-4E (Cape Canaveral Space Launch Complex)

Launch Site Names Begin with 'CCA'

SQL query

```
%sql select LAUNCH_SITE from
SPACE where LAUNCH_SITE LIKE
'%CCA%' limit 5
```

Result

In the dataframe site names are reported as their abbreviations making it a bit hard to remember them. We could create a new column presenting site names with they full names (e.g Kennedy Space Center), but we can also use SQL Like operator to search specific pattern in a columns.

Here we have 5 records where launch sites begin with the string 'CCA'

Total Payload Mass Carried by Boosters Launched by NASA

SQL query

```
%sql select
SUM(PAYLOAD_MASS__KG_) as
"Total_mass_carried_by_NASA_l
aunch" from SPACE
where CUSTOMER = 'NASA (CRS)'
```

Result

SpaceX has had different customers in their Falcon 9 launches. NASA has been one of the customers.

Total payload carried by boosters launched by NASA has been **45 596 kilos** in total.

Average Payload Mass Carried by Booster Version F9 v1.1

SQL query

```
%sql SELECT
AVG(PAYLOAD_MASS__KG_) as
Average_mass_carried_by_v1_1
FROM SPACE
WHERE UPPER(BOOSTER_VERSION)
LIKE '%V1.1%'
```

Result

SpaceX has had different booster versions in Falcon 9 rockets.

Average payload mass carried by booster version F9 v1.1 **was 2 534 kilos**.

In the dataset booster versions were in upper and lower case letters making it impossible to find specific booster version just by its name. SQL Like operator was needed with SQL uppercase function that changed all values in needed column into uppercase letters.

First Successful Ground Landing Date

SQL query

```
%sql SELECT MIN(DATE)
"First_successfull_landing_to_ground" from SPACE where
LANDING_OUTCOME = 'Success (ground pad)'
```

Result

The first successful landing outcome into ground pad by Falcon 9 was achieved **December 22. 2015**

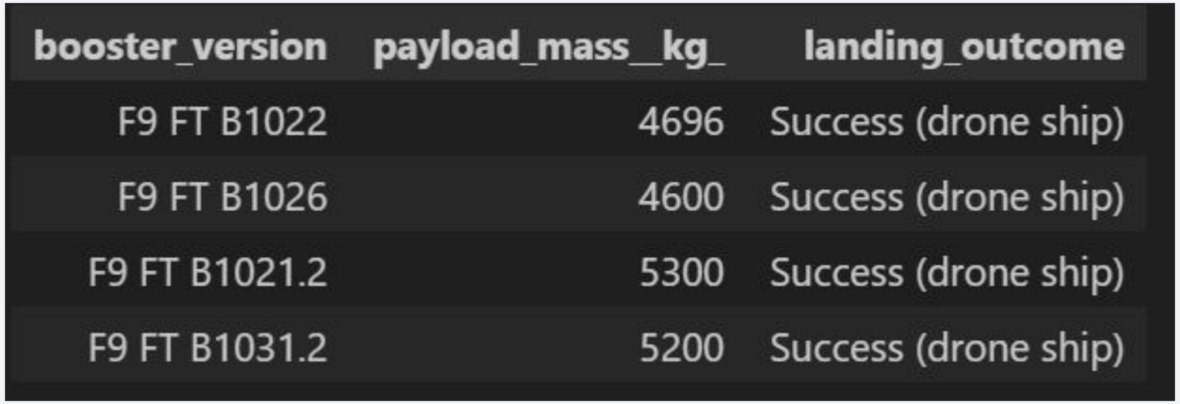
Successful Drone Ship Landings with Payload between 4000 kg and 6000 kg

SQL query

```
%sql select
BOOSTER_VERSION,PAYLOAD_MASS__KG_,
LANDING_OUTCOME from SPACE where
LANDING_OUTCOME = 'Success (drone
ship)' and PAYLOAD_MASS__KG_
BETWEEN 4000 AND 5999
```

Result

There was 4 cases fitting to this description:



booster_version	payload_mass__kg_	landing_outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

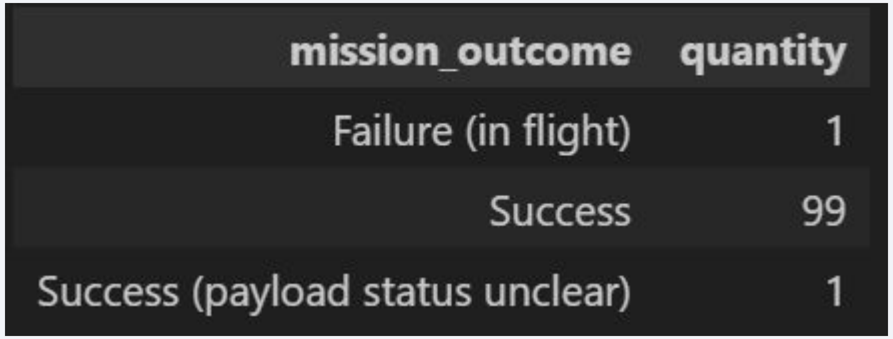
Total Number of Successful and Failure Mission Outcomes

SQL query

```
%sql select MISSION_OUTCOME,
COUNT(*) as "Quantity" FROM
SPACE grouped by
MISSION_OUTCOME
```

Result

Every Falcon 9 rocket is launched by a certain mission on mind weather or not its landing will succeed or not. From this perspective mission outcomes has been a great success so far.



mission_outcome	quantity
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Booster Versions that has Carried Maximum Payload Mass

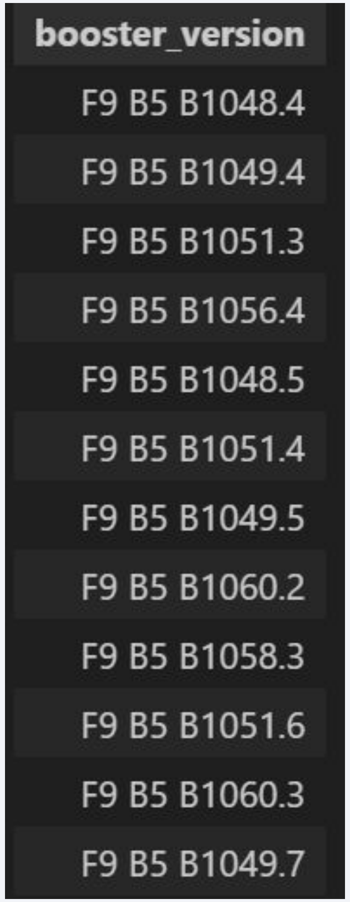
SQL query

```
%sql select BOOSTER_VERSION
from SPACE where
PAYLOAD_MASS__KG_ = (SELECT
MAX(PAYLOAD_MASS__KG_) from
SPACE)
```

Result

Maximum Payload Mass that any booster has carried so far is 15 600 kilos.

By using SQL subquery the 12 different boosters that had carried this amount were:



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

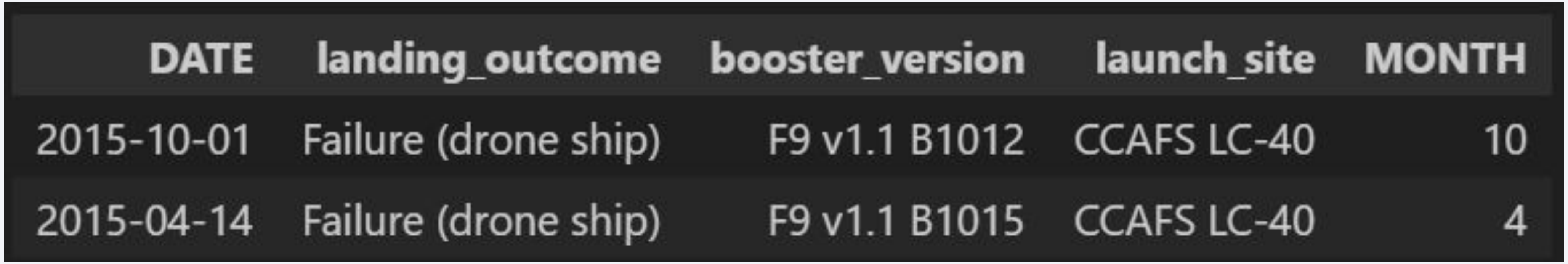
Specific Launch Records, an example 1

SQL query

```
%sql select DATE, LANDING_OUTCOME, BOOSTER_VERSION,
LAUNCH_SITE, extract(MONTH FROM DATE) as MONTH from
SPACE where LANDING_OUTCOME IN ( SELECT
LANDING_OUTCOME from SPACE where LANDING_OUTCOME =
'Failure (drone ship)') and extract(YEAR FROM DATE)
= 2015
```

Result

There has been a lot of Falcon 9 launches and we could be interested just few of them. For example we could like to know the dates of launches from one specific site that failed to land into drone ship using some specific booster version. Below the result of this kind of query:



DATE	landing_outcome	booster_version	launch_site	MONTH
2015-10-01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	10
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	4

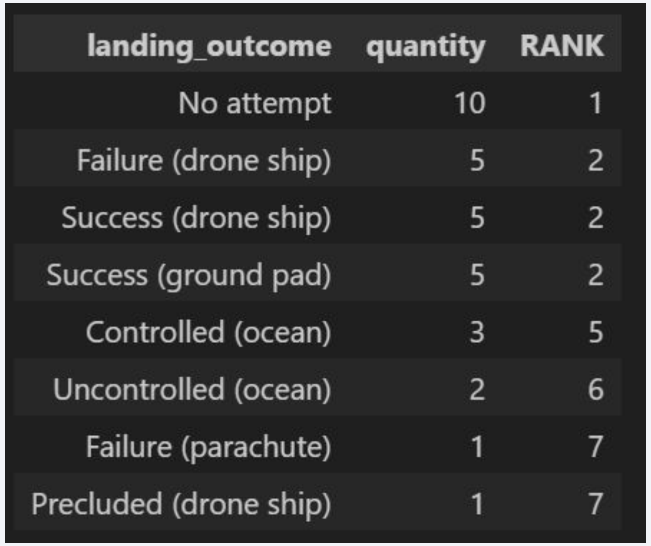
Specific Launch Records, an example 2

SQL query

```
%sql SELECT LANDING_OUTCOME, COUNT(*) as
"Quantity", RANK() OVER(ORDER BY COUNT(*)
DESC) as "Rank" FROM SPACE WHERE DATE BETWEEN
'2010-06-04' AND '2017-03-20' GROUP BY
LANDING_OUTCOME ORDER BY "Quantity" DESC
```

Result

We could also want count landing outcomes between some specific dates and other conditions. For example the count of landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order shows that most of the launches between this time interval were not even attempting to land.



landing_outcome	quantity	RANK
No attempt	10	1
Failure (drone ship)	5	2
Success (drone ship)	5	2
Success (ground pad)	5	2
Controlled (ocean)	3	5
Uncontrolled (ocean)	2	6
Failure (parachute)	1	7
Precluded (drone ship)	1	7

Section 3

Launch Sites Proximities Analysis

Launch Site Locations

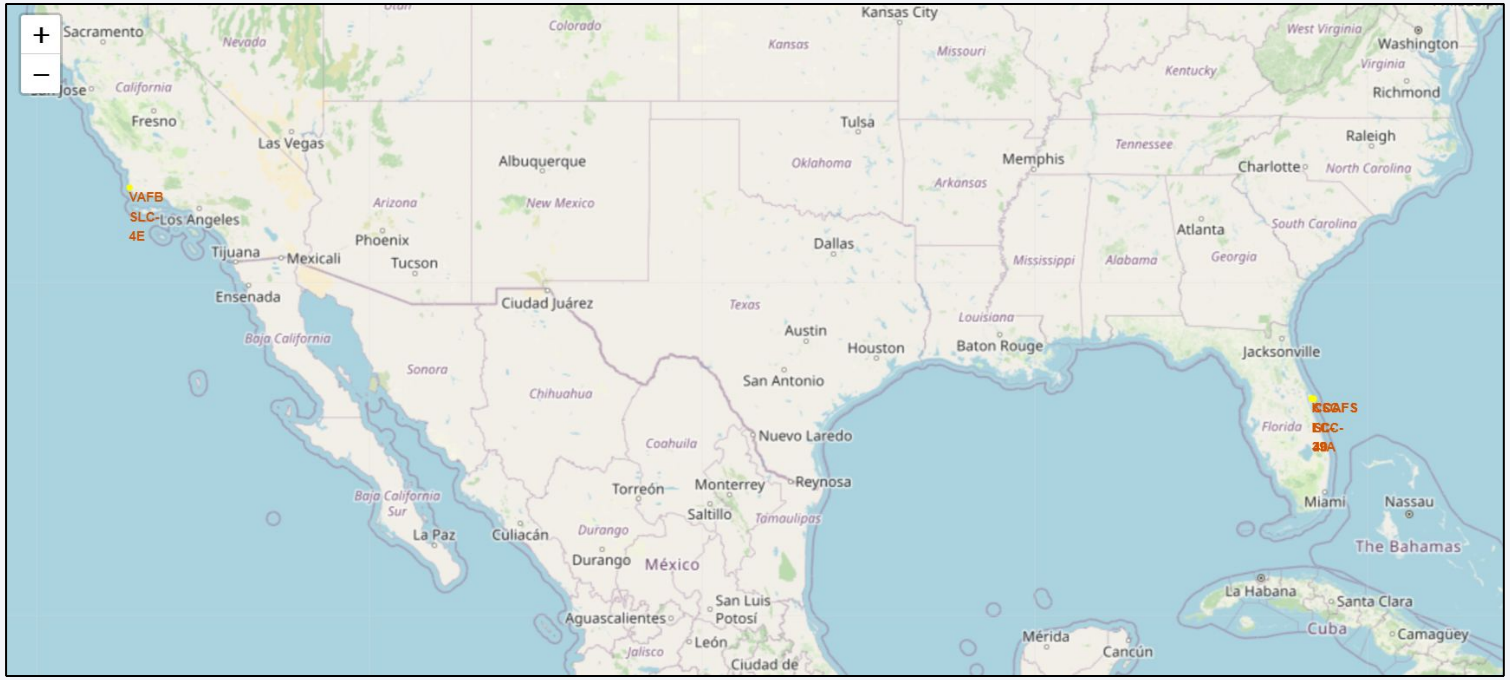
Interactive Visual Analytics enables users to direct data exploration and analytics and compared to static graphs it tells the users more compelling story.

To obtain more insights from gathered data about SpaceX Falcon 9 previous launches, two visual analytic tools were created:

- **Launch Site Geographical Data Map with Folium** and
- Data Dashboard with Plotly Dash

User of the geographical map can easily get an idea were Falcon 9 launch sites are located.

In the picture on right we can see that they are located to West and East Coast of United States.

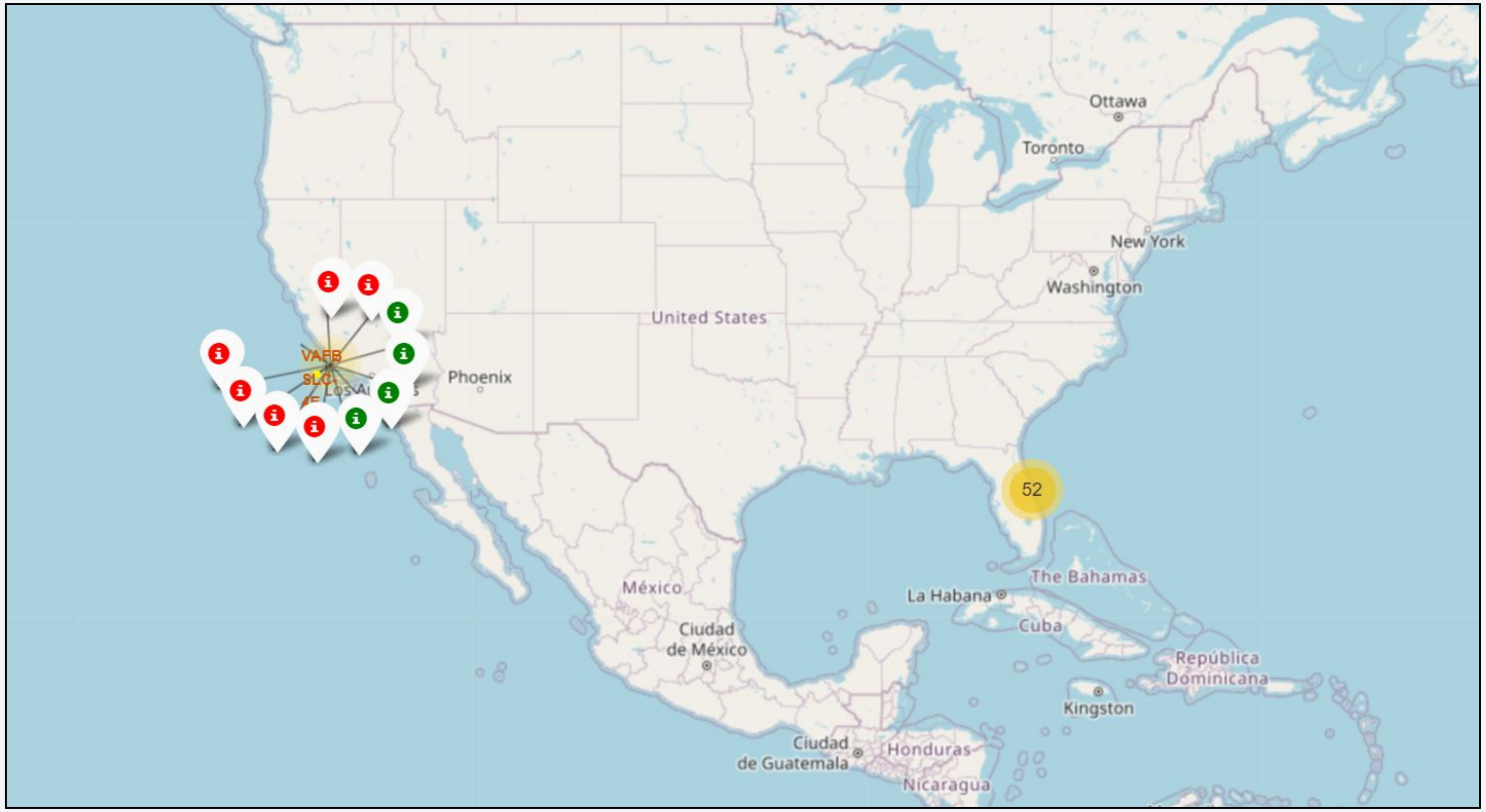


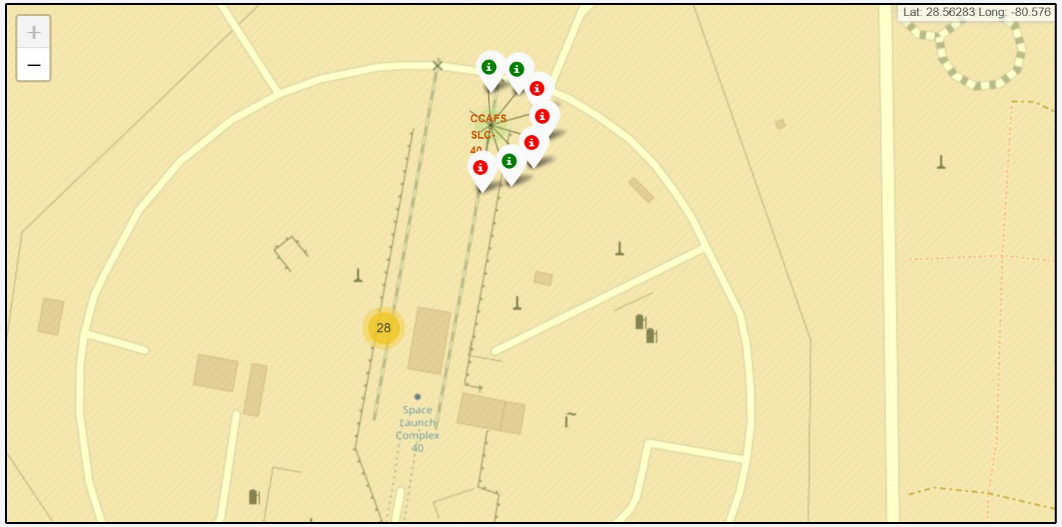
Color labeled launch outcomes

With color codes user of the geographical map can easily spot successful (colored as green) and non-successful (colored as red) launch outcomes from each launch site.

First picture above: VAFB SLC 4E sites outcomes (located in West Coast, Cape Canaveral Space Launch Complex).

Second picture below: CCAFS SLC 40 sites outcomes (located in East Coast, Vandenberg Air Force Base Space Launch Complex).



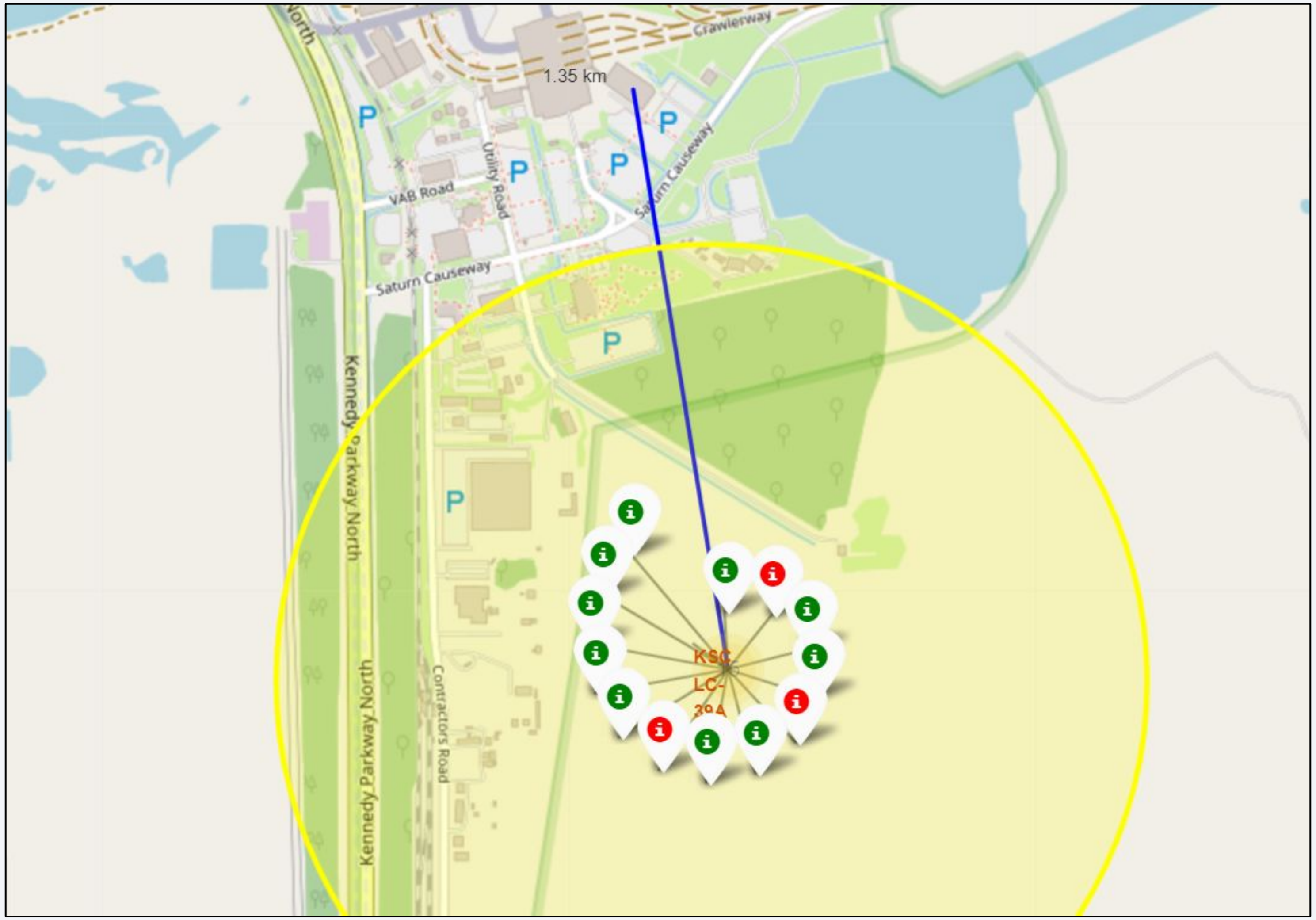


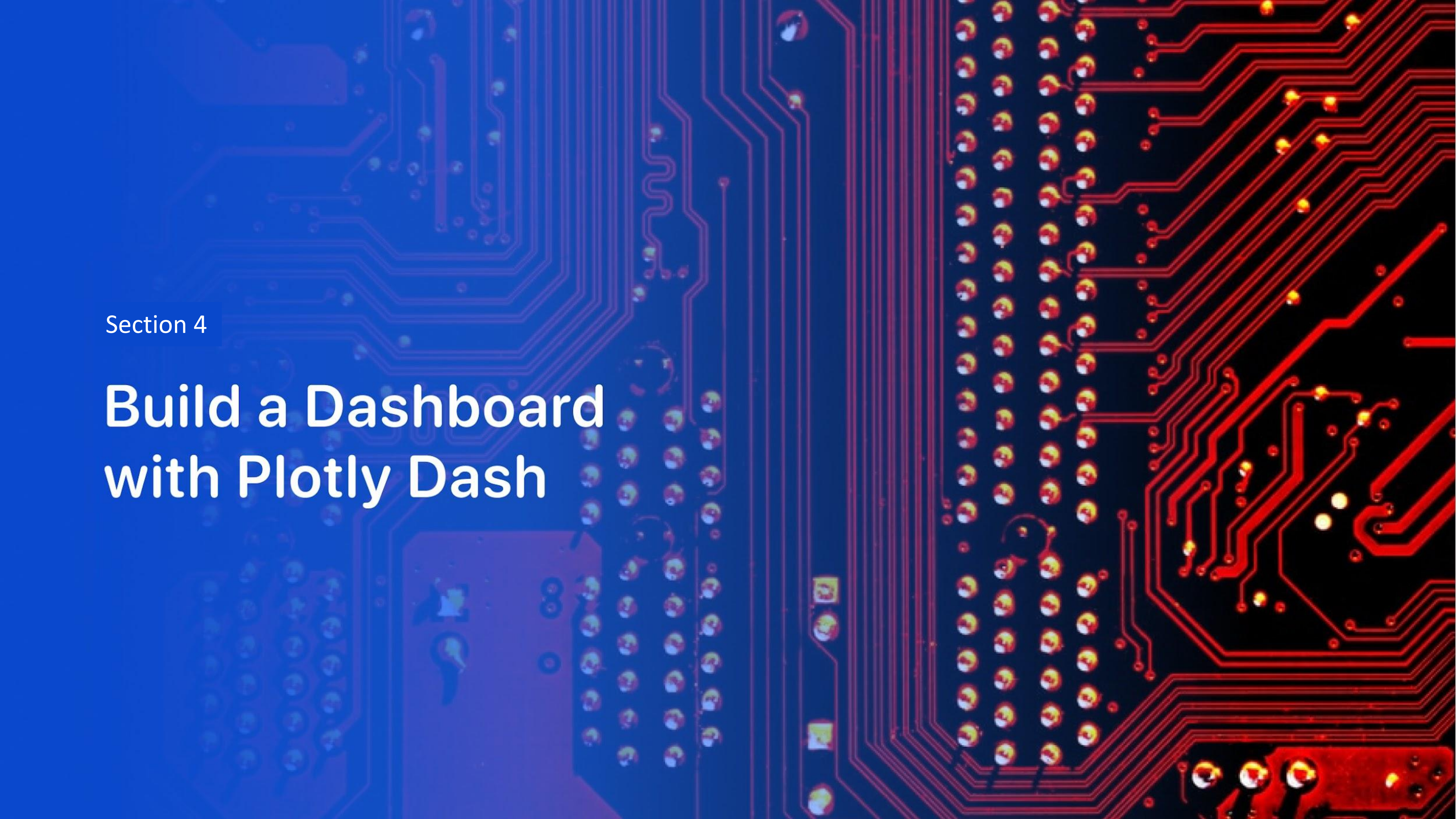
Launch site to its proximities

User of the geographical map can also check the distance between a specific launch site and subject of interest in the terrain.

Example, the picture on right:

We can see that the distance from launch site KCS LC 39A (Kennedy Space Center Launch Complex) to Launch Control Center is **1.35** kilometers.





Section 4

Build a Dashboard with Plotly Dash

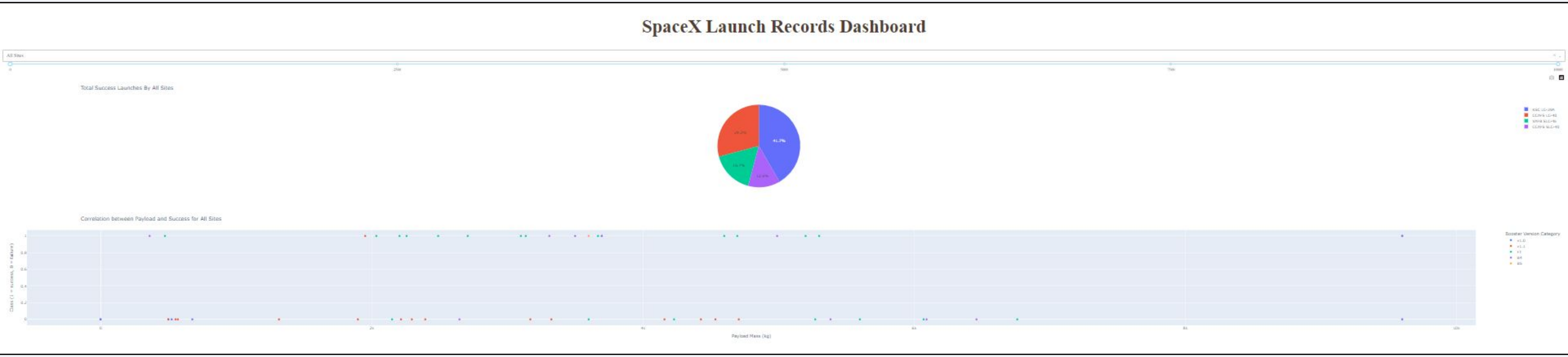
SpaceX Launch Records Dashboard

Interactive Visual Analytics enables users to direct data exploration and analytics and compared to static graphs it tells the users more compelling story.

To obtain more insights from gathered data about SpaceX Falcon 9 previous launches, two visual analytic tools were created:

- Launch Site Geographical Data Map with Folium and
- **Data Dashboard with Plotly Dash**

User of the Data Dashboard can easily choose and see the success rate of each site using Dropdown menu. With the RangeSlider user can choose a range of Falcon 9 payload mass and then see all the booster versions and if the landing was successful or not in this weight range. Below is an overview picture of the Dashboard.

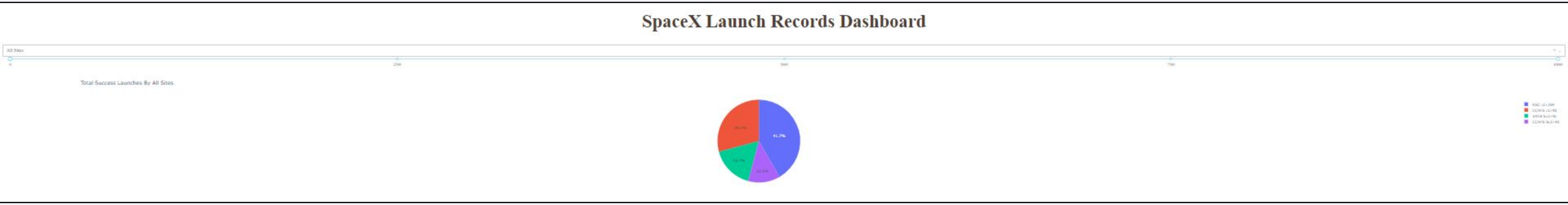


SpaceX Launch Records Dashboard

Launch success count for all sites

From the created Dashboard we can select to see for example how successful Falcon 9 launches has been per launch site.

In the picture below we can see success count for all sites in a form of a pie chart. KCL LC-39A site has biggest success rate: **41. 7 percent**.



SpaceX Launch Records Dashboard

All Sites

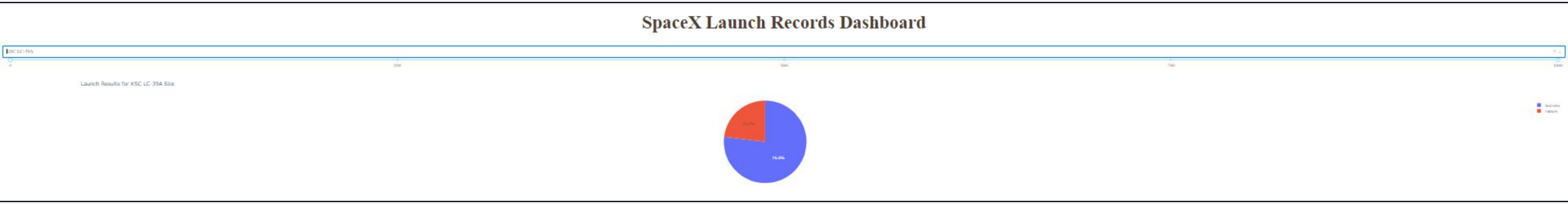
Total Success Launches By All Sites

Details from the launch site that has highest launch success ratio

In the Dashboard we can drill into more details of KCL LC-39A site.

Form the picture below we can see that this site has

- 76.9 percent of **success** and
- 23.1 percent of **failures**



SpaceX Launch Records Dashboard

KSC LC-39A

Launch Results for KSC LC 39A Site

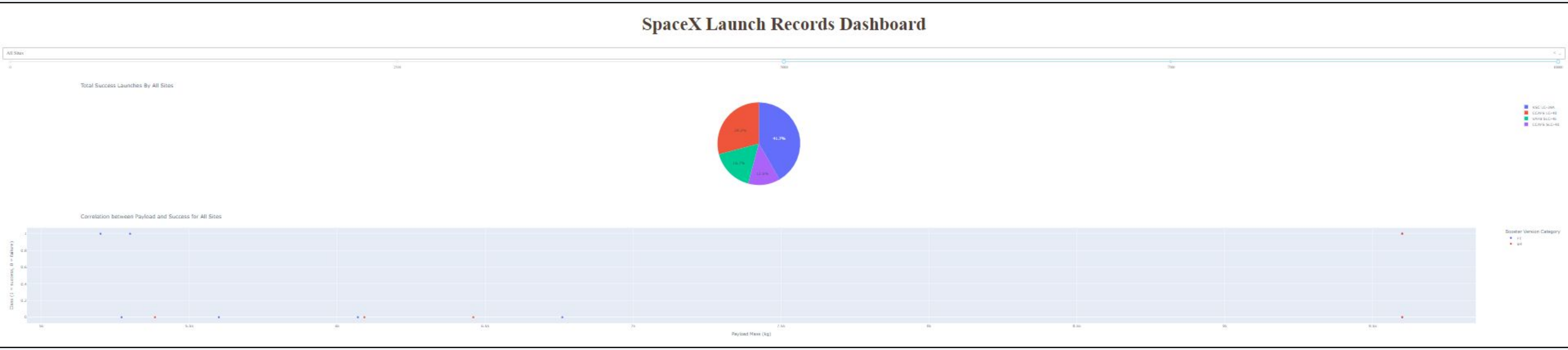
Payload Mass and Boosters Versions correlation with Success

When we use Dashboards another functionality - correlation between payload mass and success per booster version category - we can learn more about the previous Falcon 9 launches.

For example in the picture below we can see that when payload mass is set to between 5 000 - 10 000 kilos (all sites included), there has been only two kind of booster versions used with this payload mass:

- B4 and
- FT

Most of the launches with this combination (payload mass between 5 000 - 10 000 kilos with a booster version B4 or FT) has failed. From the three successful cases, two used booster version FT and one used booster version B4.



SpaceX Launch Records Dashboard

All Sites

Total Success Launches By All Sites

Correlation between Payload and Success for All Sites

Section 5

Predictive Analysis (Classification)

Predictive Analysis (Classification)

For making predictions about if future SpaceX Falcon 9 launches will be successful or not, four different machine learning models were trained and evaluated:

1. Logistic Regression,
2. Support Vector Machine,
3. Decision Tree and
4. K-Nearest Neighbour

I obtained their respective accuracy scores and confusion matrix results on both validation and test data. In the next slides I present the summary of the results.

Classification Accuracy

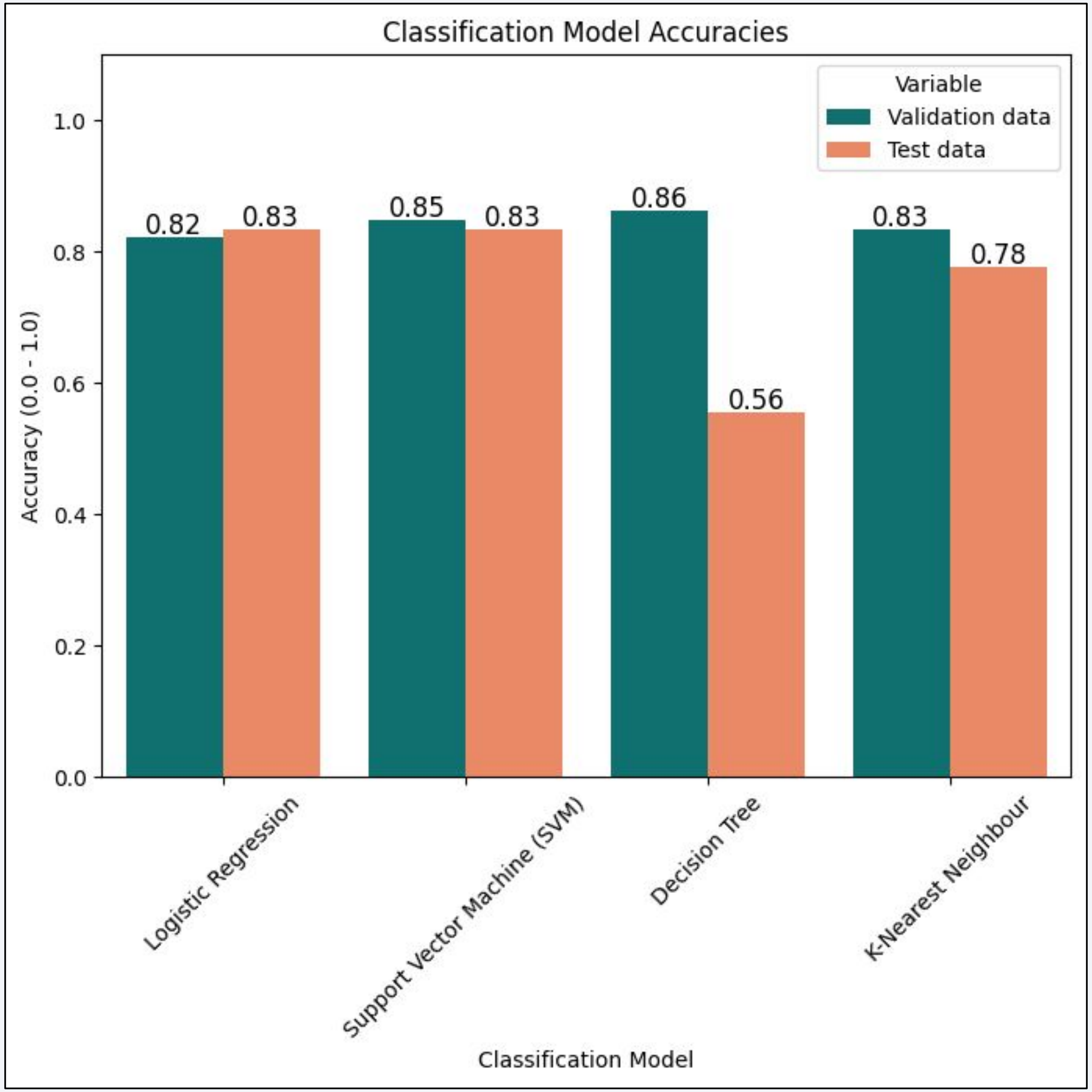
Validation data accuracy: Gives an estimate of how well mode generalizes to new unseen data

Test data accuracy: Provides a final measure of how well model is expected to perform in real-world scenario

Decision Tree Classifier achieved highest accuracy on the validation data indicating that it performed well on the data it was exposed to during hyperparameter training and model selection.

However Decision Trees accuracy on test data is lowest, which suggests that it may not generalize as well to new unseen data. Difference between the two accuracies indicates that the model has overfit the validation data.

The best balance between the two accuracies is achieved by the Logistic Regression and SVM models. Both models have similar validation and test data accuracies indicating that they generalize well to unseen data without significant drop in performance.



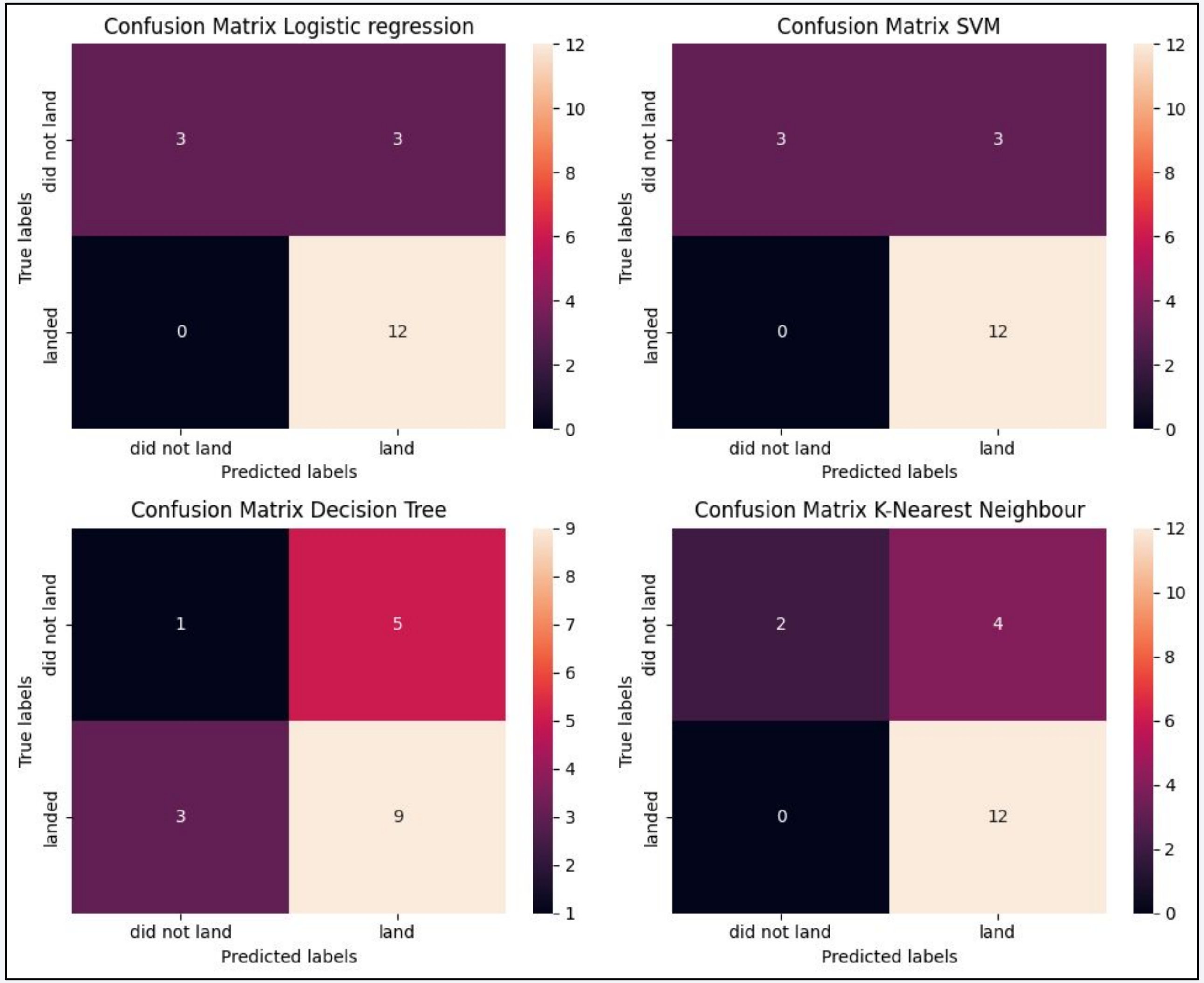
Confusion Matrix

When we split our data to train and test sets, size of our test set was 18 cases.

All models except Decision Tree has same amount of true positives (12 cases) meaning that they all predict positive class (Falcon 9 landed) in 12 cases from 18 cases equally well. These three model has also same number of false positives (0 cases) meaning that they don't predict positive class when it actually is negative.

If our aim is to create a model that can predict correctly positive cases, from this standpoint the best performing model would be either Logistic Regression or SVM. K-Nearest Neighbour would be third best option.

So in overall we can say that **Logistic Regression** or **Support Vector Machine** are the models that could be taken to use of making predictions for future outcomes.



What next?

01 Let’s assume that Logistic Regression would be the preferred model to continue with

- Find new up-to-date Falcon 9 data
- Train the Logistic Regression model with this new data: retraining can help the model stay accurate

02 Load the trained Logistic Regression model to machine learning library

- For example Scikit-Learn allows to save and load models
- New data should be a matrix or DataFrame with the same structure as the data it was trained

03 Make predictions about Falcon 9 on the new data

- When the model is loaded to a machine learning library, it can be used there for making predictions

04 Evaluate model performance, monitor and do maintenance to it

- Evaluation metrics like accuracy, F1-score and ROC-AUC can be used to evaluate models performance more deeply
- If there is a significant drop in performance, consider retraining the model or adjust hyperparameters

05 Take actions

- Based on models predictions, take actions!

Conclusions, 1

- The success of a SpaceX launch is a multifactorial problem, which we attempted to solve using Exploratory Data Analysis, static and interactive Data Visualizations and Predictive modeling
- The data collected were public and freely available using the SpaceX REST API and Web Scraping the SpaceX Wikipedia page

Conclusions, 2

- Launch success rate has significantly increased between 2013-2020, with a dip in 2018
- As the overall success rate of launches has gone up over the years, we can assume that valuable knowledge has been obtained from previous missions, so that the identified problems encountered in past missions have presumably been fixed
- Therefore the larger the amount of total launches carried out the higher the probability of mission success in the future

Conclusions, 3

- The "best performing" launch site according to the current dataset is KSC-LC-39A (Kennedy Space Center Launch Complex) with a success rate of 76.92%
- The orbit types with the highest success rates are ES-L1(100%), GEO(100%), HEO (100%), SSO (100%) and VLEO (~85%)
- However, since sample sizes are extremely small, we can deduce that VLEO is the most successful orbit type with a sample size of 14, followed by SSO with 5 total launches

Conclusions, 5

- For different orbit types Payload Mass might play an important role in determining whether a launch will be successful or not
- Generally low and medium weight payloads perform better than high payloads
- However, payloads over 7000 kg positively affect the first stage landing outcome

Conclusions, 6

- Out of all the Predictive Models we have constructed, the Logistic Regression and Support Vector Machine showed the best accuracy score on both training and test set data
- With these two classifiers we could be able to predict whether SpaceX will have a successful future

Appendix

All the codes that are made for this presentation can be found from GitHub.

Link to the repository.

Thank you!