IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

IBM Capstone Project - SpaceX

Deepak Kumar Bharti

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - Perform exploratory data analysis (EDA) using visualization and SQL
  - Perform interactive visual analytics using Folium and Plotly Dash
  - Perform predictive analysis using classification models
- Summary of all results
  - Results from exploratory data analysis
  - Interactive analytics demo with screenshots
  - Predictive analysis results

Introduction

- Project background and context

  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.

- Problems you want to find answers

  This report aims to accurately predict the likelihood of the first stage rocket landing successfully as a proxy for the cost of a launch.

  - We will try to predict if the Falcon 9 first stage will land successfully.
  - Explore the factors that influences landing of rocket.
  - Measure the effect of relationship between certain variables
  - Determine which conditions should SpaceX have to achieve the best results
  - What are the factors to ensure the successful landing each time.

Section 1

Methodology

Methodology

- Data collection methodology
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and dropping irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Plotting : Scatter Graphs, Bar Graphs to show relationships between variables to show patterns of data.
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune, evaluate classification models like Logistic Regression, Decision Tree etc.

Data Collection

API

Acquired historical launch data from Open Source REST API for SpaceX

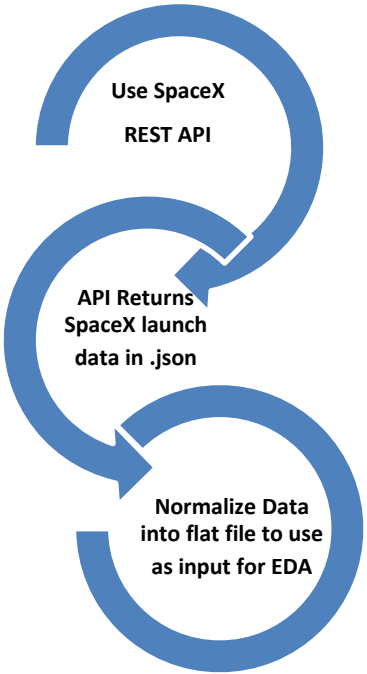
- Requested and parsed the SpaceX launch data using the GET request
- Filtered the dataframe to only include Falcon 9 launches
- Replaced missing payload mass values from classified missions with mean

Web Scraping

Acquired historical launch data from Wikipedia page 'List of Falcon 9 and Falcon Heavy Launches’

- Requested the Falcon9 Launch Wiki page from its Wikipedia URL
- Extracted all column/variable names from the HTML table header
- Parsed the table and converted it into a Pandas data frame

SpaceX API



Get Response from API

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

Convert to .json file

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

Assign list to dictionary for columns then Dataframe

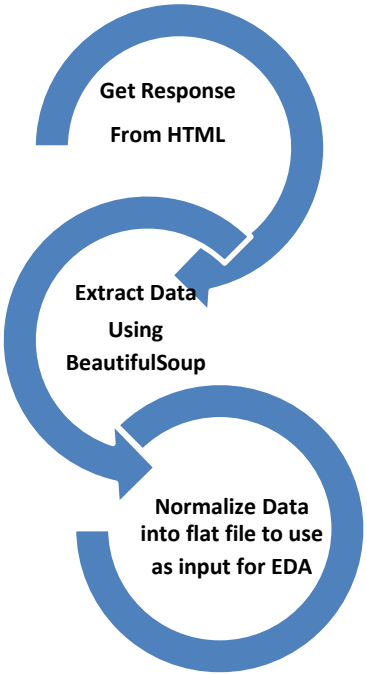
```
launch_dict = {'FlightNumber': list(data['flight_number']), 'Date': list(data['date']), 'BoosterVersion':BoosterVersion,
  'PayloadMass':PayloadMass, 'Orbit':Orbit, 'LaunchSite':LaunchSite, 'Outcome':Outcome, 'Flights':Flights, 'GridFins':GridFins,
  'Reused':Reused, 'Legs':Legs, 'LandingPad':LandingPad,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

Filter Dataframe And export to flat file

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

GitHub URL for Notebook

Data Collection Web Scraping



Get Response from HTML

```
response=requests.get(static_url)
soup=BeautifulSoup(response.text,'html.parser')
```

Get column Names

```
for row in first_launch_table.find_all('th'):
 col = extract_column_from_header(row)
 if col is not None and len(col) > 0:
 column_names.append(col)
```

Assign list to dictionary for columns then Dataframe

```
launch_dict=dict.fromkeys(column_names)
# Remove an irrelvant columndellaunch_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 columnslaunch_dict['Version
 Booster']=[]launch_dict['Booster landing']=[]launch_dict['Date']=[]launch_dict['Time']=[]
```

Filter Dataframe And export to flat file

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

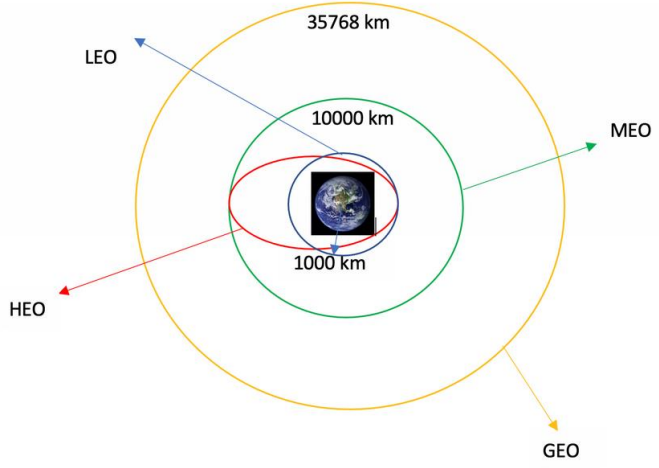
GitHub URL for Notebook

Data Wrangling

Introduction

- Explored data to determine the label for training supervised models
  - Calculated the number of launches on each site
  - Calculated the number and occurrence of each orbit
  - Calculated the number and occurrence of mission outcome per orbit type
- Created a landing outcome training label from 'Outcome' colum
  - Training label: 'Class'
  - Class = 0; first stage booster did not land successfully
    - None None; not attempted
    - None ASDS; unable to be attempted due to launch failure
    - False ASDS; drone ship landing failed
    - False Ocean; ocean landing failed
    - False RTLS; ground pad landing failed
  - Class = 1; first stage booster landed successfully
    - True ASDS; drone ship landing succeeded
    - True RTLS; ground pad landing succeeded
    - True Ocean; ocean landing succeeded

Each launch aims to an dedicated orbit, and here are some common orbit types:



GitHub URL for Notebook

EDA with Data Visualization

- Read the dataset into a Pandas dataframe
- Used Matplotlib and Seaborn visualization libraries to plot
  - FlightNumber x PayloadMass †
  - FlightNumber x LaunchSite †
  - Payload x LaunchSite †
  - Orbit type x Success rate
  - FlightNumber x Orbit type †
  - Payload x Orbit type †
  - Year x Success rate

† = with Class overlayed (1st stage booster landing outcome)

GitHub URL for Notebook

EDA with SQL

Performed following SQL queries to gather information about the 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
- Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

GitHub URL For Notebook

Building an interactive map with Folium

Launch Sites Location Analysis

- Used Python interactive mapping library called Folium
- Marked all launch sites on a map
- Marked the successful/failed launches for each site on map
- Calculated the distances between a launch site to its proximities
    - Railways
    - Highways
    - Coastlines
    - Cities

Launch Records Dashboard

- Used Python interactive dashboarding library called Plotly Dash to enable stakeholders to explore and manipulate data in an interactive and real-time way
- Pie chart showing success rate
    - Colour coded by launch site
- Scatter chart showing payload mass vs. landing outcome
    - Colour coded by booster version
    - With range slider for limiting payload amount
- Drop-down menu to choose between all sites and individual launch sites

GitHub link for Notebook

Build an interactive dashboard with Plotly Dash

The dashboard is built with Dash web framework.

Graphs

- Pie Chart showing the total launches by a certain site/all sites
- Display relative proportions of multiple classes of data.
- Size of the circle can be made proportional to the total quantity it represents.

Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster

- It shows the relationship between two variables.
- It is the best method to show you a non-linear pattern.
- The range of data flow, i.e. maximum and minimum value, can be determined.
- Observation and reading are straightforward.

Website Link

GitHub URL for Notebook

Predictive analysis (Classification)

Building Model

- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples we have
- Decide which type of machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCV objects 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

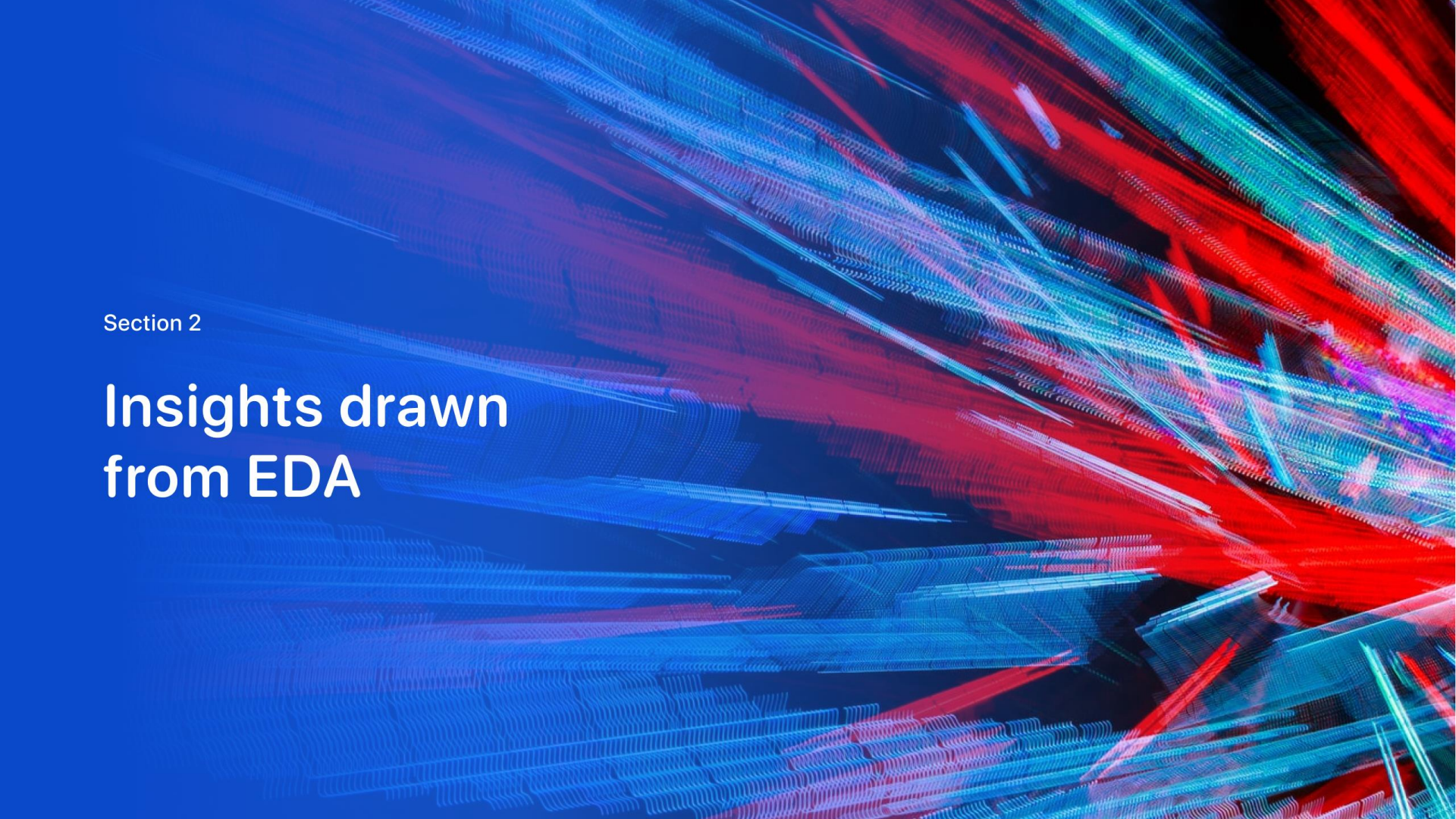
Finding the best performing Model

- 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.

GitHub URL for Notebook

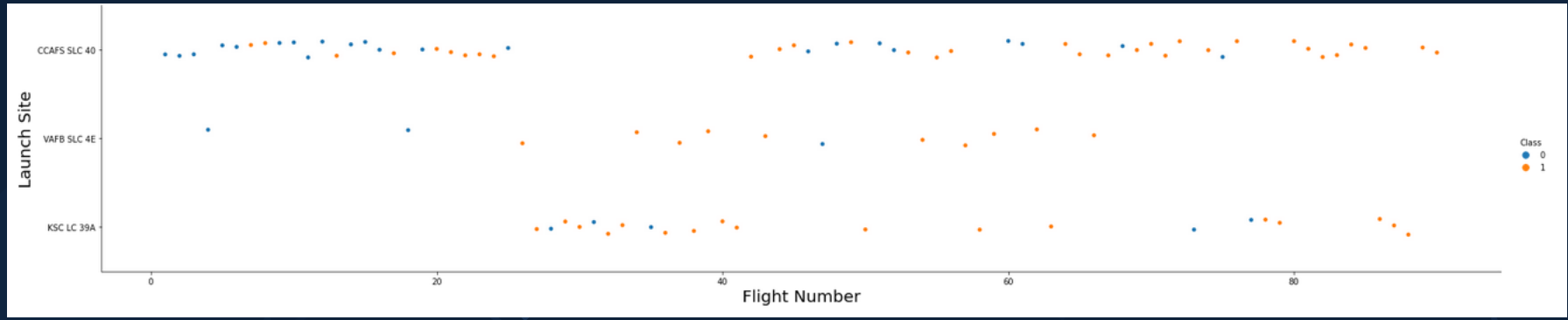
Results

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



Section 2

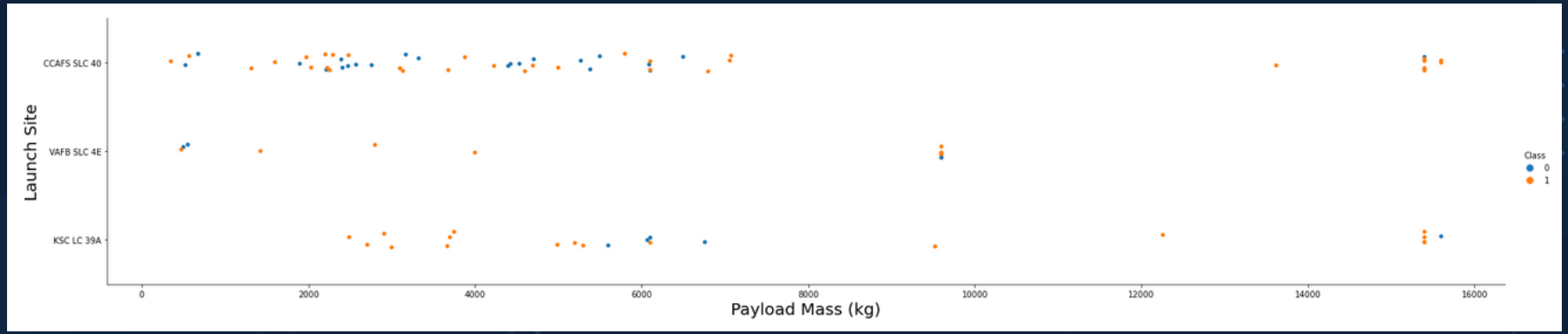
Insights drawn from EDA



Flight Number vs. Launch Site

Observations:

- More number of flights at a launch site the greater the success rate at a launch site.
- CCAFS SLC 40 appears to have been where most of the early 1st stage landing failures took place.



Payload Mass vs. Launch Site

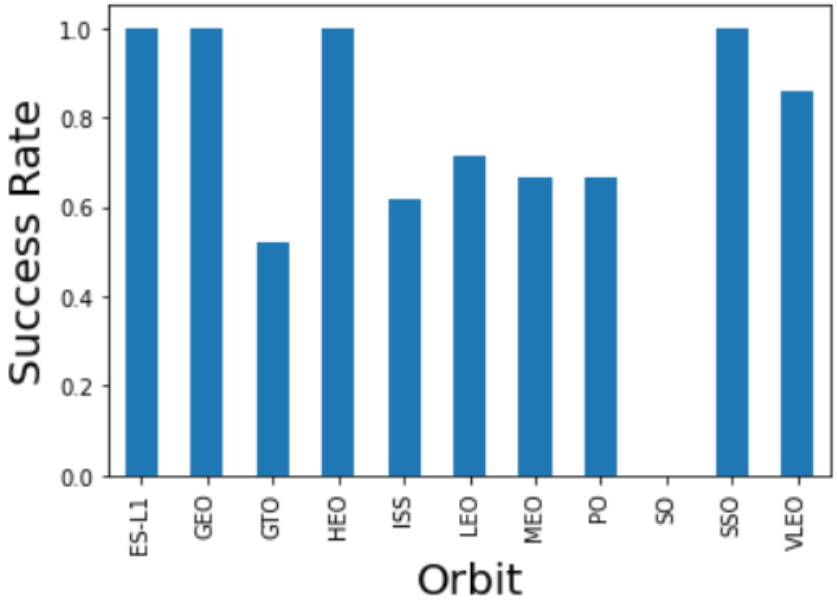
Observations:

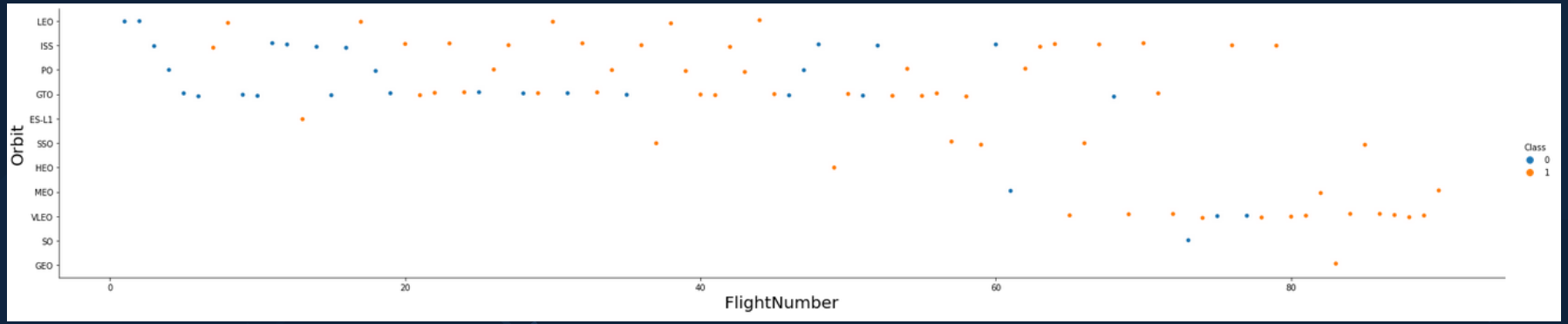
- Greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket.
- No clear pattern can be seen using this visualization to decide if the Launch Site is dependant on Pay Load Mass for a success launch.

Orbit vs. Success Rate

Observations:

- All orbit types except 'SO' have had successful 1st stage landings
- Orbit ES-L1, GEO, HEO, SSO has the best Success rate

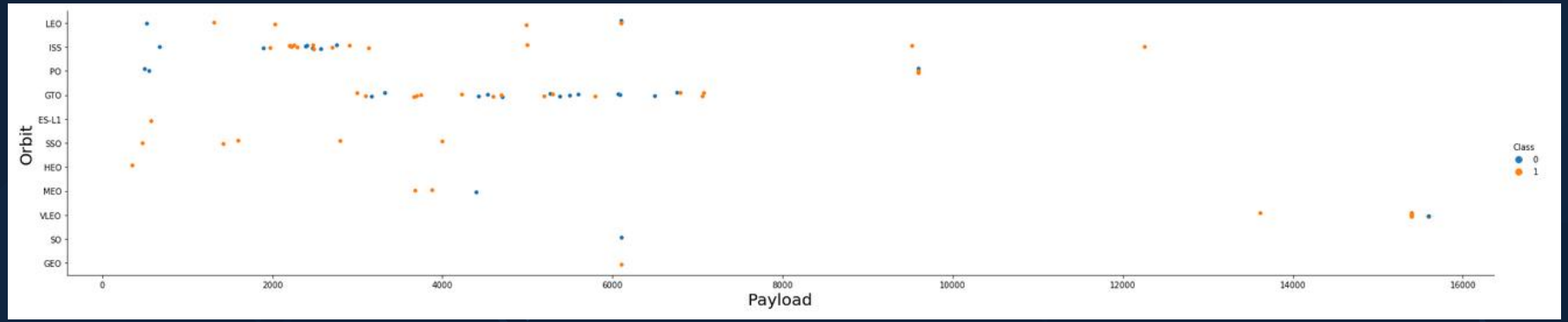




Flight Number vs. Orbit Type

Observations:

- Flight number positively correlated with 1st stage recovery for all orbit types
- We can see that LEO orbit's success is correlated with number of flights, on the other there seems to be no relation between flight number and GTO orbit



Payload vs. Orbit Type

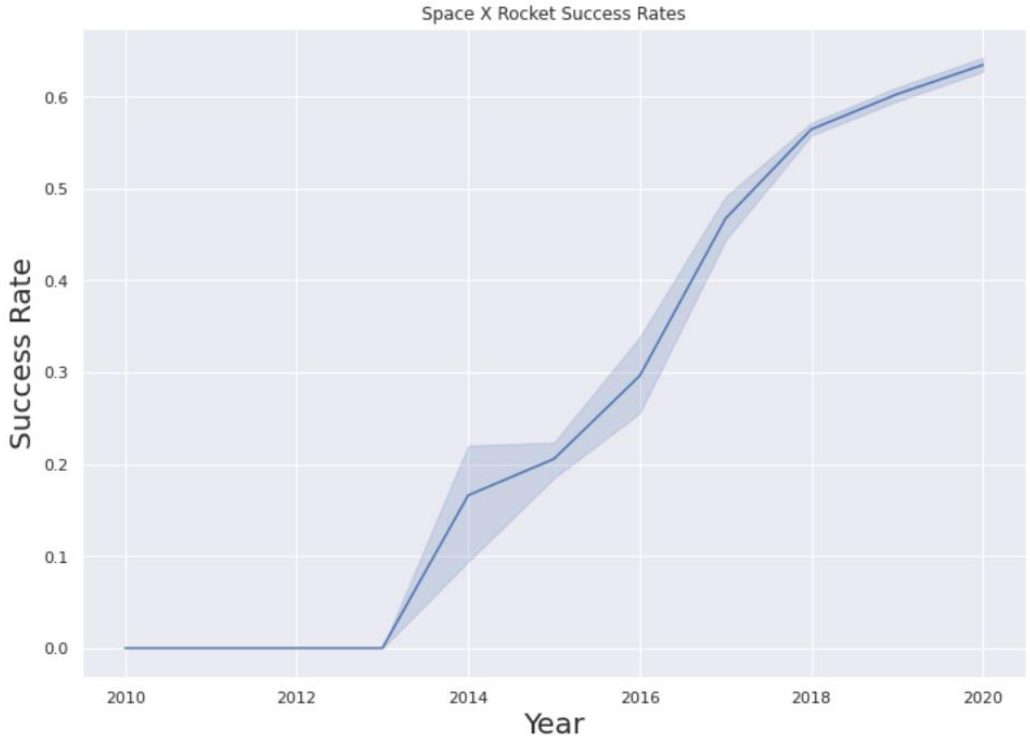
Observations:

- We can observe heavier payloads have a negative influence on GTO orbits and positive influence on ISS orbits

Launch Success Yearly Trend

Observations:

- Success rate trending positively on a yearly basis since 2013



All Launch Site Names

Names of unique launch sites

Using DISTINCT function in the SQL query we can get unique values in the LAUNCH_SITE column from table SPACEXTBL

```
1  SELECT DISTINCT LAUNCH_SITE
2  FROM SPACEXTBL;
```

SELECT DISTIN... Run time: 0.005 s

Result set 1

Find

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

Launch Site Names Begin with 'CCA'

5 records where launch sites name begin with `CCA`

Using LIKE keyword and LIMIT function in the SQL query we can get 5 LAUNCH_SITE names that begin with ‘CCA’ from table SPACEXTBL

```
1  SELECT LAUNCH_SITE
2  FROM SPACEXTBL
3  WHERE LAUNCH_SITE LIKE 'CCA%'
4  LIMIT 5;
```

SELECT LAUNC... Run time: 0.017 s

Result set 1 Find

LAUNCH_SITE
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

Total payload carried by boosters from NASA

Using SUM function in the SQL query we can get total payload mass from table SPACEXTBL

```
1  SELECT SUM(PAYLOAD_MASS__KG_) as Total_Payload
2  FROM SPACEXTBL
3  WHERE Customer = 'NASA (CRS)';
```

SELECT SUM(PA... Run time: 0.007 s

Result set 1 Find

TOTAL_PAYLOAD
45596

Average Payload Mass by F9 v1.1

Average payload mass carried by booster version F9 v1.1

Using AVG function we can calculate average payload mass and with WHERE clause we can filter Booster_version F9 v1.1 from table SPACEXTBL

```
1  SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOD_MASS
2  FROM SPACEXTBL
3  WHERE Booster_Version = 'F9 v1.1';
```

SELECT AVG(PA... Run time: 0.004 s

Result set 1 Find

AVG_PAYLOD_MASS
2928

First Successful Ground Landing Date

Dates of the first successful landing outcome on ground pad

Using MIN function we can get first date and WHERE clause will filter only successful ground landing from table SPACEXTBL

```
1  SELECT MIN(Date) AS DATE_OF_FIRST_SUCCESSFUL_GROUND_LANDING
2  FROM SPACEXTBL
3  WHERE Landing__Outcome = 'Success (ground pad)';
```

SELECT MIN(Da... Run time: 0.016 s

Result set 1 Find

DATE_OF_FIRST_SUCCESSFUL_GROUND_LANDING
2015-12-22

Successful Drone Ship Landing

Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Using WHERE clause we can filter out successful drone landing and respective payload mass from table SPACEXTBL

```
1  SELECT BOOSTER_VERSION
2  FROM SPACEXTBL
3  WHERE LANDING__OUTCOME = 'Success (drone ship)'
4      AND PAYLOAD_MASS__KG_ > 4000
5      AND PAYLOAD_MASS__KG_ < 6000;
```

SELECT BOOSTE... Run time: **0.015 s**

Result set 1 | Find

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

Total Number of Successful and Failure Mission Outcomes

Total Number of Successful and Failure Mission Outcomes

Using COUNT and GROUP BY function we can count total number of success and failures from table SPACEXTBL

```
1 SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER
2 FROM SPACEXTBL
3 GROUP BY MISSION_OUTCOME;
```

SELECT MISSIO... Run time: **0.020 s**

Result set 1 Find

MISSION_OUTCOME	TOTAL_NUMBER
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

Names of the booster which have carried the maximum payload mass

Using a subquery we can get the max payload mass and we can use the result as a filter to get the Booster Version.

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

```
 * ibm_db_sa://lpb97868:***@3883e7e4-18f5-4afe-be8c-fa31c41
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

YEAR function can get the year from Date and we use that in WHERE clause to filter out launces in 2015 and using another filter on Landing outcome we can get the desired result

```
1  SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
2  FROM SPACEXTBL
3  WHERE Landing__Outcome = 'Failure (drone ship)'
4      AND YEAR(DATE) = 2015;
```

SELECT LANDING__OUTCOME, B... Run time: **0.016 s**

Result set 1 Find

LANDING__OUTCOME	BOOSTER_VERSION	LAUNCH_SITE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes

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

Using COUNT and GROUP BY function we can count total number of success and failures from table SPACEXTBL. WHERE clause will be used to filter in the data between given dates.

```
%%sql
SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

```
 * ibm_db_sa://lpb97868:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.b
Done.
```

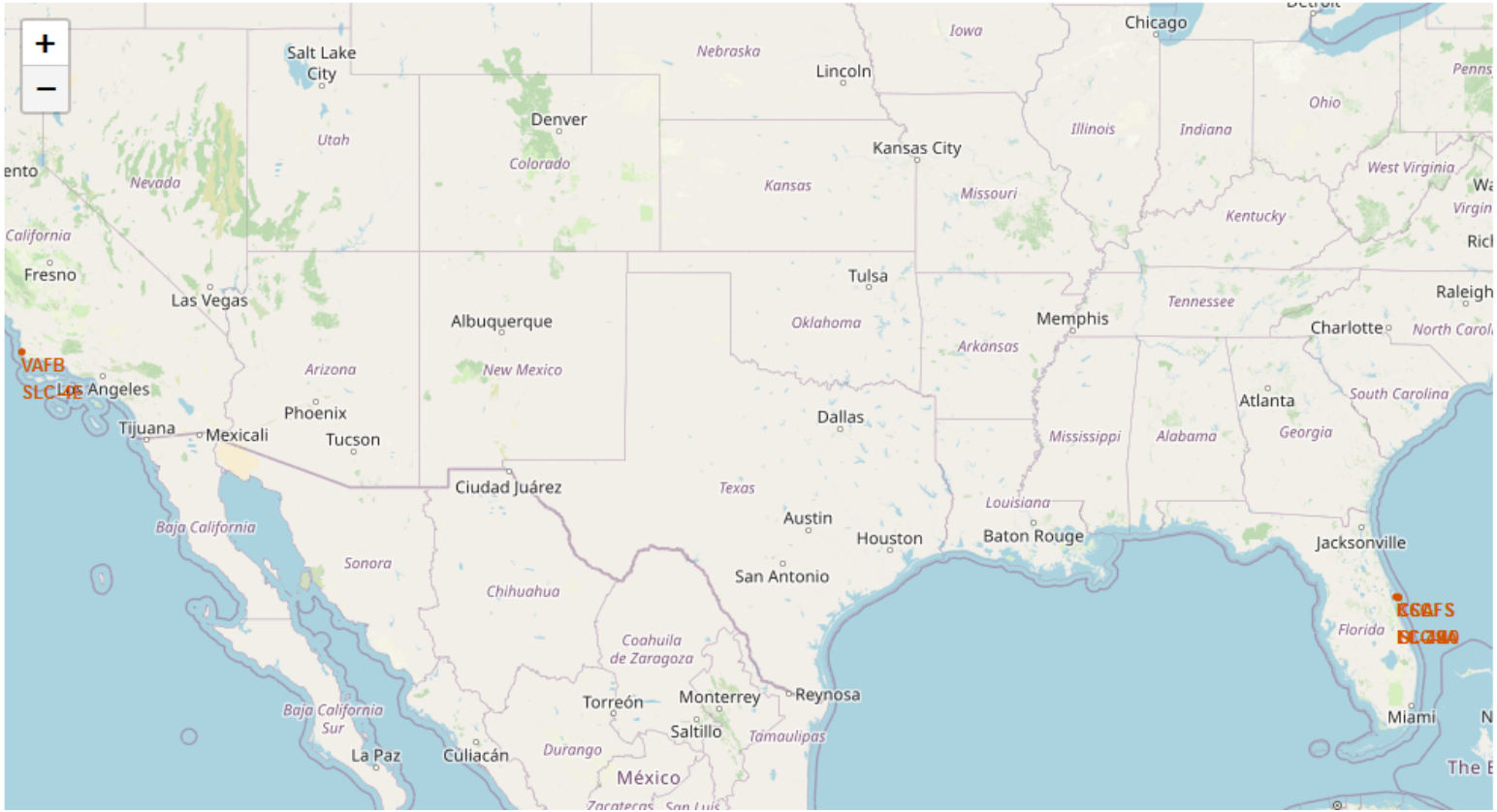
3]:

landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

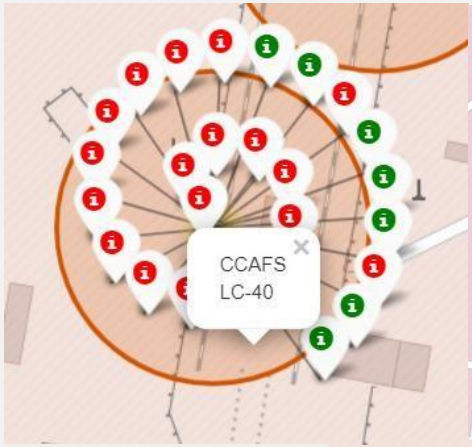
Section 4

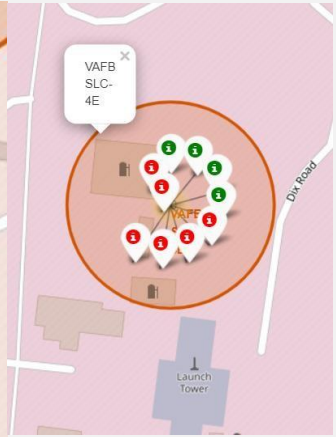
Launch Sites Proximities Analysis

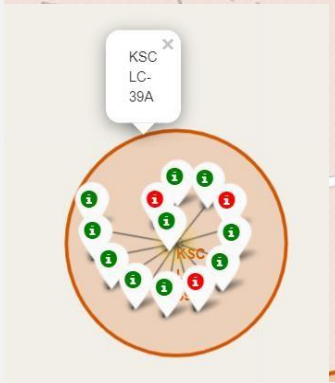
Launch sites' location markers on a global map

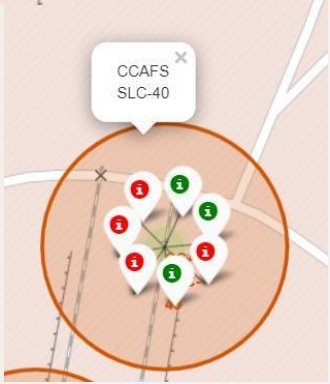


Launch sites' location markers on a global map



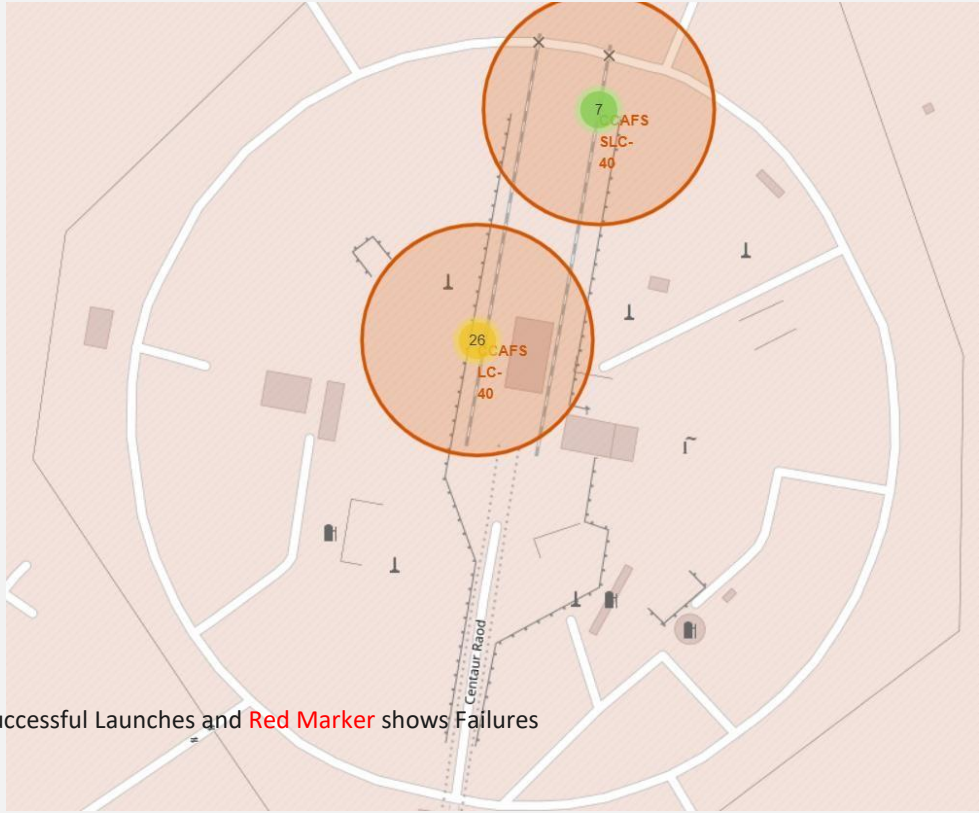




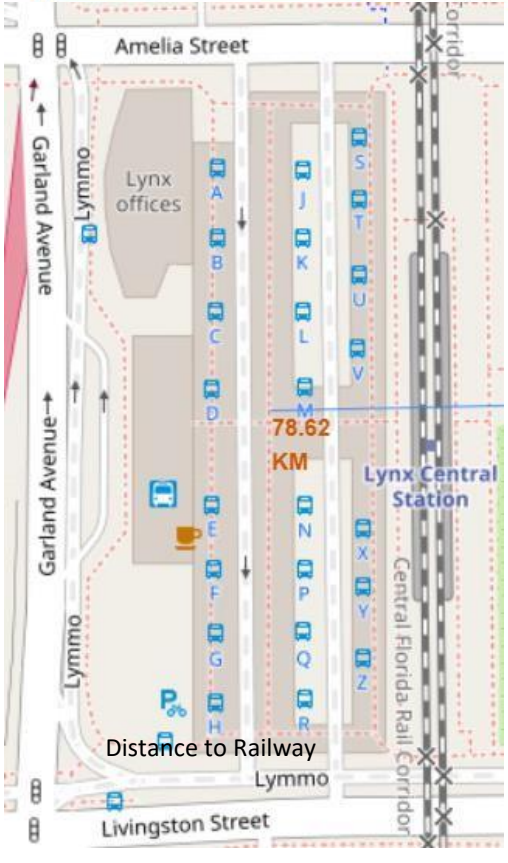


Florida Launch Sites

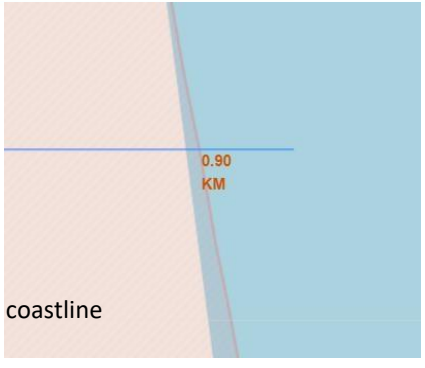
Green Marker shows successful Launches and Red Marker shows Failures



Launch site proximities



Distance to Railway



Distance to coastline

Distance to city

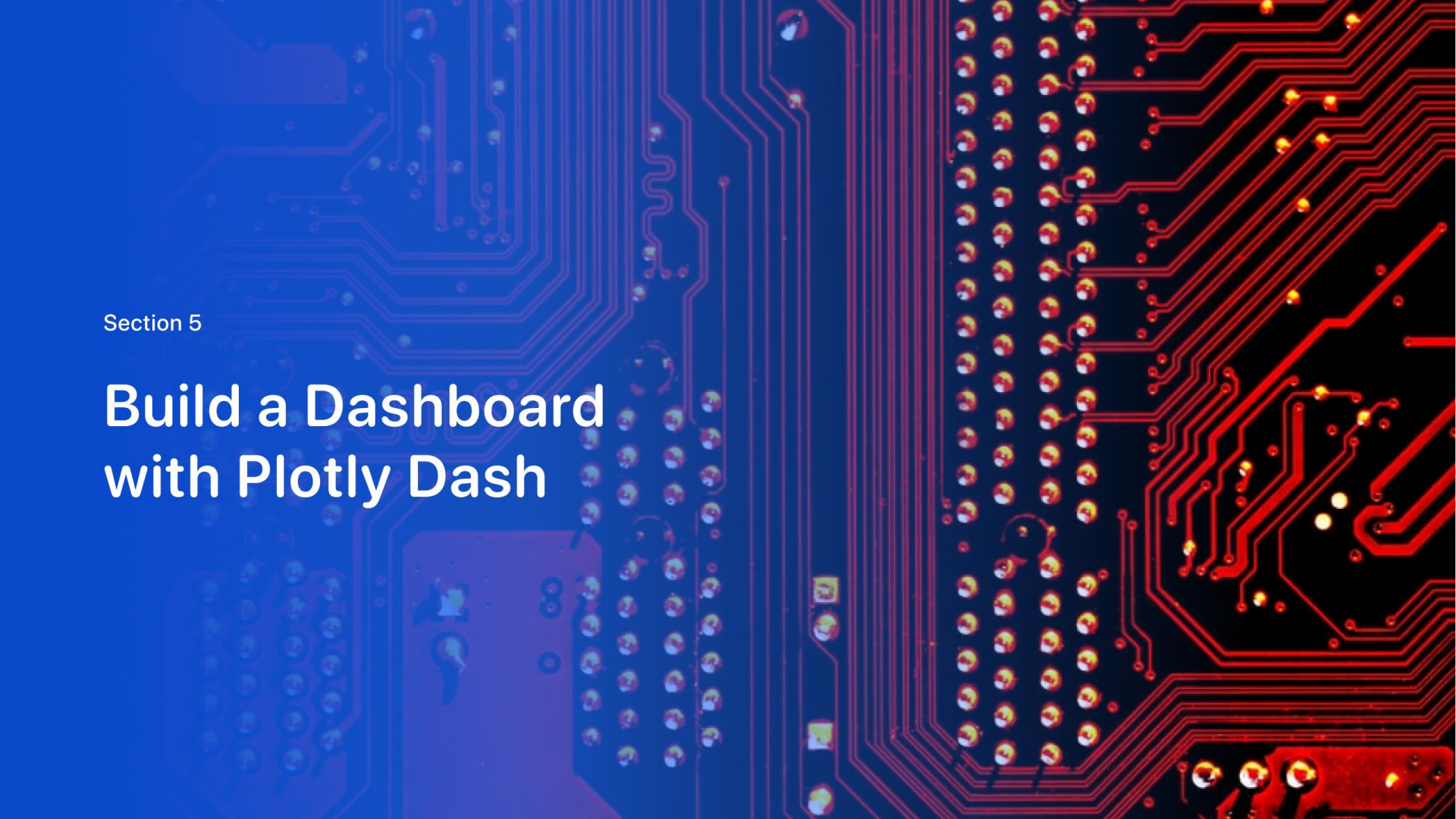
Distance to highway

Are launch sites in close proximity to railway?
No

Are launch sites in close proximity to highway?
No

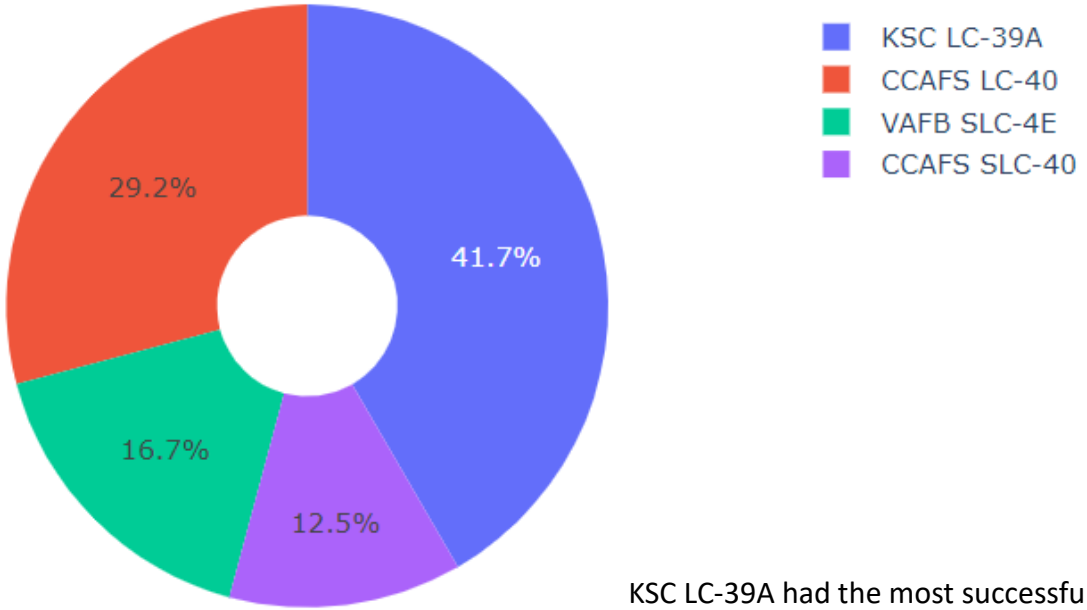
Are launch sites in close proximity to coastline?
Yes

Are launch sites in close proximity to Cities?
Yes


Section 5
Build a Dashboard with Plotly Dash

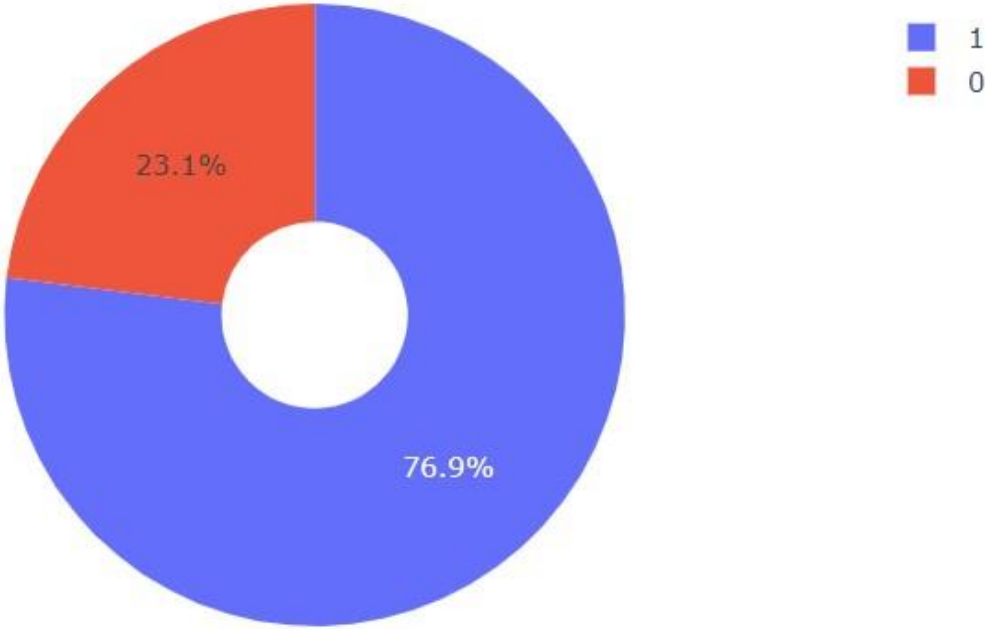
Launch sites' count of all sites

Total Success Launches By all sites



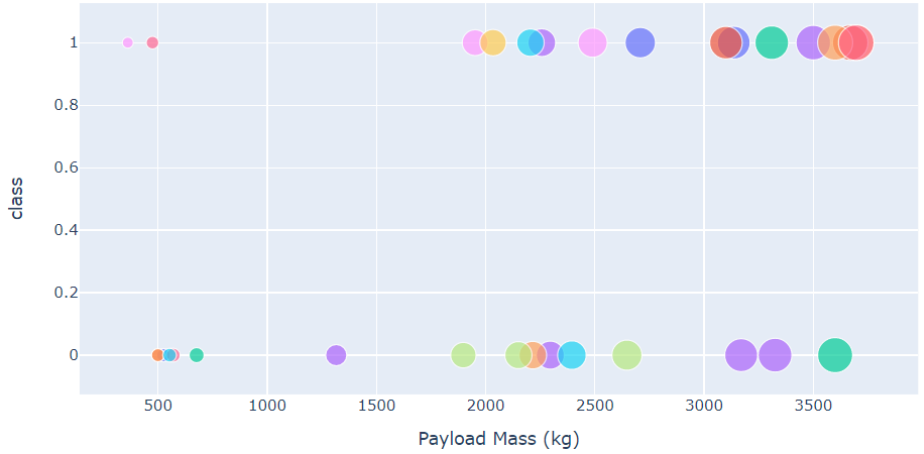
KSC LC-39A had the most successful launches from all the sites

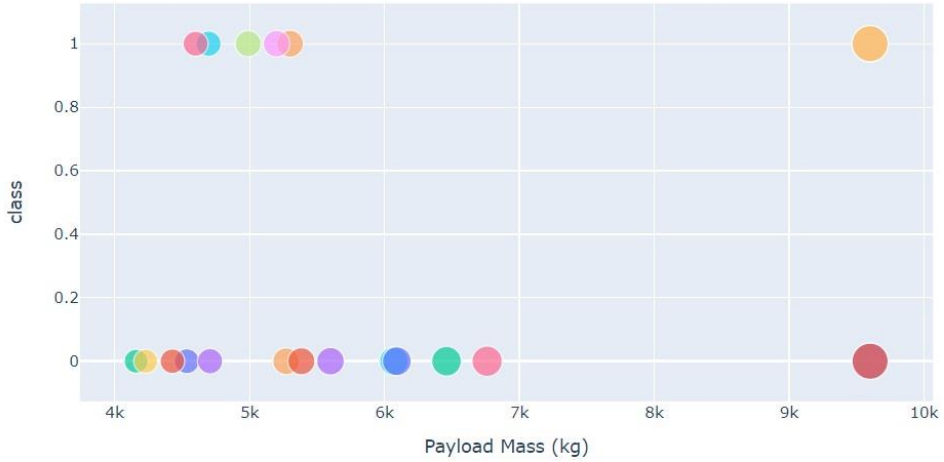
Launch site with highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs. Launch Outcome scatter plot



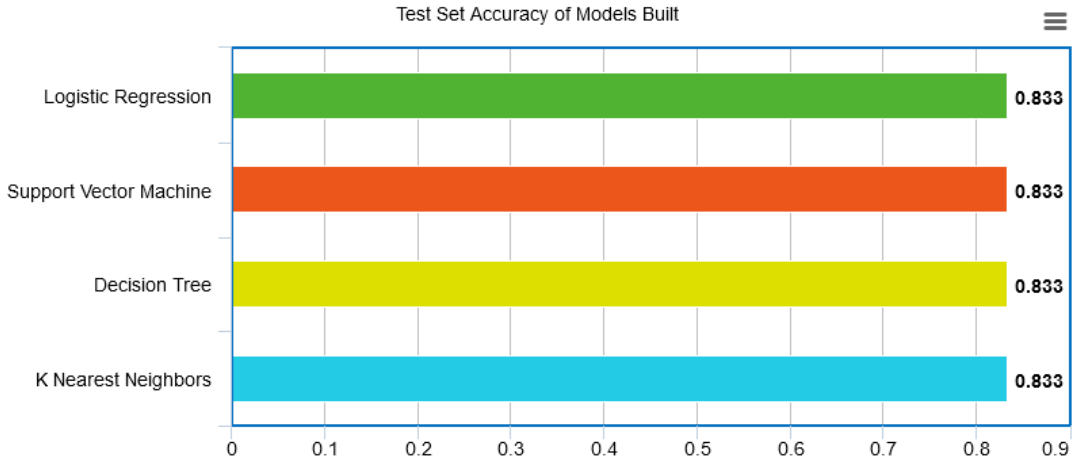


It can be observed that success rates for low weighted payloads is higher than the heavy weighted payloads

Section 6

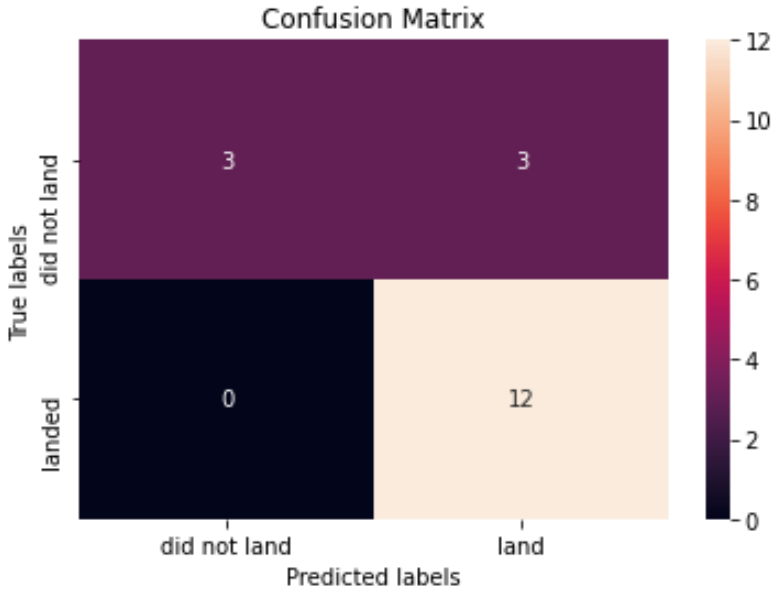
Predictive Analysis (Classification)

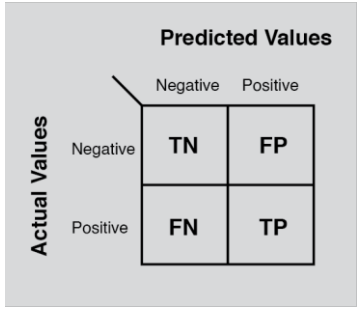
Classification Accuracy



Each of the four models built came back with the same accuracy score, 83.33%

Confusion Matrix





- Confusion matrices of the best performing models (4 way tie) are the same
- The major problem is false positives as evidenced by the models incorrectly predicting the 1st stage booster to land in 3 out of 18 samples in the test set

Conclusions

- Using the models from this report we can predict when SpaceX will successfully land the 1st stage booster with 83.3% accuracy. The Tree Classifier Algorithm seems to be the best for Machine Learning for this dataset.
- SpaceX public statements indicate the 1st stage booster costs upwards of $15 million to build
- Low weighted payloads perform better than the heavier payloads
- With a list price of $62 million per launch, sacrificing the $15+ million 1st stage, would put the SpaceX bid at upwards of $77 million
- Biggest opportunities going forward to make even more informed bids:
  - Freeze the best performing combination of model and hyperparameters and re fit using the whole dataset instead of just the training data
  - Incorporate additional launch data to the dataset and model as it becomes available
  - Subdivide the current model into two models
    - Predict if SpaceX will ATTEMPT to land the 1st stage
    - Predict if SpaceX will SUCCEED in their attempt
  - Create a related model that predicts if SpaceX will launch using a previously flown 1st stage booster
    - Would enable SpaceY to take into account when the SpaceX bid would likely include a discount

Appendix

Acknowledgments

- Thank you to Joseph Santarcangelo at IBM for creating the course and materials
- Thank you to Robert H. at IBM for assisting me with questions and troubleshooting

References

- Interview with Elon Musk where he discloses the 1st stage booster to cost upwards of $15 million
  - https://aviationweek.com/defense-space/space/podcast-interview-spacexs-elon-musk
- Explanation of why you would rebuild your model using the full dataset
  - https://datascience.stackexchange.com/a/33050
- Source of SpaceX's advertised $62 million launch price
  - https://www.spacex.com/vehicles/falcon-9/

Thank You!