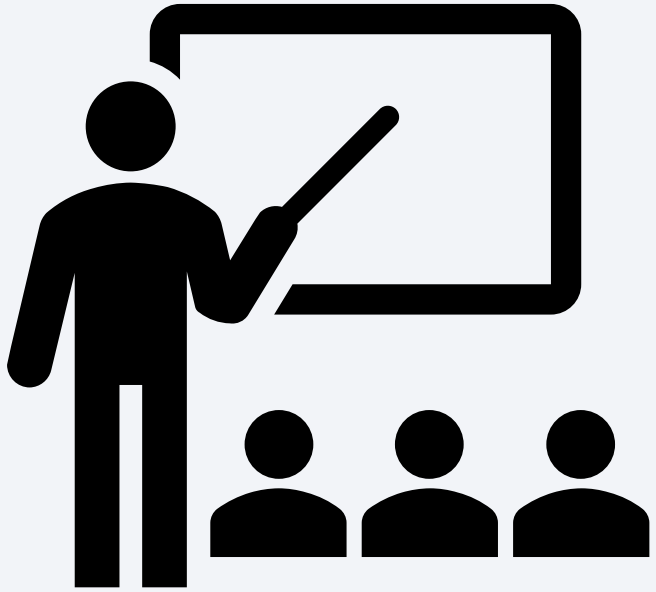
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nesibe Elibol
15/11/2021
https://github.com/etudehome/IBM-Data-Science-

Outline

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



Executive Summary

- Summary of methodologies
  - Data collection via API, SQL and Web Scraping
  - Data Wrangling and Analysis
  - Interactive Maps with Folium maps and dashboards
  - Predictive Analysis for 4 machine learning models:
    - Logistic Regression, Support Vector Machine, Decision Tree Classifier and K Nearest Neighbors.
    - All models should predict successful landings
- Summary of all results
  - Data Analysis along with Interactive Visualizations
  - Best model for Predictive Analysis

Introduction

- Companies make **space travel affordable for everyone**. One of the most successfully providers are SpaceX, resulting from their one and foremost their rocket launches which are relatively inexpensive.
- **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 successfully**, it will help us understand the cost of a launch. This information ca be used if an alternate company wants to bid against SpaceX for a rocket launch.

Hence we will predict if the Falcon 9 first stage will land successfully.

Following questions and problems we will deal with:

- Analyzing with what factors, the rocket will land successfully?
- Investigating the effect of each relationship of rocket variables on outcome
- Under which Conditions will aid SpaceX have to achieve the best results.

Section 1

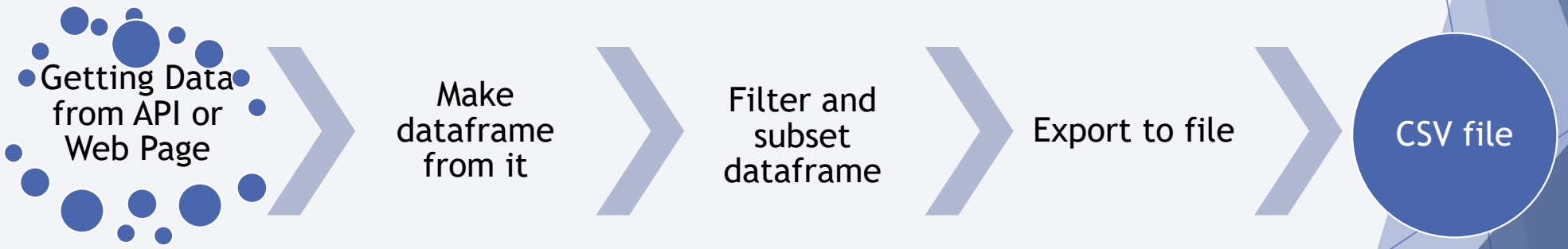
Methodology

Methodology

- Data collection methodology:
  - Via SpaceX Rest API and Web Scrapping from Wikipedia
- Perform data wrangling
  - Encoding data fields for machine learning and dropping irrelevant columns
  - Classifying landings as successful and not successful
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatter and bar graphs to show patterns between data
- Perform interactive visual analytics using Folium and Plotly Dash
  - Using Folium and Plotly Dash Visualizations
- Perform predictive analysis using classification models
  - Build and evaluate classification models used GridSearchCV

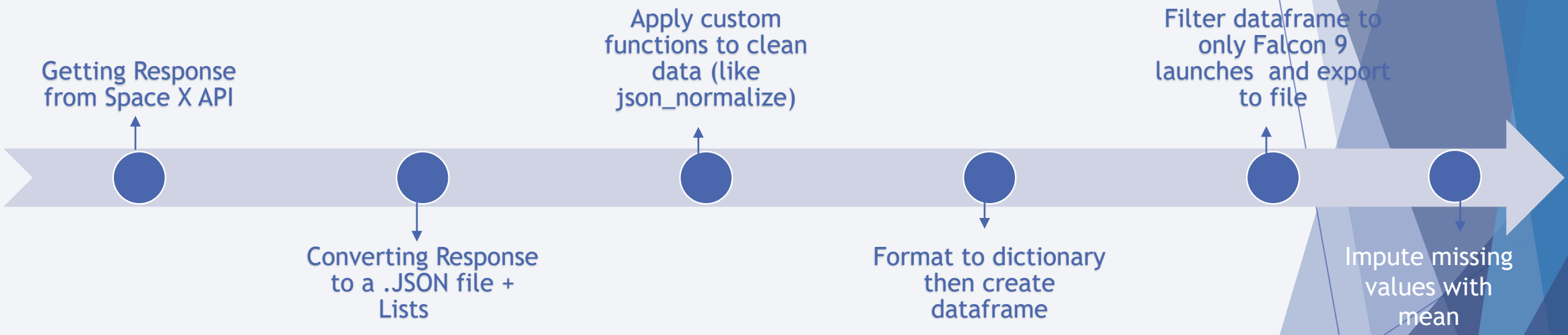
Data Collection

- Data collection is the process of gathering and measuring information on a combination of API requests from Space X public API and web scrapring data.
- Using relevant targeted variables will then enable to answer relevant questions and evaluate outcomes
- Following chart helps to understand the data collection process:



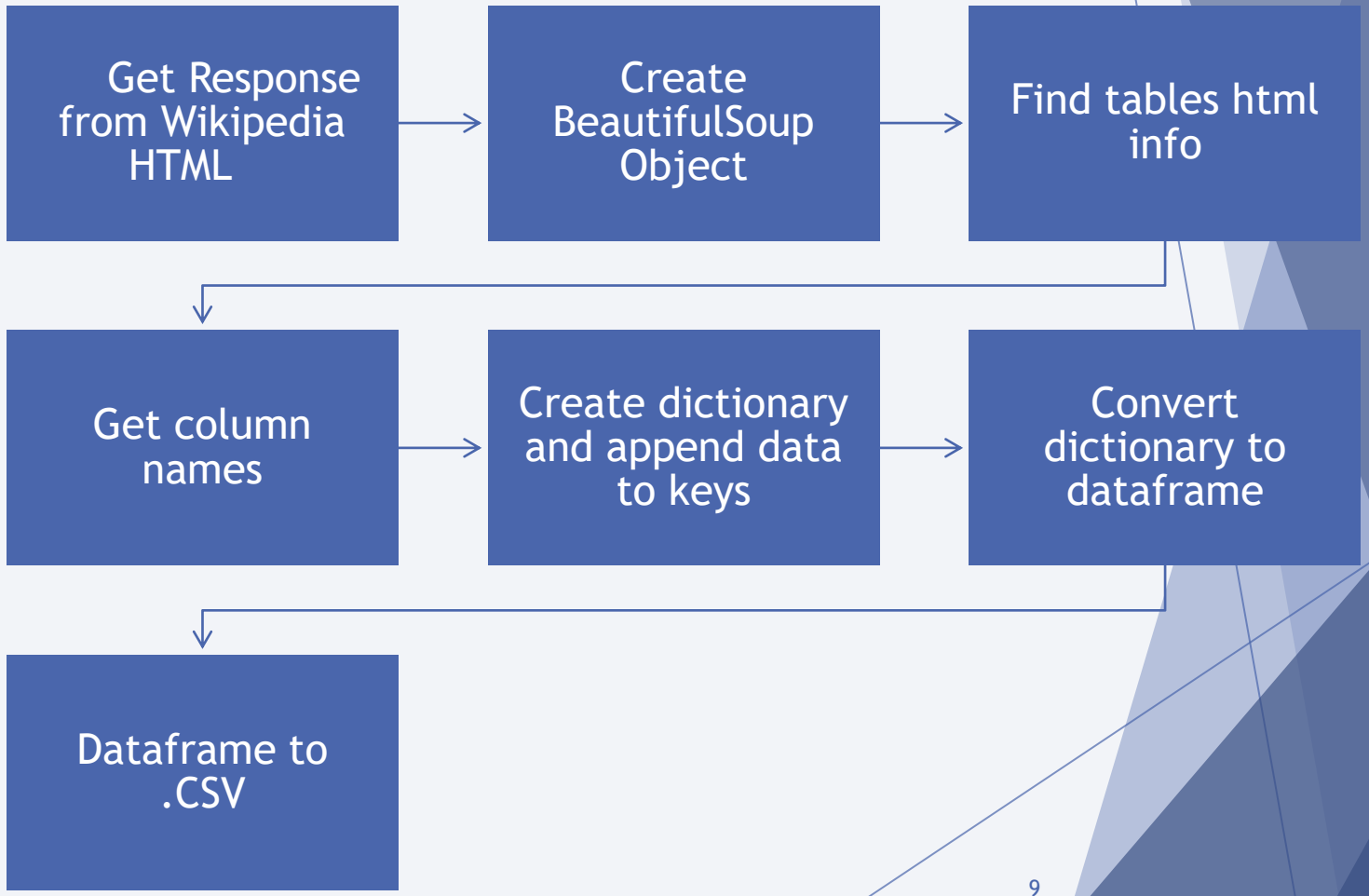
https://github.com/etudehome/IBM-Data-Science-/blob/main/Week1_data-collection-api.ipynb

Data Collection – SpaceX API



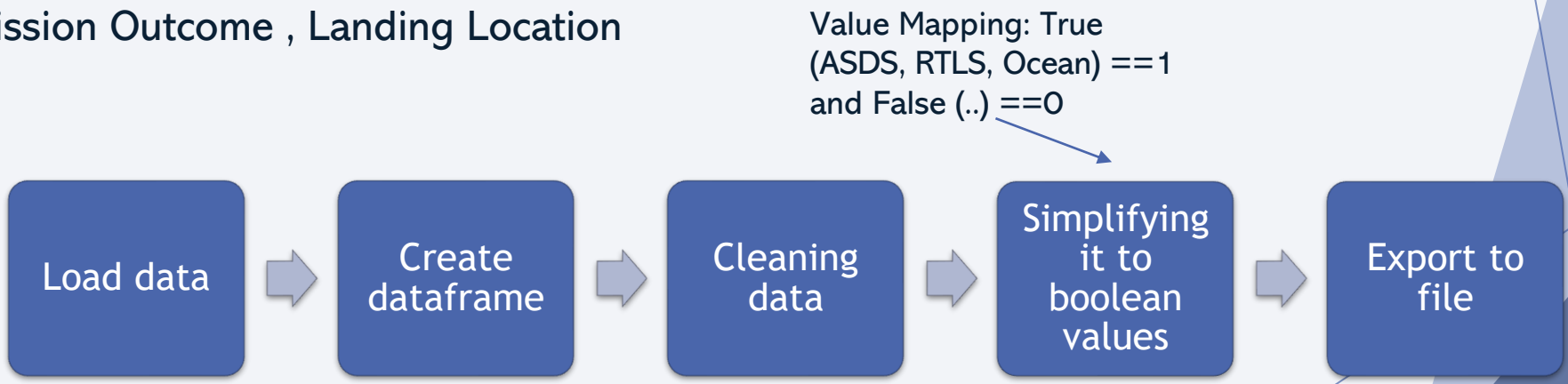
Data Collection – Web Scraping

- https://github.com/etudehome/IBM-Data-Science-/blob/main/Week1_webscraping.ipynb



Data Wrangling

- This step, called Data wrangling is the process of cleaning and unifying unstructures and complex data sets for analysis
- Here we create for these outcomes a training label, where booster successfully landed ==1 and unsuccessfully landed ==0
- Furthermore outcome consists of two components
  - Mission Outcome , Landing Location



https://github.com/etudehome/IBM-Data-Science-/blob/main/Week1_Data%20wrangling.ipynb

Data Wrangling

Data Preprocessing steps

Calculate number of launches at each site

- df["LaunchSite"].value_counts()

```
CCAFS SLC 40     55
KSC LC 39A       22
VAFB SLC 4E      13
Name: LaunchSite, dtype: int64
```

Calculate number of occurence of each orbit

- df["Orbit"].value_counts()

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

Calculate number and occurence of mission outcome per orbit type

- landing_outcomes=df["Outcome"].value_counts()

```
True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
None ASDS        2
False Ocean      2
False RTLS       1
Name: Outcome, dtype: int64
```

Create landing outcome label from Outcome column

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

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False

Export dataset as .CSV

df.to_csv("dataset_part_2.csv", index=False)

EDA – Meaning& Basic Steps

- Exploratory data analysis also known as EDA is an approach of analyzing data sets. In this step the analysis is summarized using statistical graphics and other data visualizations methods.
- Simple overview is:

- Load data
- Create dataframe
- Create Visualizations using scatter plots and bar charts
- Collect insights by comparing relationships between variables

EDA with Data Visualization

Scatter Plots: Visualized

1. Payload Mass and Flight Number
2. Fight Number and Launch Site
3. Payload Mass and Launch Site
4. Flight Number and Orbit Type
5. Payload and Orbit Type

https://github.com/etudehome/IBM-Data-Science-/blob/main/Week2_EDA-Visualization.ipynb

EDA with Data Visualization

- Bar Graph Drawn:

1. Success rate vs. Orbit Type

- With bar graphs we can easily determine which orbits have the highest probability of success.

- Line graphs are useful in that they show trends

1. Launch Success Yearly Trend

EDA with SQL

- We loaded data set into IBM's Db2 -> queried using SQL Python integration
- For example of some questions we were asked about the data we needed information about. Which we are using SQL queries to get the answers in 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 lauchned by NASA (CRS)
  - Displaying average payload mass carried by booster version F9v1.1
  - 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 failed landing_outcomes in drone ship, their booster versions, and launch sites names for the year 2015
  - Ranking 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

- https://github.com/etudehome/IBM-Data-Science-/blob/main/Week2_EDA-sql.ipynb

Build an Interactive Map with Folium

- Folium maps makes it easy to visualize data on an interactive leaflet map. We use the latitude and longitude coordinates for each launch site and added a Circle Marker around each launch site.
- It is easy to visualize the number of success and failure for each launch site with Green and Red markers on the map for key locations: Railway, Highway, Coast and City.
- Following Map Objects used:
  - Map Marker: Map object to make a mark on map
  - Icon Marker: Create an icon on map
  - Ciricle Marker: Create a circle where Marker is being placed
  - PolyLine: Create a line between points.
  - Marker Cluster Object: This is a good way to simply a map containing many markers having the same coordinate
  - AntPath: Create an animated line between points.
- https://github.com/etudehome/IBM-Data-Science-/blob/main/Week3_Folium.ipynb

Build a Dashboard with Plotly Dash

- Creating the Dashboard with Plotly Dash:

1. Pie chart showing the total success for all sites or by certain launch site
   - Percentage of success in relation to launch site
2. Scatter Graph showing the correlation between Payload and Success for all sites or by certain launch site
   - It shows the relationship between Success rate and booster version category
   - Helps understand how success varies across launch sites, payload mass, and booster version category

- https://github.com/etudehome/IBM-Data-Science/blob/main/Week3_Dashboard_plotly.py

Predictive Analysis (Classification)

Model Development process in steps:

Short version:

1. Load the data into dataframe and further work with Pandas and NumPy
2. Standardize, fit and transform data using Standard Scaler
3. Split data into training and test datasets
4. Check how many test samples has been created
5. Type of machine learning algorithms we want to use
6. Set our parameters and algorithms to GridSearchCV (cv=10)
7. Fit our datasets into the GridSearchCV objects and train our model

- https://github.com/etudehome/IBM-Data-Science-/blob/main/Week4_Classification.ipynb

Predictive Analysis (Classification)

Long Version:

```
Y=data[,Class‘],to_numpy()
Transform=preprocessing.StandardScaler()
X=transform.fit(X).transform(X)

X_train,X_test,Y_train,Y_test=
train_test_split(X,y,test_size=0.2,random_stat
e=2)
Y_test.shape
```

1. Load the data into dataframe And further work with Pandas and NumPy
2. Standardize and transform data
3. Split data into training and test datasets
4. Check how many test samples has been created
5. Type of machine learning algorithms we want to use
6. Set our parameters and algorithms to GridSearchCV
7. Fit our datasets into the GridSearchCV objects and train our model

Building the model

Evaluating the model

1. Check accuracy for each model
2. get best hyperparameters for each algorithm
3. Plot confusion matrix

```
Yhat= algorithm.predict(X_test)
Plot_confusion_matrix(Y_test,yh
at)
```

1. The model with accuracy score wins the best performing model

```
Algorthms=
{„KNN“:knn_cv.best_score_,“Deci
sionTree“:
tree_cv.best_score_,“Logistic
Regression“:logreg_cv.best_score
_}
Best_algorithm=max(algorithms,
key=lambda x:algorithms[x])
```

Finding best model

Results

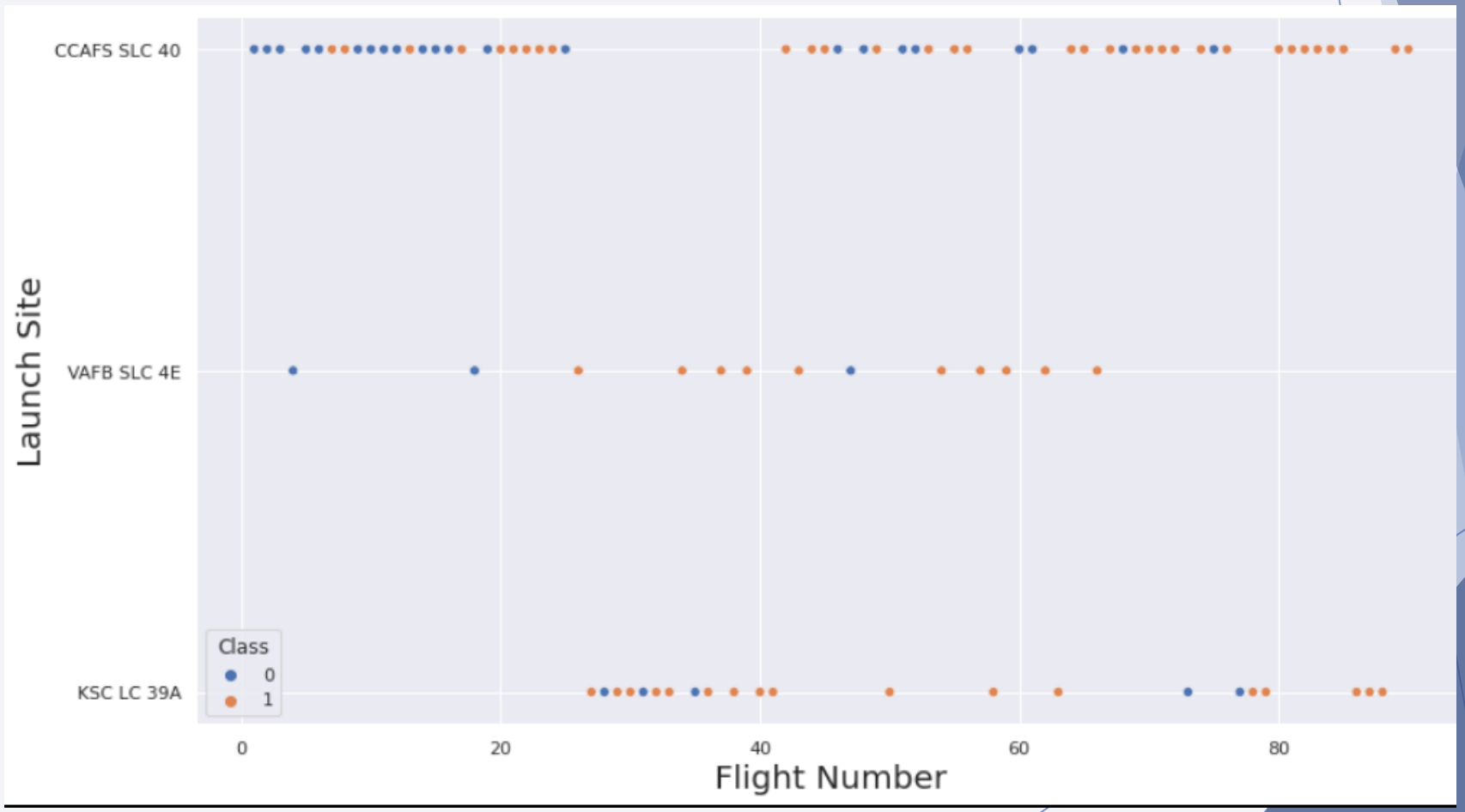
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

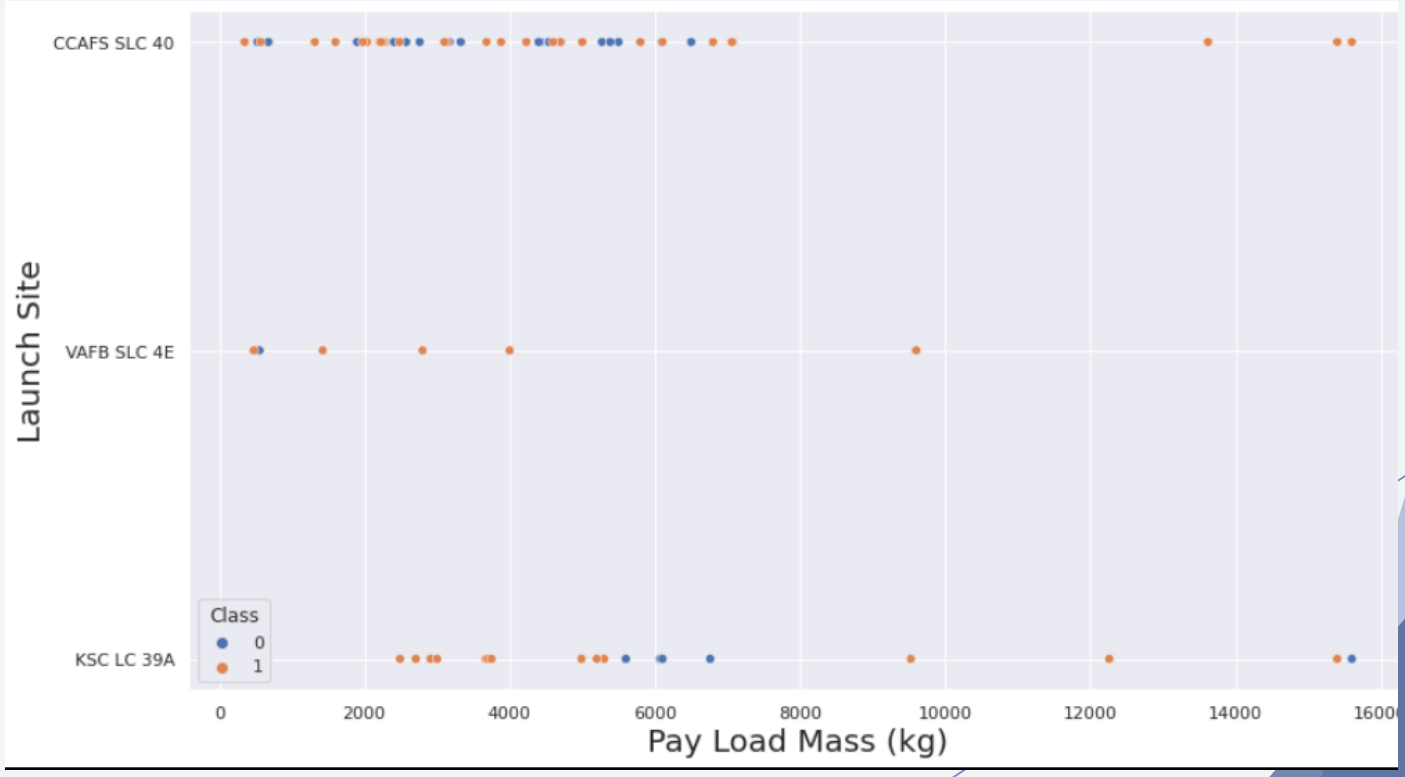
Flight Number vs. Launch Site

- With higher flight numbers at the launch site the greater the success rate at a launch site



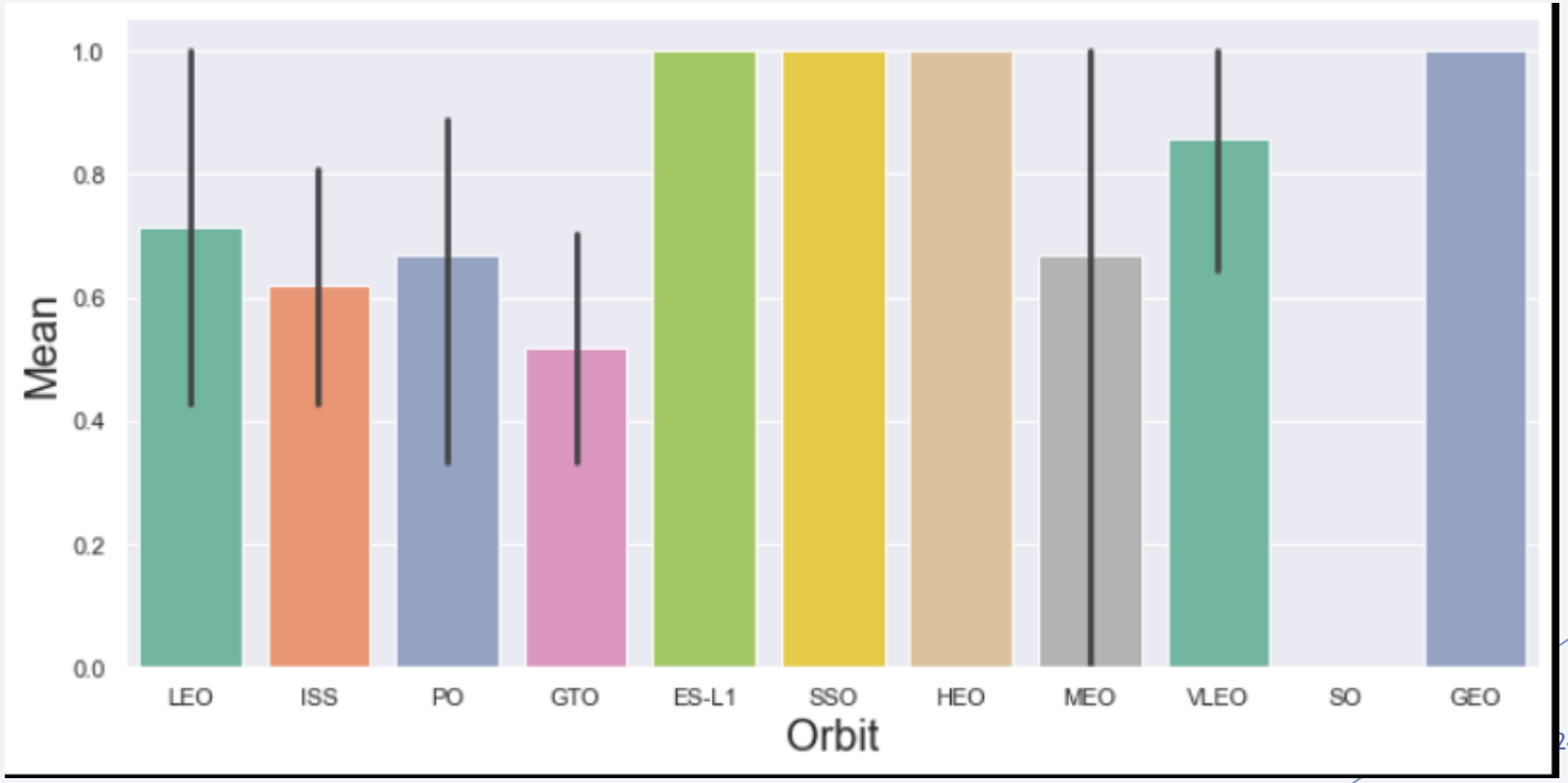
Payload vs. Launch Site

- Blue indicates unsuccessful landing and Orange indicated successful landing
- The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. It is likely a big change around flight 20 which significantly increased success rate.



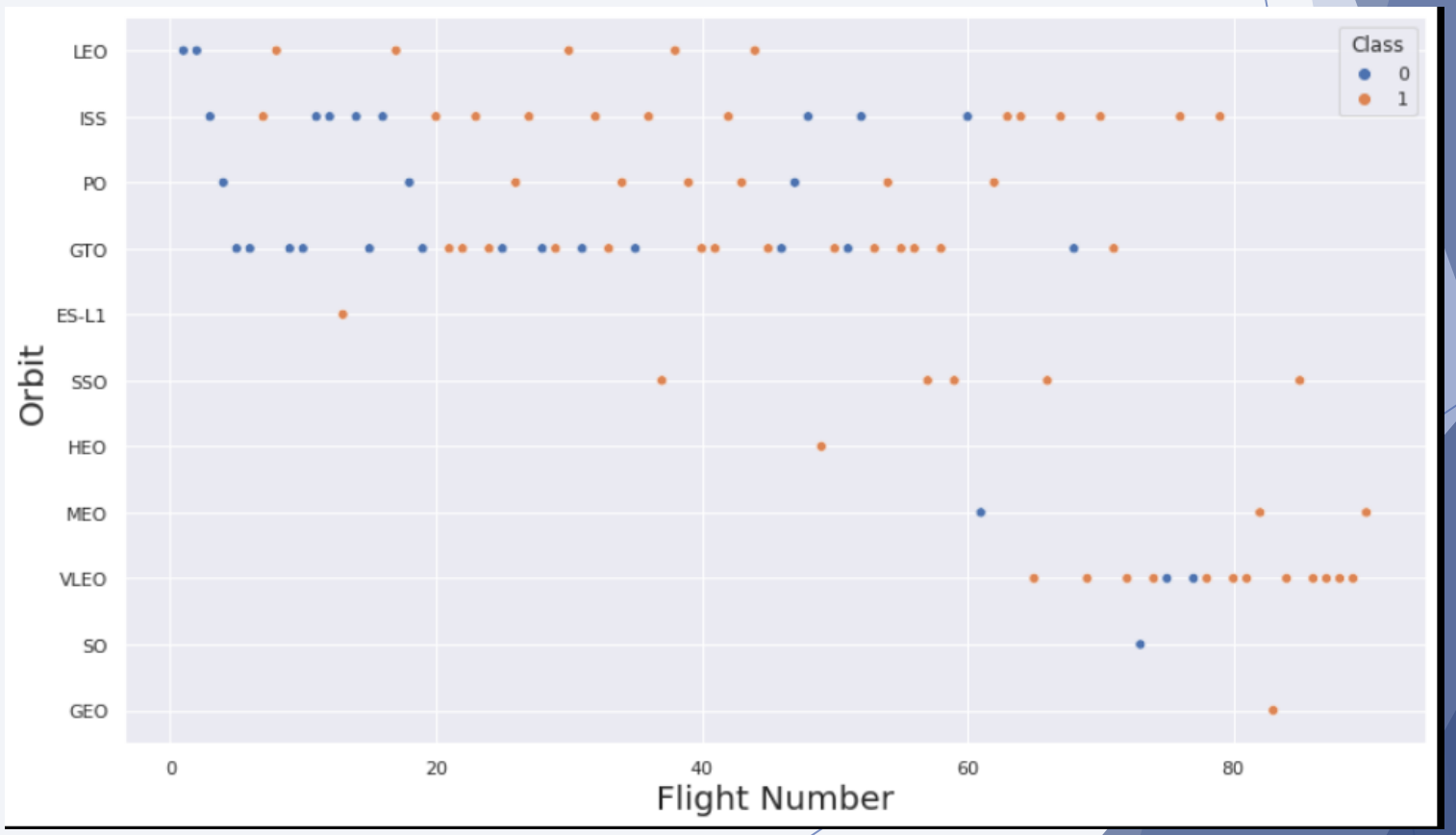
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO has highest Success rates with 100%.
- In comparison, VLEO has a decent success rate and SO has 0% success rate.



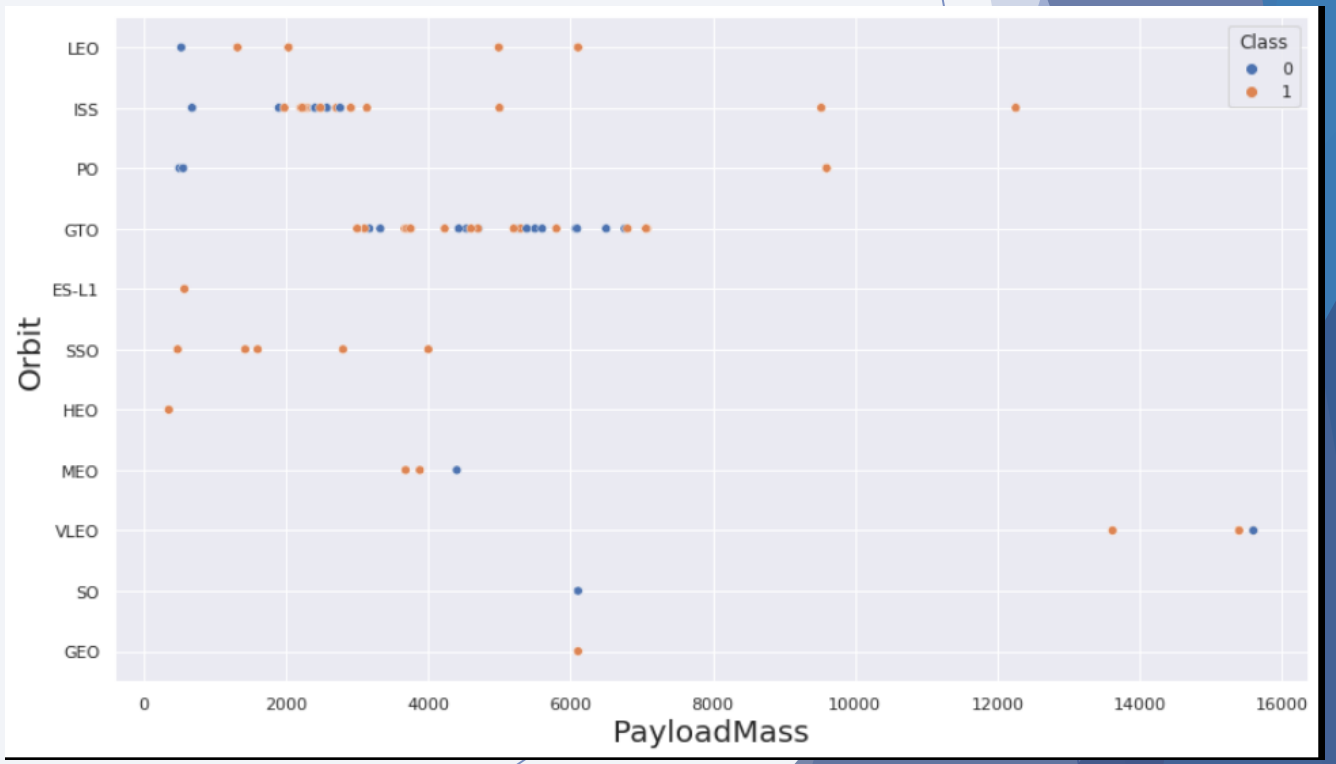
Flight Number vs. Orbit Type

- We see that for LEO orbit the success increases with the number of flights
- On the other hand, there seems to be no relationship between flight number and the GTO orbit.



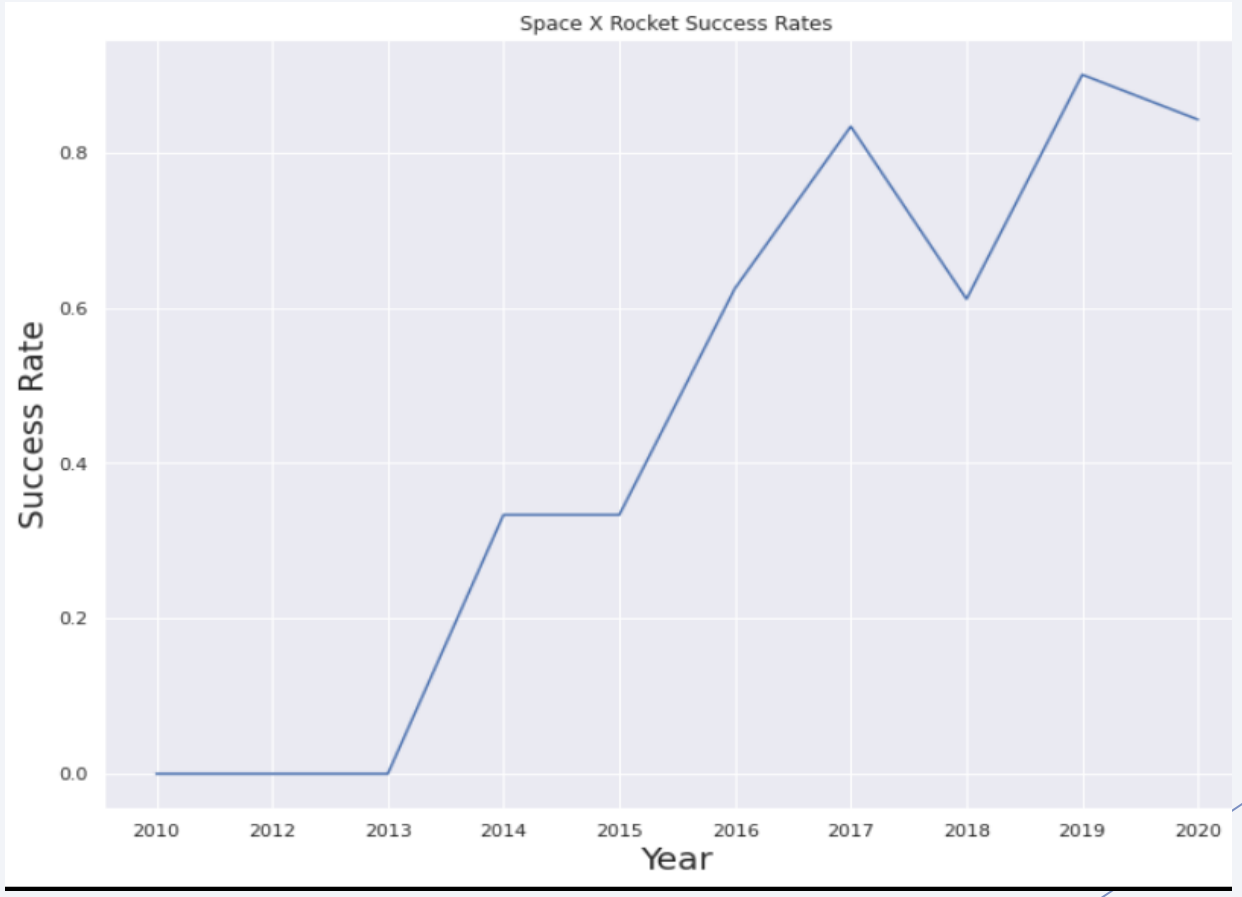
Payload vs. Orbit Type

- Payload mass seems to correlate with orbit.
- We observe that heavy payloads have a negative influence on MEO, GTO, VLEO orbits
- And positive influence on LEO, ISS



Launch Success Yearly Trend

- We can observe that the success rate since 2013 kept increasing relatively, though there is slight dip after 2018. The success rate is around 80% in the last two years.



Section 2

Insights drawn from EDA

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 tblSpaceX

```
%sql select Unique(LAUNCH_SITE) from SPACEXDATASET;
```

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

- CCAFS SLC-40 and CCAFSSLC-40 likely all represent the same
- launch site with data entry errors.

- Likely only 3 unique launch_site values: CCAFS SLC-40, KSC LC-39A,

VAFB SLC-4E

Launch Site Names Begin with 'CCA'

Query explanation: it will only show the first 5 records from tblSpaceX and LIKE keyword has a wild card with the words “CCA”.The percentage suggest that the launch Site must start with CCA

```
%sql SELECT * from SPACEXDATASET where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

DATE	Time (UTC)	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Total Payload Mass from NASA

- Query explanation: Calculate the total payload carried by boosters from NASA
- CRS stands for Commercial Resupply Services which indicates that these payloads were sent to the International Space Station (ISS).

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

sum_payload_mass_kg
45596

Average Payload Mass by F9 v1.1

- Query explanation: Calculate the average payload mass carried by booster version F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-8
Done.
```

avg_payload_mass_kg
2928

- Average payload mass of F9 1.1 is on the low end of our payload mass range

First Successful Ground Landing Date

- Query explanation: Returns the date of the first successful landing outcome on ground pad

```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81
Done.
```

first_success
2015-12-22

- First ground pad landing wasn’t until the end of 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query explanation: returns the four names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%%sql
SELECT booster_version
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4001 AND 5999;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.database
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

- Query explanation: returns a count of each mission outcome.

```
%%sql
SELECT mission_outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
GROUP BY mission_outcome;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-
Done.
```

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

- SpaceX appears to achieve its mission outcome nearly 99% of the time.
  - This means that most of the landing failures are intended.

Boosters Carried Maximum Payload

- Query explanation: returns the booster version which have carried the maximum payload mass

Highest payload mass of 15600 kg

These booster versions are very similar and start with F9 B5 B10..

```
%%sql
SELECT booster_version, PAYLOAD_MASS__KG_
FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1
Done.
```

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

2015 Launch Records

- Query explanation: returns the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, landing__outcome, booster_version, PAYLOAD_MASS__KG_, launch_site
FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.app
Done.
```

MONTH	landing__outcome	booster_version	payload_mass__kg_	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	2395	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	1898	CCAFS LC-40

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

- Query explanation: returns a list of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20

```
%%sql
SELECT landing__outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
WHERE landing__outcome LIKE 'Succes%' AND DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY no_outcome DESC;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg
Done.
```

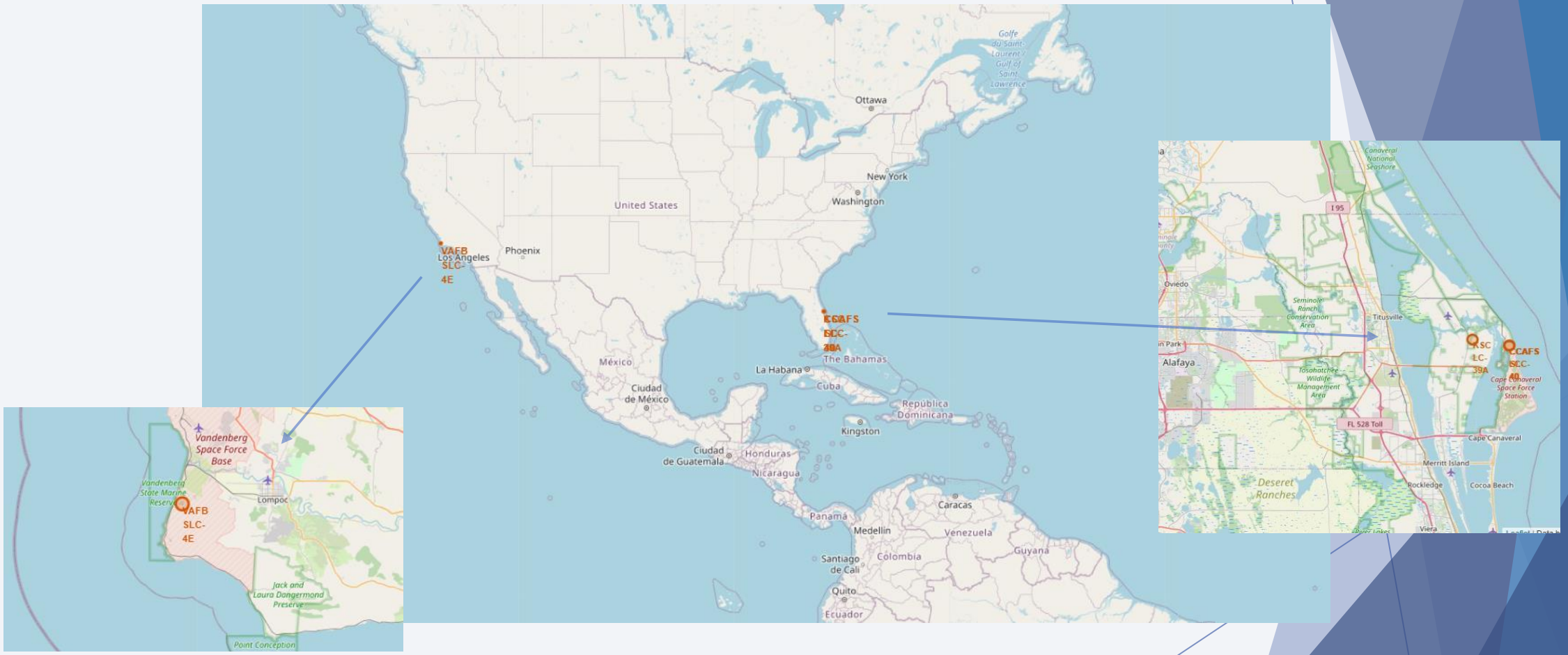
landing__outcome	no_outcome
Success (drone ship)	5
Success (ground pad)	3

- There are two types of successful landing outcomes: drone ship and ground pad landings.

Section 4

Launch Sites Proximities Analysis

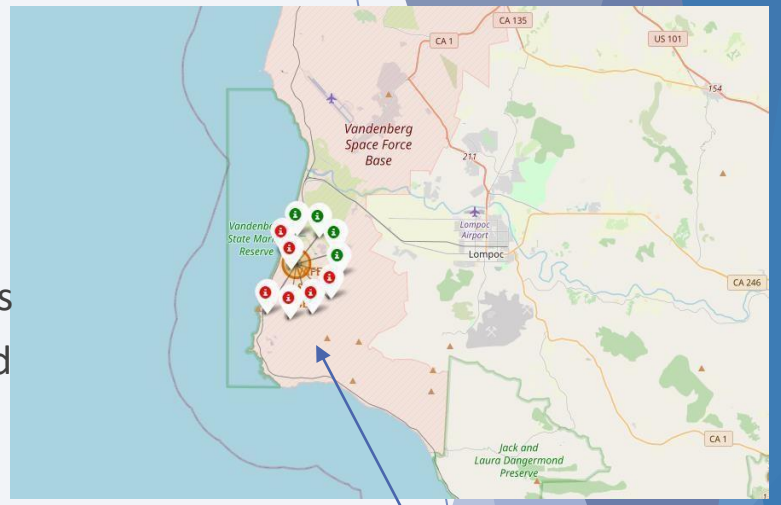
Launch Site Locations



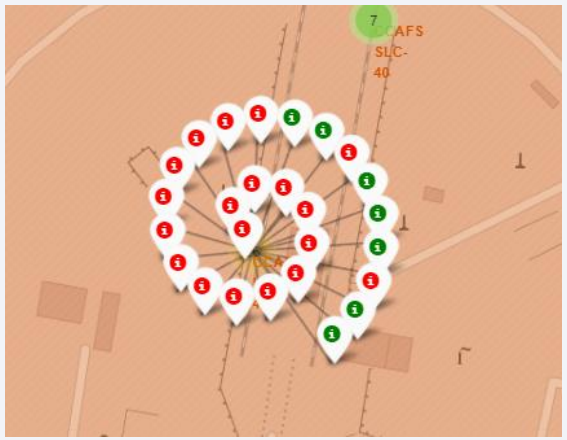
- The map shows all launch sites relative US map. The right map shows the two Florida launch sites.
- All launch sites are near the ocean.

Color labeled Launch Markers

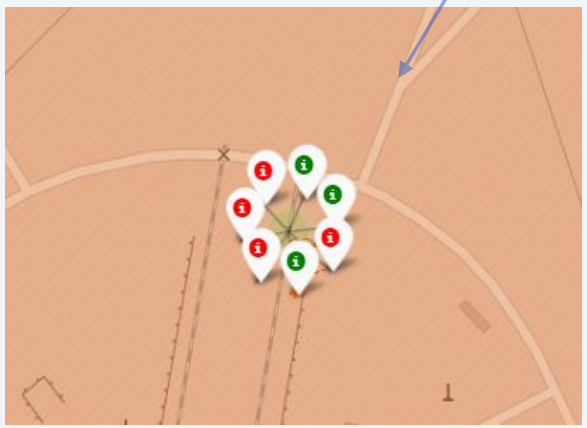
Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed landing (red icon). In this example CCAFS SLC-40 shows 3 successful landings and 4 failed landings.



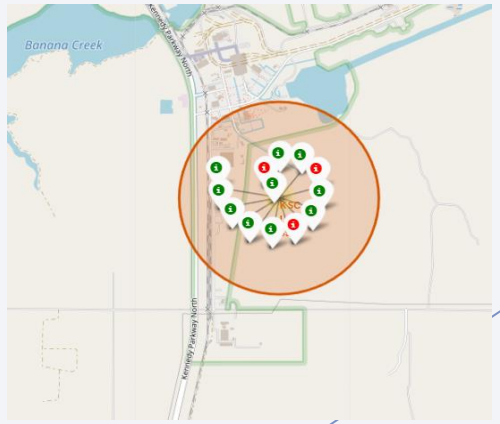
CCAFS LC-40



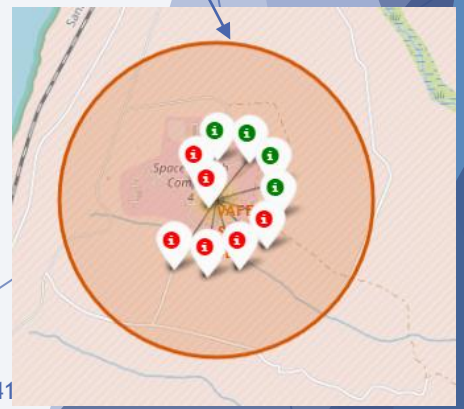
CCAFS SLC-40



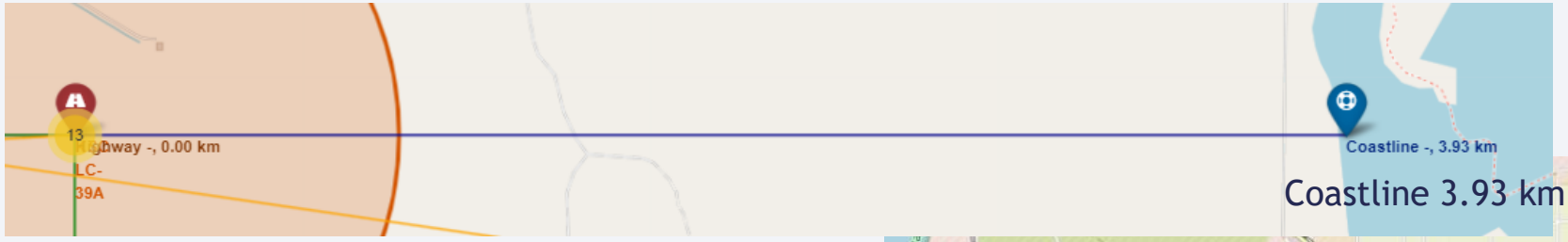
KSC LC-39A



VAFB SLC-4E

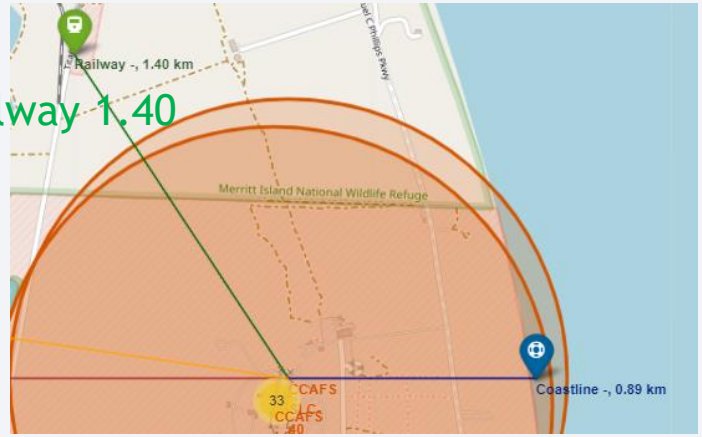


Launch Site Distances from Proximities



Coastline 3.93 km

Distance for all launch sites from coastline is less than 4 km.



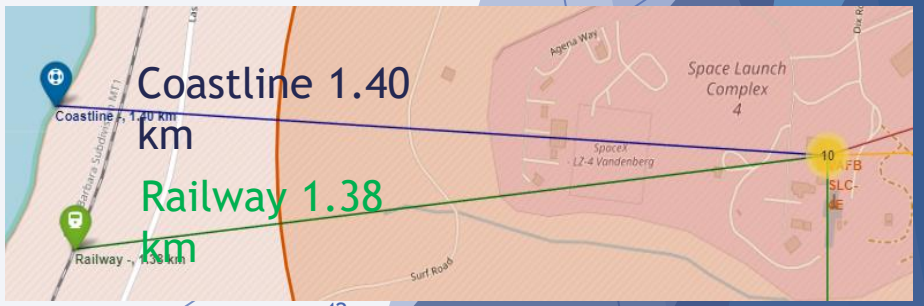
Railway 1.40 km

Highway 14.61 km

City 14.33 km

City 77.20 km

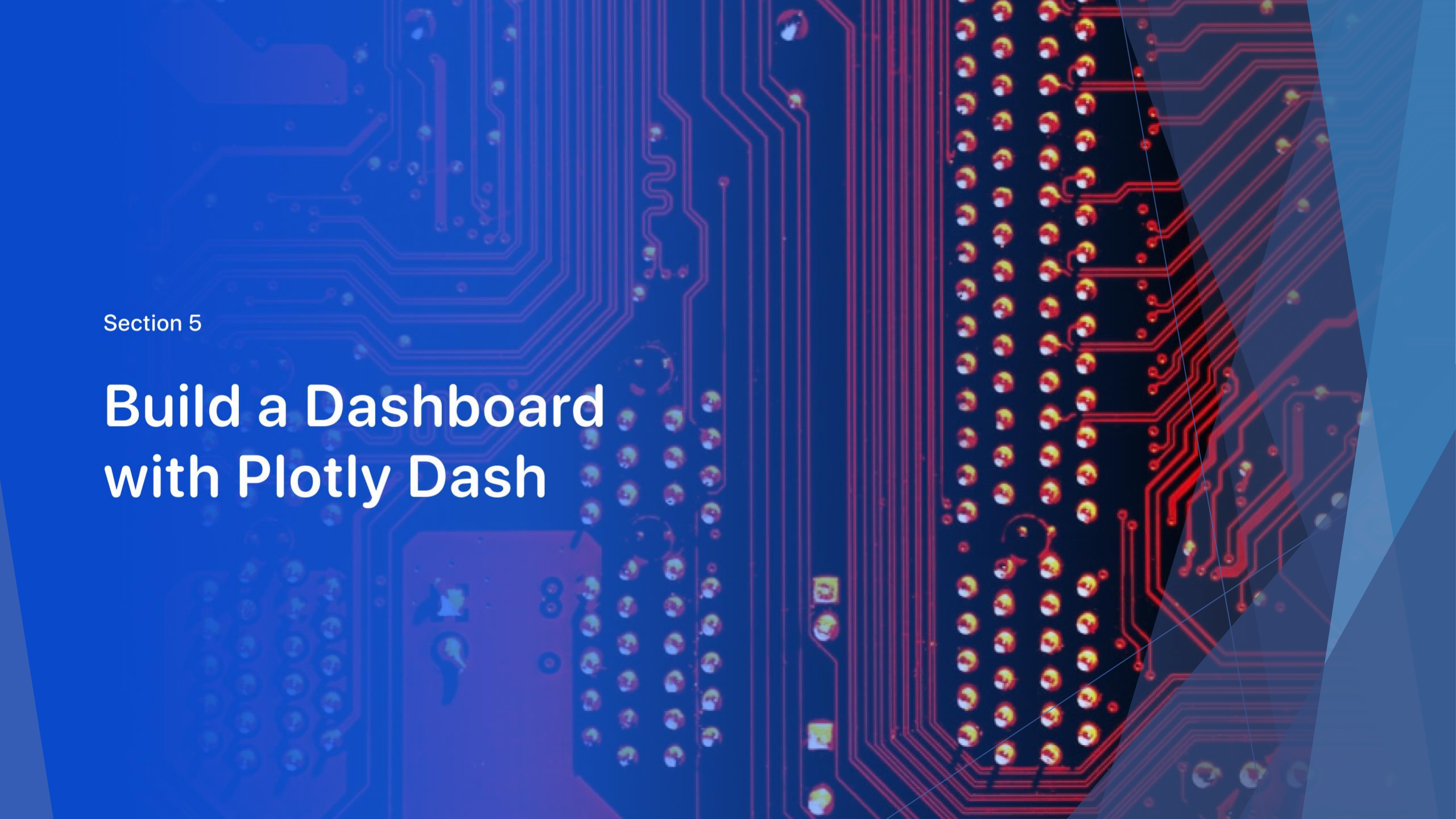
City 72.39 km



Coastline 1.40 km

Railway 1.38 km

Distance for all launch sites from cities is greater 14 km, so they are far away from cities. Launch failures can land in the sea to avoid densely populated areas.

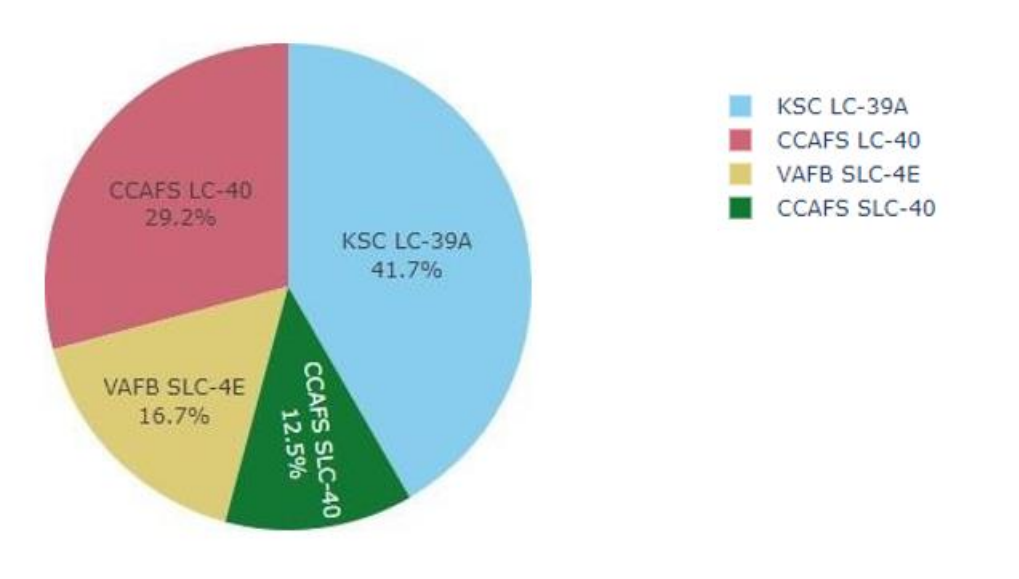


Section 5

Build a Dashboard with Plotly Dash

Launch Success Count for all Sites

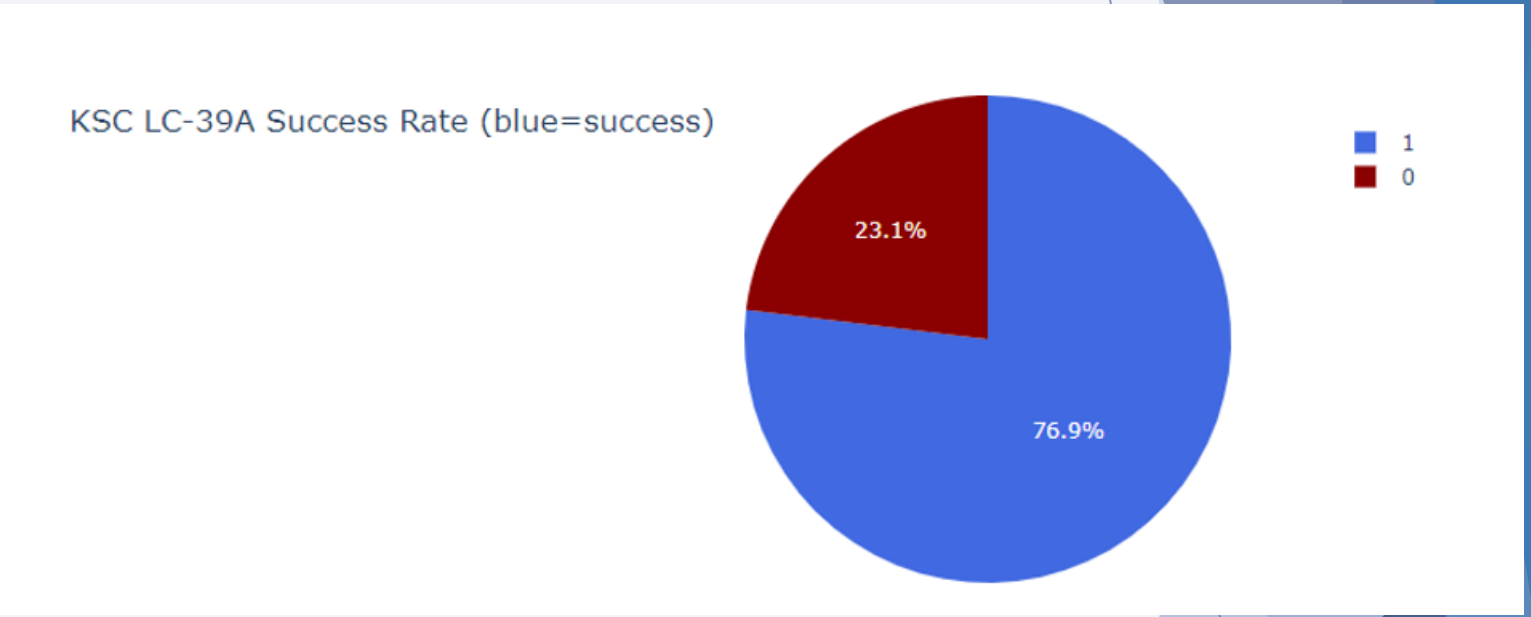
This is the distribution of successful landings across all launch sites.



- **We can see that KSC LC-39A has the most successful landings.** VAFB has the smallest share of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast.

Highest Success Rate Launch Site

KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.



Payload Mass vs. Launch Outcome Scatter Plot

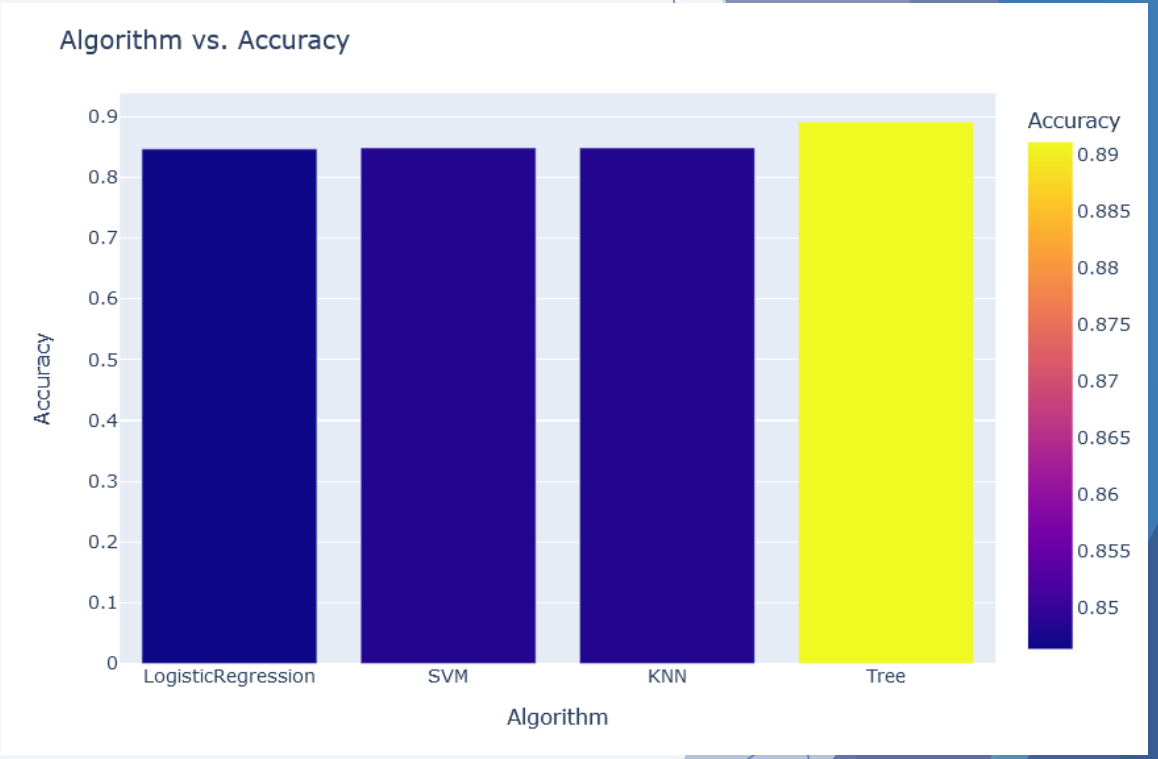
- the y-achses (class) indicated 1 for successful landing and 0 for failure.
- The payload range selector is set from 0-10000.
- We can see the success rated for low weighted payloads is higher than the heavy Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider.
- Interestingly there are two failed landings with payloads of 0 kg.

Section 6

Predictive Analysis (Classification)

Classification Accuracy

- All models had the same accuracy on the test set at 83.33% accuracy.
- The models which performed the test was **decision tree** with an accuracy of 0.90178.
- Even though the accuracies were extremely close, Decision Tree was the best.



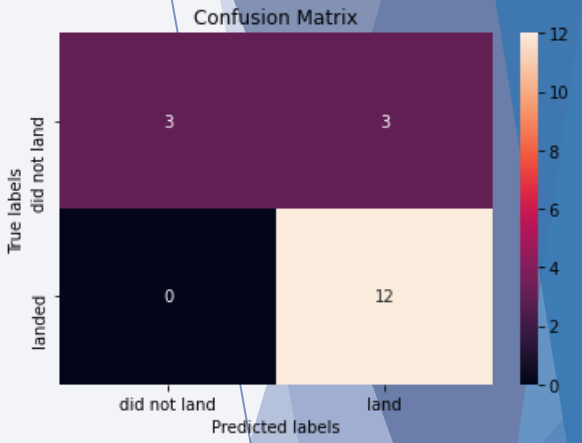
Confusion Matrix

Since all models performed the same for the test set, the confusion matrix is the same across all models.

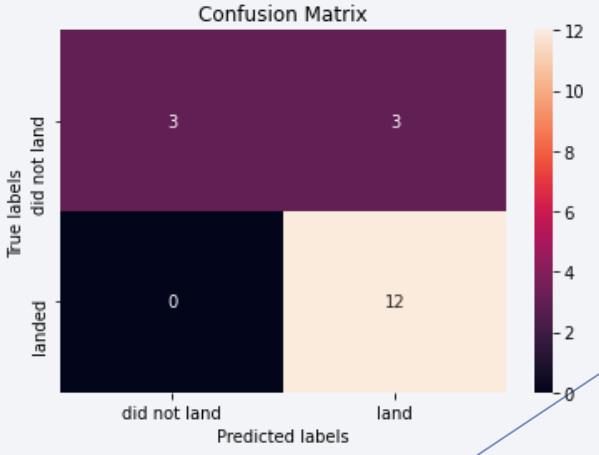
- The models predicted 12 successful landings when the true label was successful landing.
- The models predicted 3 unsuccessful landings when the true label was unsuccessful landing.
- The models predicted 3 successful landings when the true label was unsuccessful landings (false positives).

Logistic Regression

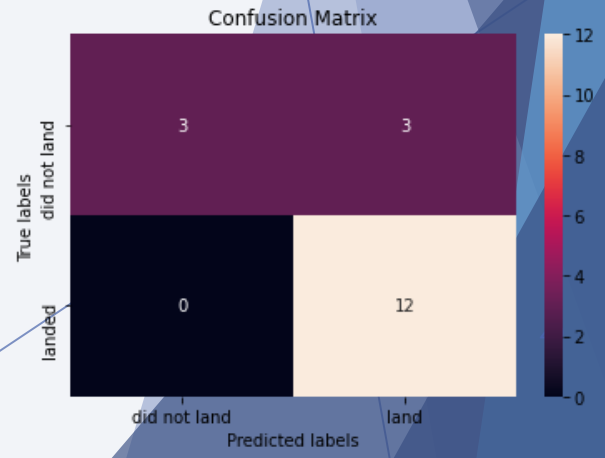
KNN



SVM



Decision Tree



Conclusions

- Our main goals was to develop a machine learning model and find the best predictor where Stage 1 launches will successfully land.
- We used data from a public SpaceX API and also web scraping SpaceX Wikipedia page
- After analyzing and visualizing the outcome, we found that the decision tree classifier is the best for machine learning for this provided dataset.
- The success rated for SpaceX launches is proportional with time and they will eventually reach the perfect launches.
- Low weights payloads perform better than the heavier payloads
- Orbits ES-L1, GEO, HEO, SSO has highest success rates
- KSC LC-39A had the most successful launches but increasing payload mass seems to have negative impact on success

Appendix

- Github respository url: https://github.com/etudehome/IBM-Data-Science-
- All materials and tasks available in IBM Data Science Cousera: https://www.coursera.org/professional-certificates/ibm-data-science

Thank you!