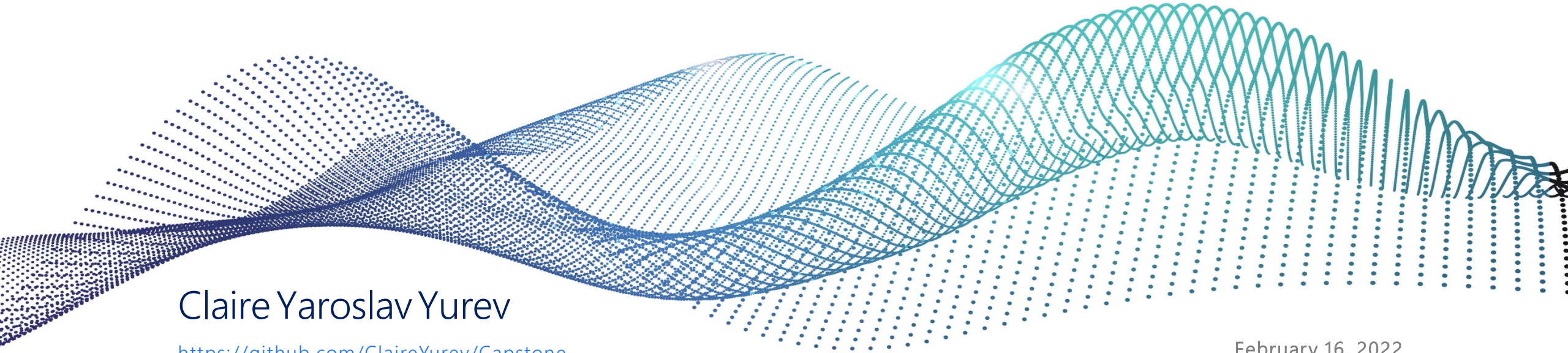
Data Science Capstone Project



Claire Yaroslav Yurev

https://github.com/ClaireYurev/Capstone

February 16, 2022

Outline

Executive Summary: Methodologies

- The goal of this project is to predict whether the Falcon 9 first stage will land successfully, thus enabling the first stage to be reused. The ability to predict a successful landing with significant accuracy is key to enabling a meaningful and productive cost planning.
- All data collection, gathering, and measurement has been performed using publicly available sources for educational use: SpaceX Open Source REST API Github Repository, SpaceX.com, as well as the historical launch records for SpaceX Falcon9 modules from Wikipedia.org.
- We utilize Data Collection, Web Scraping, Data Wrangling using Exploratory Data Analysis (EDA), and SQL to:
- We obtain data by sending requests to the SpaceX API, then clean the requested data.
- We then web scrap using BeautifulSoup by extracting launch records from a Wikipedia HTML table, parsing and converting it into a Pandas data frame. At this point we begin exploratory data analysis and determine training labels. We then take standalone SpaceX dataset in CSV format and load it directly into a new table within an IBM Cloud Db2 database, where we execute SQL queries to get a closer look at the information within the dataset.

Executive Summary: Results

- Finally we utilize Pandas, along with the Matplotlib and Seaborn libraries to continue our exploratory data analysis, where we build preliminary insights about how each of the important variables would affect the ultimate success rate. This moves us onto feature engineering where we then select the features that will be used in forthcoming success prediction.

- We then created the labels column “class” such that it classifies successful landings. We go on to explore the data using structured query language (SQL), visualization, folium maps, and dashboards. The relevant and appropriate columns are gathered in order to be used as features. Then we proceed to change all categorical variables into binary ones using one-hot encoding. Data is then standardized and we utilize GridSearchCV in order to find the best parameters for machine learning models. At last, we finally visualize the accuracy scores of all models.

- Four machine learning models were produced: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and the K-Nearest Neighbors. All models have produced similar results with accuracy rate in the range of approximately 83.33%. Each of the models has over-predicted successful landings. As a result, in order to improve model determination and accuracy, more data is needed.

Introduction

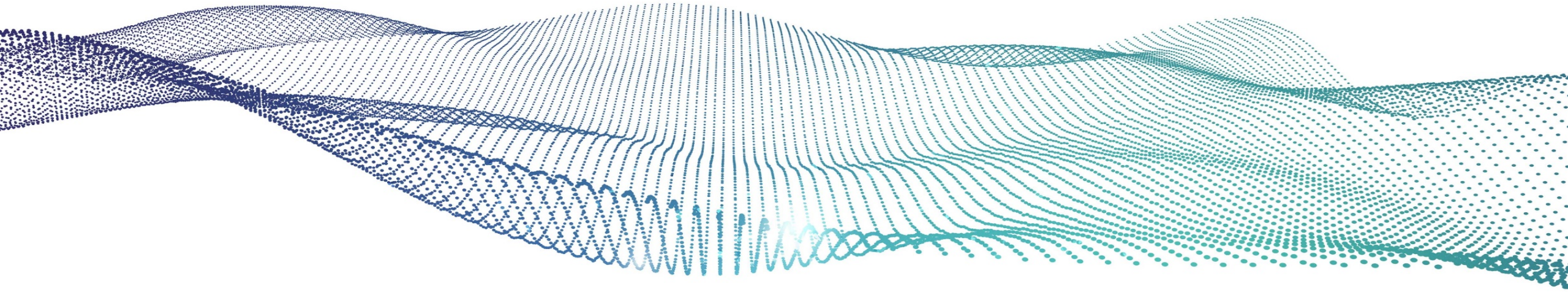
- With RocketLab, VirginGalactic, and Blue Origin, the commercial space age has arrived.
- SpaceX has the best operational pricing, at approximately $62 million vs. $165 million USD.
- SpaceX has been able to lower its costs mostly due to their ability to recover stage 1 of each rocket.
- SpaceY is a new company that is entering the competition against SpaceX.

SpaceX Falcon 9

Introduction: The Challenge

- Out job at SpaceY is to determine the price of each launch. We will do this by gathering information about SpaceX, and creating dashboards for our team. We will also determine if SpaceX will reuse the first stage of a given rocket.
- The challenge: SpaceY has commissioned us to us to train a new machine learning model and use public information to predict successful recovery of Stage 1 part of the rocket.

SpaceX Falcon 9

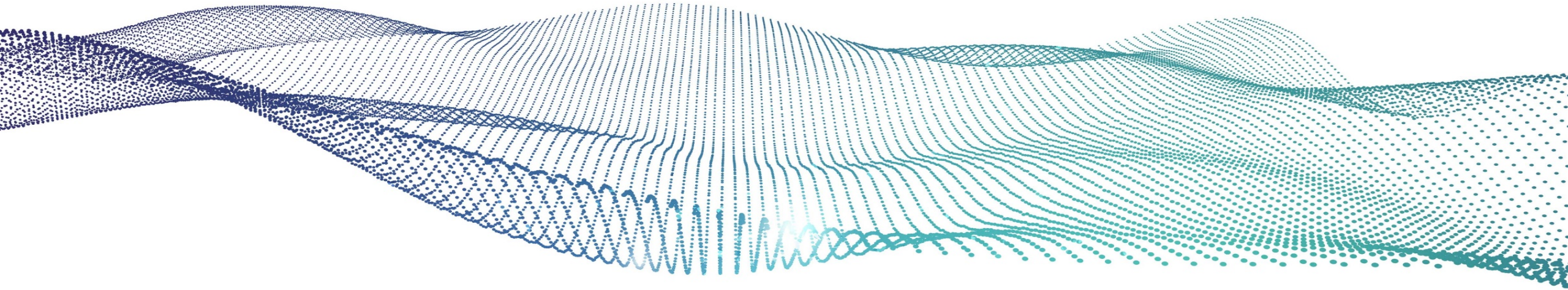


Methodology

- Data Collection and Data Wrangling Methodology
- EDA and Interactive Visual Analytics Methodology
- Predictive Analysis (Classification) Methodology

Methodology - Overview

- Data collection methodology:
  - ✓ We combine the data from SpaceX public API as well as from the SpaceX Wikipedia page
- Data wrangling methodology:
  - ✓ We classify each and every true stage 1 landing as either a successful or an unsuccessful event
- Exploratory data analysis (EDA) methodology:
  - ✓ We perform the exploratory data analysis stage through the use of Structured Query Language (SQL)
- Interactive visual analytics methodology:
  - ✓ We perform the interactive visual analytics stage of our work through the use of Folium and Plotly Dash
- Predictive analysis methodology through the use of classification models:
  - ✓ We perform the this by building the model, then tuning our model using GridSearchCV, and evaluating the results



Methodology

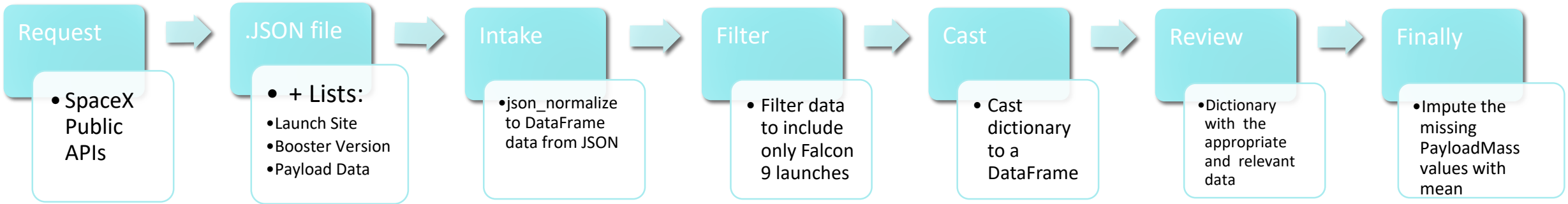
•Data Collection

Data Collection

- Our data collection process has involved a combination of both the API requests from SpaceX public API, as well as web scraping data that was structured as a table from within the SpaceX Wikipedia page.
- What follows on the next slide is the flowchart of our data collection process from the public API.
- On the subsequent slide we present the flowchart of our data collection from web scraping.
- The Data Columns sourced from within the SpaceX API:
  - ✓ FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude.
- The Data Columns sourced from web scraping the Wikipedia page:
  - ✓ Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time.

Methodology

• Data Collection: SpaceX API

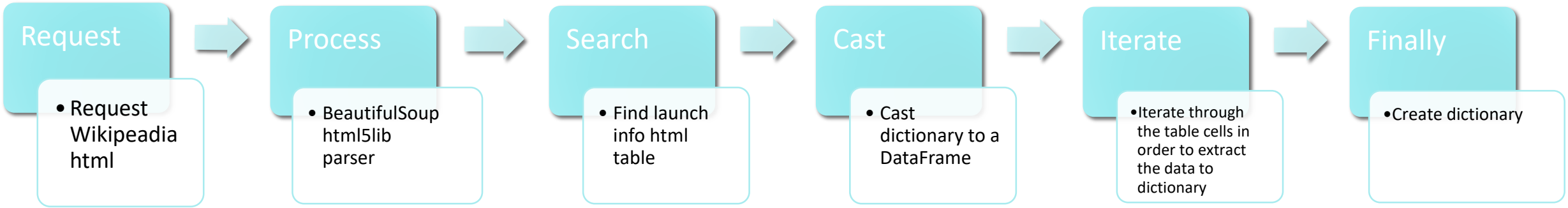


GitHub Link to the Jupyter Notebook: 🗒 **SpaceX Data Collection REST API.ipynb**

➢ *https://github.com/ClaireYurev/Capstone/blob/master/SpaceX%20Data%20Collection%20REST%20API.ipynb*

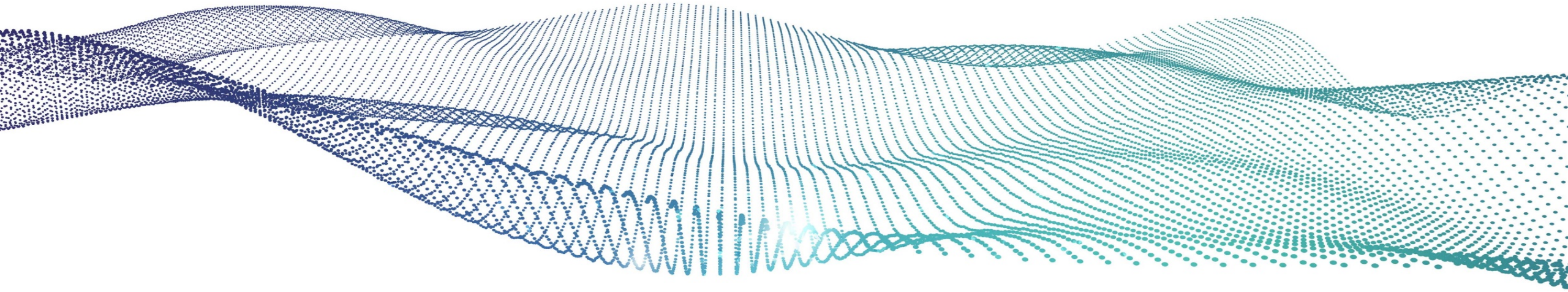
Methodology

• Data Collection: Web Scraping



GitHub Link to the Jupyter Notebook: 🗒 **Data Collection API with Webscraping.ipynb**

➢ *https://github.com/ClaireYurev/Capstone/blob/master/Data%20Collection%20API%20with%20Webscraping.ipynb*

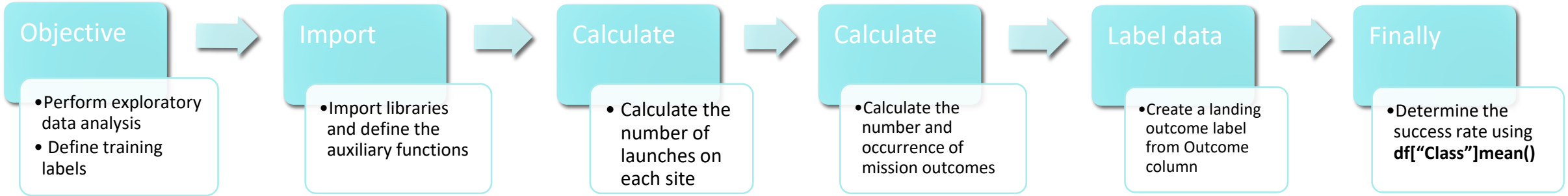


Methodology

•Data Wrangling

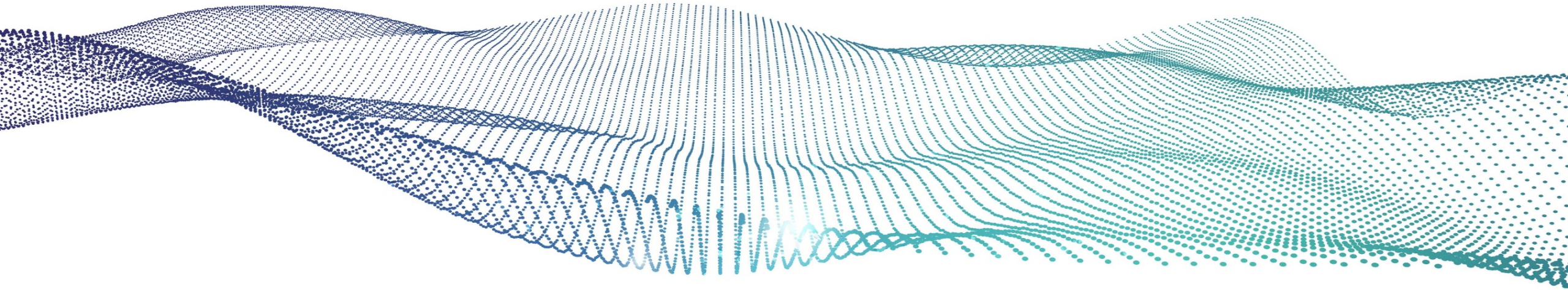
Data Wrangling - Methodology

- The data has been processed in the following manner:
- We created a training label with landing outcomes, such that a "successful" outcome = 1, & "failure" = 0.
- The outcome column has two components: "Mission Outcome" and the "Landing Location".
- A new training label column "class" has been created:
  - a successful "Mission Outcome" = 1, otherwise it is = 0.
- Value Mapping:
  - True ASDS, True RTLS, & True Ocean ~ set to → 1
  - None None, False ASDS, None ASDS, False Ocean, False RTLS ~ set to → 0



GitHub Link to the Jupyter Notebook: 🗒 [SpaceX REST API Data Wrangling.ipynb]

➢ *https://github.com/ClaireYurev/Capstone/blob/master/SpaceX%20REST%20API%20Data%20Wrangling.ipynb*



Methodology

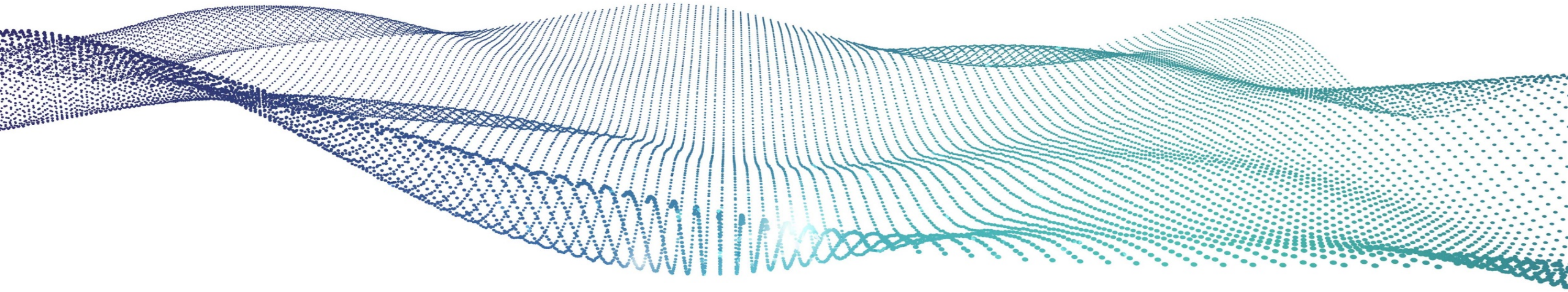
• EDA & Interactive Visual Analytics

EDA with Data Visualization - Methodology

- In this stage, Exploratory Data Analysis has been performed on variables such as: Flight Number, Payload Mass, Launch Site, Orbit, Class and Year.
- We then plotted the following charts:
- Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Success Rate vs. Orbit Type, Flight Number vs. Orbit Type, Payload vs Orbit, and Success Yearly Trend.
- We used scatter plots and bar plots to analyze and compare relationships between variables in order see if a relationship exists, so that they could be used in training the machine learning model.
- We used scatter plots with different colors (on each plot) in order to see if multiple different variables would affect the launch outcomes.
- We used bar charts (with single color) in order to visually check if there were any relationships between success rate and orbit type.

GitHub Link to the Jupyter Notebook: 🗒 **SpaceX EDA using Pandas and Matplotlib.ipynb**

➢ *https://github.com/ClaireYurev/Capstone/blob/master/SpaceX%20EDA%20using%20Pandas%20and%20Matplotlib.ipynb*



Methodology

•EDA with SQL

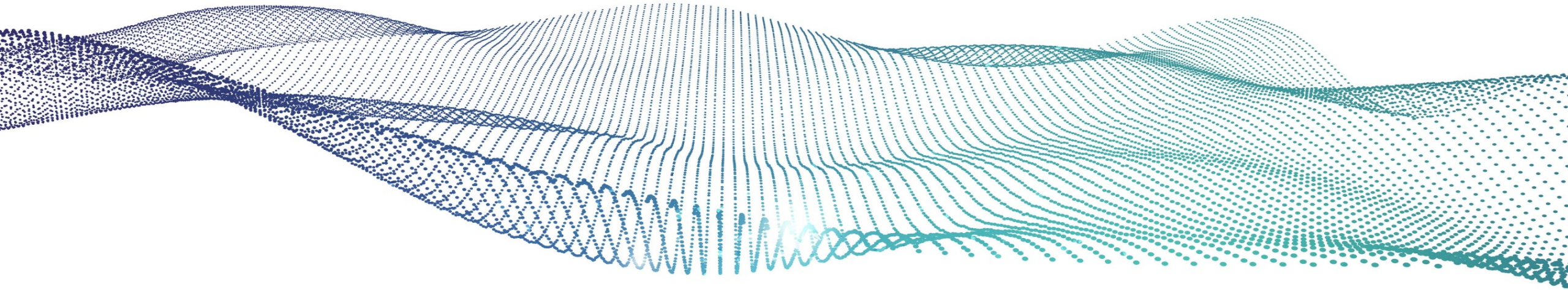
EDA with SQL

We have gone ahead and loaded the dataset into an IBM DB2 Database, querying it using SQL Python integration from within the Jupyter notebook using queries resulting in the following information:

- We displayed the names of the unique launch sites in the space mission
- We displayed 5 records where launch sites begin with the string 'CCA'
- We displayed the total payload mass carried by boosters launched by NASA (CRS)
- We displayed average payload mass carried by booster version F9 v1.1
- We listed the date when the first successful landing outcome in ground pad was achieved
- We listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- We listed the total number of successful and failure mission outcomes
- We listed the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- We listed the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- We ranked the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between 2010-06-04 & 2017-03-20

GitHub Link to the Jupyter Notebook: 🗒 **SpaceX Dataset - EDA with SQL in IBM Db2.ipynb**

➢ *https://github.com/ClaireYurev/Capstone/blob/master/SpaceX%20Dataset%20-%20EDA%20with%20SQL%20in%20IBM%20Db2.ipynb*

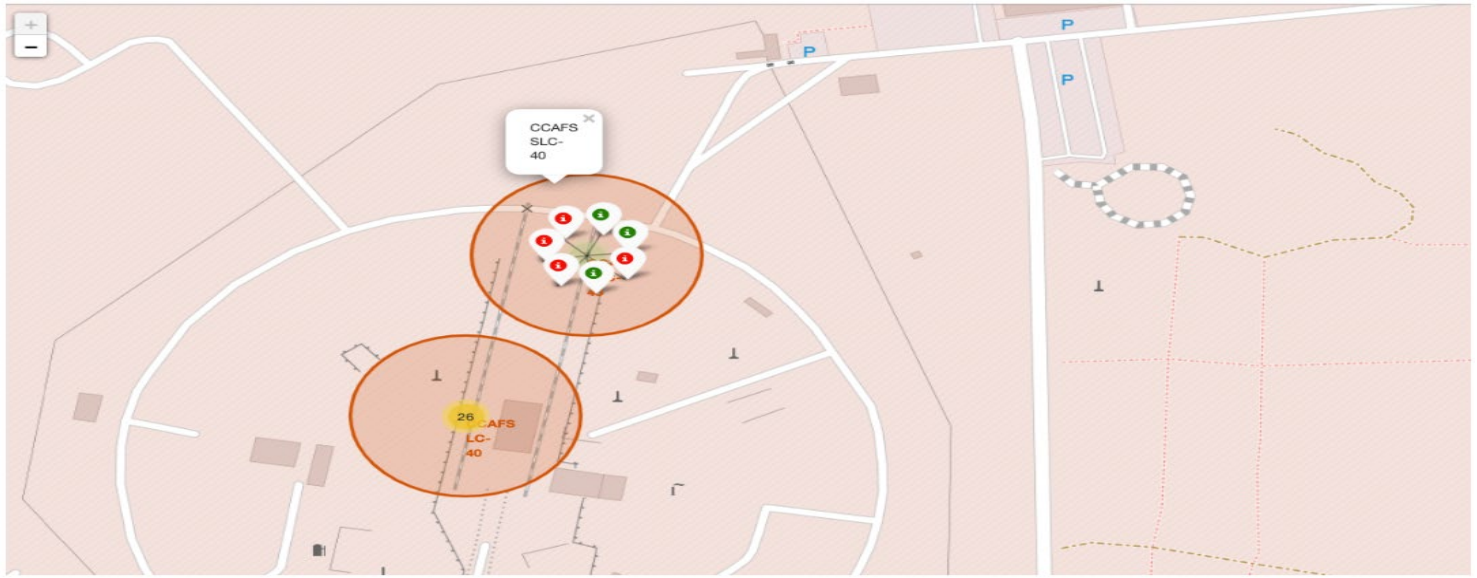


Methodology

•EDA: Interactive Map with Folium

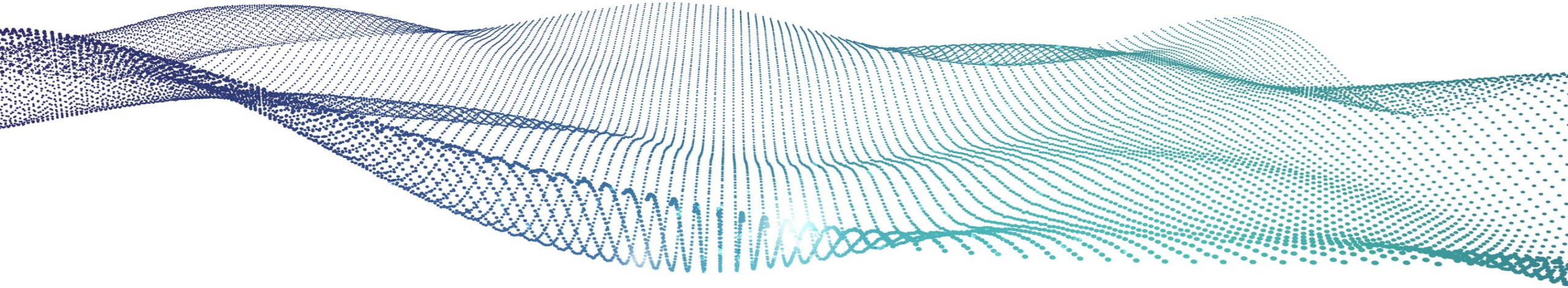
Interactive Map with Folium

- Folium maps visually denote Launch Sites, Successful and Unsuccessful landings, as well as a proximity example to key locations: Railway, Highway, Coast, and City.
- We added these objects because this allows us to better understand why launch sites are located where they are. This visualization also paints a picture of successful landings relative to location.



GitHub Link to the Jupyter Notebook: 🗒 **Interactive Visual Analytics with Folium.ipynb**

➢ *https://github.com/ClaireYurev/Capstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium.ipynb*



Methodology

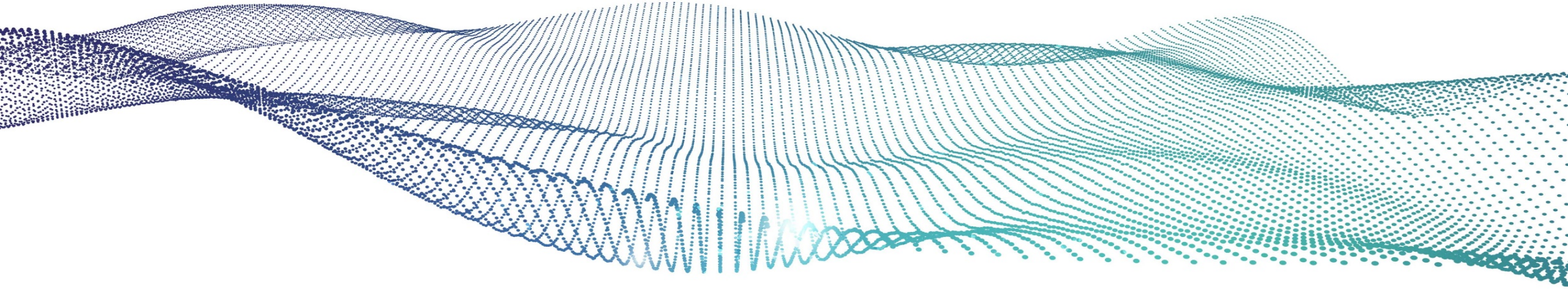
•EDA: Dashboard with Plotly Dash

Dashboard with Plotly Dash

- First, we added a pie chart and a scatter plot to the dashboard.
- Pie chart can be selected to show distribution of successful landings across all launch sites, and it can also be selected to show individual launch site success rates.
- Scatter plot charts two inputs: all sites or an individual site, and pay load mass on a slider between 0 and 10000 kilograms.
- The pie chart can be utilized to visualize launch site success rate.
- The scatter plot can help us understand visually how success varies across launch sites, pay load mass, and booster version categories.

GitHub Link to the Python File: 🗒 **SpaceX Dashboard with Plotly Dash.py**

➢ *https://github.com/ClaireYurev/Capstone/blob/master/SpaceX%20Dashboard%20with%20Plotly%20Dash.py*



Methodology

•EDA: Predictive Analysis - Classification

EDA: Predictive Analysis - Classification

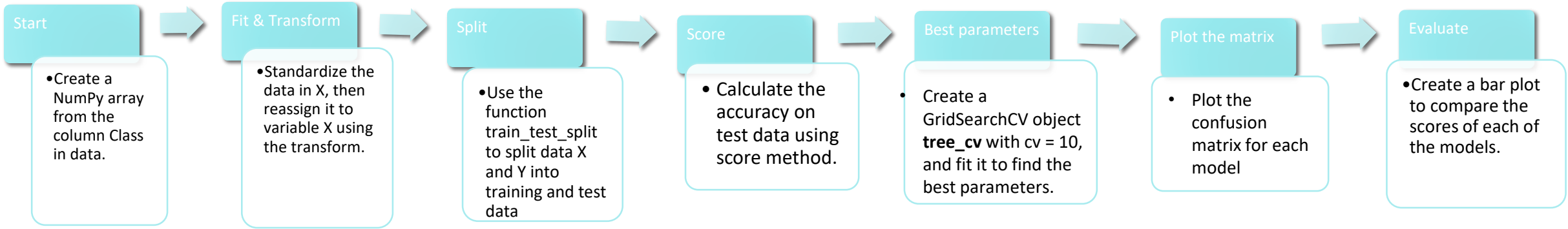
- Problem: Most unsuccessful landings are planned

- **Our Goal: Perform exploratory Data Analysis and determine Training Labels**
- We achieve this using the following process:
  1. We create a column for the class
  2. We standardize the data
  3. We then split the data into training data and test data

- We then find best Hyperparameter for SVM, Classification Trees and Logistic Regression. This enables us to find the method that performs best using test data.

Predictive Analysis

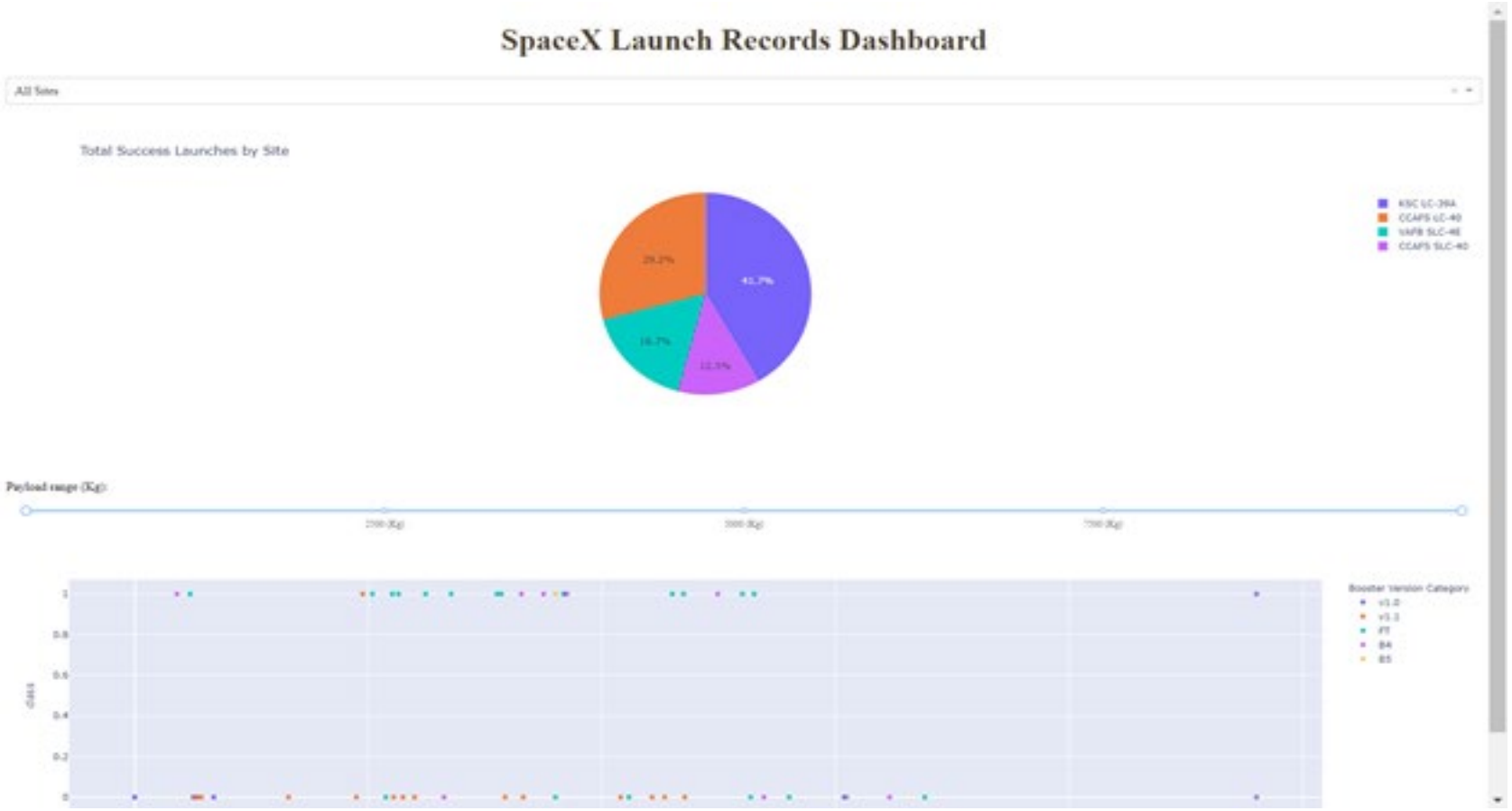
•Classification



GitHub Link to the Jupyter Notebook: 🗒 **Predictive Analysis Lab - ML Prediction.ipynb**

➢ *https://github.com/ClaireYurev/Capstone/blob/master/Predictive%20Analysis%20Lab%20-%20ML%20Prediction.ipynb*

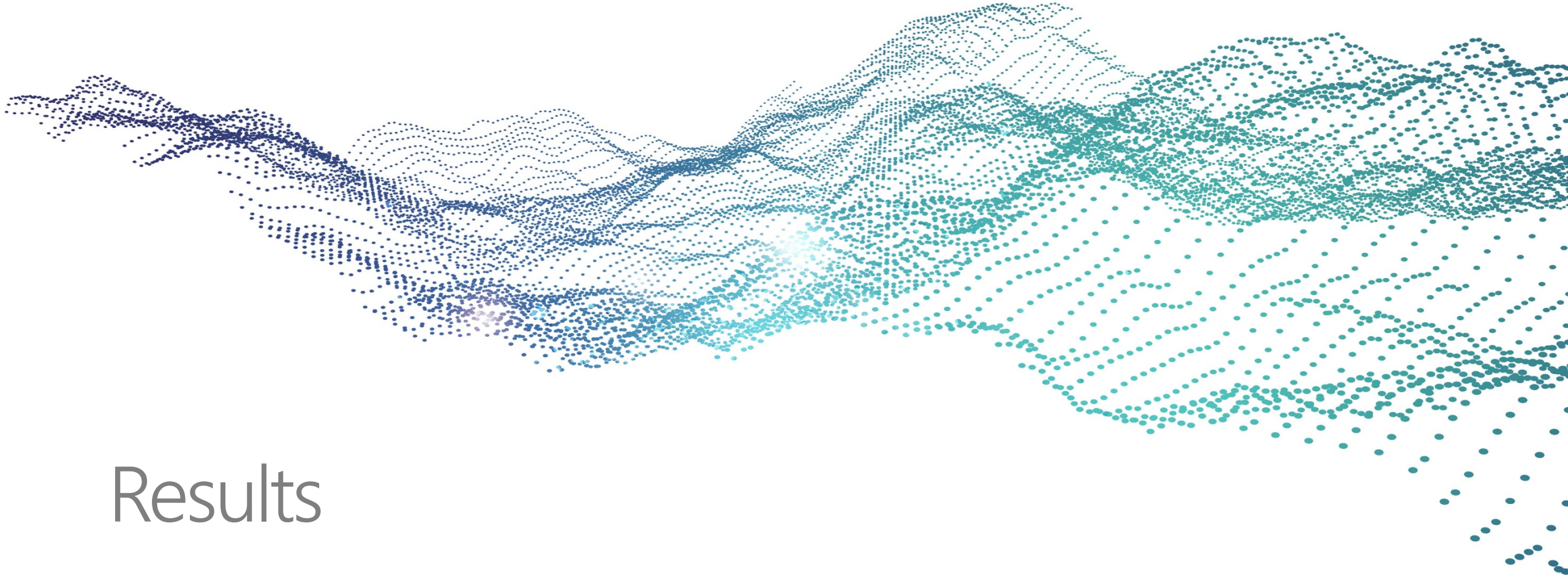
EDA: Predictive Analysis – Classification - Results



- This is the preview of Plotly dashboard. The detailed results of EDA with Visualization, SQL, Interactive Map with Folium, and model results (of ~83% accuracy) will come in the following slides.

Insights Drawn From EDA

- EDA with Visualization: results
- EDA with SQL in Db2: results
- Interactive map with Folium: results
- Dashboard with Plotly Dash: results
- Predictive analysis – classification: results

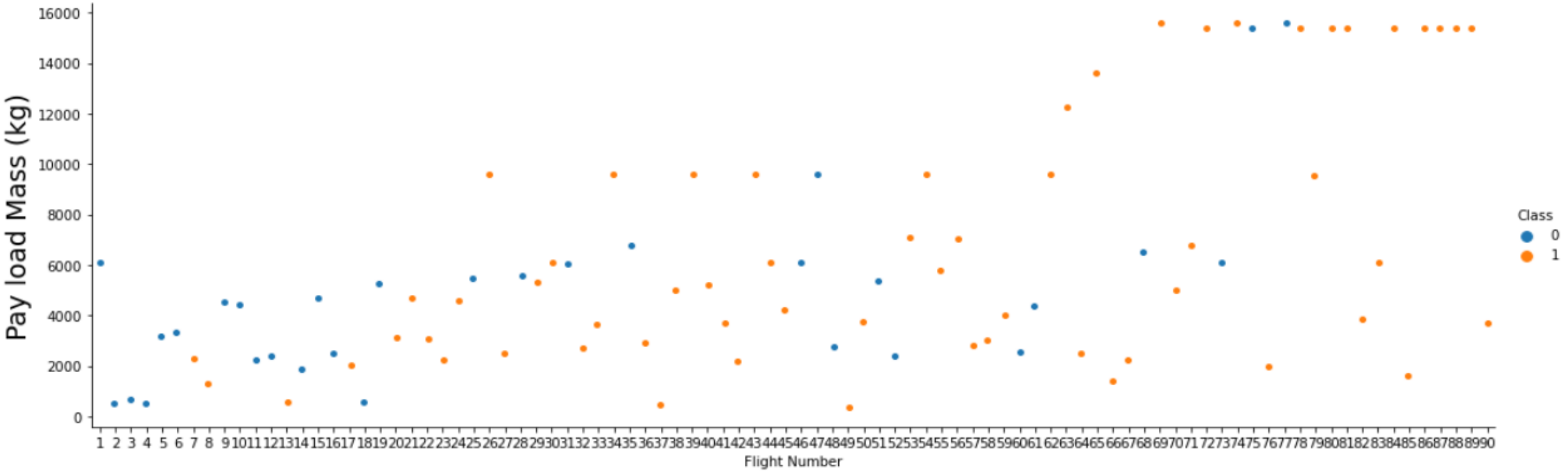


Results

•EDA with Visualization

Results: EDA with Visualization

- Exploratory data analysis with Seaborn plots: Payload Mass vs Flight Number



(*Orange* indicates successful launch, *Blue* indicates unsuccessful launch.)

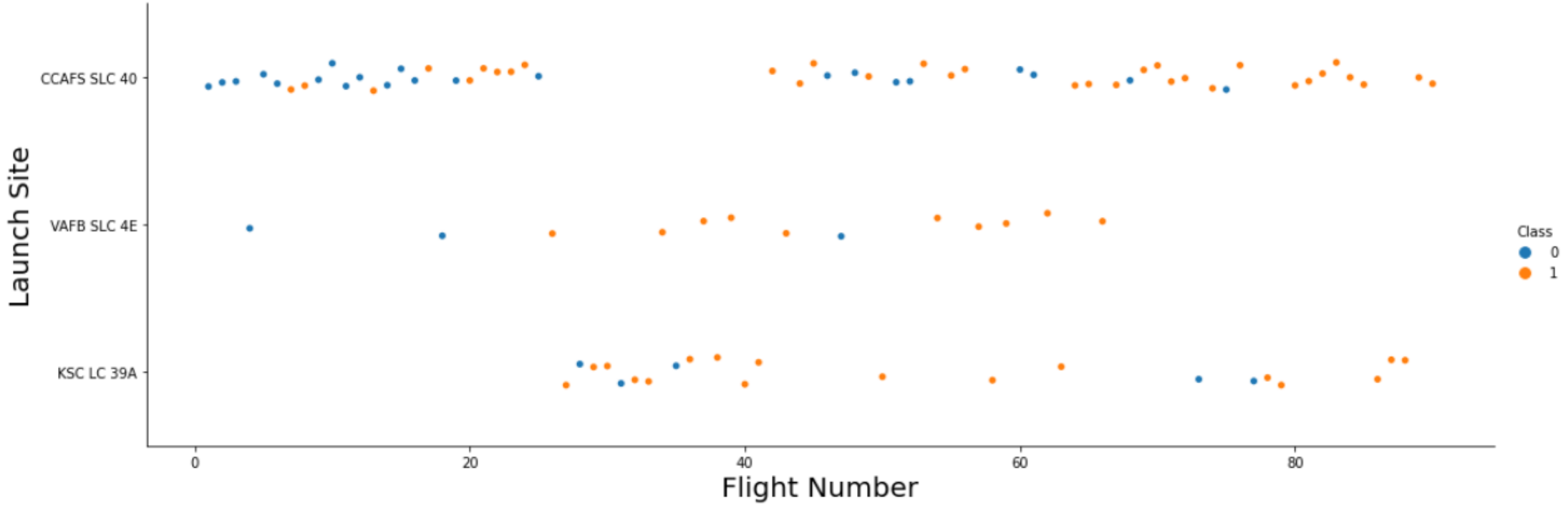
The above results suggest that as the flight number increases, the 1st stage is more likely to land successfully. Payload mass is also important: it appears the more massive the payload, the less likely the first stage will return.

GitHub Link to the Jupyter Notebook: **SpaceX EDA using Pandas and Matplotlib.ipynb**

➢ *https://github.com/ClaireYurev/Capstone/blob/master/SpaceX%20EDA%20using%20Pandas%20and%20Matplotlib.ipynb*

Results: EDA with Visualization

- Exploratory data analysis with Seaborn plots: Flight Number vs Launch Site

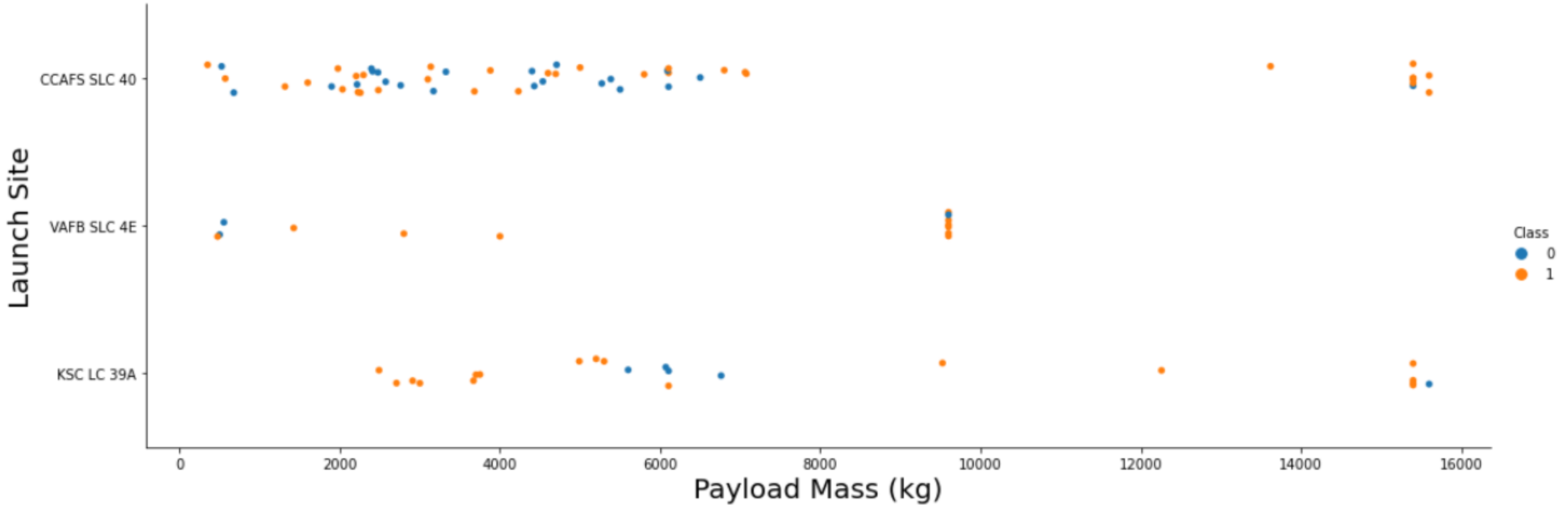


(*Orange* indicates successful launch, *Blue* indicates unsuccessful launch.)

The above results suggest an increase in success rate over time (as indicated by Flight Number). It is likely there is a big breakthrough around flight 20, which has increased success rate significantly. In terms of location, the CCAFS appears to be the primary launch site, because it has the most volume.

Results: EDA with Visualization (cont'd)

- Exploratory data analysis with Seaborn plots: Payload Mass vs Launch Site

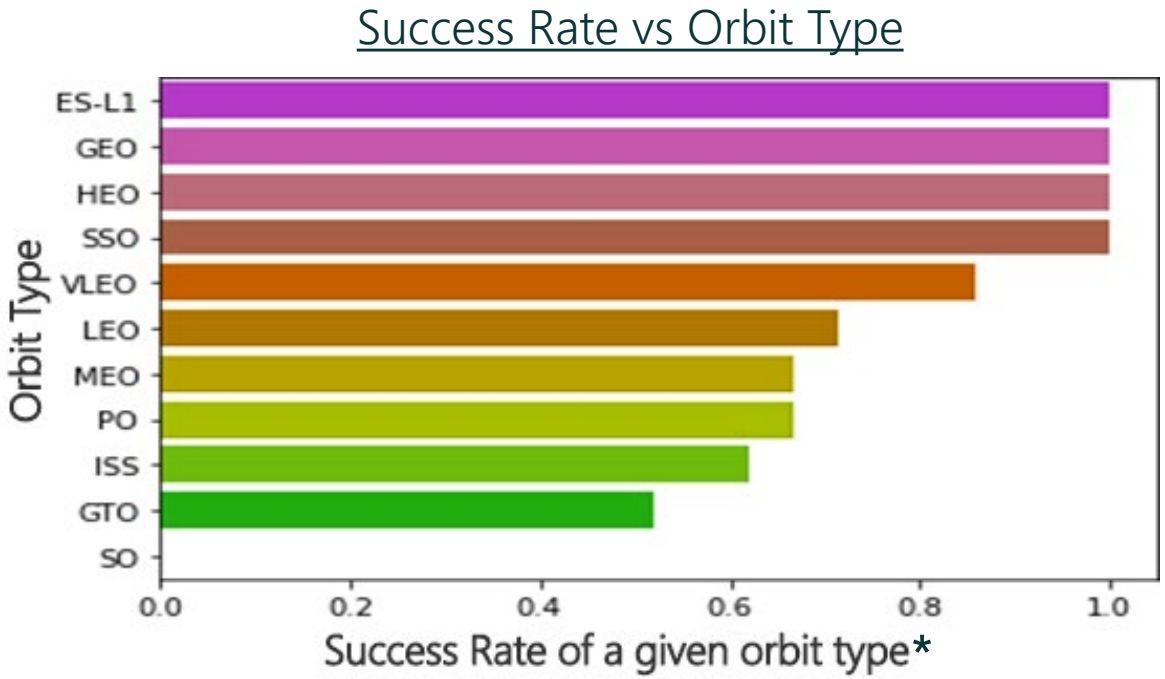


(*Orange* indicates successful launch, *Blue* indicates unsuccessful launch.)

Pay load weight (mass) appears to fall mostly within the range between 0 kilograms and 6000 kilograms.
In terms of launch site location, different launch sites appear to utilize different payload masses.

Results: EDA with Visualization (cont'd)

- Exploratory data analysis with Seaborn plots:

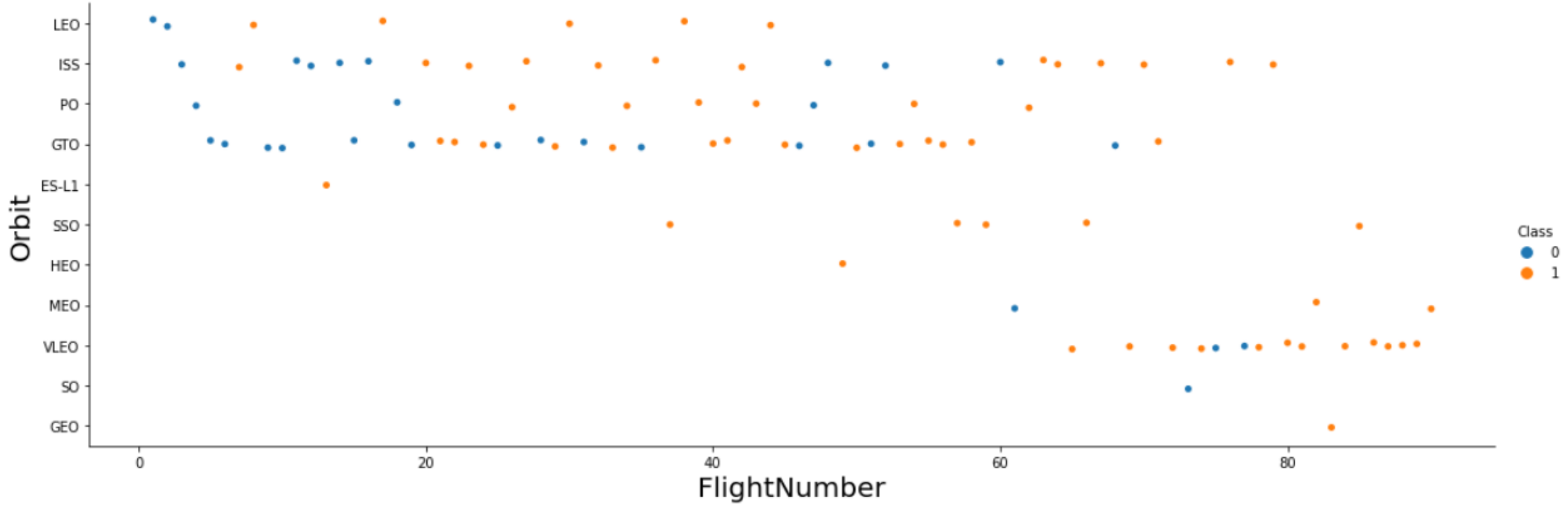


** Horizontal axis: "0.2" represents Success Rate of 20%, etc.*

- ES-L1 (1), GEO (1), HEO (1), and SSO (5) have 100% success rate (sample sizes in brackets).
- VLEO (14) has reasonably satisfactory success rate and attempts.
- SO (1) has 0% success rate.
- GTO (27) has a success rate of only approximately 50%, however it does come with the largest sample at 27.

Results: EDA with Visualization (cont'd)

- Exploratory data analysis with Seaborn plots: Flight Number vs Orbit Type

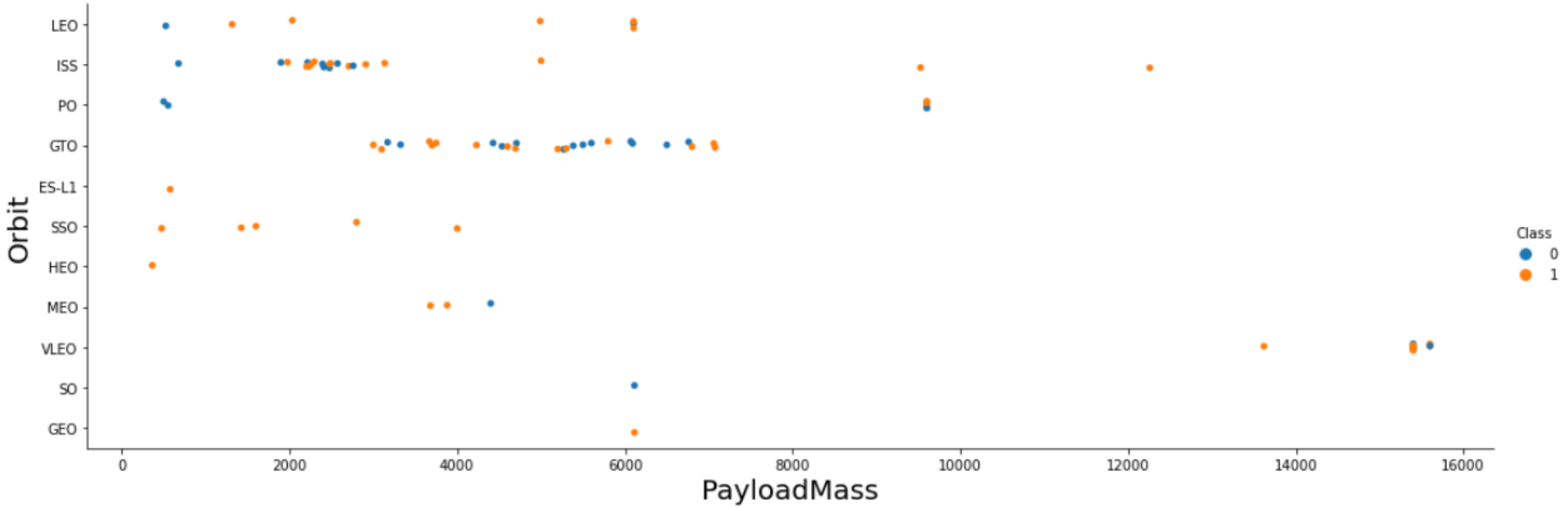


(*Orange* indicates successful launch, *Blue* indicates unsuccessful launch.)

The Launch Orbit preferences appear to have changed per Flight Number. Final Launch Outcome appears to meaningfully correlate with this preference. The data tells the story: SpaceX has begun with LEO orbits, and saw a moderate success rate. SpaceX has returned to VLEO in the more recent launches. According to the data that we observed, it would appear that SpaceX launches generally tend to perform better in lower orbits or sun-synchronous orbits overall.

Results: EDA with Visualization (cont'd)

- Exploratory data analysis with Seaborn plots: Payload Mass vs Orbit Type



(*Orange* indicates successful launch, *Blue* indicates unsuccessful launch.)

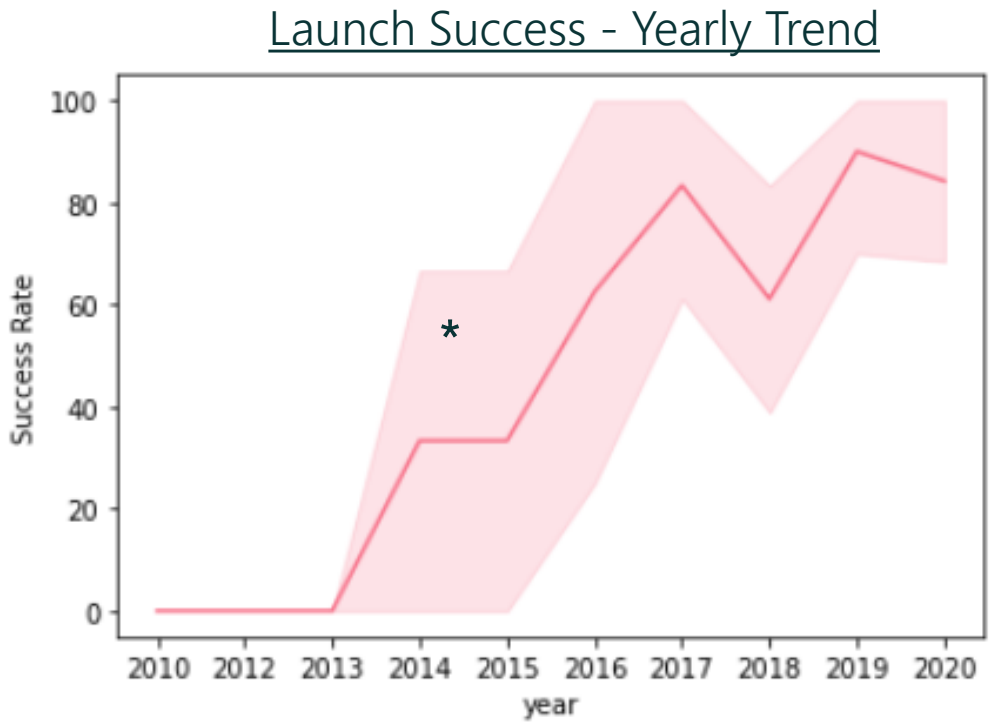
The data suggests that the Payload Mass appears to correlate with Orbit Type

According to this data, both the SSO and LEO orbit types appear to have relatively low Payload Mass.

The next-best successful orbit type is VLEO, and it has Payload Mass values in upper end of the range.

Results: EDA with Visualization (cont'd)

- Exploratory data analysis with Seaborn plots:

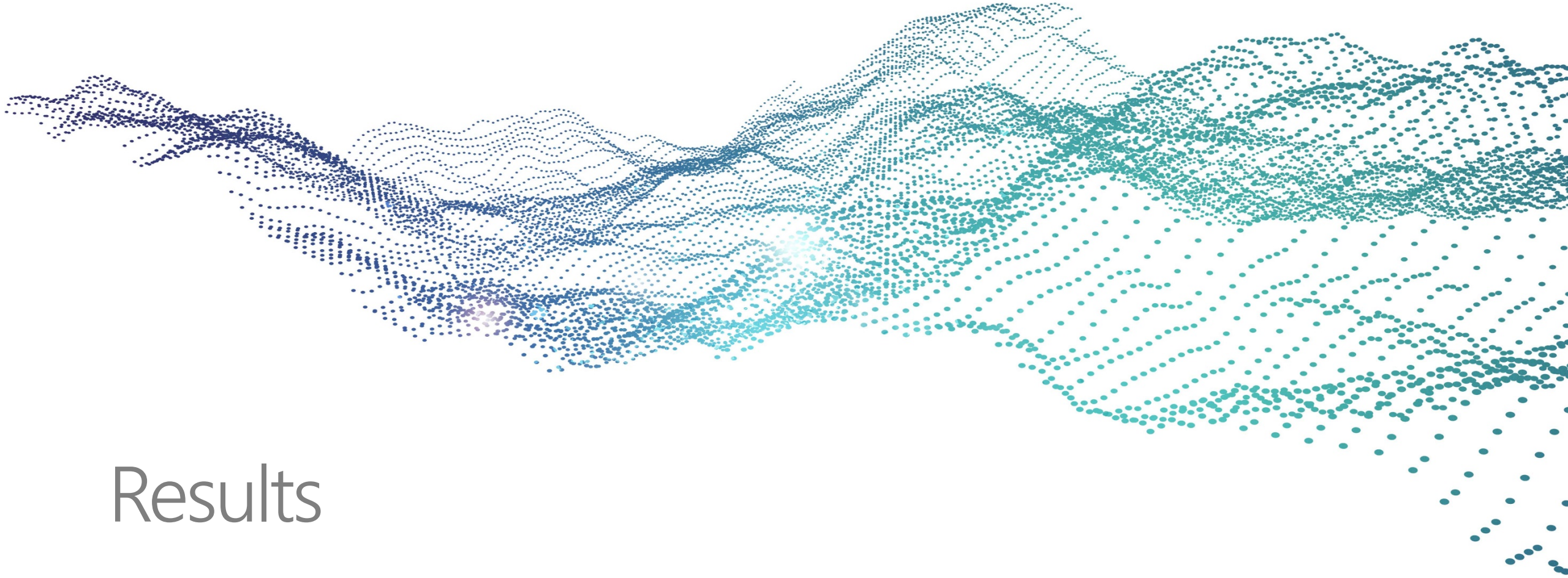


* Area shaded in light pink indicates the 95% confidence interval.

Available data suggests that success rate generally increases over time. In our dataset, we observed a continuous increase from 2013 and on, with only a small temporary dip in the year 2018. Success in the most recently available years appears to be stabilizing at approximately 80%.

GitHub Link to the Jupyter Notebook: 🗒 **SpaceX EDA using Pandas and Matplotlib.ipynb**

➢ *https://github.com/ClaireYurev/Capstone/blob/master/SpaceX%20EDA%20using%20Pandas%20and%20Matplotlib.ipynb*



Results

- EDA with SQL in Db2

SQL Results: All Launch Site Names

- All of the following exploratory data analysis is performed with SQL in Python, enabled by SQLAlchemy:

```
In [8]: %%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL;
```

```
 * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

Out[8]:

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

- *Here we query unique launch site names from the Db2 database.*
- *There are a total of **3** unique physical launch sites for rocket launches:*

  *the **(S)LC-40**, the **LC-39A**, and the **SLC-4E**.*
- *The reason the output lists **4** different launch site names is because "CCAFS LC-40" was the previous name of the "CCAFS SLC-40" launch site.*

SQL Results: Launch Site Names That Begin with “CCA”

- The following process finds 5 records where launch sites begin with `CCA`:

In [10]:
```
%%sql
SELECT *
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```

```
 * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

Out[10]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- *Here we find and display the first **five** records where the launch site name begins with ‘CCA’.*

SQL Results: Total Payload Mass

- The following process calculates the total payload carried by boosters launched by NASA:

```
In [14]: %%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

Out[14]:

payload_mass__kg_
45596

- *Here we calculate the total payload carried by boosters launched by NASA (CRS), by using the SQL command **SUM**. The result is 45,596 kilograms of 'total' payload.*
- *The CRS acronym stands for "Commercial Resupply Services", indicating that these specific payloads were delivered to the International Space Station (ISS).*

SQL Results: Average Payload Mass by F9 v1.1

- The following process calculates the average payload mass carried by booster version F9 v1.1:

```
In [15]: %%sql
         SELECT AVG(PAYLOAD_MASS__KG_) AS PAYLOAD_MASS__KG_
         FROM SPACEXTBL
         WHERE Booster_Version = 'F9 v1.1';

          * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
         Done.

Out[15]: payload_mass__kg_
                     2928
```

- *Here we calculate the* average payload mass carried by booster version F9 v1.1, using the SQL command **AVG**. *The result is 2,928 kilograms of 'average' payload mass.*
- *As far as average payload mass goes within all of the payload masses explored in our project, this average payload mass of F9 v1.1 (2,928 kilograms) is actually around the low end of the scale.*

SQL Results: First Successful Ground Landing Date

- The following process finds the dates of the first successful landing outcome on ground pad:

```
In [17]: %%sql
         SELECT MIN(Date) AS FIRST_SUCCESS
         FROM SPACEXTBL
         WHERE Landing__Outcome = 'Success (ground pad)';

          * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
         Done.
```

Out[17]:

first_success
2015-12-22

- *Here we use the SQL's **MIN** function to find the date of the first successful landing outcome on ground pad.*
- *First successful ground pad landing took place at the end of year 2015. In general, successful landings began to appear in 2014.*

SQL Results: Successful Drone Ship Landing with Payload between 4000 and 6000

(continued on the next page)

- The following process finds the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 kilograms (inclusively):

```
In [28]: %%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
 * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

Out[28]:

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

✓ *It is very important to understand the results of our query!* Here we use SQL's ***BETWEEN*** operator to select values within a given range. The ***BETWEEN*** operator is inclusive: the ***begin*** and ***end*** values are included.

- *Please see the next page for a verification check:*

➢ *Would we get different results if we ran our query on values such as 4001 and 5999 respectively, i.e. if we tried to make the **BETWEEN** operator to behave **non**-inclusively for the 4000 ~ 6000 kg range?*

SQL Results: Successful Drone Ship Landing with Payload between 4000 and 6000

(continued from the previous page)

- The following process finds the names of boosters which have successfully landed on drone ship and had payload mass greater than 4001 but less than 5999 kilograms (inclusively):

```
In [29]: %%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4001 AND 5999;

 * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

Out[29]:

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

✓ *Thus, despite the fact that the **BETWEEN** operator is inclusive, for our given dataset we have now verified with **confidence** that the query returns the **same exact results** in both cases (4000~6000 & 4001~5999).*

➢ *In other words, we ran our inclusive query on two sets of values: "BETWEEN 4001 AND 5999", as well as "BETWEEN 4000 AND 6000". In both executions, the outcome produces the same 4 booster versions: **F9 FT B1022**, **F9 FT B1026**, **F9 FT B1021.2**, and **F9 FT B1031.2**.*

SQL Results: Total Number of Successful & Failed Mission Outcomes

- The following process calculates the total number of successful and failed mission outcomes:

```
In [30]: %%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

Out[30]:

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

- *Here we find and list all of the successful and failed mission outcomes. It appears that SpaceX launches achieve its mission objective successfully approximately 99% of the time.*
- *Therefore we can interpret that vast majority of the failed landings have indeed been intended.*
- *It is of note that one launch has an "unclear" payload status and another one failed in flight.*

SQL Results: Boosters Carrying Maximum Payload

- The following process lists names of the booster versions that have carried the maximum payload mass:

In [31]:
```
%%sql
SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

Out[31]:

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

- *Here we find and list all of the distinct versions of boosters that have carried the maximum payload mass (15,600 kilograms).*
- *The resulting data indicates that all of these maximum-payload-carrying versions are relatively close to one another, as every single one of them is in fact of the F9 B5 B10xx.x kind.*
- *As such, this data suggests that such high (maximum) payload mass has a relationship with respect to the booster version that is being used.*

SQL Results: 2015 Launch Records

- The following lists failed **landing_outcomes** in drone ships, their booster versions, and launch site names for 2015:

In [33]:
```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, LANDING__OUTCOME, BOOSTER_VERSION, PAYLOAD_MASS__KG_, LAUNCH_SITE
FROM SPACEXTBL
WHERE Landing__Outcome = 'Failure (drone ship)'
    AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

Out[33]:

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

- *Here we find and list the failed outcomes for the drone ships, their booster versions, and launch site names during 2015, splitting them up for convenience by the month of occurrence.*
- *As a result, there were only two such occurrences during the entire year of 2015.*

SQL Results: Outcome Ranking Between 2010-06-04 and 2017-03-20

- The following ranks the count of landing outcomes (such as 'Failure' (drone ship), or 'Success' (ground pad), etc.) between the dates of 2010-06-04 and 2017-03-20 respectively, in descending order:

In [42]:
```
%%sql
SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE LANDING__OUTCOME LIKE 'Success (%' OR LANDING__OUTCOME LIKE 'Failure (%' AND DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

```
 * ibm_db_sa://xcj78248:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

Out[42]:

landing__outcome	total_number
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Failure (parachute)	2

- *Here we find and list all of the successful and failed landings between 2010-06-04 and 2017-03-20, inclusively.*
- *The results are displayed in 4 different groups, by the outcome and context:*

*"**Success (drone ship)**" (14 occurrences) , "**Success (ground pad)**" (9 occurrences),*

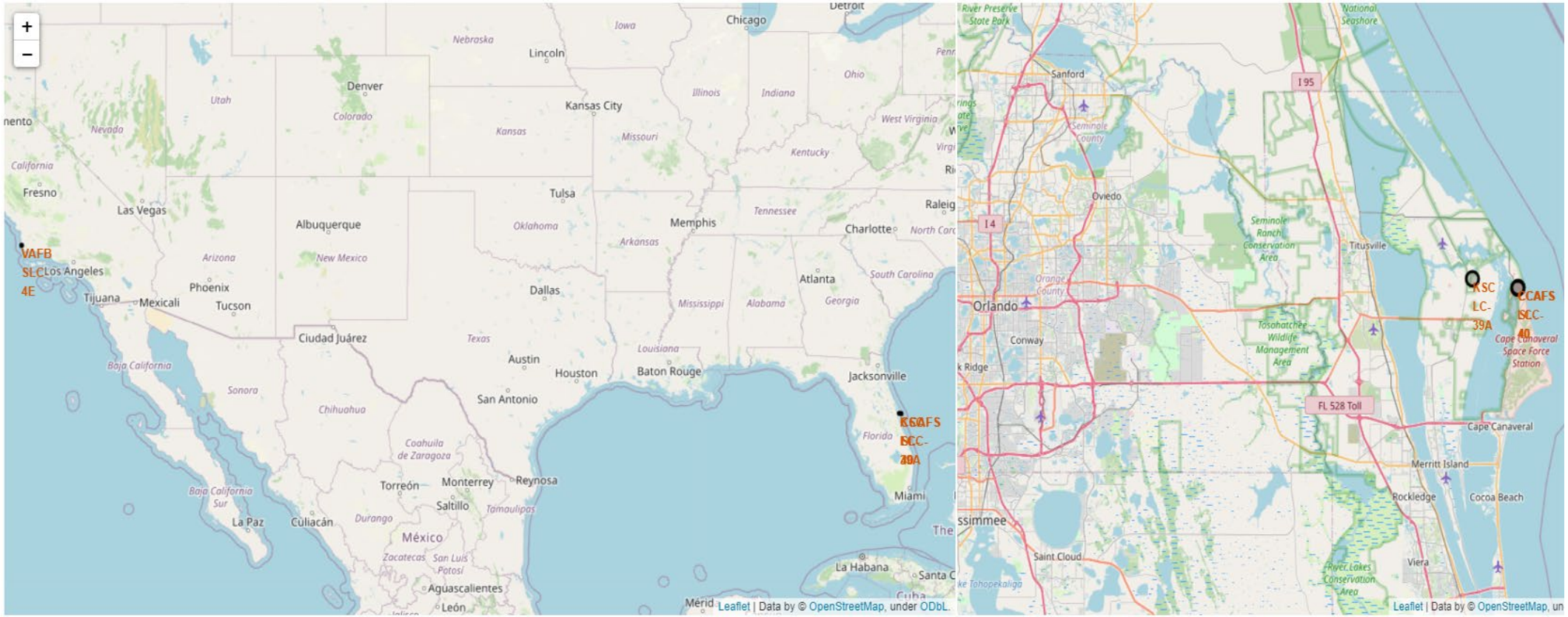
*"**Failure (drone ship)**" (5 occurrences), and "**Failure (parachute)**" (2 occurrences).*

- *As a result, we observe a total of 23 successful landings, with only 7 failed ones.*
- *This indicates an approximately **76.66%** average success ratio for this time period.*

Results

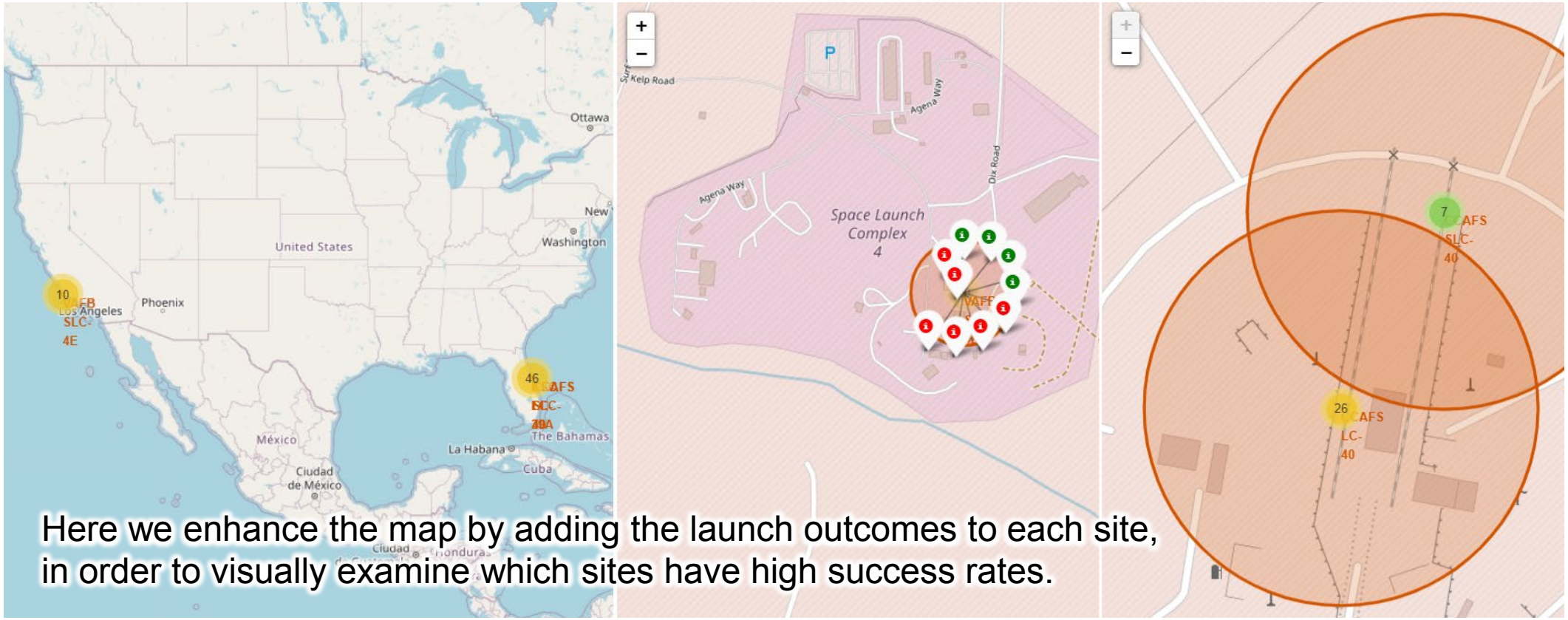
•EDA: Interactive Map with Folium

Results - Insights from EDA: All Launch Site Locations



The left section of the map displays all of the launch sites examined in this presentation. The segment of the map on the right shows the two launch sites on the East Coast in slightly more detailed zoom level.

Results - Insights from EDA: All Launch Site Locations

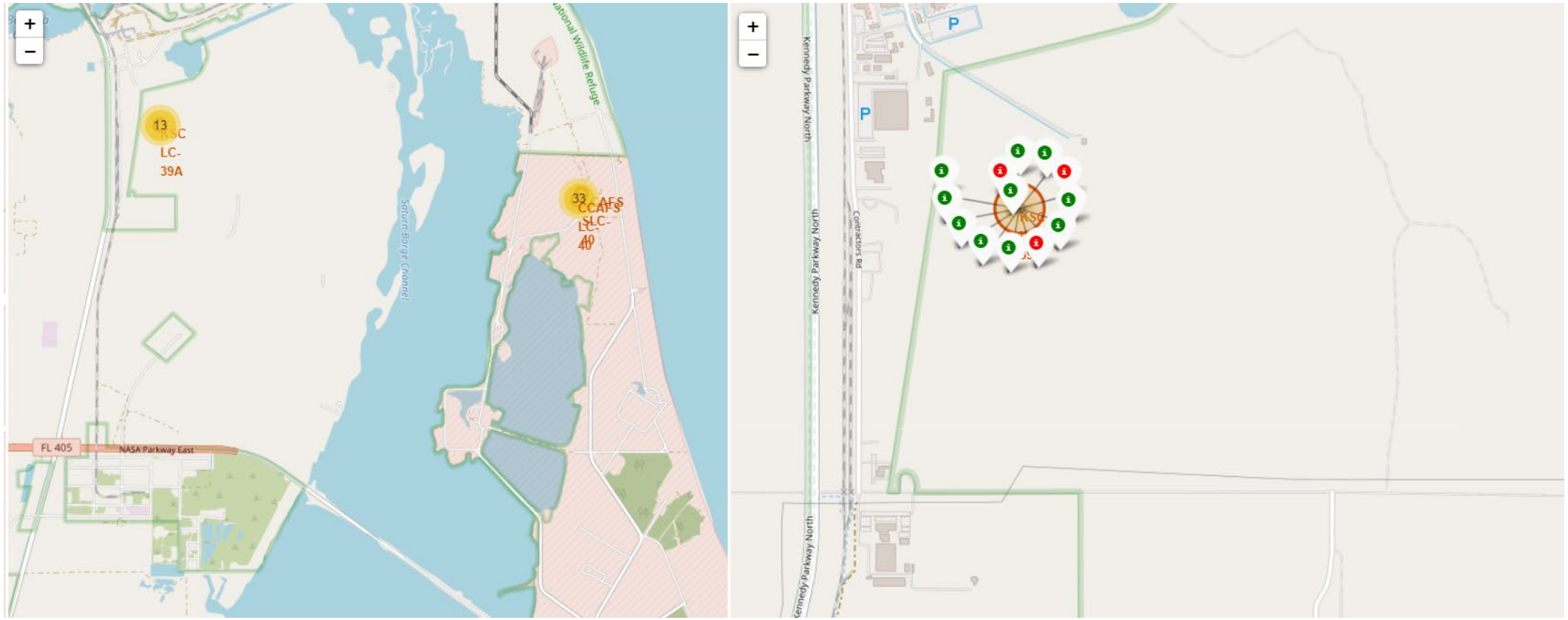


Clusters on the Folium map can be clicked to show each successful (*green*) or failed (red) landing. In this example, the segment in the middle shows VAFB SLC-4E with its 4 successful landings and 6 failed landings.

Results - Insights from EDA: All Launch Site Locations

Clicking around the East Coast, we can see the CCAFS SLC-40 and the CCAFS-LC40 launch sites, as pictured above. The map we enhanced allows for us to click on each site, and observe successful and failed launches.

Results - Insights from EDA: All Launch Site Locations

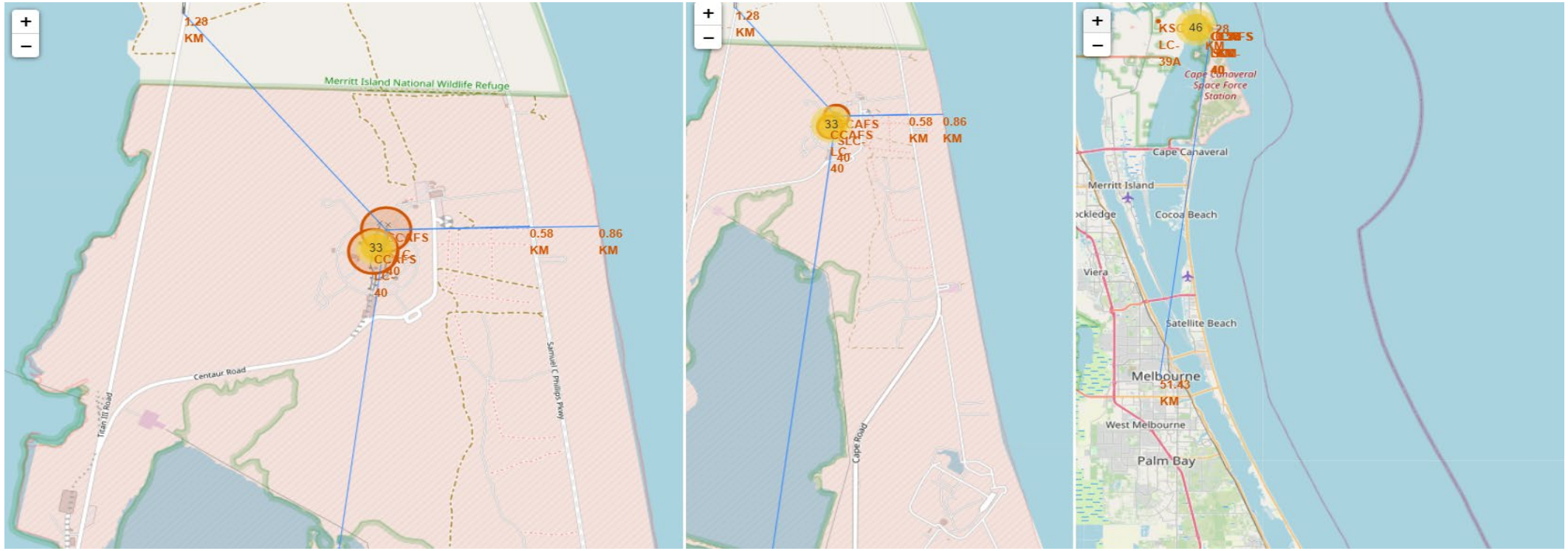


Finally, we observe the KSC-LC39A launch site, pictured above with its successful and failed launches.

GitHub Link to the Jupyter Notebook: 🗒 **Interactive Visual Analytics with Folium.ipynb**

- *https://github.com/ClaireYurev/Capstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium.ipynb*

Results - Insights from EDA: Key Location Proximities



As the above example shows, launch sites are near railroads due to their transport needs, near freeways for easier access by people and truck transport, and are also close to the coast, while being relatively far from populated areas. This is so that any potential launch failures can occur over water, avoiding loss of life.

GitHub Link to the Jupyter Notebook: **Interactive Visual Analytics with Folium.ipynb**

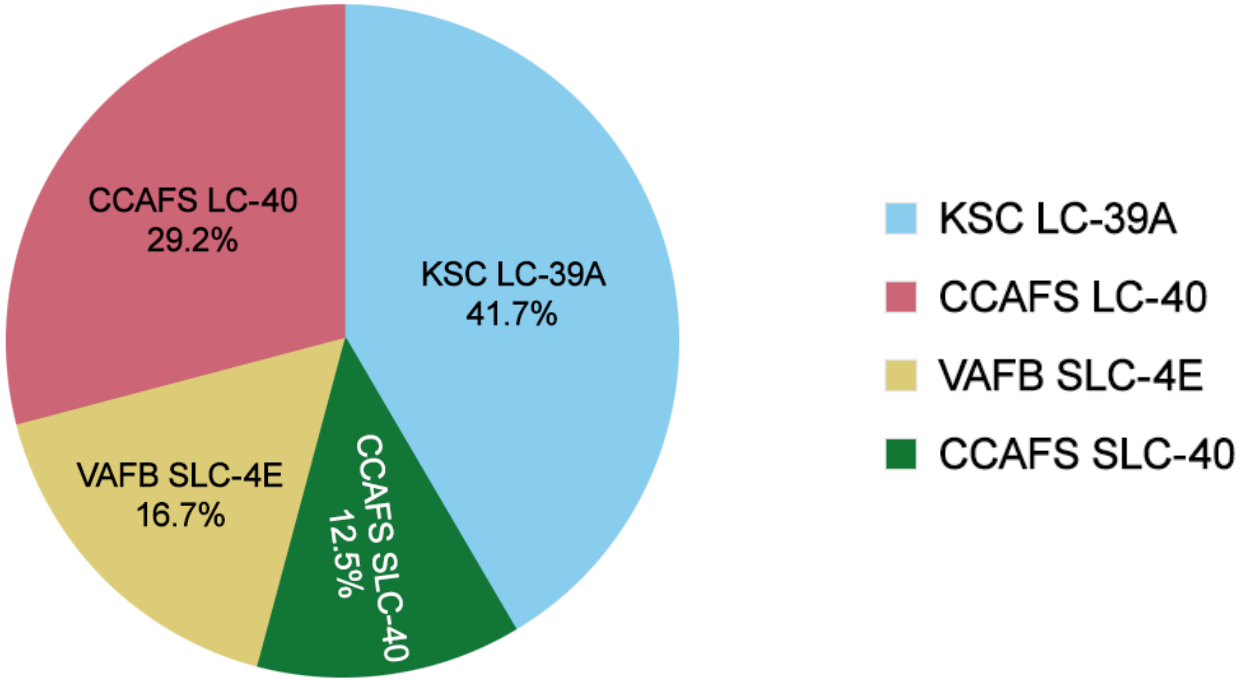
- *https://github.com/ClaireYurev/Capstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium.ipynb*

Results

- EDA: Dashboard with Plotly Dash

Insights from EDA: Plotly Dash dashboard - Results

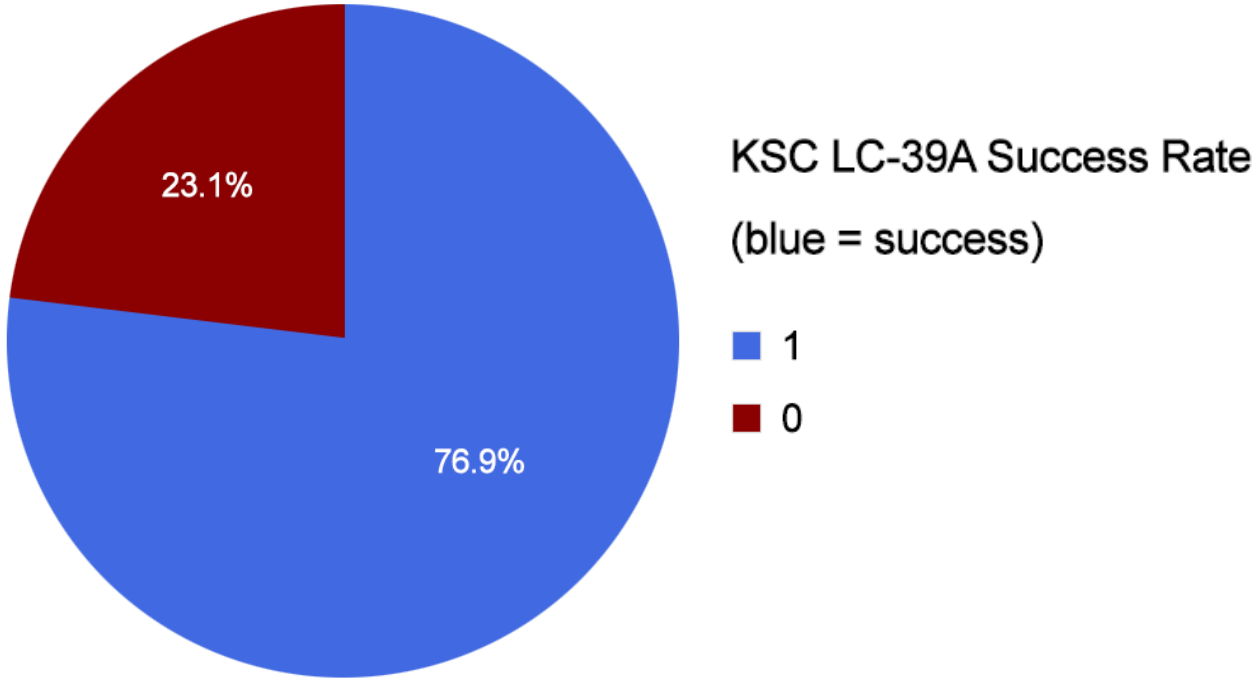
- Successful Launches Across Launch Sites:



The above is the distribution of successful landings across all launch sites. The "CCAFS LC-40" happens to be the former name of "CCAFS SLC-40", meaning that KSC and CCAFS sites appear to have the same amount of successful landings. However it should be noted that the majority of successful landings took place prior to the site name change. VAFB site has the least proportion of successful landings. This can possibly be due to an otherwise smaller sample, as well as a potentially higher difficulty of performing a successful launch on the west coast.

Insights from EDA: Plotly Dash dashboard - Results

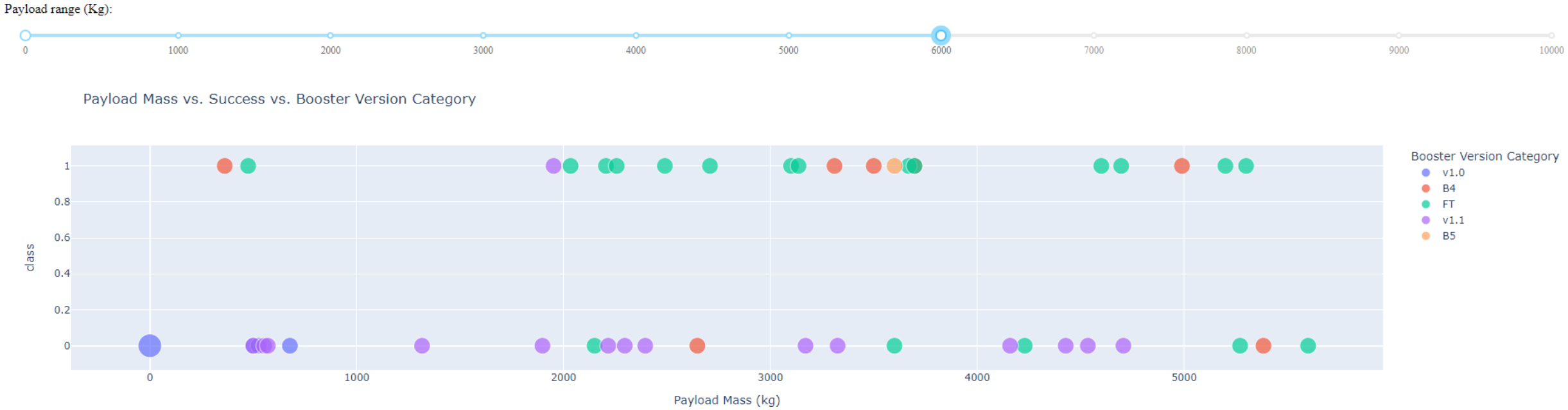
- Highest Success Rate of a Launch Site:



The data shows that site KSC LC-39A has the highest success rate among all sites reviewed here, with 10 successful landings and 3 unsuccessful landings.

Insights from EDA: Plotly Dash dashboard - Results

- Payload Mass vs Success vs Booster Version Category:



Our Plotly dashboard offers a user-controllable slider that can select any value of Payload Range (in kilograms) within the available range. This, however is set to have a range between 0 and 10,000 (as opposed to the maximum payload mass of 15,600 kilograms). The vertical axis, "Class" indicates the value of "1" for successful landings, and value of "0" for failed landing attempts. This scatter plot depicts Booster Version Category in different colors and the corresponding number of launches in varying point size. Of note is the fact that we can observe two *failed* landings within our range with payloads of "zero" kg.

Predictive Analysis

- Classification

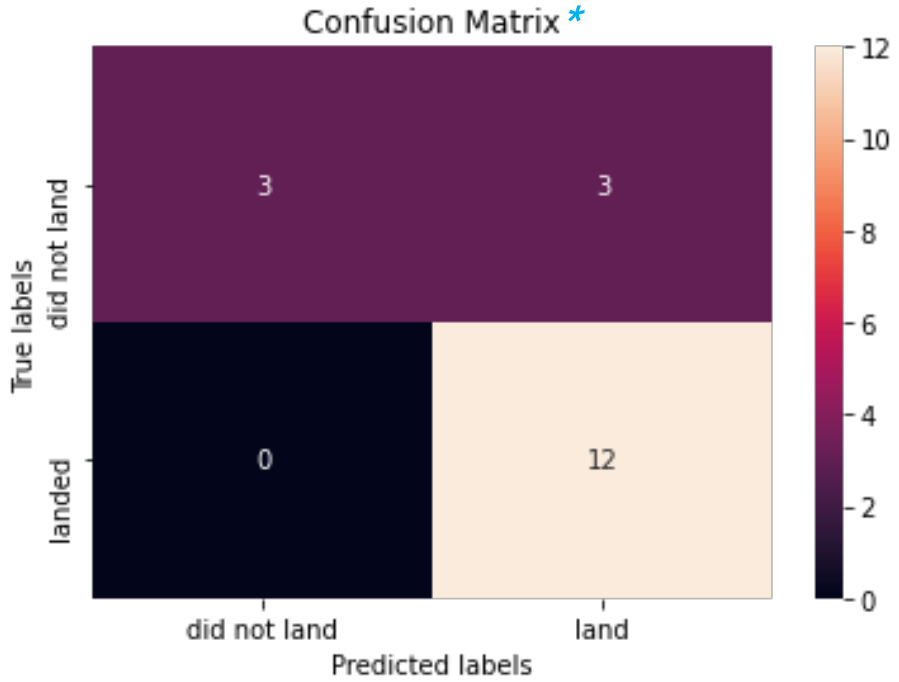
Predictive Analysis (Classification) - Results

- Classification Accuracy: Model Accuracy Score Comparison in a Bar Chart

All of the built models produced the same accuracy level on the test set at approximately 83.33%. Important caveat to this is the fact that the overall test group size is relatively small, with the sample size of only 18. On the other hand, this does also have the potential to produce a relatively large variance in accuracy results, such as that of the Decision Tree Classifier model in repeated runs. In order to establish the best model with greater certainty, we would need considerably more data.

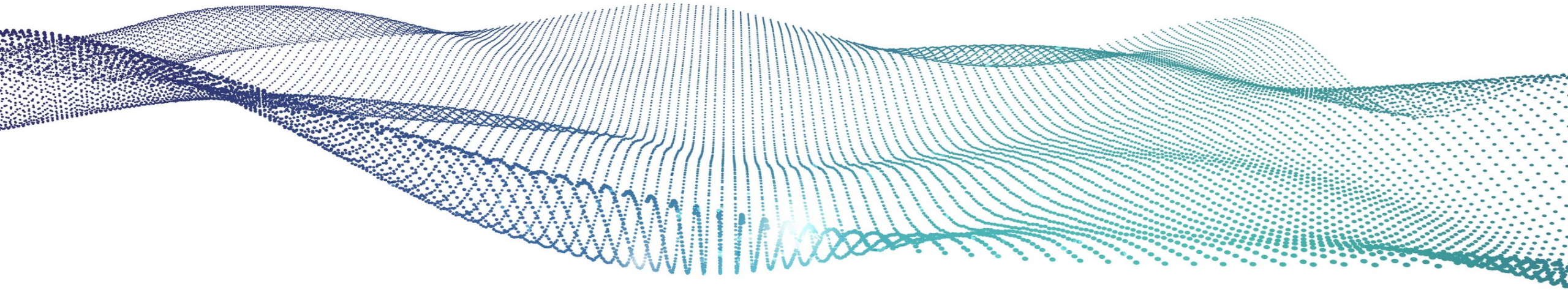
Predictive Analysis (Classification) - Results

- Confusion Matrix of the best-performing model



*Correct predictions are displayed in this diagram sequentially running in a diagonal order, from top-left to bottom-right.

Because all of the models have displayed the same performance with respect to the test set - the confusion matrices are the same across all models as well, and are represented by the matrix above. Our models predicted twelve successful landings while the actual label was “successful landing”. Our models further predicted 3 unsuccessful landings while the real label was “unsuccessful landing”. Furthermore, our models predicted 3 successful landings while the actual label was “unsuccessful landings”, in other words producing a *false positive* result in this case. As such, our models over-predict successful landings.



Conclusion

Conclusion

- ✓ Our goal was to develop a machine learning model for SpaceY, a company competing with SpaceX.
- ✓ The purpose of our model was to predict the success rate of Stage 1 landing, to save around $100 million.
- ✓ We utilized the data from a public SpaceX API and have web scraped data from SpaceX Wikipedia page.
- ✓ We then created data labels and stored data into an IBM Cloud DB2 SQL database.
- ✓ We created a dashboard in order to visualize the data and discover any insights from it.
- ✓ We designed, created, and trained a machine learning model with an accuracy of approximately 83%.
- ✓ Allon Mask of SpaceY can now use the results from this model to infer a prediction, with relatively high accuracy, of whether a launch will have a corresponding successful Stage 1 landing in order to save costs.
- ✓ Next steps: if possible, an effort should be made to collect more data, and use it to train the different models, with the goal of determining the best machine learning model in terms of predicting successful Stage 1 landings for SpaceY.

Acknowledgements

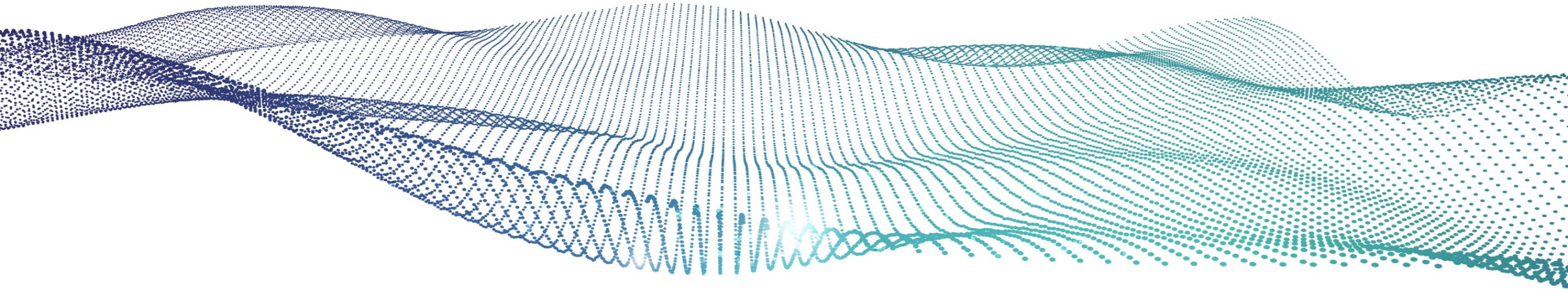
Acknowledgements

A **big thank you** to every instructor and staff member who has worked to make this curriculum possible!

❤ Special acknowledgement and gratitude goes out to each contributor, in no particular order:

- Svetlana Levitan, Senior Developer Advocate
- Eric Thomas, Ph. D, Michigan State University
- Polong Lin, Data Scientist
- Rav Ahuja, Global Program Director
- Joseph Santarcangelo, Ph. D, Data Scientist
- Aije Egwaikhide, Senior Data Scientist
- Romeo Kienzler, Chief Data Scientist, Course Lead
- Saishruthi Swaminathan, Data Scientist & Developer Advocate, IBM CODAIT
- Murtaza Haider, Ph.D, Sr. Data Scientist, Ryerson University
- Hima Vasudevan, Data Scientist
- Saeed Aghabozorgi, Ph.D., Sr. Data Scientist
- Azim Hirjani, Cognitive Data Scientist
- Yan Luo, Ph.D, Data Scientist and Developer
- Alex Aklson, Ph.D., Data Scientist

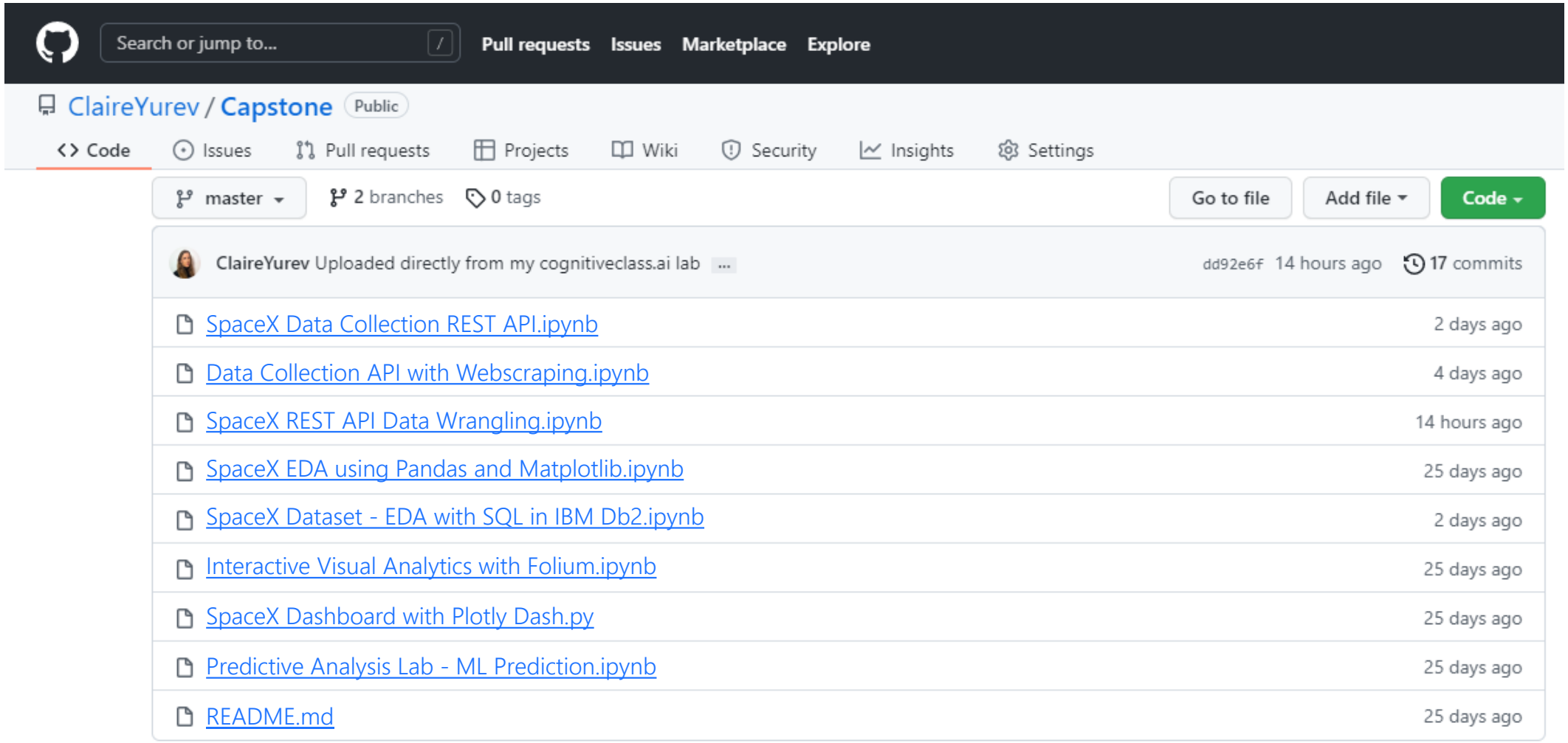
... and many, many more. Thank you to each and every one of you!



Appendix

Appendix

The following is an interactive list of all resources produced for and included in this presentation, with corresponding GitHub links:



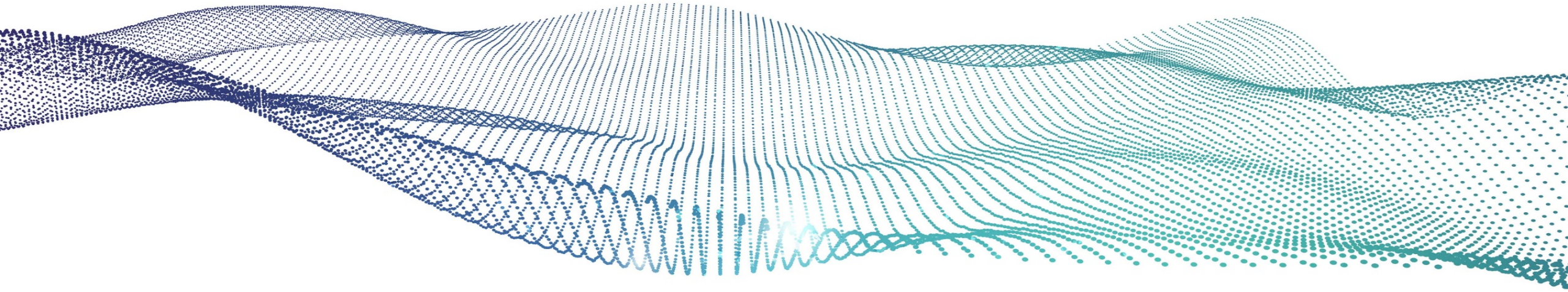
Additional Resources

The following is a non-exhaustive list of additional resources that were especially helpful to this project:

- Helpful documentation for the IBM Cloud, be it Data, Watson Studio, SQL, or Jupyter:
  - https://dataplatform.cloud.ibm.com/docs/

- A great minimal web 'IDE' for on-the-go Python debugging:
  - https://pythonsandbox.com/

- A useful guide for utilizing SQL in Python:
  - https://realpython.com/python-sql-libraries/



End of the Presentation

Thank you for your time!

Claire Yaroslav Yurev • 2022