IBM Developer
SKILLS NETWORK

WINNING SPACE RACE WITH DATA SCIENCE

Joanna Grace S. Saladino
November 2021

OUTLINE

IBM Developer
SKILLS NETWORK

- EXECUTIVE SUMMARY
- INTRODUCTION
- METHODOLOGY
- RESULT
- CONCLUSION
- APPENDIX

IBM Developer
SKILLS NETWORK

EXECUTIVE
SUMMARY

EXECUTIVE SUMMARY

Recent breakthroughs on reusable launch system has unbolt several possibilities on space exploration especially on space flight commercialization. Launch vehicles are traditionally designed for single flight use making them expensive costing around 165 million for each flight. But by recovering and reusing the part of the vehicles that carries the entire rocket, its payload, and all the unused fuel, the cost drives down to 62 million dollars. Reusable launch system focuses on developing new set of technologies that may be recovered and reused many times and it has been proven to be possible and is significantly more affordable than traditional launch system.

This kind of system was first used, albeit they haven't mastered it yet, for the first stage of SpaceX's Falcon 9. Falcon 9 is a two-stage rocket where the first stage is responsible in helping the rocket escape the Earth's atmosphere and once beyond the Earth's atmosphere the pneumatic stage separation system releases the first stage from the second stage. A single Merlin engine then fires and propels the second stage into orbit. After that, the first stage re-enters the Earth's atmosphere, lands on a landing pad, and ready to be reflown again.

EXECUTIVE SUMMARY

IBM Developer SKILLS NETWORK

Apparently, the first stage does most of the work and is much larger than the second stage. And while there were numerous instances where the first stages landed successful there were also instances where it does not land. Other times, SpaceX will sacrifice the first stage due to the mission parameters like payload, orbit, and customer. The goal of this study is to determine the price of each launch by predicting the landing outcome of the first stage. The landing outcome not only determines the total cost of the current flight mission but also the next one since it affects the reusability of the first stage. Failed landing, as one can expect, will expedite more cost since the first stage needs to be rebuilt.

Mission specific factors (such as booster version, payload mass, launch site, landing pad, and dedicated orbit), vehicle specific factors (such as grid fins, legs, reused count, and serial number), and location based factors (such as latitude, longitude, proximities) were analyzed and used as features or explanatory variables. SpaceX have 3 unique launch sites: Cape Canaveral Space Launch Complex 40 CCAFS SLC 40, Vandenberg Air Force Base Space Launch Complex 4E (VAFB SLC 4E), and Kennedy Space Center Launch Complex 39A KSC LC 39A. Overall, the launch success is trending upward with a slight decline on the year 2018 and 2020. A large majority of the successful launches, 41.7% to be specific, where launched from KSC LC 39A. 29.2% where launched in CCAFS LC 40, 16.7% where launched in VAFB SLC 4E and 12.5% where launched in CCAFS SLC 4.

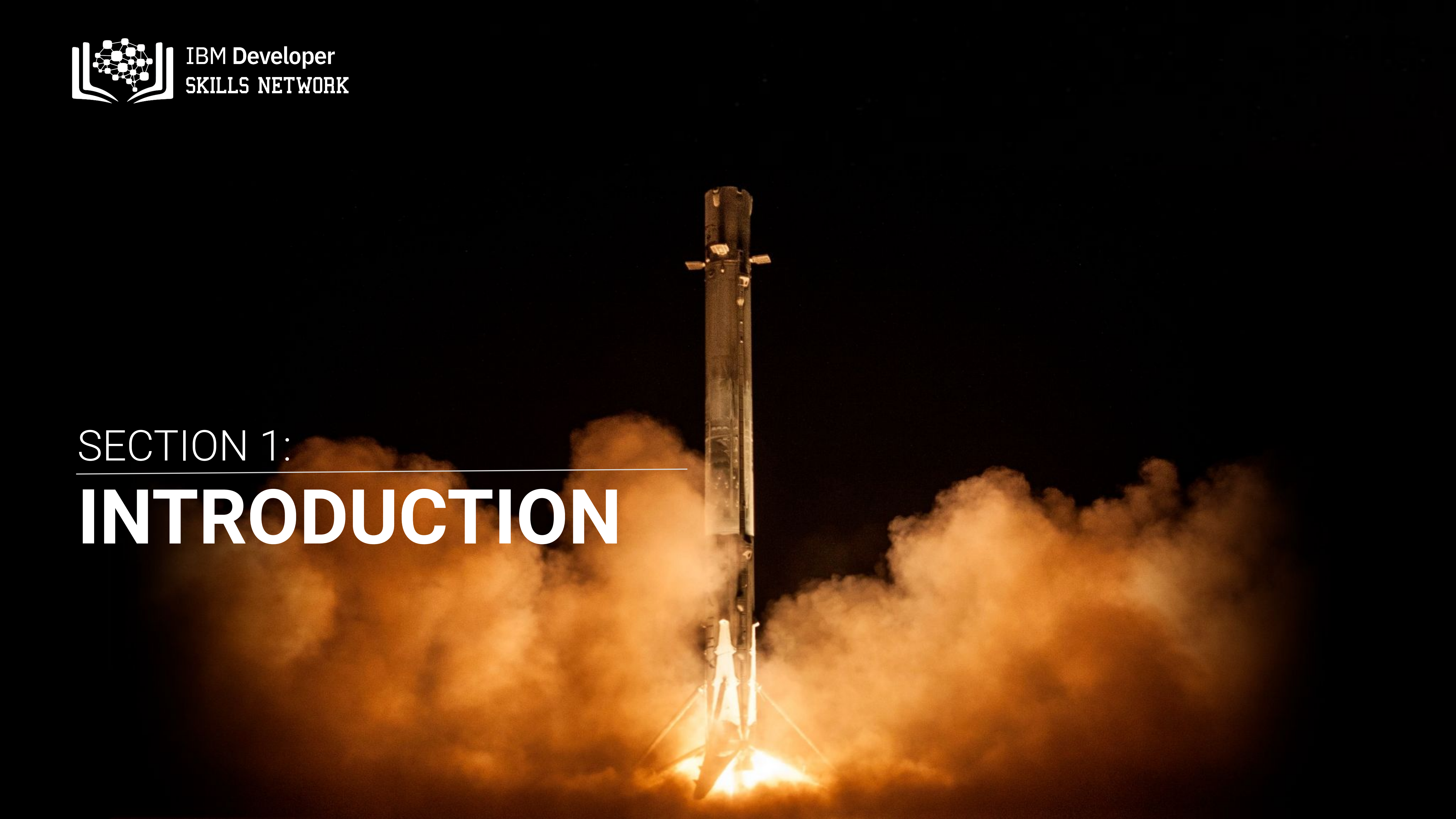
EXECUTIVE SUMMARY

IBM Developer SKILLS NETWORK

With heavy payloads (payload mass greater than 10000), a positive landing rate is expected for PO, ISS, and LEO orbit. However, there seems to be no apparent relationship between payload mass and success rate for GTO. In addition to that, none of the heavy payload missions were launched from VAFB SLC 4E.

Launching sites are usually close to coastline and are of close proximity to the equator line. Apparently, launch site are usually located on coastline so that, in the event, where the Falcon 9 failed to launch or land there'll be less damage on the hinterland. Moreover, launching near the equator line helps the rocket to get an initial boost equal to the velocity of Earth surface. This "boost" gradually reduces as we move to the pole which explains why locations near the equator line is the optimal location for rocket launches. They are also of close proximity to railways and highways but far from city proper. Trains/railroads are often used to move rocket boosters and highways are used for seamless transportation around the site complex which is why it is necessary for launch sites to be near this kind of platforms.

FInally, we used four machine learning classifiers as our base models - logistic regression, support vector machines, decision trees, and k-nearest neighbor. Decision trees obtained an 88% accuracy on the test set and thereby outperformed the other three models which each have an 83% accuracy.



SECTION 1:

INTRODUCTION

Space exploration attracts interest of an increasing number of governments, private sector; start-ups and large enterprises to exploit the commercial potential of exploration activities. In May of 2020, SpaceX crossed an important threshold in space exploration as it is the first private company to send human into space.

This, along with the fact that SpaceX utilizes reusable launch system where they aim to develop a set of new technologies that may be reused many times marks not only a tremendous technological achievement, but also the first indication that the commercialization of space travel could be close at hand due to the relatively lower cost of each flight.

The objective of SpaceX’s reusable launch system include returning a launch vehicle first stage to the launch site in minutes and to return a second stage to the launch pad following orbital realignment with the launch site and atmospheric reentry in up to 24 hours.

SpaceX's long term goal is that both stages of their orbital launch vehicle will be designed to allow reuse a few hours after return. This system was used for the first stage of Falcon 9.

FALCON 9

IBM Developer
SKILLS NETWORK

SpaceX's Falcon 9 is the first orbital class rocket capable of reuse. It’s a two-stage rocket which is mainly used for the reliable and safe transport of people and payloads into the Earth’s orbit.

Beyond the simple launch, Falcon 9 is capable of recovering and reflying the first stage which in turn drives down the cost of space access.

FIRST STAGE

The first stage carries the entire rocket, its payload, and all the unused fuel. It is the biggest, most powerful, and most expensive section of the rocket.

It incorporates nine Merlin engines and aluminum-lithium alloy tanks containing liquid oxygen and rocket-grade kerosene (RP-1) propellant, generating more than 1.7 million pounds of thrust.

The Merlin engine is responsible in helping the rocket escape the Earth’s atmosphere. Once beyond Earth’s atmosphere, the pneumatic stage separation system releases the first stage from the second stage. A single Merlin engine fires, propelling stage two into orbit.

SECOND STAGE

The second stage, powered by a single Merlin Vacuum Engine, delivers Falcon 9’s payload to the predetermined orbit.

The second stage has a lot less to transport, and it doesn't have to fight through the thick lower atmosphere, so it usually has just one engine.

GOAL OF THE STUDY

The first stage does most of the work and is much larger than the second stage. This stage is quite large and expensive. Unlike other rocket providers, SpaceX's Falcon 9 can recover the first stage. Sometimes the first stage does not land. Other times, SpaceX will sacrifice the first stage due to the mission parameters like payload, orbit, and customer.

The goal of this study is to determine the price of each launch by predicting the landing outcome of the first stage. The landing outcome not only determines the total cost of the current flight mission but also the next one since it affects the reusability of the first stage. Failed landing, as one can expect, will expedite more cost since the first stage needs to be rebuilt.

We start by gathering information about the previous launches of SpaceX especially the Falcon series. The raw data are then cleaned, processed. and transformed into a format that is amenable for supervised machine learning algorithm. Going back to the problem, we want to determine if Falcon 9 will land successfully or not, which can be predicted using binary classification models such as logistic regression, support vector machine, decision trees, and k nearest neighbor.

There are several factors that could influence the landing outcome such as payload mass, orbit type, booster type, etc. In addition to this, its initial location of the rocket and its proximities could also affect the rocket. So apart from analyzing how the different factors relates to each other and how they could influence the outcome of the first stage landing, we also studied the geographical profile of the different launch sites.

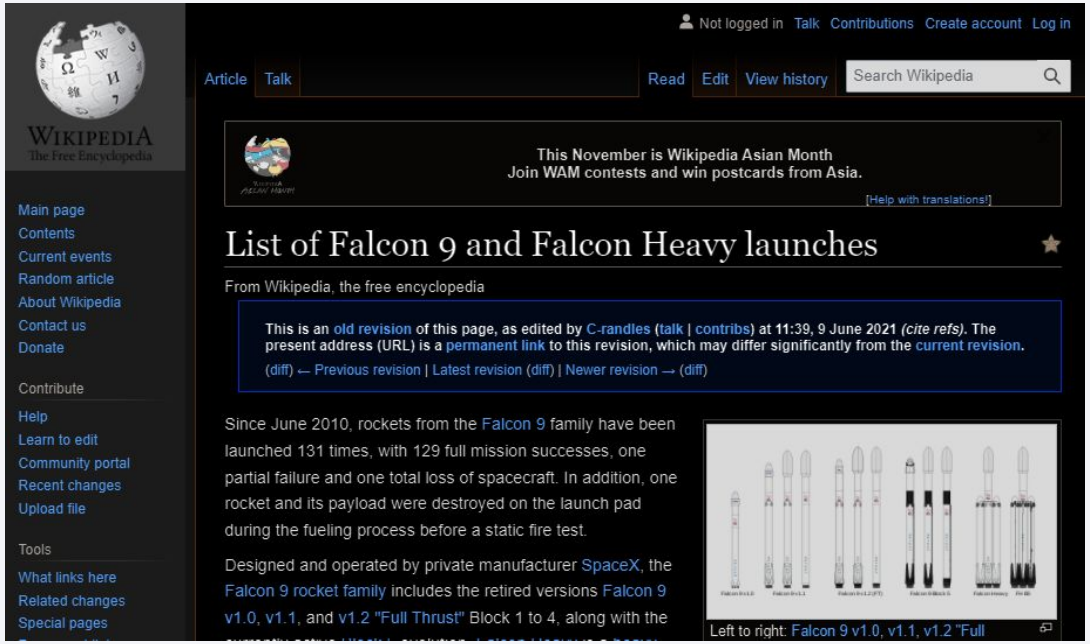
IBM Developer
SKILLS NETWORK

SECTION 2:

METHODOLOGY

DATA COLLECTION

Data were collected from SpaceX API and by webscrapping Falcon 9's historical launch records from wikipedia.



DATA COLLECTION: SpaceX API

1 **Request and parse launch data from SpaceX API using GET request**

```
# Request  and parse launch data from SpaceX API using GET request

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

2 **Use the API again to extract more relevant information**

```
# Use the API again to extract more relevant information

def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])

def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])

def getPayloadData(data):
    for load in data['payloads']:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])

def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

[Data Collection API]()

DATA COLLECTION: SpaceX API

3 **Format the dataframe to contain the following columns:**

FlightNumber', 'Date', 'BoosterVersion', 'PayloadMass', 'Orbit', 'LaunchSite', 'Outcome', 'Flights', 'GridFins', 'Reused', 'Legs', 'LandingPad', 'Block', 'ReusedCount', 'Serial', 'Longitude', 'Latitude'

4 **Filter the dataframe to only include Falcon 9 launches**

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	7	B1060	-80.603956	28.608058
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	7	B1058	-80.603956	28.608058
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	9	B1051	-80.603956	28.608058
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	7	B1060	-80.577366	28.561857
93	90	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	3	B1062	-80.577366	28.561857

[Data Collection API]

DATA COLLECTION: Webscrapping

1. Request the Falcon 9 Launch Wiki page from its URL using BeautifulSoup

```python
import requests
from bs4 import BeautifulSoup
import re
import unicodedata
import pandas as pd

static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"

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

response = requests.get(static_url)

# Use BeautifulSoup() to create a BeautifulSoup object from a response text content

soup = BeautifulSoup(response.content)
```

2. Extract all column/variable names from the HTML table header and create an empty dictionary using the columns as keys

```python
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 Collection: Webscrapping]

DATA COLLECTION: Webscrapping

3 **Fill up the empty launch_dict with launch records extracted from the table rows**

4 **Create a dataframe by concatenating the launch_dict**

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

[Data Collection: Webscrapping]

DATA WRANGLING

As mentioned on the outset, **our main goal is to predict if the Falcon 9 first stage will land successfully**. Clearly, this is a **classification problem** where the target variable have two classes (0 if the landing failed and 1 if it successful). This section focuses on transforming the raw data into a format that is amenable for classification problem.

1 - successful landing

0 -un successful landing

[Data Wrangling]

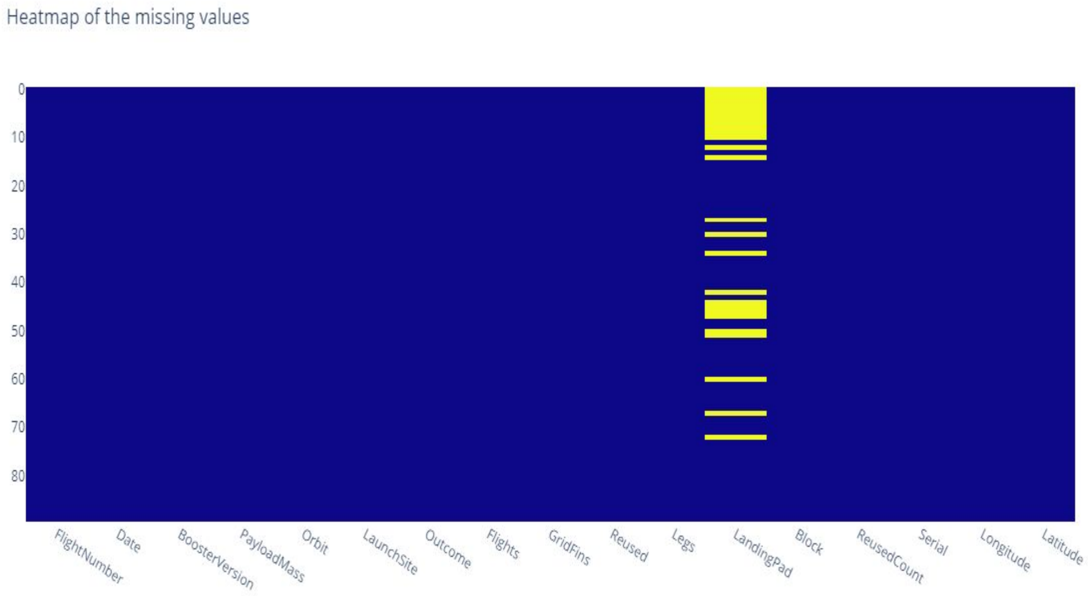
DATA WRANGLING

But before that … let’s do some simple data analysis to familiarize ourselves with the content of the data

Percentage of missing values

The figure visualizes the missing value occurrence using heatmap: yellow indicates the data is missing and blue if it isn’t.

Clearly, all features/columns have complete values except LandingPad which has 40.625% missing values.



[Data Wrangling](#)

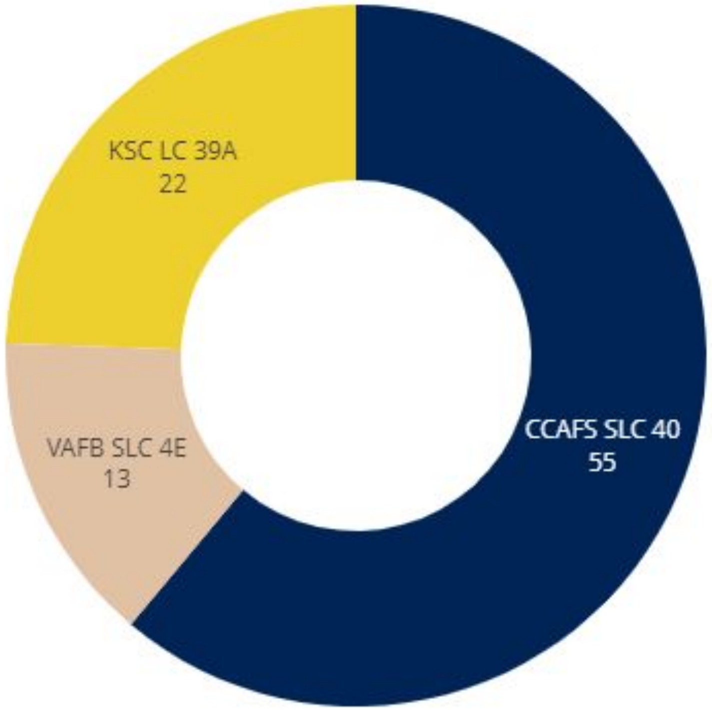
DATA WRANGLING

Number of launches per site

SpaceX have 3 unique launch sites namely:

- Cape Canaveral Space Launch Complex 40 CCAFS SLC 40,
- Vandenberg Air Force Base Space Launch Complex 4E (VAFB SLC 4E), and
- Kennedy Space Center Launch Complex 39A KSC LC 39A

55 Falcon 9 flights where launch on CCAFS SLC 40, 13 where launch on VAFB SLC 4E, and 12 where launch in KSC LC 39A



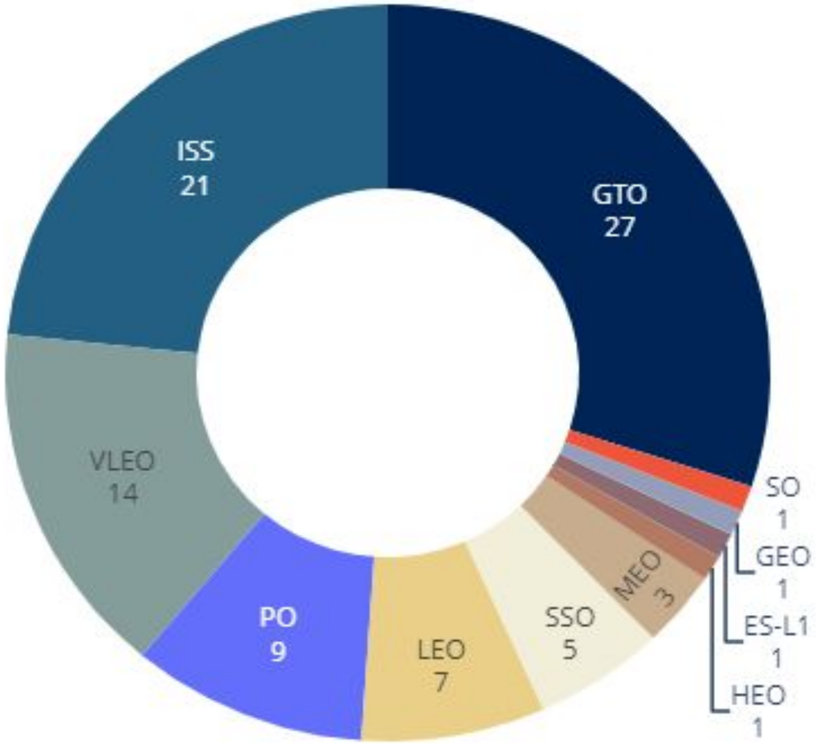
[Data Wrangling]

DATA WRANGLING

Number of launches for each orbit type

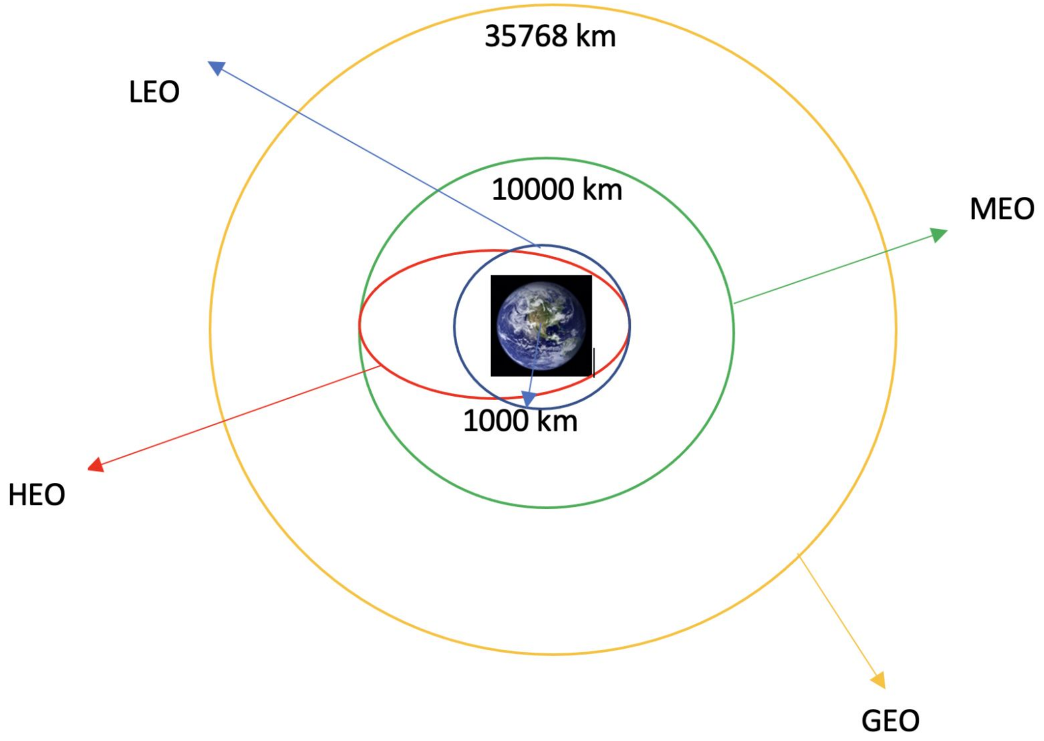
Each launch aims to an dedicated orbit, and here are some common orbit types: Low Earth orbit (LEO), Very Low Earth Orbits (VLEO), GTO, Sun-synchronous orbit (SSO), ES-L1, highly elliptical orbit (HEO), ISS, MEO, HEO, GEO, PO

Based from the results, a large number of launches were dedicated to GTO, ISS, VLEO, nad PO



[Data Wrangling]

DATA WRANGLING



Catalog of Earth Satellite Orbits

DATA WRANGLING

Number and occurence of mission outcome per landing pad

The outcome of each launch are represented by the following:

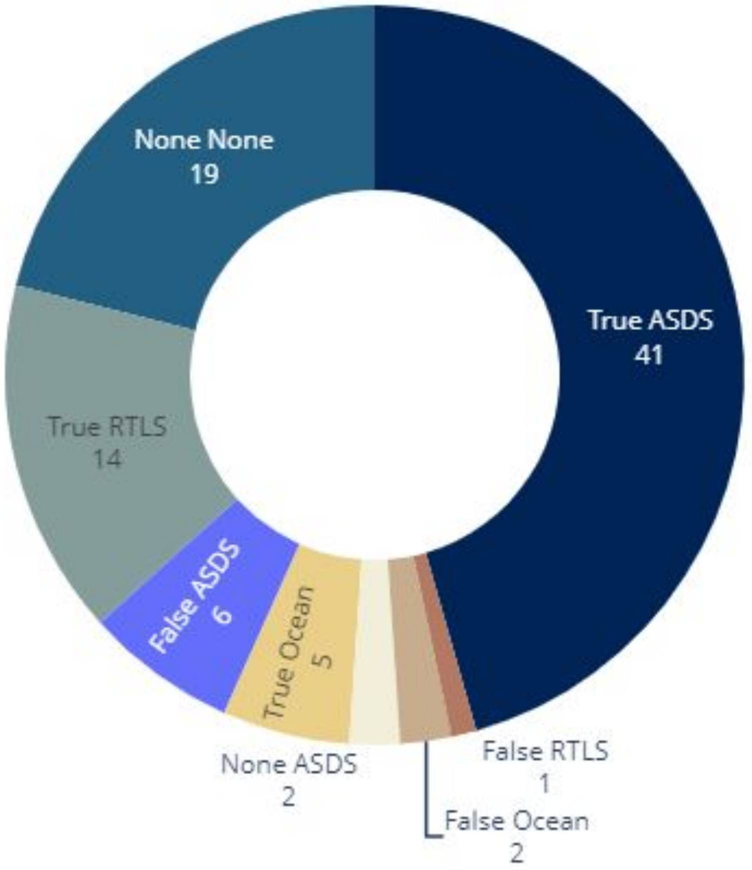
- None None
- True ASDS/ False ASDS
- True RTLS / False RTLS
- True/False Ocean

There were 2 components : TRUE/FALSE and ASDS/RTLS/OCEAN.

TRUE means successful landing (so FALSE means unsuccessful landing). ASDS means drone ship, RTLS means ground pad, OCEAN means region on the ocean.

For example, True ASDS means successful landing on the drone ship.

None None represents failure to land.



Data Wrangling

DATA WRANGLING: Feature engineering

First, we'll do some pre-processing on the explanatory variables

Prediction models cannot work directly with (text) string-type categorical data so it is important to assign numerical values to string/categories and use this numerical values in the prediction. This can be done by **applying one-hot encoding to each categorical column**

Categorical columns : orbit type, launch site, landing pad, and serial.

```
# Feature engineering
# features is a dataframe containing the 'FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite',
'Flights', 'GridFins', 'Reused', 'Legs', 'LandingPad', 'Block', 'ReusedCount', 'Serial

# One hot encoding

orbit_dummies = pd.get_dummies(features['Orbit'])
launchsite_dummies = pd.get_dummies(features['LaunchSite'])
landingpad_dummies = pd.get_dummies(features['LandingPad'])
serial_dummies = pd.get_dummies(features['Serial'])

features_one_hot = pd.concat([ features[[
        'FlightNumber', 'PayloadMass',
        'Flights', 'GridFins', 'Reused',
        'Legs', 'Block', 'ReusedCount'
        ]],
        orbit_dummies ,
        launchsite_dummies,
        landingpad_dummies,
        serial_dummies ], axis =1)

features_one_hot= features_one_hot.astype('float64')
```

[EDA with Visualization]

*** It is implemented on the last part of the EDA with Visualization notebook instead of the Data Wrangling notebook*

DATA WRANGLING: Transforming landing outcome into binary classification

The ‘Outcome’ column contains the outcomes of each launch and there’s 8 distinct values namely: True ASDS, None None, True RTLS, False ASDS, True Ocean, False Ocean, None ASDS, and False RTLS.

Since we want to transform it into a binary classification problem where 1 indicates successful landing and 0 if not, we need to separate good outcomes from bad outcomes, ie

Good outcomes (anything that has TRUE)

- **True ASDS, True RTLS, True Ocean**

Bad outcomes (anything that has FALSE or NONE)

- **False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None’**

```python
# Transforming landing outcome into binary classification

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

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

# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

landing_class= df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1 )
```

[Data Wrangling]

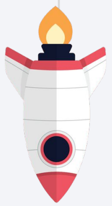
EDA with DATA VISUALIZATION

1. Visualize the relationship between Flight Number and Launch Site
2. Visualize the relationship between Payload and Launch Site
3. Visualize the relationship between success rate of each orbit type
4. Visualize the relationship between flight number and orbit type
5. Visualize the relationship between payload and orbit type
6. Visualize launch success yearly trend

[EDA with Data Visualization]()

EDA with SQL

1. Unique launch sites in the space mission
2. 5 records where launch sites begin with the string ‘CCA’
3. total payload mass carried by boosters launched in NASA
4. Display average payload mass carried by booster version F9 v1.1
5. Date when the first successful landing outcomes in group pad was achieved
6. name of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000



[EDA with SQL]

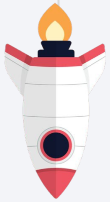
EDA with SQL

7 total number of successful and failure mission outcomes

8 names of booster versions which have carried the maximum payload mass

9 list of failed lancing outcomes in drone ship, their booster versions, and launch site name for the year 2015

10 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



[EDA with SQL]

BUILD AN INTERACTIVE MAP WITH FOLIUM

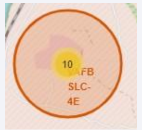
The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories.

This section focuses on analyzing the existing launch sites locations and its proximities.

[Interactive map with Folium]

BUILD AN INTERACTIVE MAP WITH FOLIUM

With the help of the folium python library, map objects such as **circles, markers, lines, and numbers** are used to analyze the location of the different launch site



The circles are used to show all the locations that are within **1 km** radius of all the launch sites

A line extending from a specific launch site to its proximities (coastline, railways, highways, city, etc.) and a number is used to indicate the **distance** (in km) between the two locations

For each launch site, a marker cluster is added to indicate the outcome of the launch. Green, if its a success, and red if it failed.

Interactive map with Folium

Build a Dashboard with Plotly Dash

There's two visualization in the dash application

1. An **interactive pie chart** which shows the total success launches for ALL sites and the success rate of the selected launch sites
2. An **interactive scatter plot** with 'Payload Mass (kg) as the x axis, 'Class' as the y-axis and "Booster Version Category" for the colors.

```
# Build a dash application
# Success pie chart

@app.callback(
     Output(component_id='success-pie-chart',component_property='figure'),
     [Input(component_id='site_dropdown',component_property='value')]
)
def update_graph(site_dropdown):
    if (site_dropdown == 'All Sites'):
        df  = spacex_df[spacex_df['class'] == 1]
        fig = px.pie(df, names = 'Launch Site',hole=.3,title = 'Total Success Launches By all sites')
    else:
        df  = spacex_df.loc[spacex_df['Launch Site'] == site_dropdown]
        fig = px.pie(df, names = 'class',hole=.3, title = 'Total Success Launches for site
'+site_dropdown)
    return fig
```

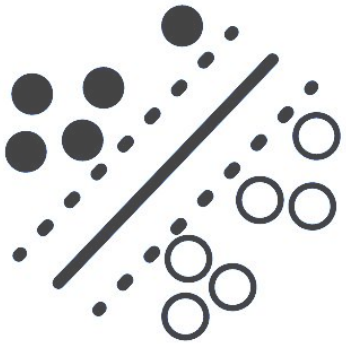
```
# Build a dash application
# Success scatter plot

@app.callback(
     Output(component_id='success-payload-scatter-chart',component_property='figure'),
     [Input(component_id='site_dropdown',component_property='value'),
     Input(component_id="payload_slider", component_property="value")]
)
def update_scattergraph(site_dropdown,payload_slider):
    if site_dropdown == 'All Sites':
        low, high = payload_slider
        df  = spacex_df
        mask = (df['Payload Mass (kg)'] > low) & (df['Payload Mass (kg)'] < high)
        fig = px.scatter(
            df[mask], x="Payload Mass (kg)", y="class",
            color="Booster Version",
            size='Payload Mass (kg)',
            hover_data=['Payload Mass (kg)'])
    else:
        low, high = payload_slider
        df  = spacex_df.loc[spacex_df['Launch Site'] == site_dropdown]
        mask = (df['Payload Mass (kg)'] > low) & (df['Payload Mass (kg)'] < high)
        fig = px.scatter(
            df[mask], x="Payload Mass (kg)", y="class",
            color="Booster Version",
            hover_data=['Payload Mass (kg)'])
    return fig
```

[Dash App]

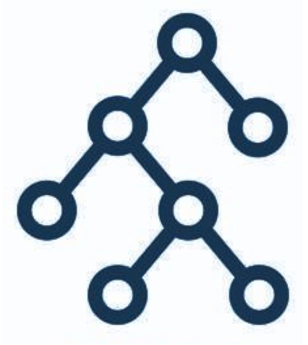
Predictive Analysis

We used four classifiers in predicting whether the Falcon 9 will have a successful landing or not.

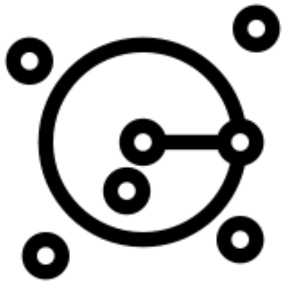
Logistic Regression



Support Vector Machines



Decision Trees



KNN

[Machine Learning Prediction]

Predictive Analysis

1 Standardize the data

2 Split the data into training and testing data using the function train_test_split.

3 For each classifier, specify the list of values for each parameter

4 Create a GridSearchCV and fit the best parameters using the training data

5 Display the hyperparameters and score

6 Calculate the accuracy on the test data

6 Display the confusion matrix

Machine Learning Prediction

IBM Developer
SKILLS NETWORK

SECTION 3:

RESULTS

Exploratory data analysis results
Launch sites analysis and proximities
Interactive analytics demo in screenshots
Predictive analysis results

WHAT IS THE RELATIONSHIP BETWEEN FLIGHT NUMBER AND LAUNCH SITE?

[EDA with Data Visualization]()

INSIGHT/S

SUCCESS RATE

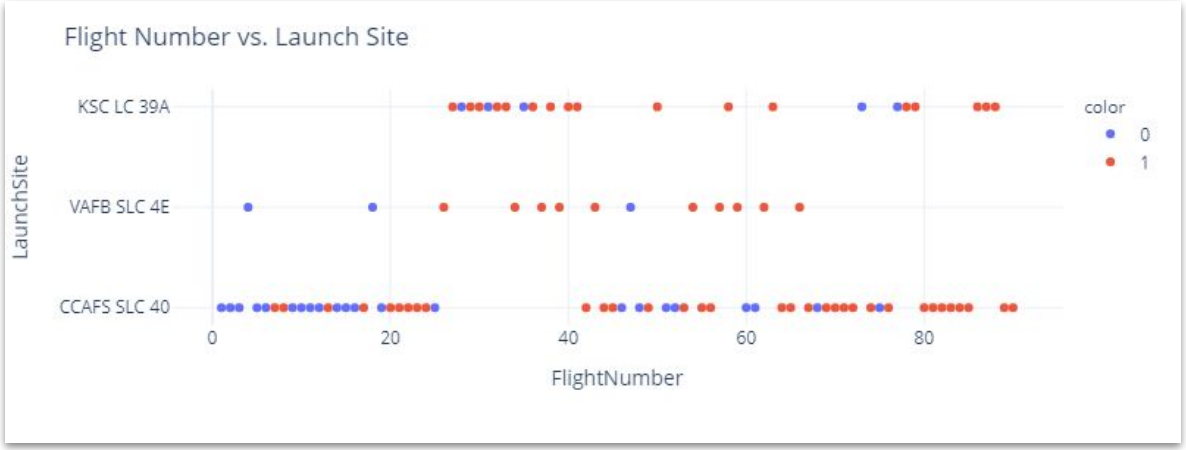
The different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

INSIGHT/S

We see that rate of successful landing of the first stage increases as the number of flights increases. Earlier launches are part of the experimental stage and are justifiably prone to vulnerabilities. This explains the high rate of failure on the first few flights.

EARLIER LAUNCHES

In addition to that, it seems like a large number of earlier launches were conducted on Cape Canaveral Space Launch Complex 40 CCAFS SLC 40.



The scatter plot shows how the flight number (x -axis) and the launch site (y-axis) relate to each other and how this two feature possibly affect the outcome of the launch.

WHAT IS THE RELATIONSHIP BETWEEN PAYLOAD AND LAUNCH SITE?

EDA with Data Visualization

INSIGHT/S

Falcon 9 heavy launches (payload mass greater than 10000) are all conducted at **CCAFS SLC 40 site and KSC LC 39A.** In other words, VAFB-SLC have no rockets launched for heavy payload.

The scatter plot below shows how the mass of the payload (x -axis) and the launch site (y-axis) relate to each other and how this two feature possibly affect the outcome of the launch.

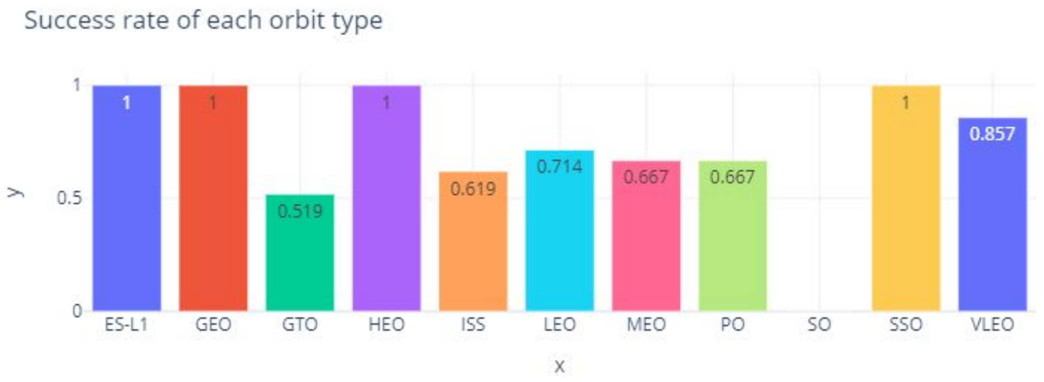
WHAT IS THE RELATIONSHIP BETWEEN SUCCESS RATE AND ORBIT TYPE?

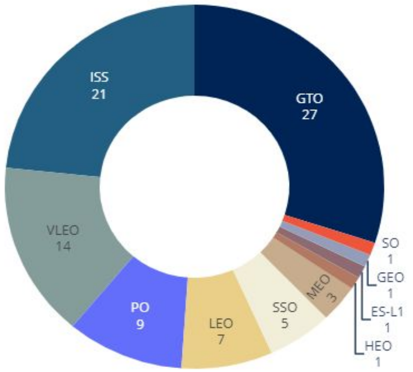
INSIGHTS

ES-L1, GEO, HEO, and SO have one dedicated flight each and they all resulted in a successful landing except SO which explains the 100% (or 1) success rate on the first 3 orbit and 0% success rate on SO.

INSIGHTS

GTO having a total of 27 dedicated flights have a very low success rate ~51.9% together with ISS orbit which has a total of 21 total dedicated flights





The bar plot shows the average success rate of each orbit type and the pie chart shows the total number of launches for each orbit

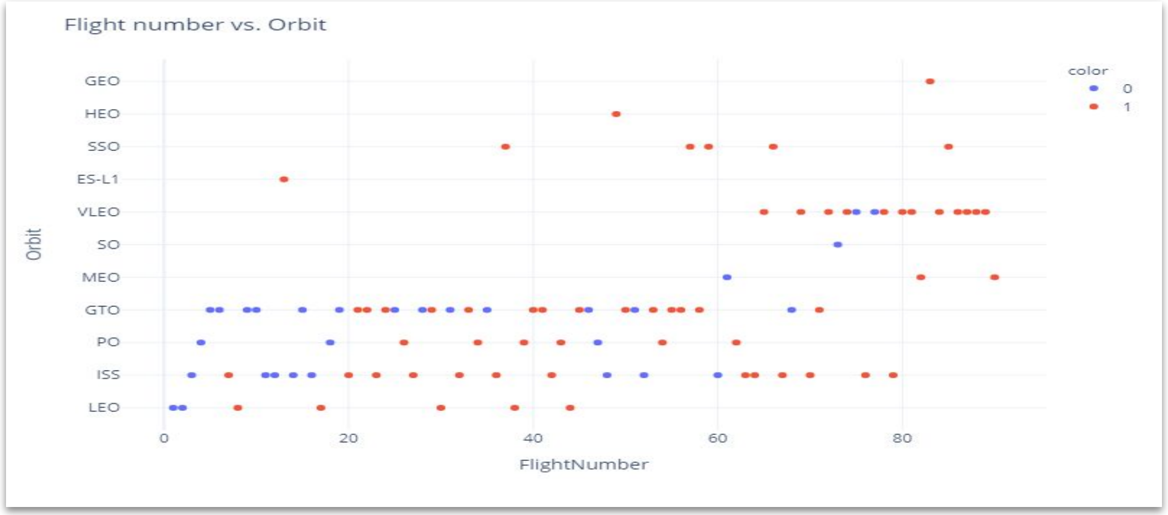
[EDA with Data Visualization]

WHAT IS THE RELATIONSHIP BETWEEN FLIGHT NUMBER AND ORBIT TYPE?

[EDA with Data Visualization]

INSIGHT/S

In the LEO and MEO orbit, all the flight mission was successful after the first few flights. On the other hand, there seems to be no apparent relationship between flight number when in VLEO, GTO, PO, and ISS orbit.



The scatter plot below shows how the flight number (x -axis) and the orbit type (y-axis) relate to each other and how this two feature possibly affect the outcome of the launch.

WHAT IS THE RELATIONSHIP BETWEEN PAYLOAD AND ORBIT TYPE?

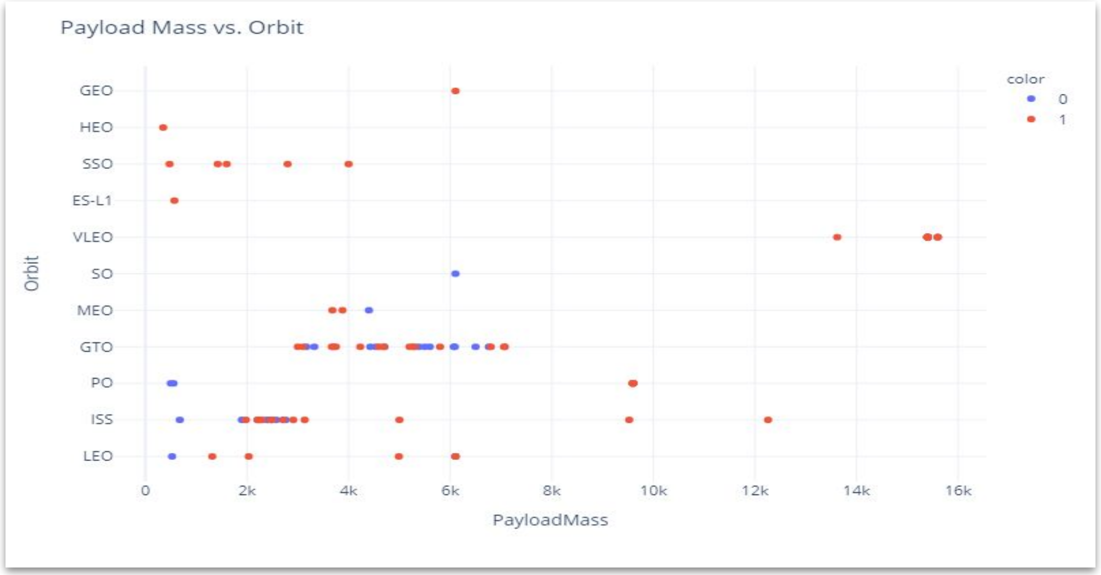
EDA with Data Visualization

INSIGHTS

With heavy payloads, a positive landing rate is expected for PO, ISS, and LEO orbit. However, there seems to be no apparent relationship between payload mass and success rate for GTO.

INSIGHTS

As for MEO, the launches almost have the same payload mass and the there's only three so we can't infer an insight right now.



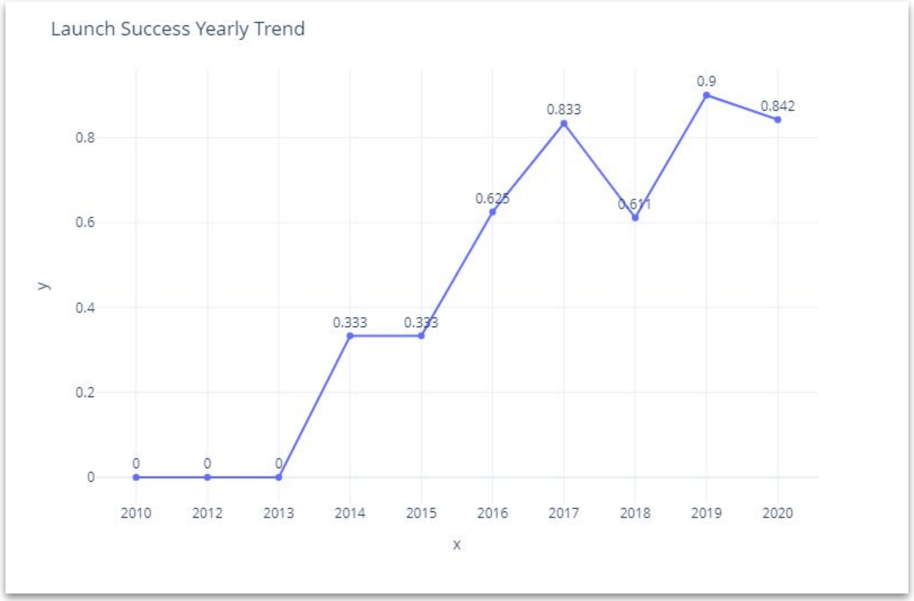
The scatter plot below shows how the payload (x -axis) and the orbit type (y-axis) relate to each other and how this two feature possibly affect the outcome of the launch.

AVERAGE SUCCESS RATE FROM YEAR 2010 TO 2020

EDA with Data Visualization

INSIGHTS

The launch success rate is trending upward with a slight decline on the year 2018 and 2020.



Average success rate of Falcon 9 launches throughout the years

All Launch Site Names

```
# Unique launch sites in the space mission

%sql
select distinct(launch_site)
from GMH61120.SPACEXTBL
```

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

EDA with SQL

Launch Site Names Begin with 'CCA'

```
# 5 records where the launch sites begin with the string ‘CCA’

%sql
select *
from GMH61120.SPACEXTBL
where launch_site LIKE 'CCA%'
limit 5
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

EDA with SQL

Total Payload Mass

```
# Total payload mass carried by boosters launched by NASA (CRS)

%sql
select SUM(payload_mass__kg_)
from GMH61120.SPACEXTBL
where customer = 'NASA (CRS)'
```

1
45596

[EDA with SQL]

Average Payload Mass by F9 v1.1

```
# average payload mass carried by booster version f9 v1.1

%sql
select AVG(payload_mass__kg_)
from GMH61120.SPACEXTBL
where booster_version = 'F9 v1.1'
```

1
2928

[EDA with SQL]

First Successful Ground Landing Date

```
# Date when the first successful landing outcome in ground pad was achieved

%sql
select MIN(DATE) as first_successful_landing_date
from GMH61120.SPACEXTBL
where landing__outcome = 'Success (ground pad)'
```

first_successful_landing_date
2015-12-22

EDA with SQL

Successful Drone Ship Landing with Payload between 4000 and 6000

```
# Names of the booster which have success in drone ship and have payload mass graeter than 4000 but less than 6000

%sql
select distinct(booster_version)
from GMH61120.SPACEXTBL
where (payload_mass__kg_ BETWEEN 4000 and 6000) and (landing__outcome = 'Success (drone ship)')
```

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

[EDA with SQL]

Total Number of Successful and Failure Mission Outcomes

```
# total number of successful and failure mission outcomes

%sql
select mission_outcome, count
from GMH61120.SPACEXTBL
group by mission_outcome
```

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

EDA with SQL

Boosters Carried Maximum Payload

```
# names of the booster_versions which have carried the maximum payload mass

%sql
select booster_version, payload_mass__kg_
from GMH61120.SPACEXTBL
where payload_mass__kg_ = (
  select MAX(payload_mass__kg_)
  from GMH61120.SPACEXTBL
  )
```

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

EDA with SQL

2015 Launch Records

```
# all failed landing outcomes in drone ship, their booster versions,
# and launch site names for in year 2015

%sql
select DATE, booster_version, launch_site, landing__outcome
from GMH61120.SPACEXTBL
where (landing__outcome = 'Failure (drone ship)') and (DATE LIKE '2015%' )
```

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

EDA with SQL

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

```
# rank the count of landing outcomes between date 2010-06-04 and 2017-03-20

%sql
select landing__outcome, count, rank() over (order by count desc) as rank
FROM (
  select landing__outcome, count(landing__outcome) as  count
  from GMH61120.SPACEXTBL
  where DATE >= '2010-06-04' and DATE <='2017-03-20'
  group by landing__outcome
  )
```

landing__outcome	COUNT	RANK
No attempt	10	1
Failure (drone ship)	5	2
Success (drone ship)	5	2
Controlled (ocean)	3	4
Success (ground pad)	3	4
Failure (parachute)	2	6
Uncontrolled (ocean)	2	6
Precluded (drone ship)	1	8

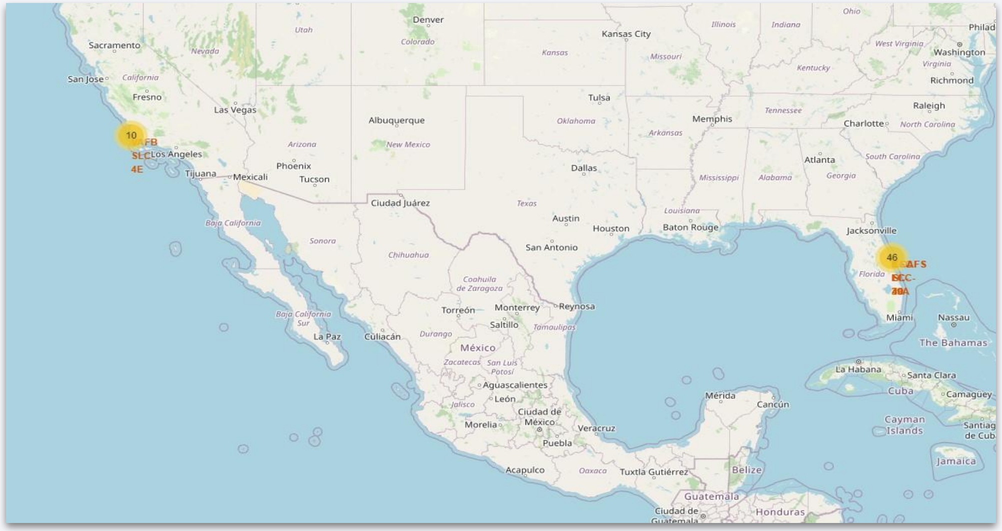
IBM Developer
SKILLS NETWORK

SECTION 3:

RESULTS

Exploratory data analysis results
Launch sites analysis and proximities
Interactive analytics demo in screenshots
Predictive analysis results

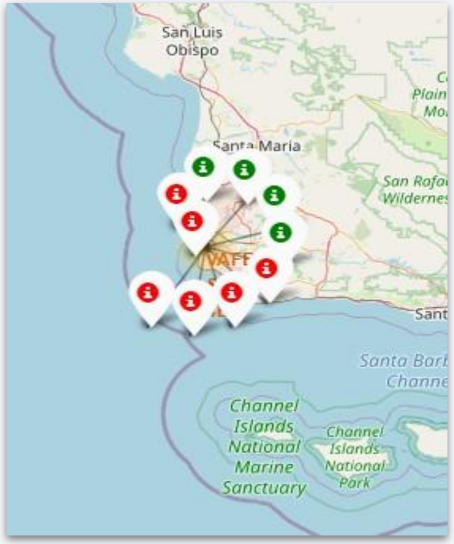
Geographical data visualization of the different launch sites



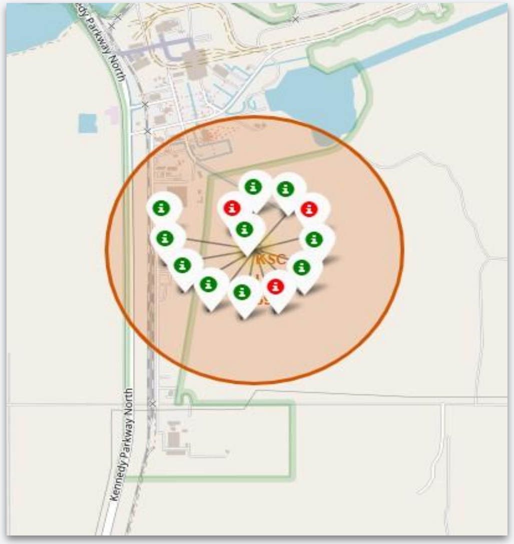
The black dash line is the equator line . The yellow circles show the location of the different launch sites. The circle covers locations that are within 1 km distance from the sites.

The maps above shows the location of the launch sites. Launching sites, based on the circled areas above, are close to coastline and are of close proximity to the equator line. Apparently, launch site are usually located on coastline so that, in the event, where the Falcon 9 failed to launch or land there'll be less damage on the hinterland. Moreover, launching near the equator line helps the rocket to get an initial boost equal to the velocity of Earth surface. This "boost" gradually reduces as we move to the pole which explains why locations near the equator line is the optimal location for rocket launches.

Launch outcome for each site

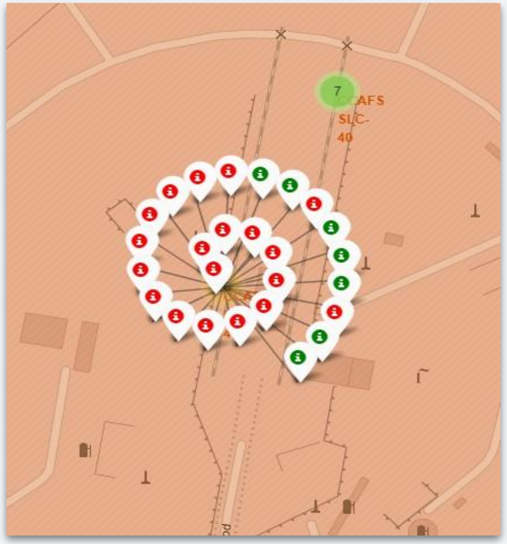


VAFB SEC 4E



KSC LC 39A

CCAFS LC 40

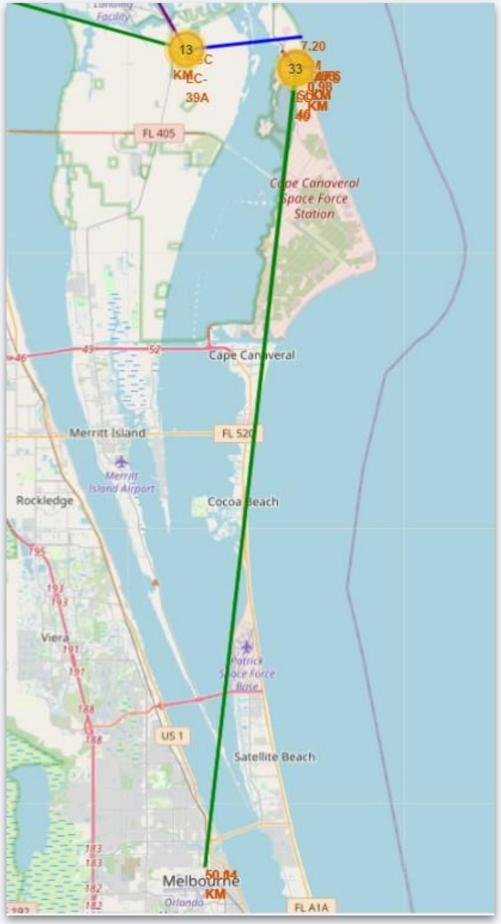


CCAFS LC 40

For each launch site, a marker cluster is added to indicate the outcome of the launch. Green, if its a success, and red if it failed.

From the color-labeled markers in marker clusters, we can see that CCAFS LC 40 have a relatively very low success rate and that KSC LC 39A have a relatively high success rate.

CCAFS: Distance lines to different proximities

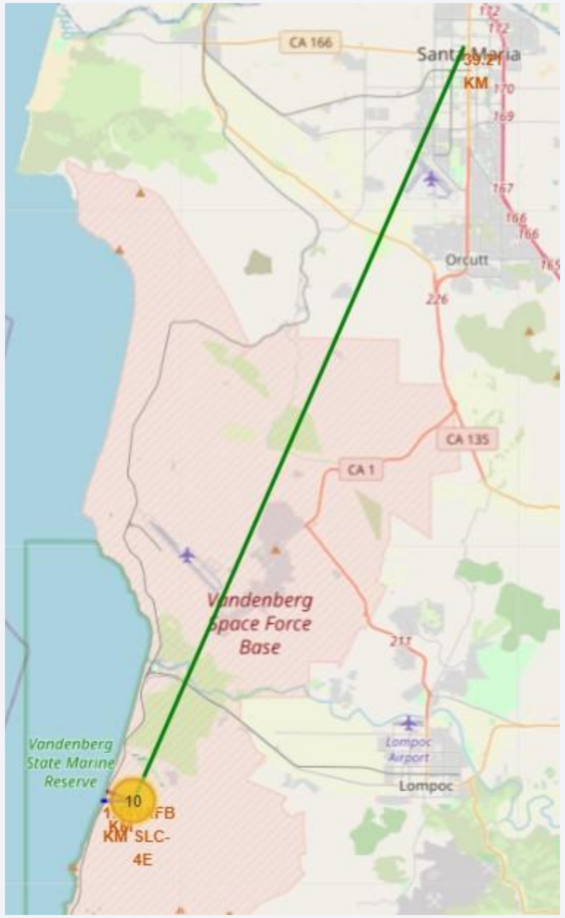


The maps on the left shows the nearest coastline (blue line), railroad (purple line), highway (yellow), and the nearest city (green)

We see that the closest coastline, railroad, and highway from CCAFS LC 40 is almost within the 1 km radius distance of the said site.

However, Melbourne, the city closest to CCAFS LC 40 is located very far, ~50.84 km, from the site.

VAFB: Distance lines to different proximities

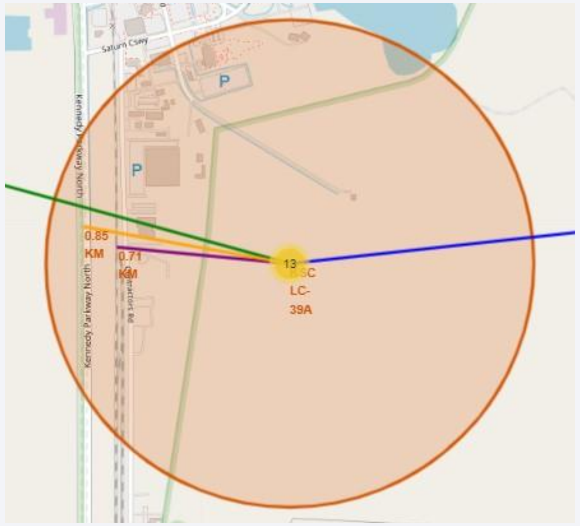


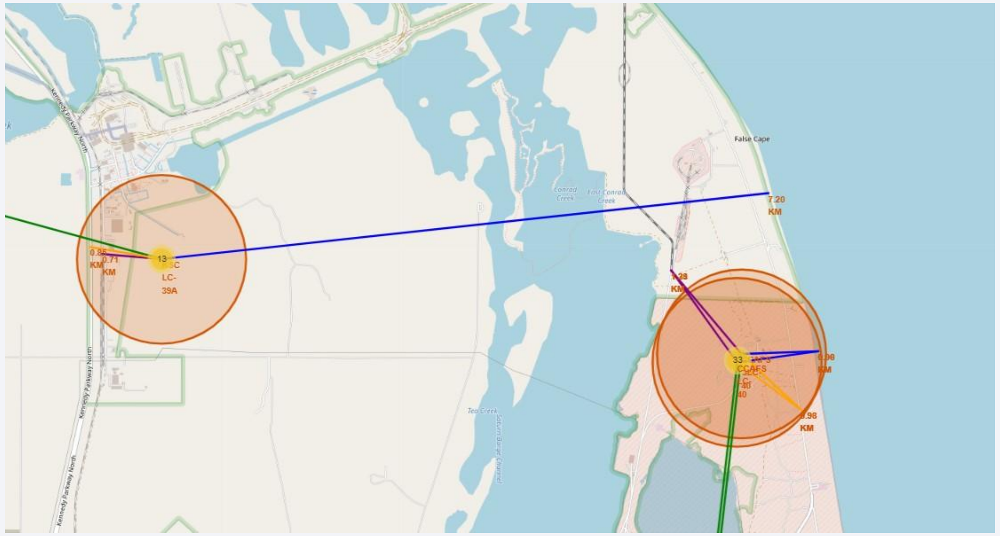
The maps on the left shows the nearest coastline (blue line), railroad (purple line), highway (yellow), and the nearest city (green)

Similarly, we see that VAFB SEC 4E is very close to a railway and to a coastline and by close we mean within the 1.5km radius. It seems that there's wasn't a highway near to it but there were still roads which allows seamless transport around the area.

The nearest city to it is Santa Maria which is 39.21 km away from it.

KSC: Distance lines to different proximities





The maps on the left shows the nearest coastline (blue line), railroad (purple line), highway (yellow), and the nearest city (green)

Note: On the third map, KSC LC 39A is the one that's on the left and CCAFS LC 40 is the one on the right.

KSC LC 39A is close to a railway and highway (less than 1 km distance) but it is quite far from the coastline (~ 7 km). The closest city to it that I could find is Titusville which is 16.32 km from the site

Insights

The optimal location for launching sites is somewhere near the equator line. In general, this three sites are in close proximity to railways, highways, and coastline. But they keep a certain distance away from cities.

Trains/railroads are often used to move rocket boosters and highways are used for seamless transportation around the site complex which is why it is necessary for launch sites to be near this kind of platforms. They are also usually located on coastline so that there'll be less damage on the hinterland in case the launching/landing failed.

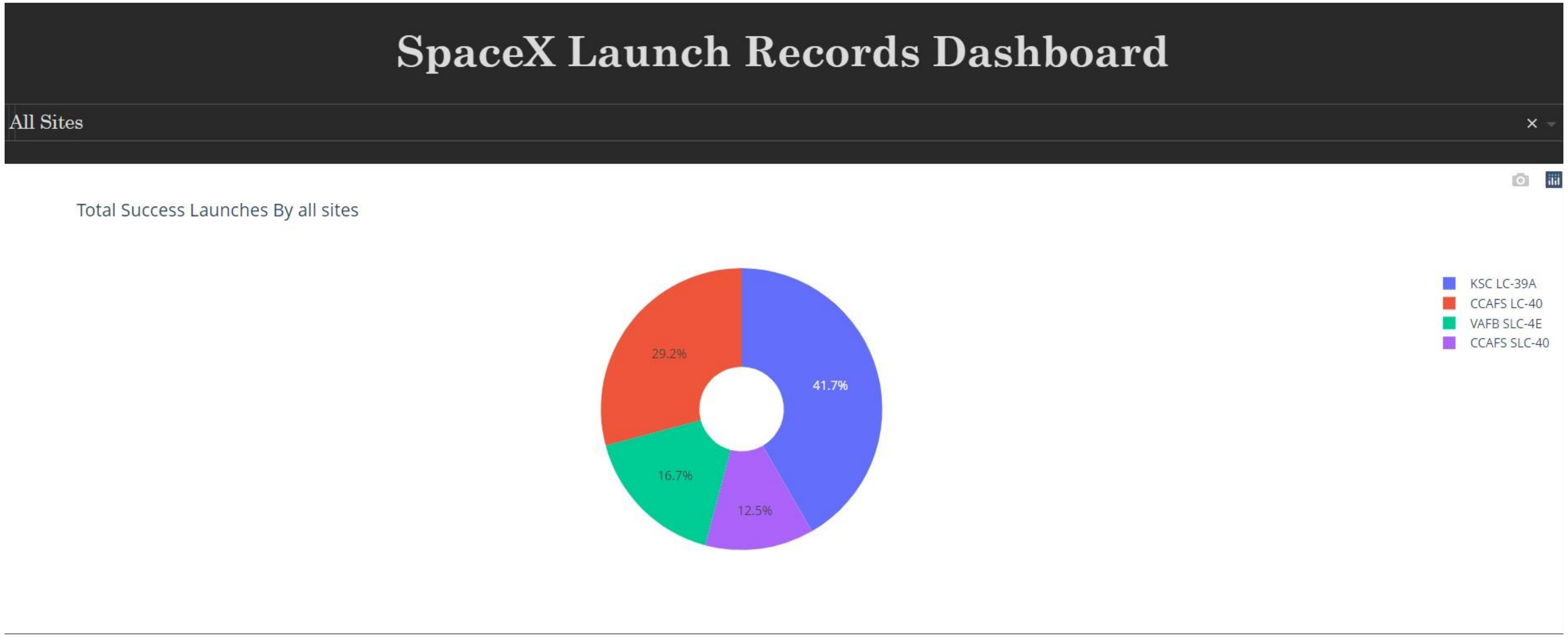
IBM Developer
SKILLS NETWORK

SECTION 3:

RESULTS

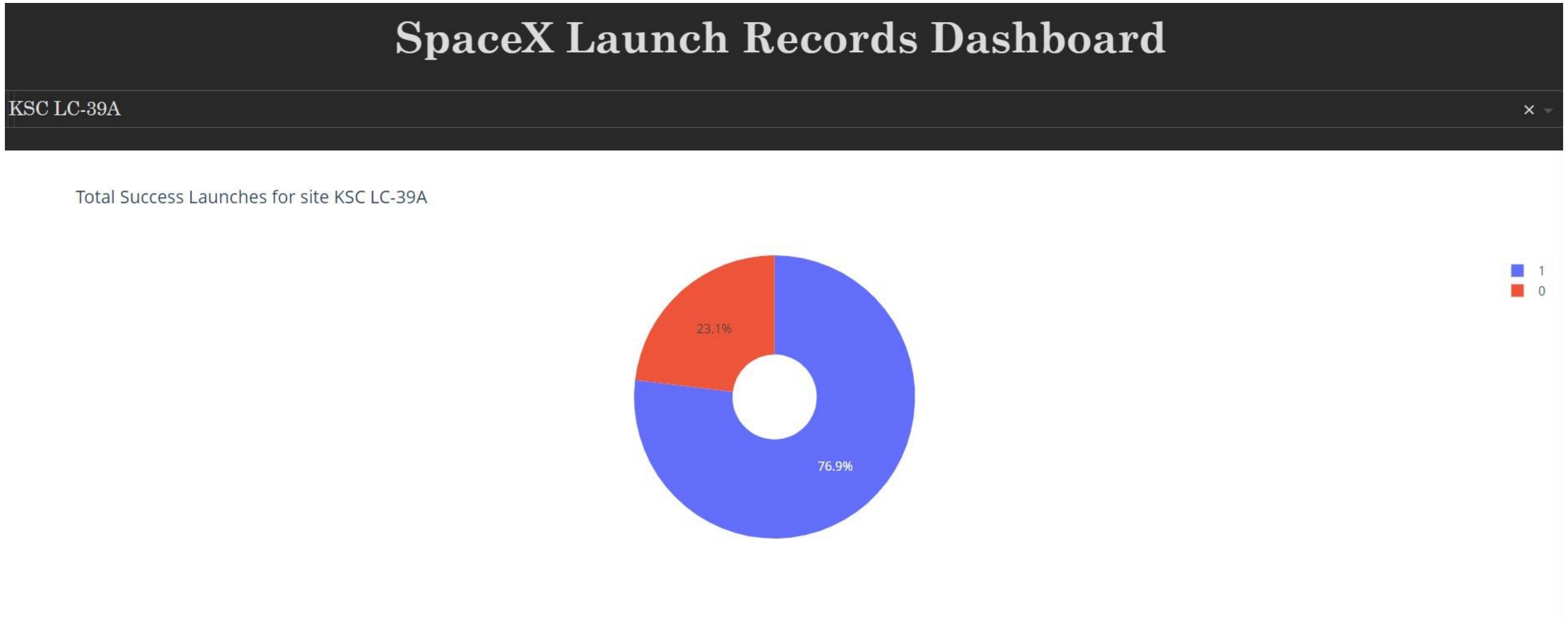
Exploratory data analysis results
Launch sites analysis and proximities
Interactive analytics demo in screenshots
Predictive analysis results

Total success launches by all sites

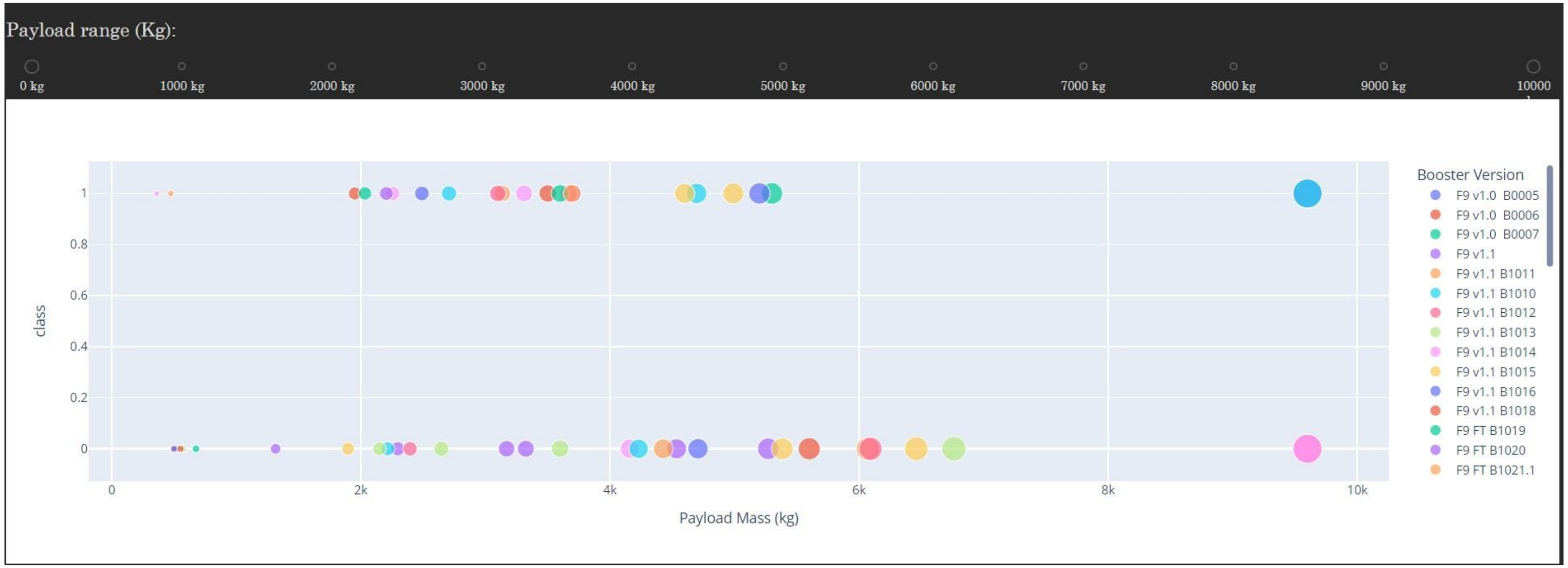


A large majority of the successful launches, 41.7% to be specific, where launched from KSC LC 39A. 29.2% where launched in CCAFS LC 40, 16.7% where launched in VAFB SLC 4E and 12.5% where launched in CCAFS SLC 4 .

Total success launches for a selected site



Scatter plot: Payload mass vs. Class



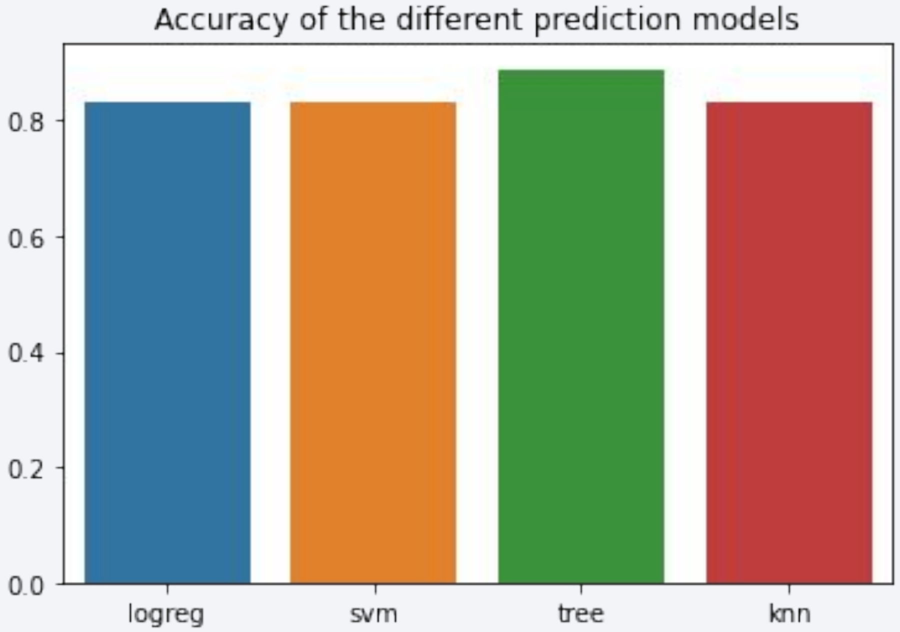
IBM Developer
SKILLS NETWORK

SECTION 3:

RESULTS

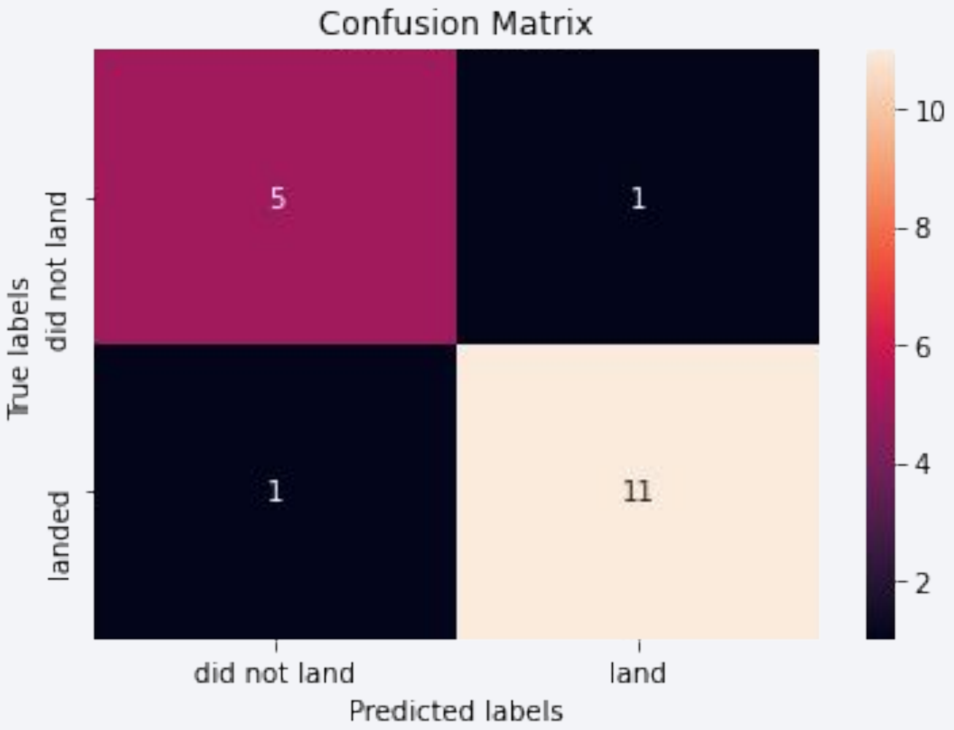
Exploratory data analysis results
Launch sites analysis and proximities
Interactive analytics demo in screenshots
Predictive analysis results

Classification Accuracy



The decision tree have the highest accuracy with 88%. The other three models have 83% accuracy.

Confusion Matrix



The test set have 18 instances, 6 of this instances resulted to failed landing and 12 resulted to successful landing. Five of the failed landing and eleven of the successful landing was predicted correctly by the trained decision tree model. The best parameters is shown below.

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 14, 'max_features':
'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'best'}
accuracy : 0.875
```

IBM Developer
SKILLS NETWORK

SECTION 4:

CONCLUSION

CONCLUSION

- Launch vehicles are traditionally designed for single flight use making them expensive costing around 165 million for each flight. But by recovering and reusing the part of the vehicles that carries the entire rocket, its payload, and all the unused fuel, the cost drives down to 62 million dollars. Reusable launch system focuses on developing new set of technologies that may be recovered and reused many times and it has been proven to be possible and is significantly more affordable than traditional launch system. This system was first used on SpaceX's falcon 9.
- The landing outcome not only determines the total cost of the current flight mission but also the next one since it affects the reusability of the first stage. Failed landing, as one can expect, will expedite more cost since the first stage needs to be rebuilt.
- Mission specific factors (such as booster version, payload mass, launch site, landing pad, and dedicated orbit), vehicle specific factors (such as grid fins, legs, reused count, and serial number), and location based factors (such as latitude, longitude, proximities) were analyzed and used as features or explanatory variables.
- SpaceX have 3 unique launch sites: Cape Canaveral Space Launch Complex 40 CCAFS SLC 40, Vandenberg Air Force Base Space Launch Complex 4E (VAFB SLC 4E), and Kennedy Space Center Launch Complex 39A KSC LC 39A.

CONCLUSION

- Overall, the launch success is trending upward with a slight decline on the year 2018 and 2020.
- A large majority of the successful launches, 41.7% to be specific, where launched from KSC LC 39A. 29.2% where launched in CCAFS LC 40, 16.7% where launched in VAFB SLC 4E and 12.5% where launched in CCAFS SLC 4.
- The rate of successful landing of the first stage increases as the number of flights increases. Earlier launches are part of the experimental stage and are justifiably prone to vulnerabilities. This explains the high rate of failure on the first few flights.
- With heavy payloads (payload mass greater than 10000), a positive landing rate is expected for PO, ISS, and LEO orbit. However, there seems to be no apparent relationship between payload mass and success rate for GTO. In addition to that, none of the heavy payload missions were launched from VAFB SLC 4E.
- Launching sites are usually close to coastline and are of close proximity to the equator line. They are also of close proximity to railways and highways but far from city proper.
- we used four machine learning classifiers as our base models - logistic regression, support vector machines, decision trees, and k-nearest neighbor. Decision trees obtained an 88% accuracy on the test set and thereby outperformed the other three models which each have an 83% accuracy.


IBM Developer
SKILLS NETWORK
THANK YOU