IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Robin Bram
3/16/2022

Outline

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

Executive Summary

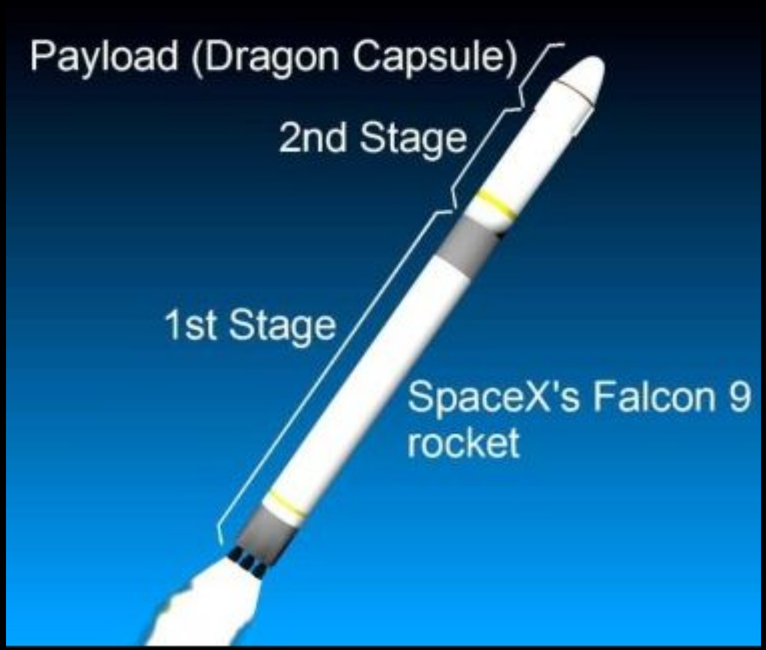
- Background
  - SpaceY is a new company trying to compete in the rocket launch business where launches cost anywhere between $65M to $165M. SpaceY asked us to predict weather the Phase 1 of a rocket could be reused in order to price a launch.
  - In order to create a classification model to predict launch outcomes, data from an API and web scraping were ingested. The data was then cleaned and analyzed using SQL queries, Matplotlib and Seaborn visualizations, Plotly interactive dashboard and Folium geolocation visualization in order to select features for the model. 5 logistic regression prediction models were created using Scikit-learn and the best models were chosen using confusion matrices.

- Findings
  - The SVM, Logistic Regression and KNN can predict the outcome of a launch for SpaceY with 83% accuracy and 80% precision.

- Recommendation
  - Its risky to price the launch into space with the classification model because the model precision is 80%. 3/15 times the model wrongly predicts the launch will be successful when it actually fails. SpaceY will loose $100M on each wrong prediction.
  - Since the cost of Phase 1 depends on the number of times it is reused (average cost), not whether the outcome of one launch is successful, Multiple Linear Regression model would be a better model for SpaceY to use in pricing.
  - By iterating on this process further including cleaning the data and refining the feature selection, we can improve accuracy and precision.

Introduction

- Commercial space travel is here and affordable for everyone!
- SpaceY is a new company and wants to compete with SpaceX on pricing.
- SpaceX advertises Falcon 9 rocket launches on its website for a cost of $62 million dollars. SpaceX costs are lower than other providers because they can recover the first stage of the rocket launch.
- Competitors to SpaceX advertise Falcon 9 rocket launches for a cost of $165 million dollars.
- In this data science capstone, I will use machine learning to determine whether SpaceX will reuse the first phase that will enable SpaceY to price of a launch to travel into Space.



Payload: Purpose of mission. Could include satellites, telescopes, Space Station Modules, Cargo, and even cr spacecraft

Staging: Staging is the combination of several rocket sections, or stages, that fire in a specific order and then detach, so a ship can penetrate Earth’s atmosphere and reach space

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Used get request with API and Beautiful Soup with Web Page to ingest data and create Pandas dataframes.
- Perform data wrangling
  - Used Pandas and Numpy to transform the messy and incomplete data into actionable information for analysis.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Used Scikit-Learn to create LR, SVM, KNN and Decision Tree models and tune with GridSearchCV.

Data Collection

- API
  - Gathered data on historical SpaceX launches using SpaceX API and the Get Command. The data in the API was filtered to only include Falcon 9 launches. Falcon 9s can lift heavier payloads.
- Web Scraping
  - Gathered data on historical SpaceX launches using web scraping.   The Web Scraping provided additional information not found in API including customer, time and booster version. To make the answers consistent with the API, the data was pulled in a pre-selected date range. (121 rows)
- High Level Data Collection Process/Goal
  - Parsed each of the data sets, converted the data into a pandas dataframe and cleaned the data to be used in Exploratory Data Analysis

Data Collection – SpaceX API

- Parsed API data, converted the data into a pandas dataframe and clean the data to be used in EDA

- GitHub URL for the Jupyter notebook: https://github.com/robinbramdata/IBM-Data-Science-Professional-Certificate/blob/master/Data-Collection-API-Lab.ipynb

Request and parse the API data using the GET request

response = requests.get(spacex_url)

Convert the json data into a flat file

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

Filter the DF to only include F9s, pull more API data by using unique IDs in the DF and store the lists in global variables

BoosterVersion = []

Create dictionary using global variables and DF lists columns

launch_dict = {
'Date': list(data['date']),
'BoosterVersion':BoosterVersion, etc

Convert dictionary into a Dataframe

df = pd.DataFrame(launch_dict)

Deal with missing values by replacing them with the mean

data_falcon9['PayloadMass'].fillna(data_falcon9['PayloadMass'].mean(), inplace=True)

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857

Data Collection – Scraping

- Parsed an HTML table from Wikipedia and converted it into a pandas dataframe to be used in EDA.
- GitHub URL for the Jupyter notebook: https://github.com/robinbramdata/IBM-Data-Science-Professional-Certificate/blob/master/Data%20Collection%20with%20Web%20Scraping.ipynb

Request and parse the the HTML Page using the get request

response =requests.get("https:/…")

Create a BeautifulSoup object from the response text content

soup = BeautifulSoup(response.text, 'html.parser')

Extract the column names from the html Table headers

column_names = []
headers = first_launch_table.find_all('th')

Create an empty dictionary with keys from the extracted column names

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

Fill the dictionary using Beautiful Soup and predefined functions.

launch_dict['Booster landing'].append(booster_landing)

Convert dictionary into a Dataframe

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

	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

Data Collection – Data Wrangling

- Using the API data, created the outcome variable (Y) with numeric value 0 or 1 that is necessary for creation of a classification models. The class labels are mapped to 1 for the positive class or outcome and 0 for the negative class or outcome. The models predict the probability that an example belongs to class 1.
- GitHub URL for the Jupyter notebook: https://github.com/robinbramdata/IBM-Data-Science-Professional-Certificate/blob/master/EDA%20Lab%202%20Data%20Wranginling.ipynb

Check % of missing values in attributes of Data Frame

*(df.isnull().sum()/df.count()*100)*

Check the data type of each column

df.types

Check the occurrences by category for the attributes in the Data Frame.

df.LaunchSite.value_counts()

Create a variable to store outcome of each launch 0=bad outcome and 1=Otherwise

landing_class = np.where(df['Outcome'].isin(set(bad_outcomes)), 0, 1)

EDA with Data Visualization

- Created data visualizations with the API data to examine the relationship between variables and outcome in order to select the features that will best predict in a machine learning model
- Machine learning models require all input and output variables to be numeric. Therefore, features selected were encoded to numbers.

- GitHub URL for the Jupyter notebook: https://github.com/robinbramdata/IBM-Data-Science-Professional-Certificate/blob/master/EDA%20with%20Data%20Visualization%20(Pandas%20and%20Matplotlib.ipynb

- Scatterplot
  - Flight Number vs. Payload Mass
  - Payload vs. Launch Site
  - Flight Number vs. Orbit type
  - Payload Vs. Orbit type
- Barplot
  - Success Rate (Mean) vs Orbit type
- Line Plot
  - Launch Success yearly trend

EDA with SQL

- Imported the wikipedia web scraping data into DB2 and queried the data from Jupyter notebook to examine the relationship between variables and outcome in order to select the features that will best predict in a machine learning model
- GitHub URL for the Jupyter notebook: https://github.com/robinbramdata/IBM-Data-Science-Professional-Certificate/blob/master/EDA%20with%20SQL.ipynb

- Distinct, Count and group
  - Names of the unique launch sites in the space mission
  - Landing outcome
  - Launch sites
  - Mission outcomes
- Order, sum and average
  - Launch sites beginning with the string 'CCA'
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
- Where clause and groups
  - The date when the first successful landing outcome in ground pad was achieved.
  - Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Total number of successful and failed mission outcomes
  - Names of the booster_versions which carried the maximum payload mass.
  - Failed landing_outcomes in drone ship, their booster versions, and launch site names for the year 2015
- Rank
  - Ranked 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

Build an Interactive Map with Folium

- Using the API data, mapped successful/failed launches by launch site and the distances of sites to proximities to be able to find some geographical patterns about launch sites in order to select the features that will best predict in a machine learning model

- GitHub URL for the Jupyter notebook: https://github.com/robinbramdata/IBM-Data-Science-Professional-Certificate/blob/master/Data%20Vis%20-%20Folium%201_22_2022.ipynb

Objects added to the map

- Added folium.Circle and folium.Marker for each launch site on the site map
- Added a marker_cluster with green or red markers for every outcome by site and by launch to see which sites have high success rates
- Added a MousePosition on the map to get coordinate for a mouse over a point on the map so you use the coordinates to create a folium.Marker to show the distance
- Drew a PolyLine between a launch site and the selected coastline point

Build a Dashboard with Plotly Dash

- Using the Wikipedia information, added interactions to explore which site, payload range and booster version have the highest successful launches.
- GitHub URL for the Jupyter notebook: https://github.com/robinbramdata/IBM-Data-Science-Professional-Certificate/blob/master/spacex_dash_app.txt

Plots/graphs and interactions added to a dashboard

- Dropdown menu so the user could choose a launch site or all sites.
- A pie chart driven by the drop-down menu that shows launch successes by site as a % of all launches and success rate for each individual site.
- An slider that enables choice of range of the amount of Payload mass he/she wants to analyze.
- A scatter plot driven by the slider that shows how the amount of Payload mass affects the results of the landing by booster version.

Predictive Analysis (Classification)

- Created machine learning classification models using Logistic Regression, SVM, Decision Tree and KNN
- Used GridSearchCV to choose the best Hyperparameters for SVM, Classification Trees and Logistic Regression
- Evaluated the models by calculating the accuracy and generating a confusion matrix.
- GitHub URL for the Jupyter notebook: https://github.com/robinbramdata/IBM-Data-Science-Professional-Certificate/blob/master/Machine%20Learning%20Prediction%20lab.ipynb

Convert the Column Class (Y) into a NumPy array

Y = data['Class'].to_numpy()

Scandardize and transform the training data so it has a distribution of mean value 0 and standard deviation of 1

ttransform = preprocessing.StandardScaler()
transform.fit_transform(X)

Use Sklearn to split data arrays into training and testing data

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
print ('Train set:', X_train.shape, Y_train.shape)

Use Sklearn Logistic Regression and GridSearchCV to create model using best parameters

logreg_cv = GridSearchCV(lr,parameters,cv=10)
logreg_cv.fit(X_train, Y_train)
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)

Run LR Model on test data and calculate the accuracy

print("test set accuracy :",logreg_cv.score(X_test, Y_test))

Generate Confusion that shows the correct and also incorrect values in number count

yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)

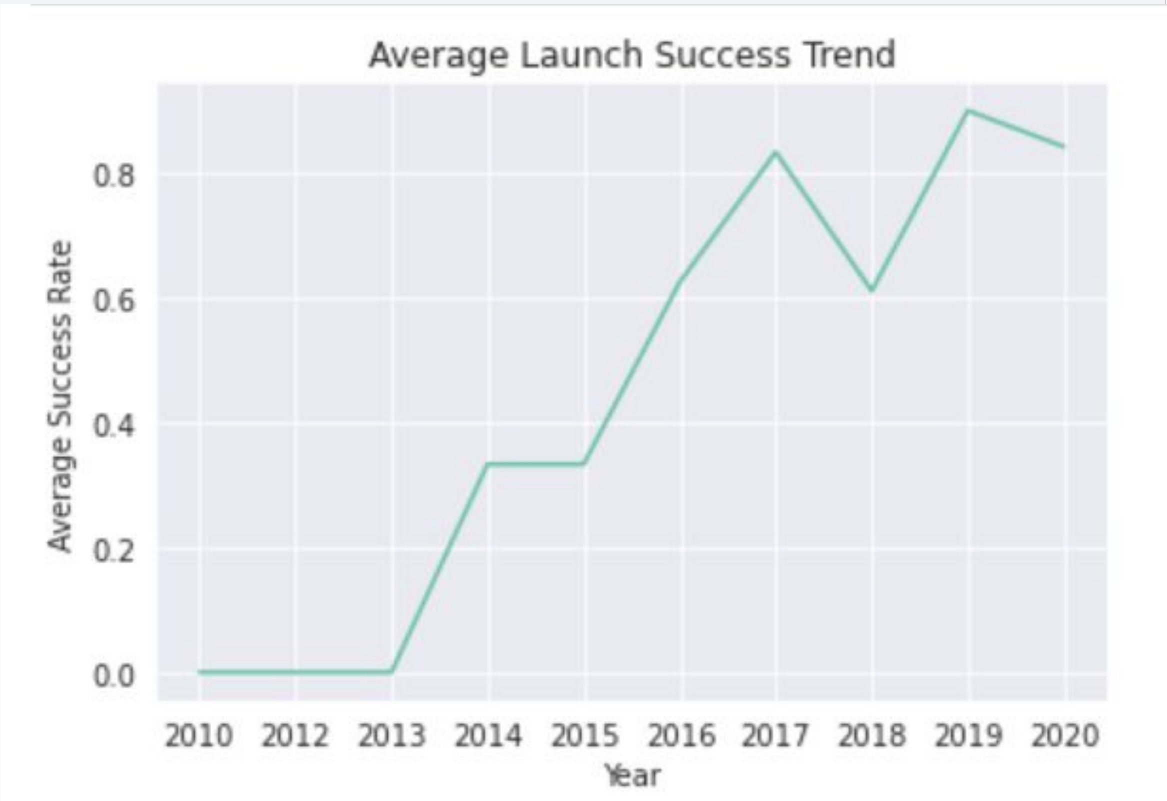
Repeat through machine learning algorithms
-LR
-Support Vector Machine
-Decision Tree
-K Nearest Neighbor

Compare and choose best model

print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))

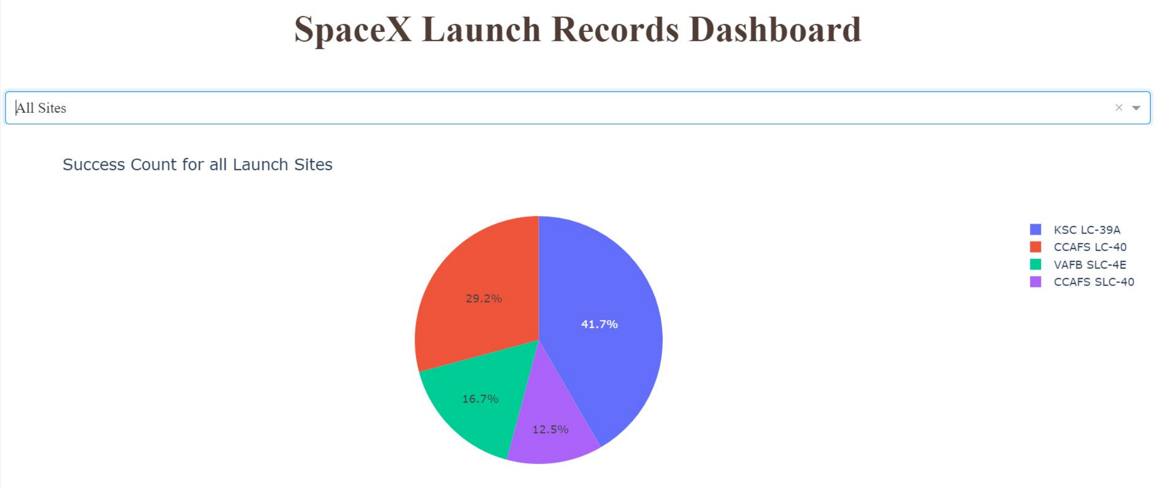
Results - EDA

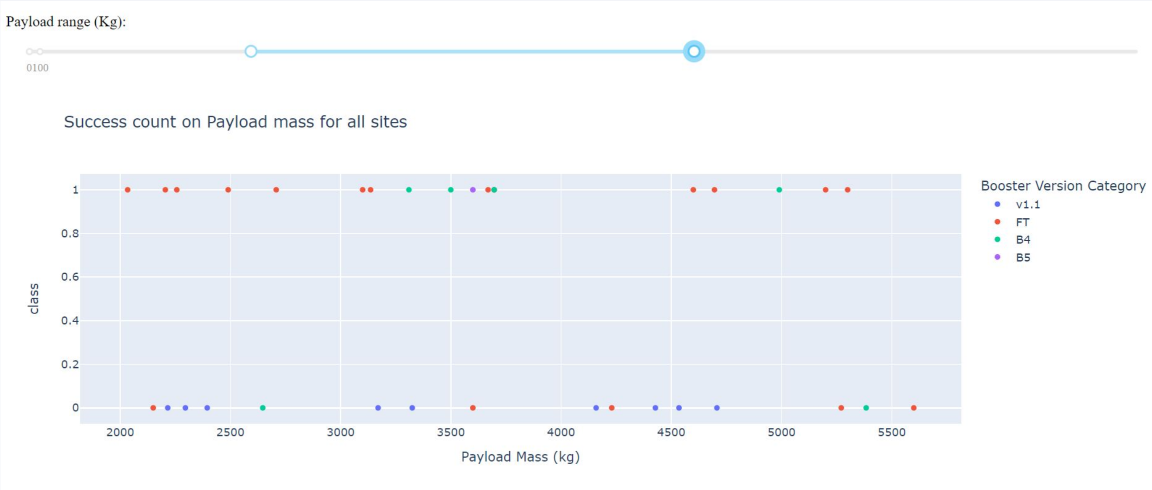
- Outcomes are improving over time represented by flight number.
  - Launches from CCAFS SLC 40 locations increasing over time
  - Launches from ISS and VLEO orbits increasing over time
- Payload Mass is increasing over time.
  - ISS, VLEO and Polar Orbits can successfully handle heavier payloads
- 12 Launches (out of 69) that were successful and had maximum payload of 15,600KG were by Booster Version version F9 B5 B10.
- Drone landings were most successful during 2010-2017.
- Feature Selection: FlightNumber, PayloadMass, Orbit, LaunchSite. Flights, GidFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial



Results - Interactive Analytics

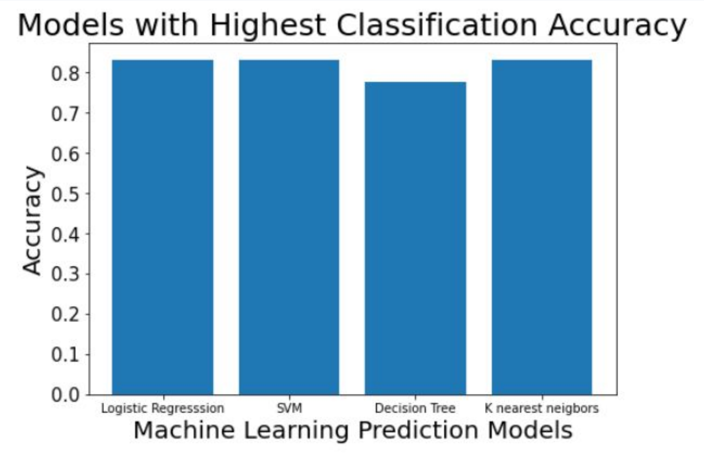
- The pie chart displays the % of success launches by site. We know from folium location analysis that there is only 3 sites and thus the success rate is equal for the KSC-LC39A and CCAFS SLC-40 locations.
- The scatter plot driven by the slider shows booster version FT performs best at higher payloads and V1.1 doesn't perform well at higher payload Masses.

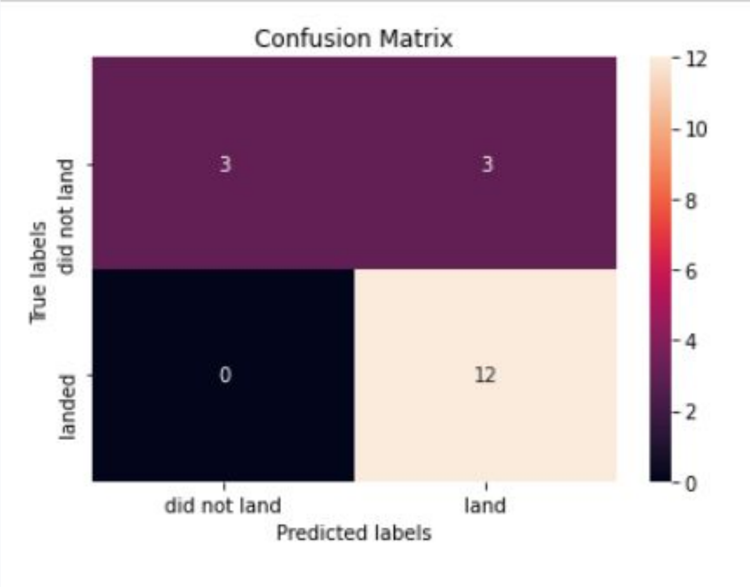




Results - Predictive Analysis

- The Logistic Regression, SVM and K Nearest Neighbor all had accuracy of 83% and precision of 80%.
- There is a risk that our prediction model will underestimate the number that did not land.



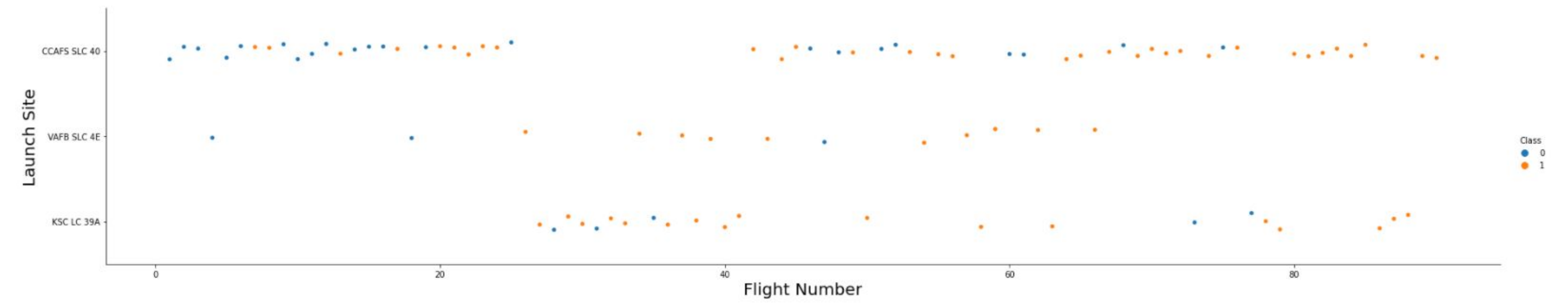


Section 2

Insights drawn from EDA

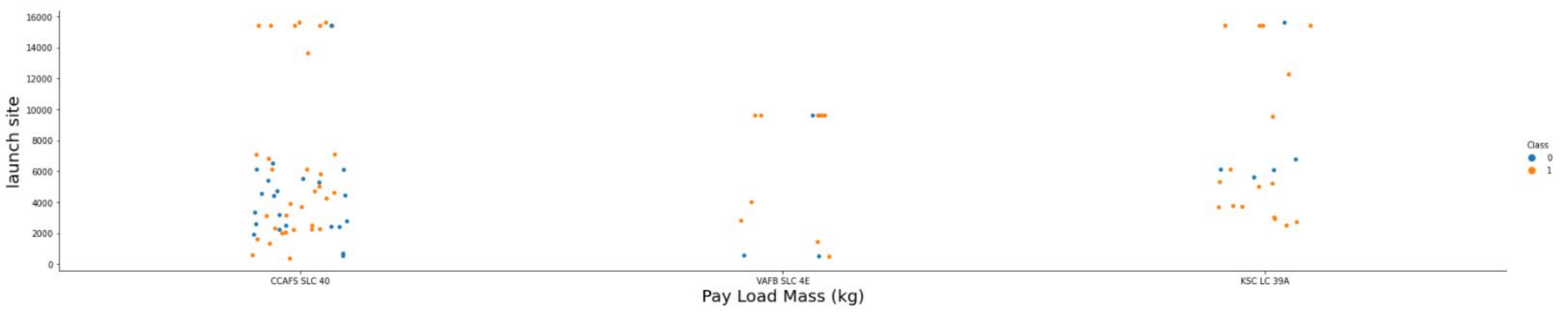
Flight Number vs. Launch Site

- CCAFS-SLC 40 Launch site has had the most launches and as flight attempts have increased, the success rate has gone up to around 9:1
- VAFB-SLC 4E has been consistent in its success rate at about 5:1



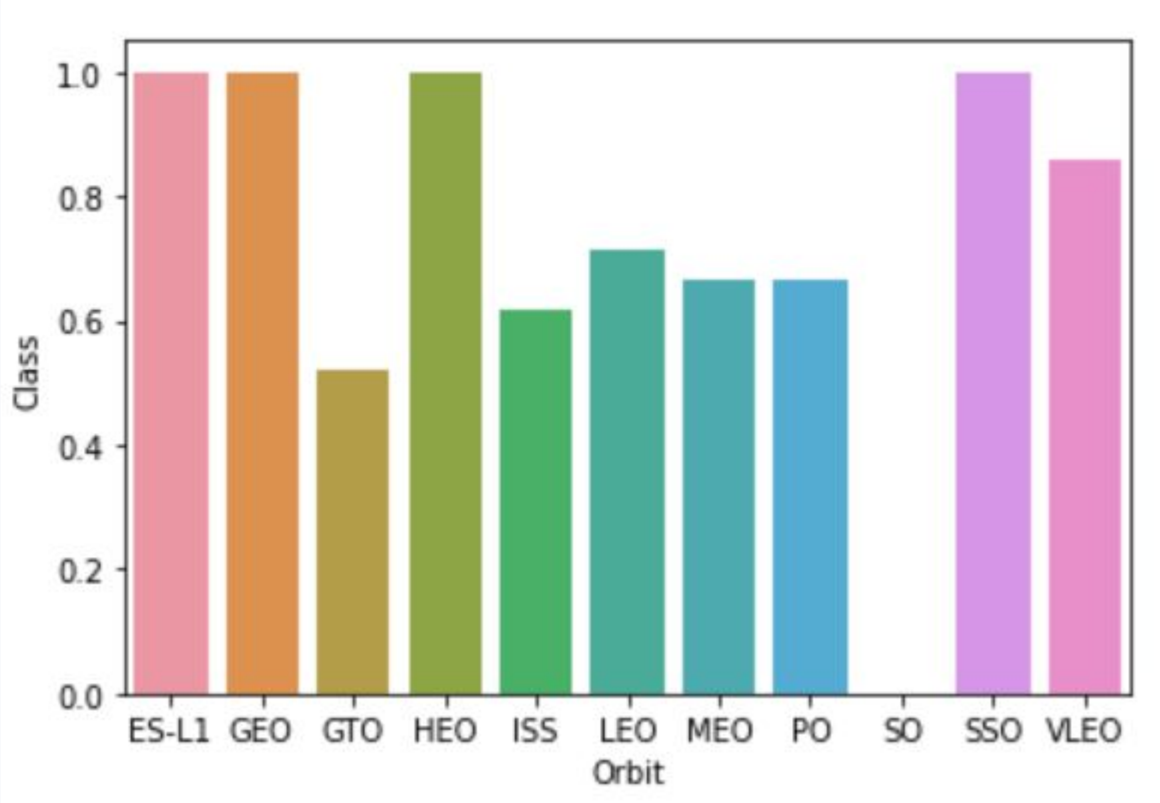
Payload vs. Launch Site

- CCAFS SLC 40 and KSC LC 39A positive outcome increase as Pay Load Mass goes up
- VAFB-SLC 4E launch site has no rockets launched for heavy payload mass(greater than 10,000KG)



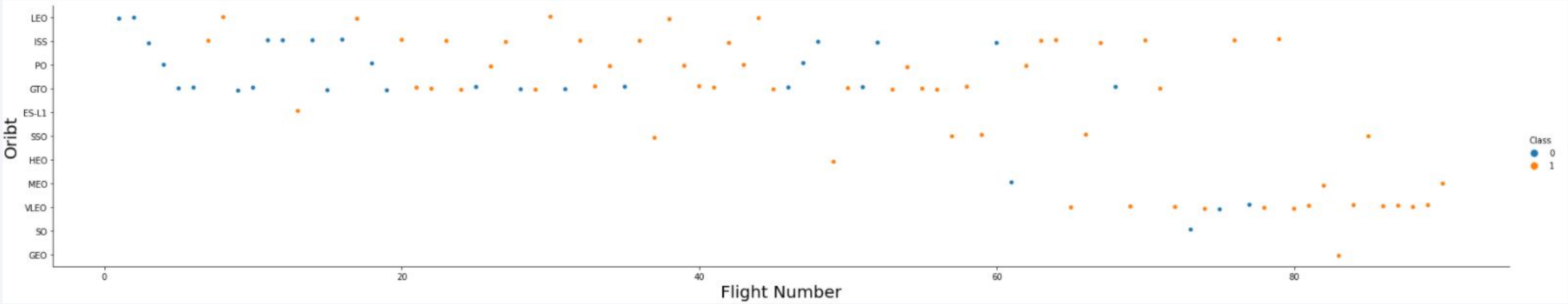
Success Rate vs. Orbit Type

- The success rates over 80% use the ES-LI, GEO, HEO, SSO and VLEO orbits.
- GTO had the lowest success rate at 50%



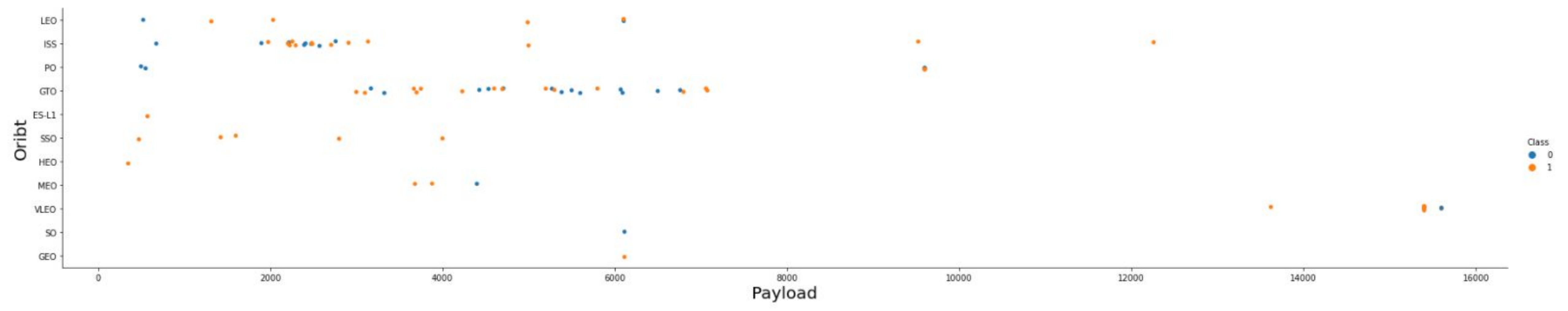
Flight Number vs. Orbit Type

- In the ISS, VLEO and LEO orbit, the success rate improves as the number of flights increase
- There seems to be no relationship between flight number in the GTO orbit.
- All SSO flights were successful.



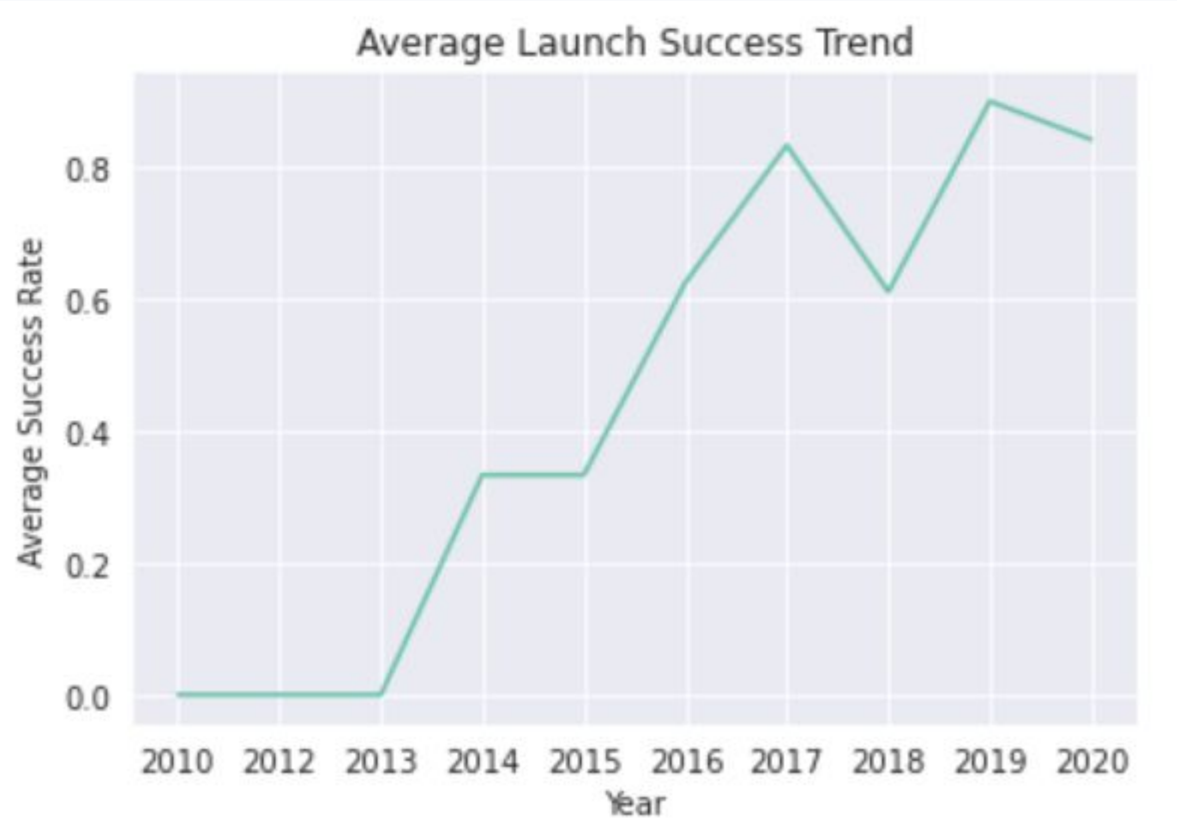
Payload vs. Orbit Type

- Heavier payloads over 8,000 KG were only run in Polar, VLEO and ISS oribts.
- GTO does not improve as payload increases
- SSO orbit always successful but at smaller payloads



Launch Success Yearly Trend

- Average launch success rate has had an overall upward trend since 2013.
- Success rate dipped in 2018 and 2020



All Launch Site Names

- There are 4 unique launch site names from the web scrape data. But CCAFS SLC-40 and CCAFS LC-40 are likely the same and only separated by a typo. This will be proven later on the location analysis.

```
%%sql
select distinct launch_site as distinct_launch_site
from SPACEXTBL
```

```
 * ibm_db_sa://dlr55524:***@b1bc1829-6f45-4cd4-bef4-10c
Done.
```

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

Launch Site Names Begin with 'CCA'

- The mission outcomes were all successful for the 5 `CCA` shown below in the LEO(ISS), GTO and LEO orbits

```
%%sql
select * from SPACEXTBL where launch_site LIKE 'CCA%'
order by launch_site desc
limit 5
```

```
 * ibm_db_sa://dlr55524:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
Done.
```

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2017-12-15	15:36:00	F9 FT B1035.2	CCAFS SLC-40	SpaceX CRS-13	2205	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2018-04-02	20:30:00	F9 B4 B1039.2	CCAFS SLC-40	SpaceX CRS-14	2647	LEO (ISS)	NASA (CRS)	Success	No attempt
2018-03-06	05:33:00	F9 B4 B1044	CCAFS SLC-40	Hispasat 30W-6 PODSat	6092	GTO	Hispasat NovaWurks	Success	No attempt
2018-01-31	21:25:00	F9 FT B1032.2	CCAFS SLC-40	GovSat-1 / SES-16	4230	GTO	SES	Success	Controlled (ocean)
2018-01-08	01:00:00	F9 B4 B1043.1	CCAFS SLC-40	Zuma	5000	LEO	Northrop Grumman	Success (payload status unclear)	Success (ground pad)

Total Payload Mass

- The total payload carried by boosters for customer NASA is 45,596 KG
- A follow up is to check how many payloads make up the 45,596 KG but the database I was working in is no longer available.

```
%%sql
SELECT sum(payload_mass__kg_) as sum_payload_mass__kg_NASA_CRS from spacextbl
where customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://dlr55524:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqc
Done.
```

sum_payload_mass__kg_nasa_crs
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is 2,928 KG

```
%%sql
SELECT avg(payload_mass__kg_) as payload_mass__kg_booster_F9v11 from spacextbl
where booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://dlr55524:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqd
Done.
```

payload_mass__kg_booster_f9v11
2928

First Successful Ground Landing Date

- The first successful landing outcome on ground pad was December 22, 2015

```
%%sql
select DATE, landing__outcome
from SPACEXTBL
where DATE = (select min(DATE) from SPACEXTBL where landing__outcome like '%Success (ground pad)%');
```

```
 * ibm_db_sa://dlr55524:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomair
Done.
```

DATE	landing__outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- There were 4 FT boosters that successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

```
%%sql
select booster_version, landing__outcome, payload_mass__kg_
from SPACEXTBL
where landing__outcome like '%Success (drone ship)%' and payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000
```

```
 * ibm_db_sa://dlr55524:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- There were 10 failed outcomes
- There were 69 successful outcomes
- There were 22 no attempt

```
%%sql
select landing__outcome,count(landing__outcome)
from SPACEXTBL
group by landing__outcome
```

```
 * ibm_db_sa://dlr55524:***@b1bc1829-6f45-4cd4-bef4-
Done.
```

landing__outcome	2
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	22
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- There were 12 Boosters with the same maximum payload of 15,600 KG with the booster version F9 B5 B10XX.X.

```
%%sql

select booster_version, landing__outcome, payload_mass__kg_
from SPACEXTBL
where payload_mass__kg_ = (select max(payload_mass__kg_) from spacextbl)
```

```
 * ibm_db_sa://dlr55524:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgt
Done.
```

booster_version	landing__outcome	payload_mass__kg_
F9 B5 B1048.4	Success	15600
F9 B5 B1049.4	Success	15600
F9 B5 B1051.3	Success	15600
F9 B5 B1056.4	Failure	15600
F9 B5 B1048.5	Failure	15600
F9 B5 B1051.4	Success	15600
F9 B5 B1049.5	Success	15600
F9 B5 B1060.2	Success	15600
F9 B5 B1058.3	Success	15600
F9 B5 B1051.6	Success	15600
F9 B5 B1060.3	Success	15600
F9 B5 B1049.7	Success	15600

2015 Launch Records

- In 2015 there were 2 failed drone ship landings at Launch site CCAFS LC-40 by booster version F8 v1.1 B10XX and with payload ~ 2,000KG.

```
%%sql

select booster_version, landing__outcome, payload_mass__kg_, launch_site, DATE
from SPACEXTBL
where landing__outcome LIKE '%Failure (drone ship)%' and  Date between '2014-12-31' and '2016-01-01'
```

```
 * ibm_db_sa://dlr55524:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.c
Done.
```

booster_version	landing__outcome	payload_mass__kg_	launch_site	DATE
F9 v1.1 B1012	Failure (drone ship)	2395	CCAFS LC-40	2015-01-10
F9 v1.1 B1015	Failure (drone ship)	1898	CCAFS LC-40	2015-04-14

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

- There were 10 landing that weren’t attempted between 2010 and 2017
- The most successful landings at that time were by drone ship.

```
%%sql

select landing__outcome, count(landing__outcome) as count_landing_outcome
from SPACEXTBL
where Date between '2010-06-04' and '2017-03-20'
group by landing__outcome
order by count_landing_outcome desc
```

```
 * ibm_db_sa://dlr55524:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0
Done.
```

landing__outcome	count_landing_outcome
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 3

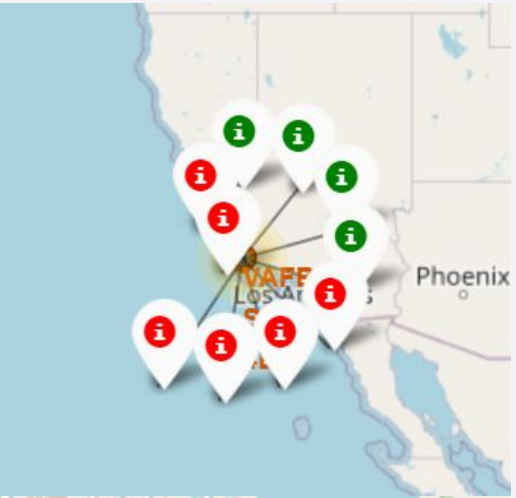
Launch Sites Proximities Analysis

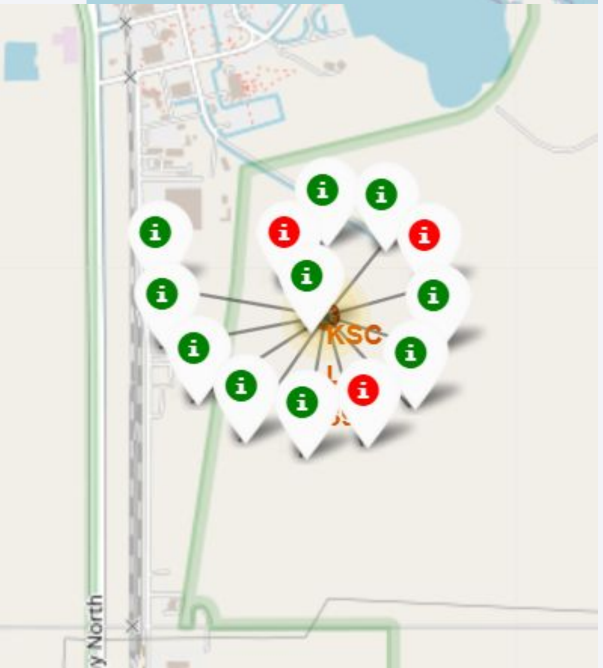
Location of all launch Sites

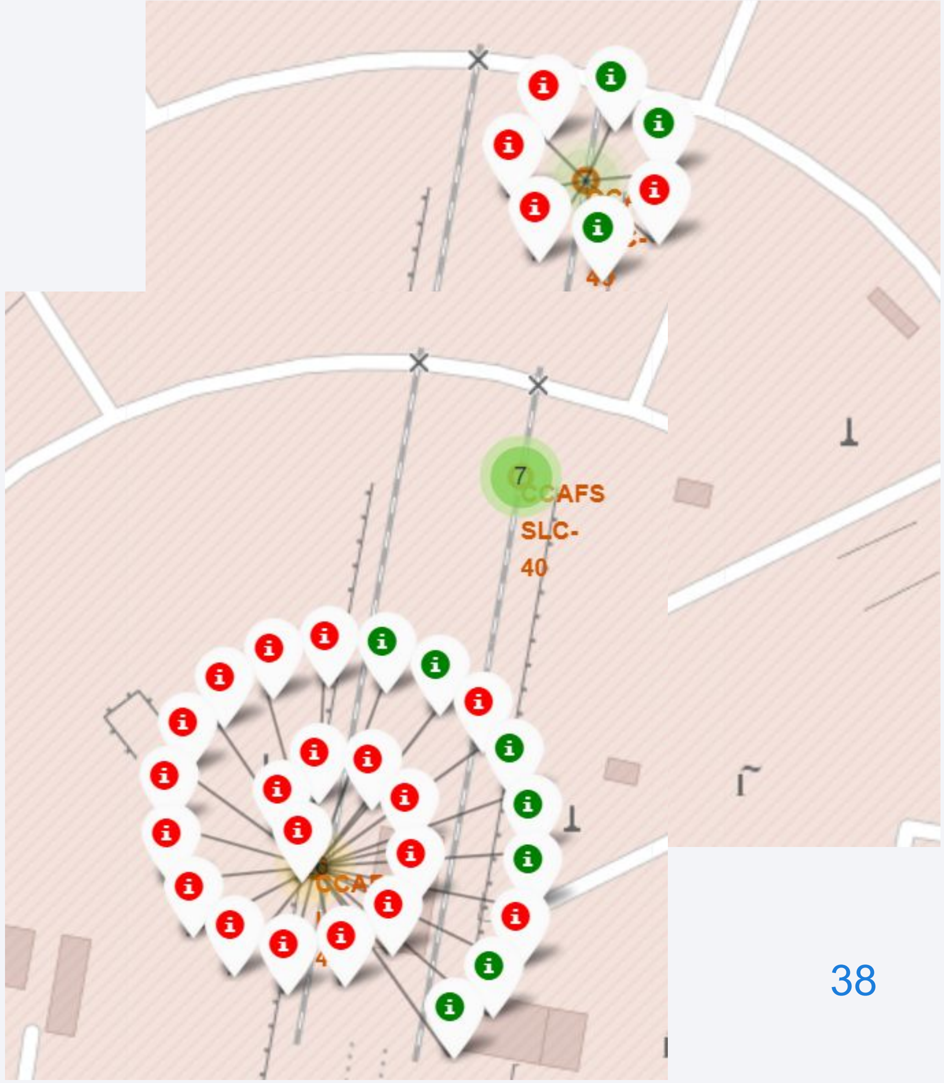
- We have proven that CCAFS SLC-40 and CCAFS LC-40 the same place because their geolocation is on the same exact spot.
- There are only 3 launch sites used and they are all near the ocean
- The launch sites are somewhat proximite to the equator line. Launch speeds would be greater near the equator. The extra rotation speed gives extra velocity for spaceships going east.

Launch Outcomes by Site

- Successful launch outcomes in green were the greatest at SLC LC 39A

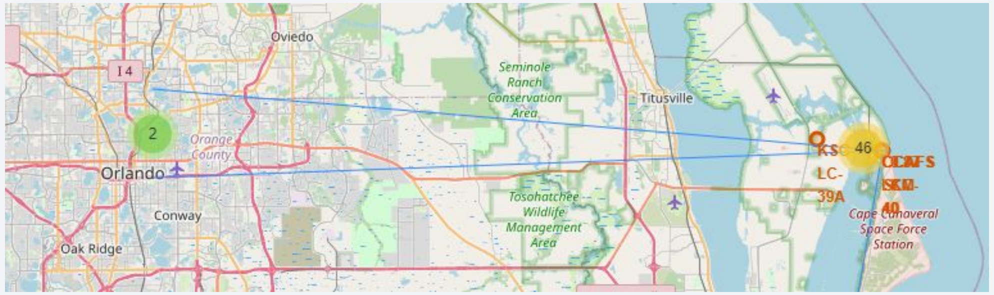






Launch Site Proximity to important elements

- CCAFS SLC-40 site is closest to the ocean (.39 KM from the ocean)
- The highway is 19 KM away from CCAFS SLC-40
- The railway and city are 76 KM away from CCAFS SLC-40

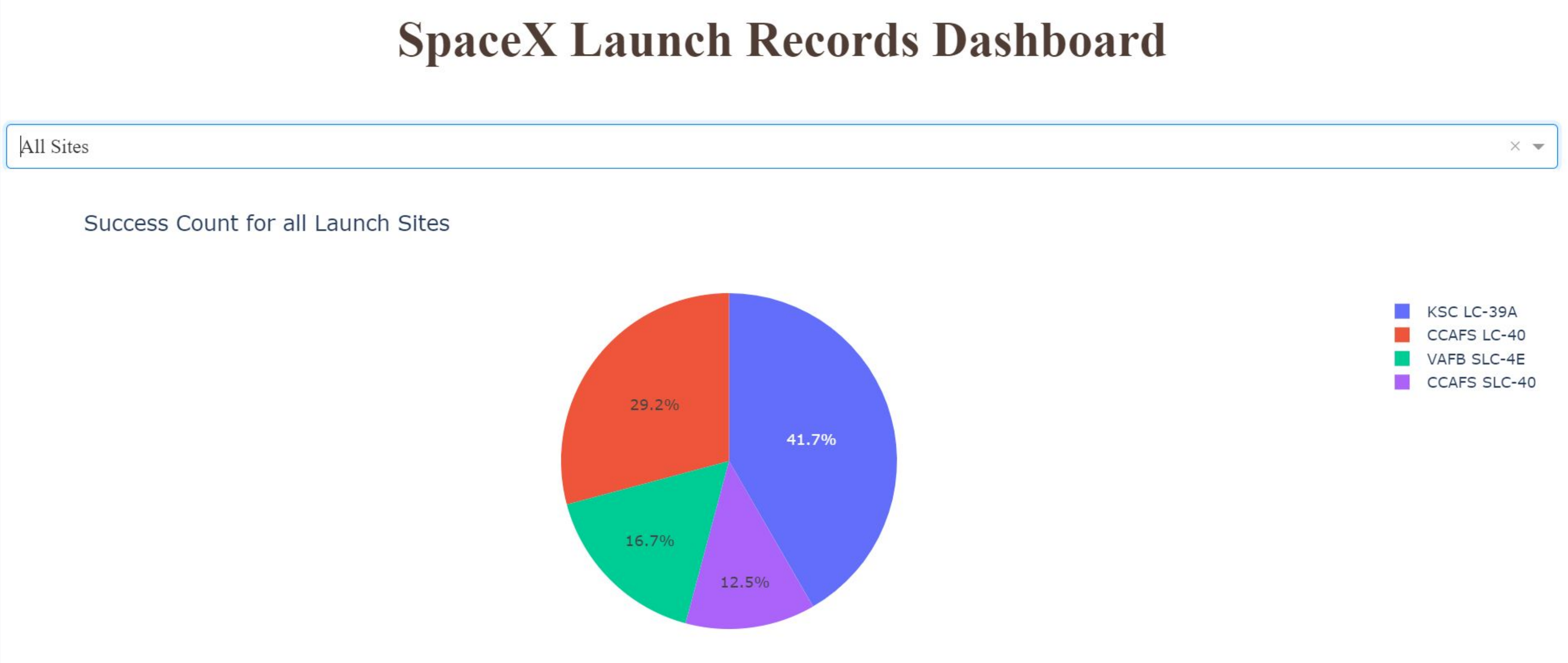


Section 4

Build a Dashboard with Plotly Dash

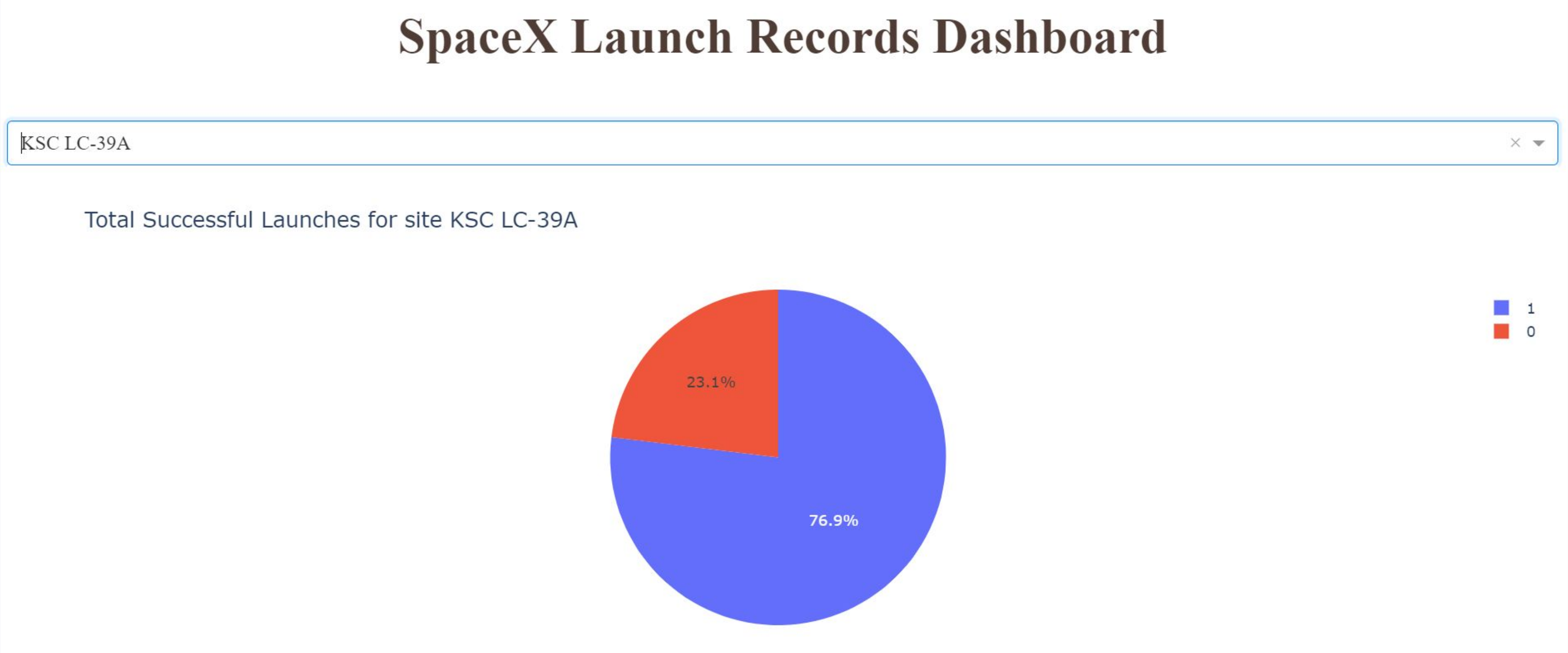
Successful Launches By Site

- The KSC-LC39A and CCAFS SLC-40 (if you combine the typo) both have a success rate of 41.7%.



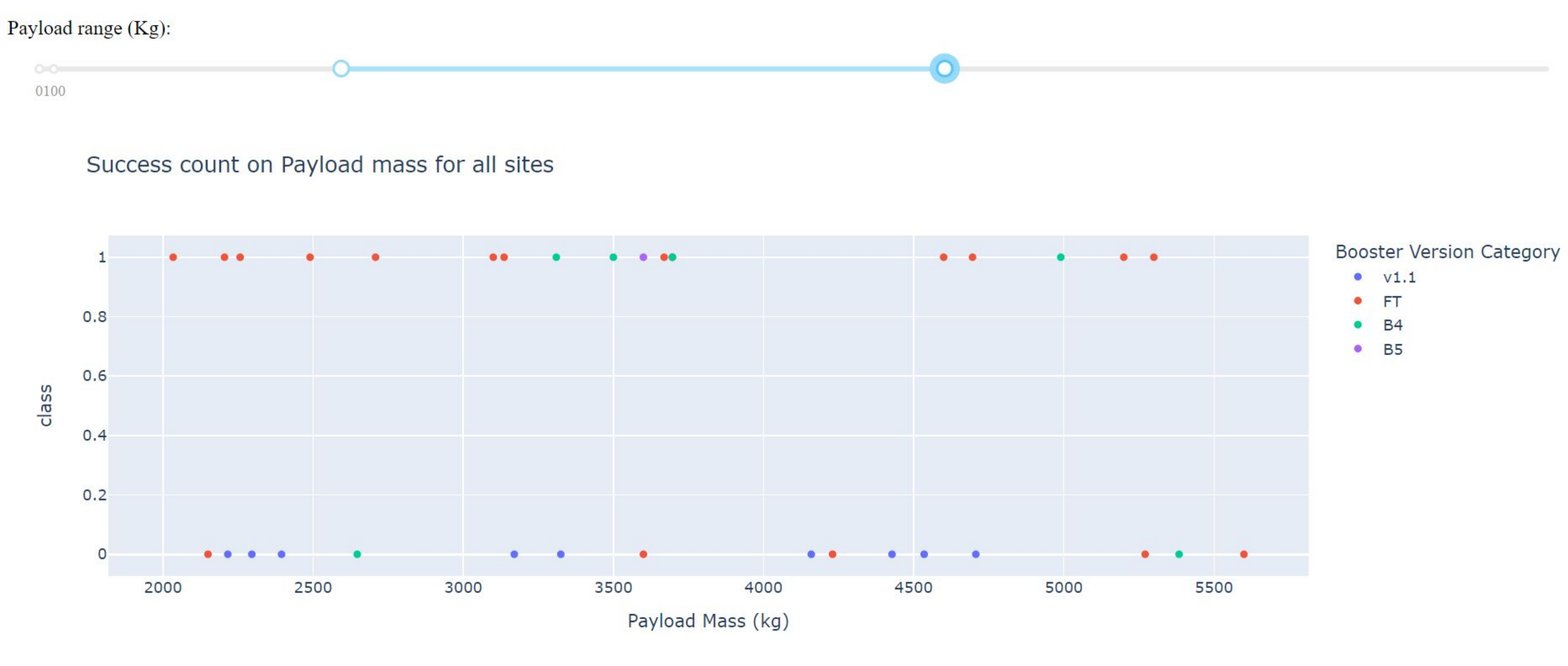
Highest on Site Success Rate

- The KSC-LC39A and CCAFS SLC-40 have the highest success rate with 76.9% successful launches from the site.



Payload Mass and Booster Version Impact on Outcome

- V1.1 doesn't perform well at higher payload Masses. FT appears to have the highest success rate at higher Payload Masses.

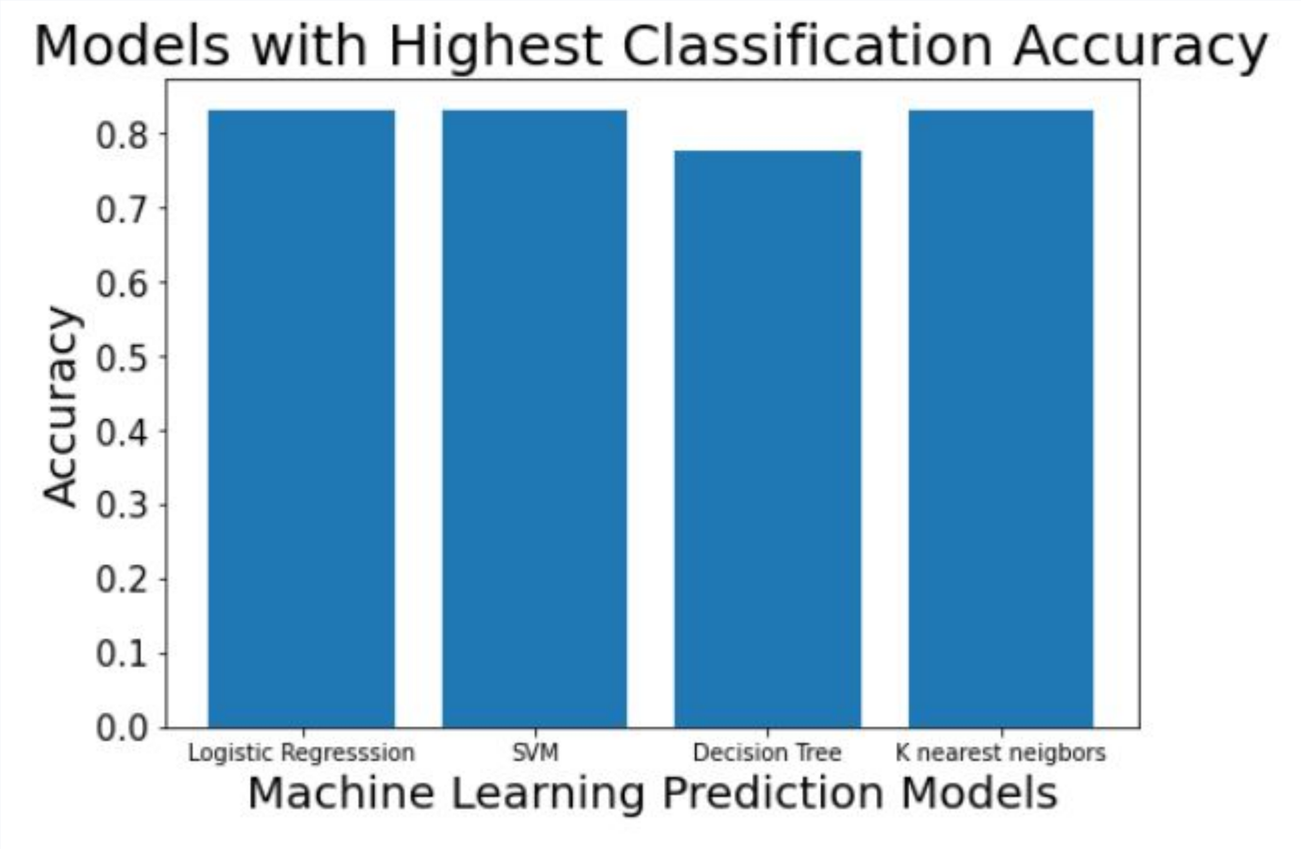


Section 5

Predictive Analysis (Classification)

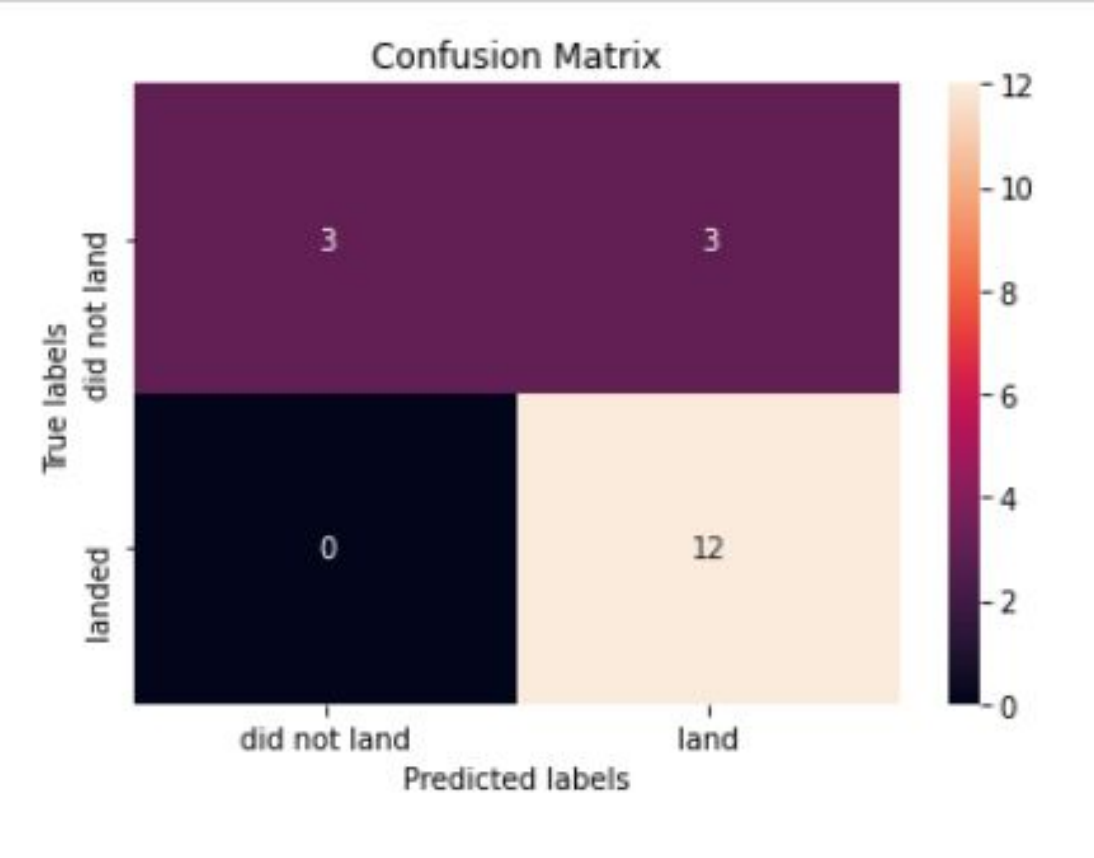
Classification Accuracy

- Logistic Regression, SVM and K Nearest Neighbor all had the highest accuracy at .833



The Space X Falcon 9 First Stage Landing Confusion Matrix

- 72 flights: 54 to train model and 18 to test model.
- There were 12 True Positives
- There were 3 True Negatives
- There were 3 that were predicted to land did not land. (Type 1 error/ False Positive)
- There were no landing predicted not to land that landed (No type 2 errors / False Negatives)
- Accuracy: 15/18-.83 (total correct over total)
- Precision:12/15=.80 (positive number over total positives)
- The classification model predicted with greater accuracy than precision



Conclusions

- Good accuracy in machine learning is subjective.
- In the case of SpaceY, If you predict fewer flights crashing than actually do, you will underestimate the cost of launches. While you may win bids, you may risk losing your business by underestimating costs.
- The recommendation for SpaceY is to continue to improve their classification model by using some of recommended techniques
  - Lower the threshold values to lower the false positive rate (FPR).
  - Giving positive samples a very large weight during training.
  - Data augmentation of positive samples, so making the positive dataset 100 time bigger or something.
  - Adding more data
  - Treat missing and Outlier values
  - Feature Engineering
  - Feature Selection
  - Multiple algorithms
  - Algorithm Tuning
  - Ensemble methods

In the EDA with Data Visualization lab, we selected the following variables for our prediction model.

- FlightNumber
- PayloadMass
- Flights
- Block
- ReusedCount
- Orbits:
  - Orbit_ES-L1
    Orbit_GEO
    Orbit_GTO
    Orbit_HEO
    Orbit_ISS ...
- Serials:
  - Serial_B1058
    Serial_B1059
    Serial_B1060
    Serial_B1062
- GridFins_False
- GridFins_True
- Reused_False
- Reused_True
- Legs_False
- Legs_True

Appendix

- GitHub repository: https://github.com/robinbramdata/IBM-Data-Science-Professional-Certificate/tree/master

- A special thanks to IBM for creating this phenomenal course that breaks down what to learn into digestible pieces that layer on top of eachother for a deeper level of understanding. I don't think I have taken a better class including my years at Yale. https://www.coursera.org/professional-certificates/ibm-data-science

Thank you!

Appendix

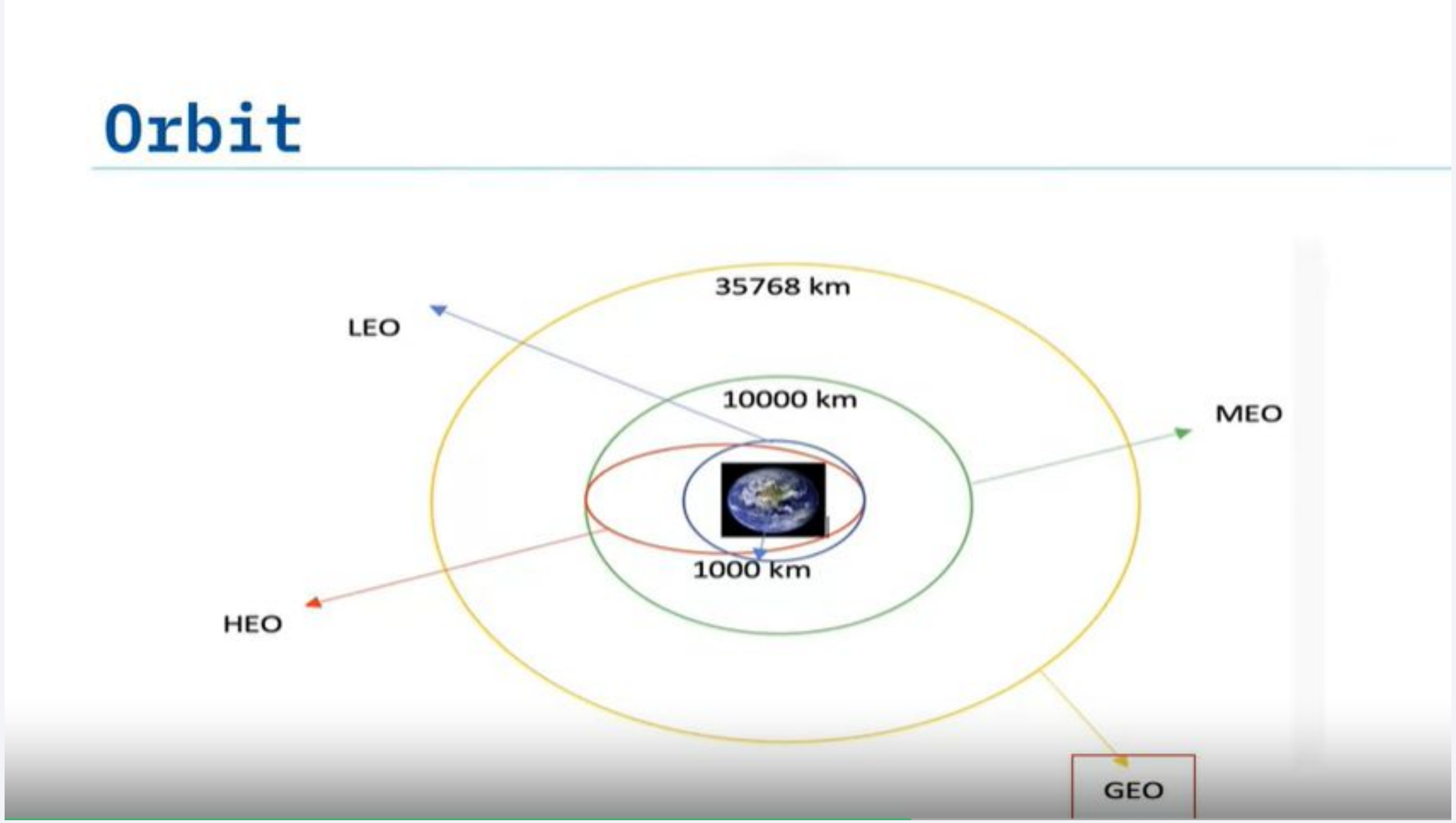
- Definitions
- API Code snippets
- Beautiful Soup Code Snippets
- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Introduction - Definitions

- Payload: Purpose of mission. Could include satellites, telescopes, Space Station Modules, Cargo, and even crew spacecraft
- Staging: Staging is the combination of several rocket sections, or stages, that fire in a specific order and then detach, so a ship can penetrate Earth's atmosphere and reach space
- Fins: Grid fins are a type of flight control surface used on rockets and bombs.
- Legs: Enables the phases of the rocket to be land and be reusable.
- Data Wrangling: Data wrangling is the process of transforming and mapping data from one "raw" data form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics.
- Outcome: Whether first stage successfully landed True ASDS / Landed to drone ship, False ASDS / Didn't land to drone ship.

Introduction - Definitions (Cont)

- Orbit: An orbit is a regular, repeating path that one object in space takes around another one. There are three main "bands" of orbit around the Earth: low Earth orbit (LEO), medium Earth orbit (MEO), and geostationary orbit (GEO).



Introduction - Definitions (Cont)

Orbits

- LEO: Low Earth orbit (LEO)is an Earth-centred orbit with an altitude of 2,000 km (1,200 mi) or less (approximately one-third of the radius of Earth),[1] or with at least 11.25 periods per day (an orbital period of 128 minutes or less) and an eccentricity less than 0.25.[2] Most of the manmade objects in outer space are in LEO [1].
- VLEO: Very Low Earth Orbits (VLEO) can be defined as the orbits with a mean altitude below 450 km. Operating in these orbits can provide a number of benefits to Earth observation spacecraft as the spacecraft operates closer to the observation[2].
- GTO A geosynchronous orbit is a high Earth orbit that allows satellites to match Earth's rotation. Located at 22,236 miles (35,786 kilometers) above Earth's equator, this position is a valuable spot for monitoring weather, communications and surveillance. Because the satellite orbits at the same speed that the Earth is turning, the satellite seems to stay in place over a single longitude, though it may drift north to south," NASA wrote on its Earth Observatory website [3] .
- SSO (or SO): It is a Sun-synchronous orbit also called a heliosynchronous orbit is a nearly polar orbit around a planet, in which the satellite passes over any given point of the planet's surface at the same local mean solar time [4] .

Introduction - Definitions (Cont)

Orbits(Cont)

- ES-L1 :At the Lagrange points the gravitational forces of the two large bodies cancel out in such a way that a small object placed in orbit there is in equilibrium relative to the center of mass of the large bodies. L1 is one such point between the sun and the earth [5] .
- HEO A highly elliptical orbit, is an elliptic orbit with high eccentricity, usually referring to one around Earth [6].
- ISS A modular space station (habitable artificial satellite) in low Earth orbit. It is a multinational collaborative project between five participating space agencies: NASA (United States), Roscosmos (Russia), JAXA (Japan), ESA (Europe), and CSA (Canada) [7]
- MEO Geocentric orbits ranging in altitude from 2,000 km (1,200 mi) to just below geosynchronous orbit at 35,786 kilometers (22,236 mi). Also known as an intermediate circular orbit. These are "most commonly at 20,200 kilometers (12,600 mi), or 20,650 kilometers (12,830 mi), with an orbital period of 12 hours [8]
- HEO Geocentric orbits above the altitude of geosynchronous orbit (35,786 km or 22,236 mi) [9]
- GEO It is a circular geosynchronous orbit 35,786 kilometres (22,236 miles) above Earth's equator and following the direction of Earth's rotation [10]
- PO It is one type of satellites in which a satellite passes above or nearly above both poles of the body being orbited (usually a planet such as the Earth [11]

Thank you!