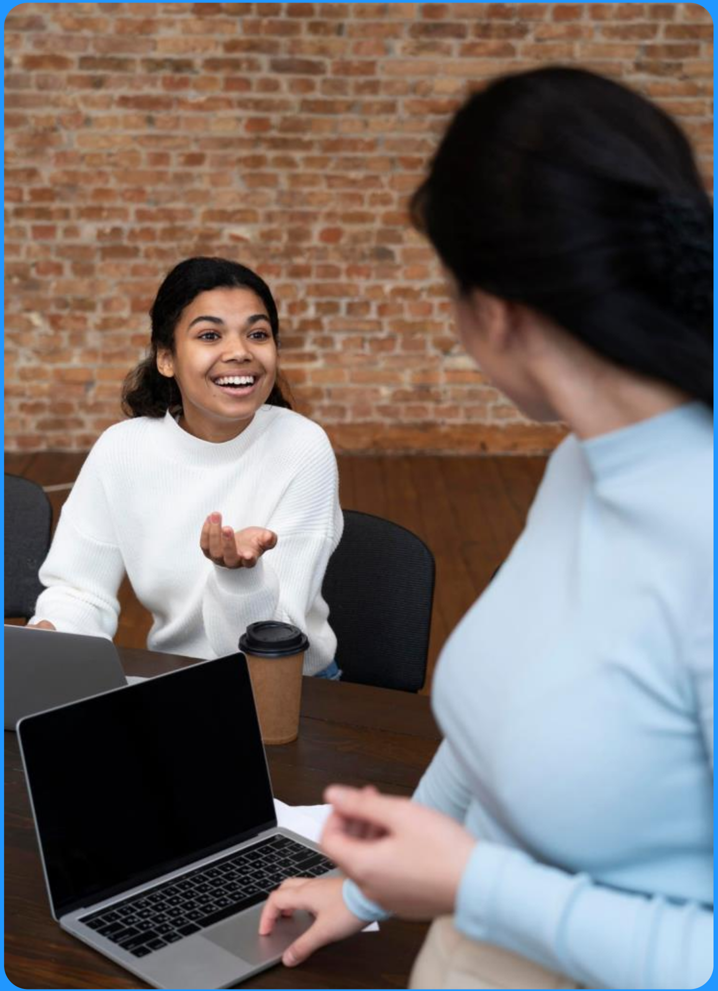
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Klaudia Dogoda
06.06.2024



Outline

1) Executive Summmary
2) Introduction
3) Methology
4) Results
5) Conclusion
6) Appendix

Executive Summary

01. Summary of methologies

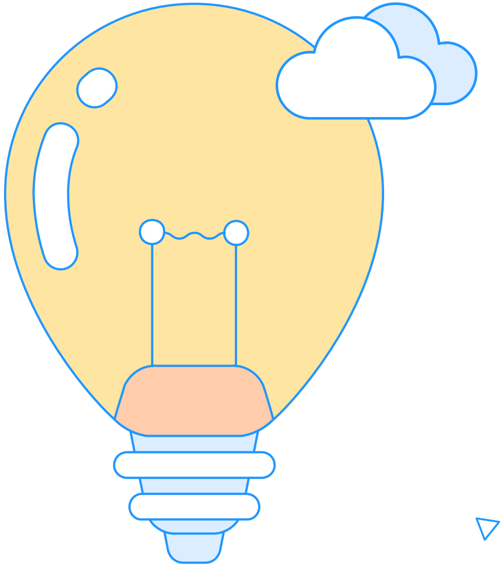
- Data collection
- Data wrangling
- Data visualization
- Working with SQL statements
- Building an interactive map with python packages and Dashbords

02. Summary of all results

- Optimal model was chosen
- Visualisations done for analysis of results

Introduction

We predicted if the Falcon 9 first stage will land successfully. 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. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch



Section 1

Methodology

Methology

- **Data collection methodology:**

SpaceX Rest API and Web Scrapping from Wikipedia

- **Performed data wrangling**

One Hot Encoding data fields for Machine Learning,dropping irrelevant columns

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

Plotting : Scatter Graphs, Bar Graphs to show relationships between variables to show patterns of data.

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

This was used to achive goal of getting interacting visualizations

- **Performed predictive analysis using classification models**

How to build, tune, evaluate classification models

Methology

Data Collection methology

- SpaceX Rest API
- Web Scrapping

Assumptions

Despite being red, Mars is a cold place, not hot. It's full of iron oxide dust

Issues

Mercury is the closest planet to the Sun and also the smallest one

Dependencies

Jupiter is the biggest in the Solar System and the fourth-brightest

Data collection

01. 02. 03. 04.

Phase 1

Collecting the data with API

Phase 2

Converting the data frame with JSON

Phase 3

Pre-processing (updating columns and rows)

Phase 4

Filtering data for Falcon 9 launches

Data Collection – Space X API

1 .Getting Response from API

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

2.Converting to json

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

3. Pre-processing:

```
# Lets take a subset of our dataframe keeping only the features we want and the 
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_

# We will remove rows with multiple cores because those are falcon rockets with 
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single v
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extractin
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

https://github.com/klaudiadg/Falcon_9

Data Collection – Space X API - part 2

4. Assign list to a dictionary

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
df = pd.DataFrame(launch_dict)
```

5. Filter on Falcon 9 launches

```
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

6. Export to csv

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

https://github.com/klaudiadg/Falcon_9

Data Collection – web scrapping

1. HTML response

```
falcon9_data = requests.get(static_url)
print(falcon9_data.status_code)
```

```
200
```

2. Beautiful Soup Object

```
soup = BeautifulSoup(falcon9_data.text, "html.parser")
```

3. Finding all tables

```
html_tables = soup.find_all('table')
```

4. Getting column names

```
]: ths = first_launch_table.find_all('th')
```

```
]: column_names = []
for table in ths:
    try:
        name = extract_column_from_header(table)
        if ( name is not None and len(name)>0):
            column_names.append(name)
    except:
        pass
```

5. Dictionary creation

```
launch_dict= dict.fromkeys(column_names)
# Remove an irrelvant column
del launch_dict['Date and time ( )']
# Let's initial the launch_dict with eac
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
```

Data Collection – web scrapping – part 2

6. Appending data to keys

```
# Customer
# TODO: Append the customer into launch_dict with key `Customer`
if row[6].a is not None:
    customer = row[6].a.string
else:
    customer = row[6].string
launch_dict['Customer'].append(customer)
#print(customer)

# Launch outcome
# TODO: Append the launch_outcome into launch_dict with key `Launch outcome`
launch_outcome = list(row[7].strings)[0]
launch_dict['Launch outcome'].append(launch_outcome)
```

7. Converting to dataframe

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

8.Export to csv

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.We mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful

Performed Analysis on EDA dataset

Calculated numbers of launches

Calculated numer and occurances for each orbit

Calculated numer and occurance for missions per orbit types

Export of dataset

Created a landing outcom lable from outcome column

Worked out success rate

https://github.com/klaudiadg/Falcon_9

EDA with Data Visualization

Scatter Graphs:

- Flight Number VS. Payload Mass
- Flight Number VS. Launch Site
- Payload VS. Launch Site
- Orbit VS. Flight Number
- Payload VS. Orbit Type
- Orbit VS. Payload Mass

Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation . Scatter plots usually consist of a large body of data

Bar Graph:

- Mean vs Orbit

A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes.

Line Graph:

- Success Rate vs Year

Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded

https://github.com/klaudiadg/Falcon_9

EDA with SQL

SQL queries used to gather information about dataset:

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date where the successful landing outcome in drone ship was achieved.
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster_versions which have carried the maximum payload mass
- Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- Ranking the count of successful landing_outcomes between dates in descending order

Build an Interactive Map with Folium

To visualize the Launch Data into an interactive map the Latitude and Longitude Coordinates at each launch site were taken and added a Circle Marker around each launch site with a label of the name of the launch site.We assigned the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster().
Using Haversine's formula we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks

Build a Dashboard with Plotly Dash

Dash and html componetnts were used as they are important to ctrate graphs, tables, dropdowns.

Pandas library was used to simplify the work by using datafames.
Plotly was used to create graphs. Pie chart and scatter was used to plot. Rangslider was used to select payload mass range. Dropdowns were used for launch sites.

https://github.com/klaudiadg/Falcon_9

Predictive Analysis (Classification)

BUILDING MODEL

- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Decide which type of machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets and train our dataset.

EVALUATING MODEL

Check accuracy for each model
Get tuned hyperparameters for each type of algorithms
Plot Confusion Matrix

IMPROVING MODEL

Feature Engineering
Algorithm Tuning

FINDING THE BEST PERFORMING CLASSIFICATION MODEL

The model with the best accuracy score wins the best performing model
In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.

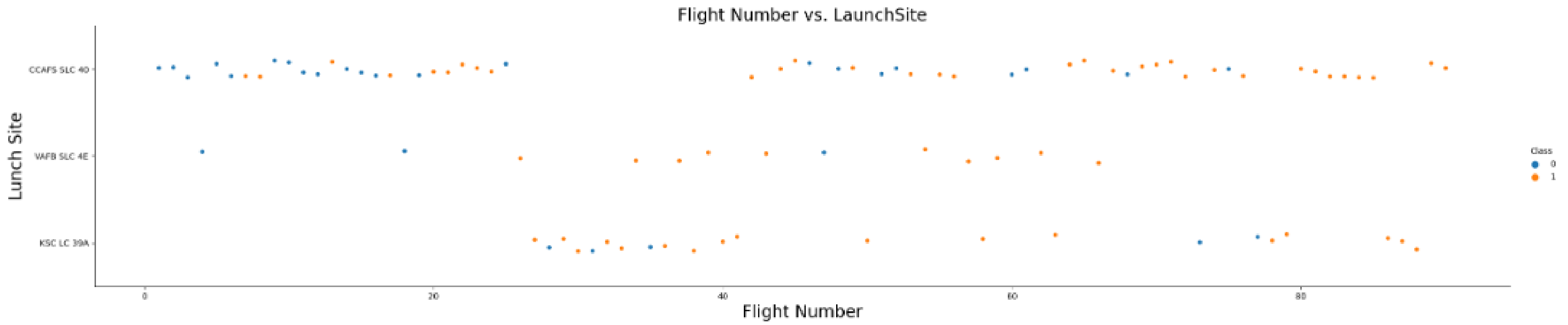
Results

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

Section 2

Insights drawn from EDA

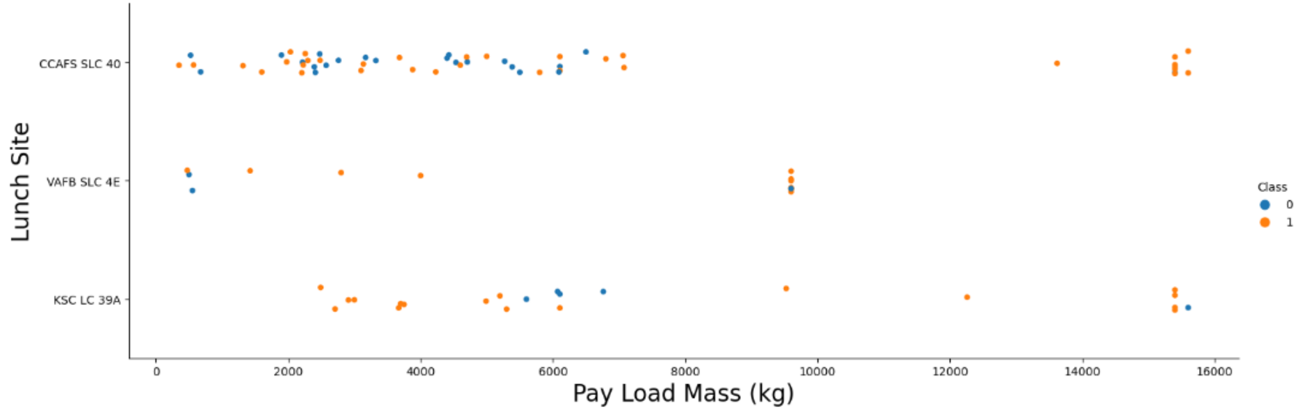
Flight Number vs. Launch Site



The more amount of flights at a launch site the greater the success rate at a launch site

https://github.com/klaudiadg/Falcon_9

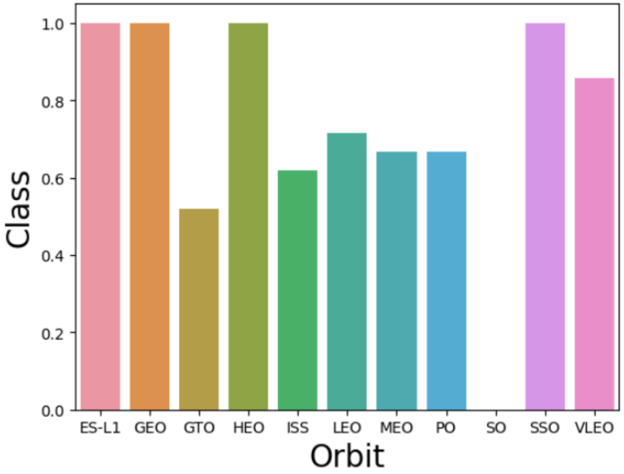
Payload vs. Launch Site



With increase of payload, the success rate is increasing with the saunach sites.

https://github.com/klaudiadg/Falcon_9

Success Rate vs. Orbit Type

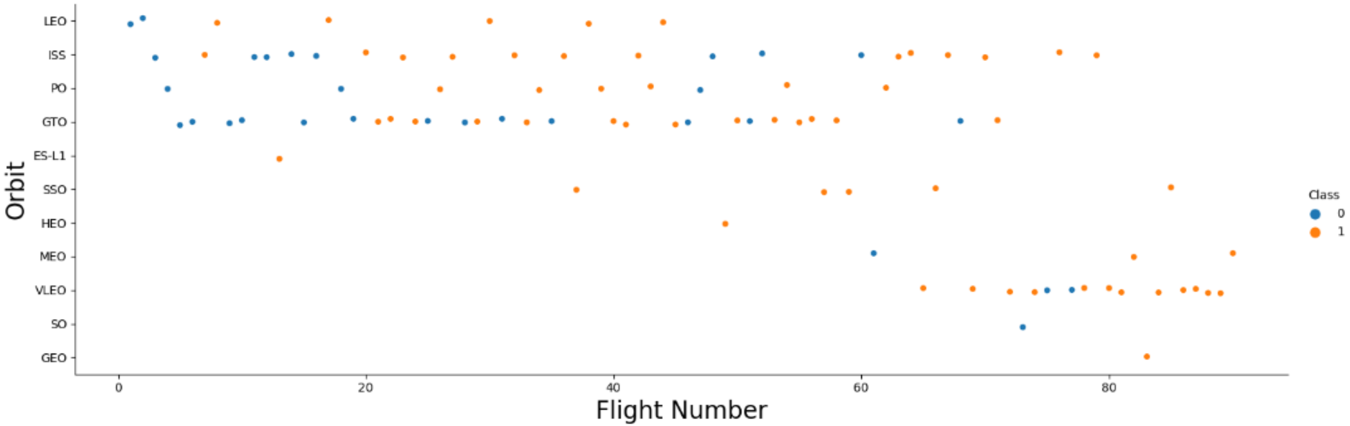


ES-L1, GEO, HEO, SSO have success rate of 100%. SO has success rate of 0%.

https://github.com/klaudiadg/Falcon_9

Flight Number vs. Orbit Type

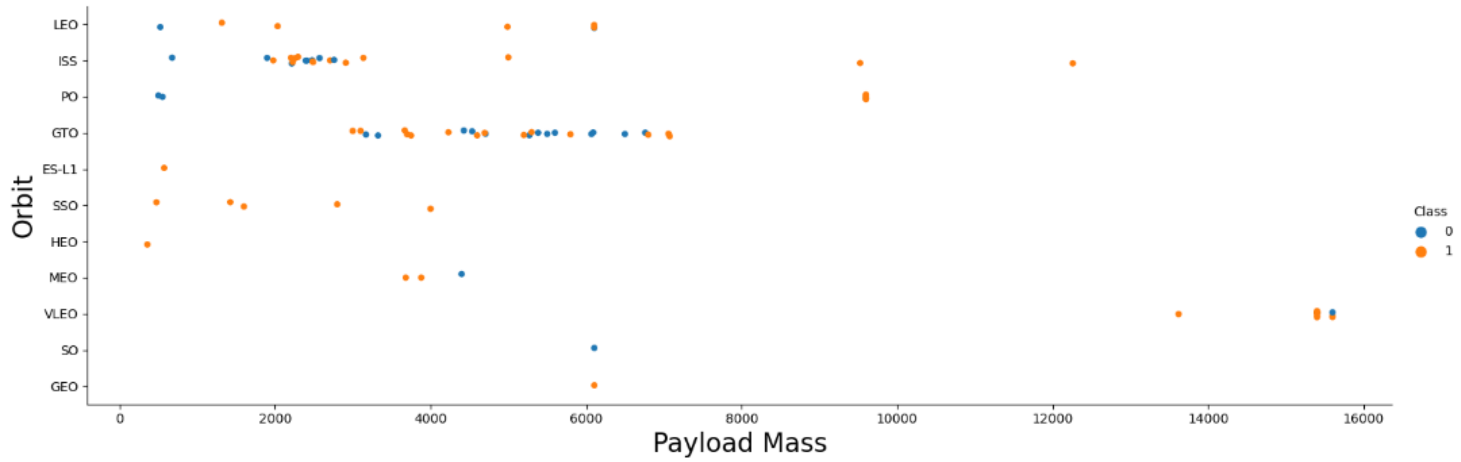
There is no relationship between flight numer and Orbit



https://github.com/klaudiadg/Falcon_9

Payload vs. Orbit Type

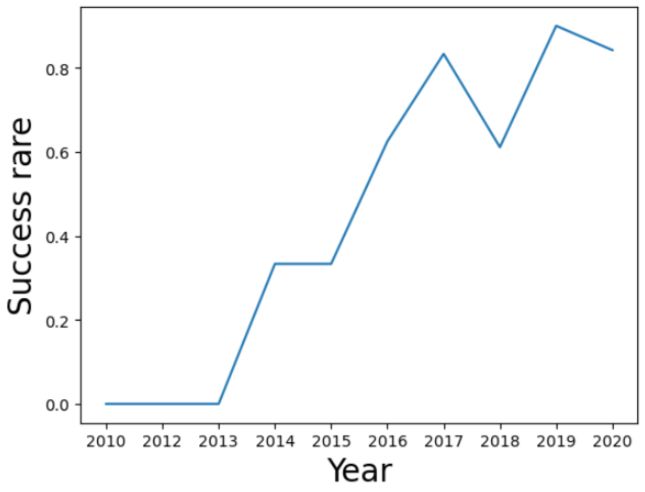
The payload mass between 2000 and 3000 is affecting the ISS, between 3000 and 7000 is affecting GTO.



https://github.com/klaudiadg/Falcon_9

Launch Success Yearly Trend

Since 2013 there's a increase in success rate, which dropped in 2018 but the trend is still going up.



https://github.com/klaudiadg/Falcon_9

All Launch Site Names

- Using the word DISTINCT in the query means that it will only show Unique values in the Launch_Site column from SpaceX

```
%%sql
select distinct Launch_Site from SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

- Using TOP 5 in the query means that it will only show 5 records from SpaceX and LIKE keyword has a wild card with the words 'CCA%' the percentage in the end suggests that the Launch_Site name must start with KSC.

```
%%sql
select  *
from SPACEXTABLE
    where Launch_Site like ('CCA%')
LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Using the function SUM summates the total in the column PAYLOAD_MASS_KG_The WHERE clause filters the dataset to only perform calculations on Customer NASA (CRS)

```
%%sql
select  sum(PAYLOAD_MASS__KG_)
from SPACEXTABLE
where customer = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

https://github.com/klaudiadg/Falcon_9

Average Payload Mass by F9 v1.1

Using the function AVG works out the average in the column PAYLOAD_MASS_KG_The WHERE clause filters the dataset to only perform calculations on Booster_version F9 v1.1

```
%%sql
select  avg(PAYLOAD_MASS__KG_)
from SPACEXTABLE
where Booster_Version = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

https://github.com/klaudiadg/Falcon_9

First Successful Ground Landing Date

Using the function MIN works out the minimum date in the column DateThe WHERE clause filters the dataset to only perform calculations on Landing_Outcome Success (drone ship)

```
%%sql
select min(Date)
from SPACEXTABLE
where Landing_Outcome = 'Success (ground pad)'
```

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

min(Date)
2015-12-22

https://github.com/klaudiadg/Falcon_9

Successful Drone Ship Landing with Payload between 4000 and 6000

The WHERE clause filters the dataset to Landing_Outcome = Success (drone ship) The AND clause specifies additional filter conditions payload mass between 4000 and 6000

```
%%sql
select distinct Booster_Version
from SPACEXTABLE
where 1=1
and Landing_Outcome = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

Group by and sum of the mission outcome column was used to get the total numer of the outcomes

```
%%sql
select Mission_Outcome, count(Mission_Outcome) as total_number
from SPACEXTABLE
group by Mission_Outcome
```

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

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

https://github.com/klaudiadg/Falcon_9

Boosters Carried Maximum Payload

Using the word DISTINCT in the query means that it will only show Unique values in the Booster_Version column from SpaceX
GROUP BY puts the list in order set to a certain condition.
DESC means its arranging the dataset into descending order

```
%%sql
select distinct Booster_Version, max(PAYLOAD_MASS__KG_)
from SPACEXTABLE
group by Booster_Version
ORDER BY MAX(PAYLOAD_MASS__KG_)  DESC
```

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

Booster_Version	max(PAYLOAD_MASS__KG_)
F9 B5 B1060.3	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1056.4	15600
F9 B5 B1051.6	15600
F9 B5 B1051.4	15600
F9 B5 B1051.3	15600

2015 Launch Records

Month numer was dirived from Date, then the filtering was done on 2015 year and Failure (dron ship) for landing outcome

```
%%sql
select substr(Date,6,2) as month_name , landing_outcome, Booster_Version, Launch_Site
from SPACEXTABLE
where substr(Date,0,5) = '2015'
and landing_outcome = 'Failure (drone ship)'
```

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

month_name	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

Function COUNT counts records in column
WHERE filters data on date between the seleced ones

```
%%sql
select Landing_Outcome , count(Landing_Outcome) as count_of_lading_outcome, DENSE_RANK() OVER(ORDER BY  count(Landing_Outcome) de
from SPACEXTABLE
where Date between '2010-06-04' and '2017,03,20'
group by (Landing_Outcome)
order by count_of_lading_outcome desc
```

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

Landing_Outcome	count_of_lading_outcome	Rank
No attempt	9	1
Failure (drone ship)	5	2
Success (drone ship)	4	3
Controlled (ocean)	3	4
Uncontrolled (ocean)	2	5
Success (ground pad)	2	5
Failure (parachute)	2	5
Precluded (drone ship)	1	6

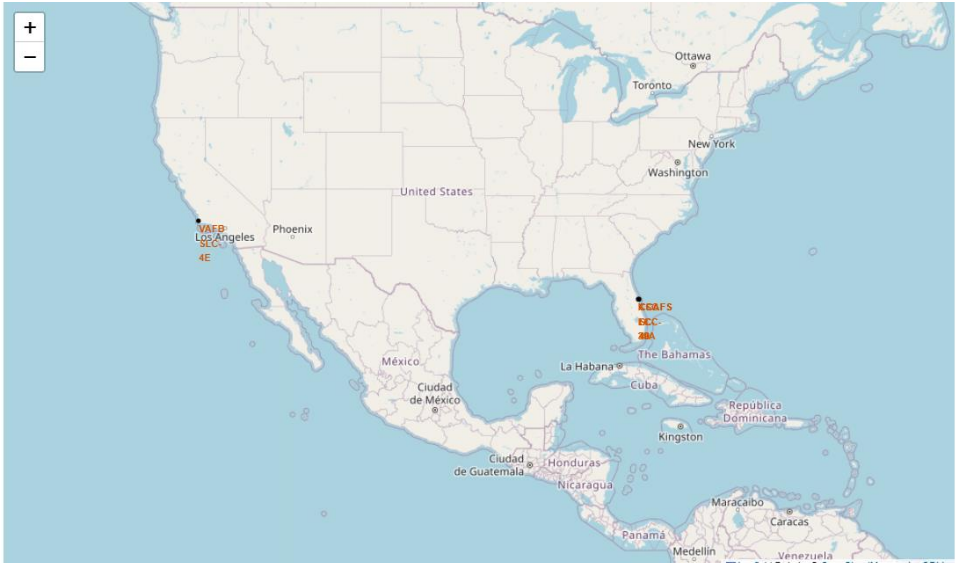
https://github.com/klaudiadg/Falcon_9

Section 3

Launch Sites Proximities Analysis

All launch sites global map markers

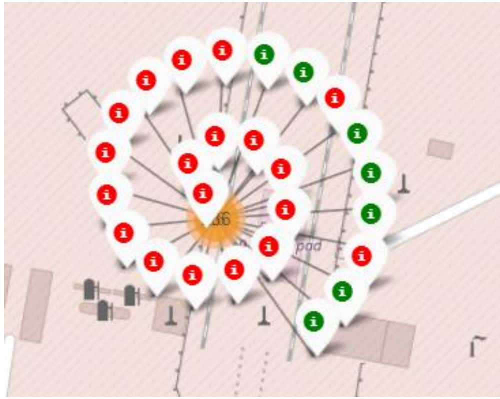
All launches in USA



https://github.com/klaudiadg/Falcon_9

Color labeled Launch Outcomes

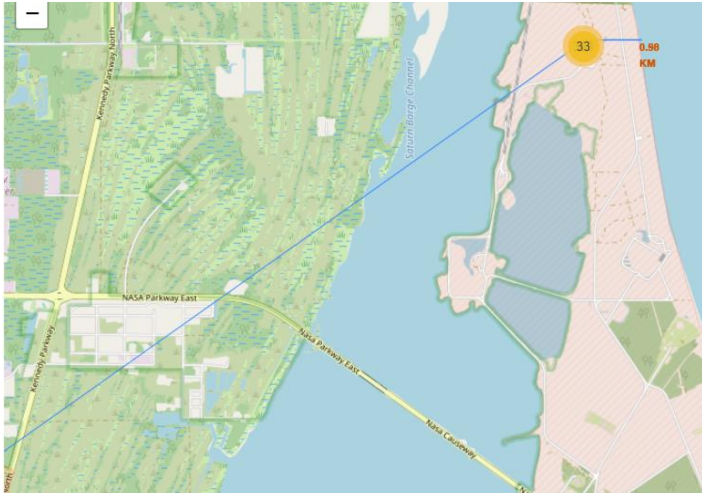
Green markers present successsful, red – failure launches



https://github.com/klaudiadg/Falcon_9

Launch Sites to Proximites locations

Launch Sites were close to coastline, further from the railways and highways



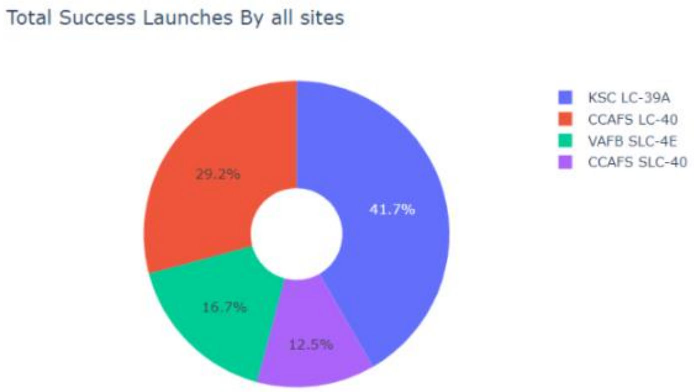
https://github.com/klaudiadg/Falcon_9

Section 5

Build a Dashboard with Plotly Dash

Lunch Success

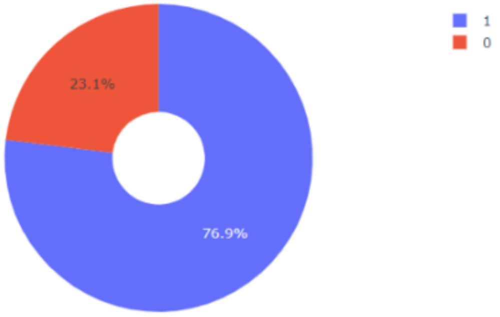
KSC has the highest core, next one is CCASF



https://github.com/klaudiadg/Falcon_9

Launch Site with highest score

KSC LC has highest core with 76.9% success rate



https://github.com/klaudiadg/Falcon_9

Payload vs Launch Outcome

Payload 0-5000 kg

Payload 6000-10000kg

Rates for low weighted payloads is higher than the heavy weighted payloads

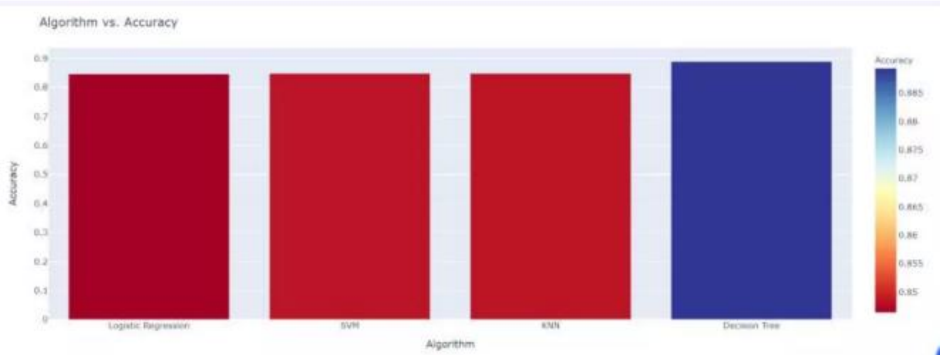
https://github.com/klaudiadg/Falcon_9

Section 6

Predictive Analysis (Classification)

Classification Accuracy

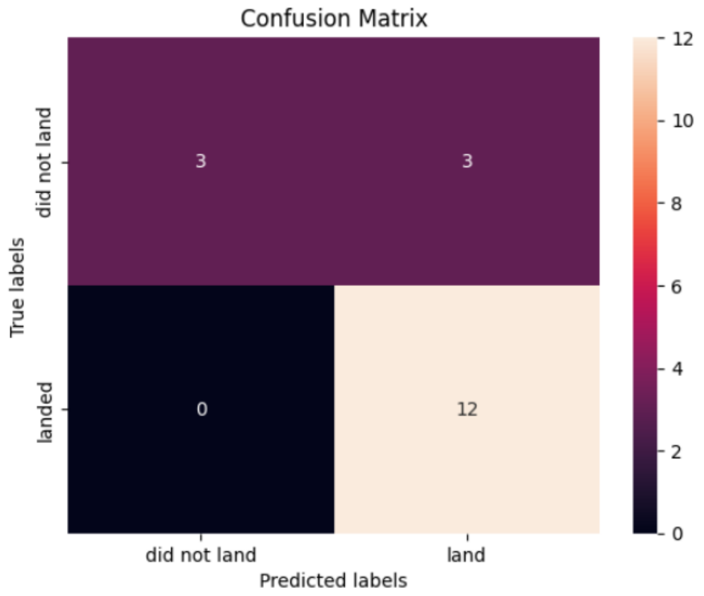
The decision tree has the highest accuracy at 0.8



https://github.com/klaudiadg/Falcon_9

Confusion Matrix

Examining the confusion matrix, we see that Tree can distinguish between the different classes. We see that the major problem is false positives



https://github.com/klaudiadg/Falcon_9

Conclusions

- The highest core site was KSC LC-39A
- The payoad of 0-5000 kg was more diverese than higher payloads
- Decision Tree was optimal model with accuracy 0.89
- The calculations of launches sites distance to it's proximities was perfomed.

https://github.com/klaudiadg/Falcon_9

Appendix

All code can be found on my git hub:
https://github.com/klaudiadg/Falcon_9

Thank you!