IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Adam Gyönyör
09.03.2022

Disclaimer

- This presentation serves nonprofit educational purposes only.
- It might contain and/or refer to copyrighted material not authorized for use by the owner.
- The use of potentially copyrighted material in this presentation falls under Fair Use:
  - Copyright Disclaimer under section 107 of the Copyright Act 1976, allowance is made for “fair use” for purposes such as criticism, comment, news reporting, teaching, scholarship, education and research. Fair use is a use permitted by copyright statute that might otherwise be infringing. Non-profit, educational or personal use tips the balance in favor of fair use.
- All rights and credits go directly to their rightful owners.
- No copyright infringement intended.

Each reference is hyperlinked in place to enable easy access to their contents. No academic formatting is used.

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendices

Executive Summary

Objective

Can rockets be reused?

This is the main question upon considering how to significantly reduce costs associated with reaching orbit – or in simple terms: how to make spaceflight cheap.

Yes

With this seeming to be the correct answer, the goal is clear: rockets must be reliable enough to be used for multiple launches.

Analysis

Based on available information on historical launches, the question on the left has been investigated, with the answer 'yes' being considered the desired outcome. The table below lists the steps of the analysis in order.

Methodology	Means	Tool	Result	
Data Collection	API	SpaceX REST API	Dataframe	(of Falcon 9 launches)
Data Collection	Web Scraping	Beautiful Soup	Dataframe	(of launch & landing outcomes)
Data Wrangling	Notebook	Pandas	Class label	(required for ML model)
Exploratory Data Analysis	Visualization	Seaborn	Relationship	(between pairs of features)
Exploratory Data Analysis	Database	SQL	Insight	(into data characteristics)
Visual Analytics	Spatial Analysis	Folium	Characteristics	(of launch sites)
Visual Analytics	Dashboard	Plotly Dash	Success Rates	(of booster landings)
Predictive Analysis	Machine Learning	Scikit-learn	Classifier	(of landing outcome)

Outcome

Findings

- Most useful features: Landing Legs, Grid Fins
- Classifier accuracy: 0.83

Implications

With the given data, none of the classifiers were able to yield a score above 0.9. Enhancing the models by using pipelines and performing feature selection did not improve the overall results. Consequently, more data – on new launches – seem to be required for better model performance.

Introduction

Background

- Due to their single-use designs, launch vehicles have historically always been particularly expensive.
- Therefore, at the dawn of commercial spaceflight, mission sustainability quickly established itself as the crucial business goal.
- It became evident: The objective is simple - but not easy. Launch vehicles must be capable of reliably performing multiple missions.

Objective

- The reusability of SpaceX's Falcon 9 rockets is being investigated in this data science project, by analyzing publicly available data on Falcon 9 launches.
- The main question: **Can Falcon 9 rockets be reused?**
- The expected answer: **Yes**

Section 1

Methodology

Methodology

Executive Summary

• Data collection	API, Web Scraping
• Data wrangling	Pandas, Numpy
• Exploratory data analysis	Visualization, SQL
• Interactive visual analytics	Folium, Plotly Dash
• Predictive analysis	Scikit-learn classifiers

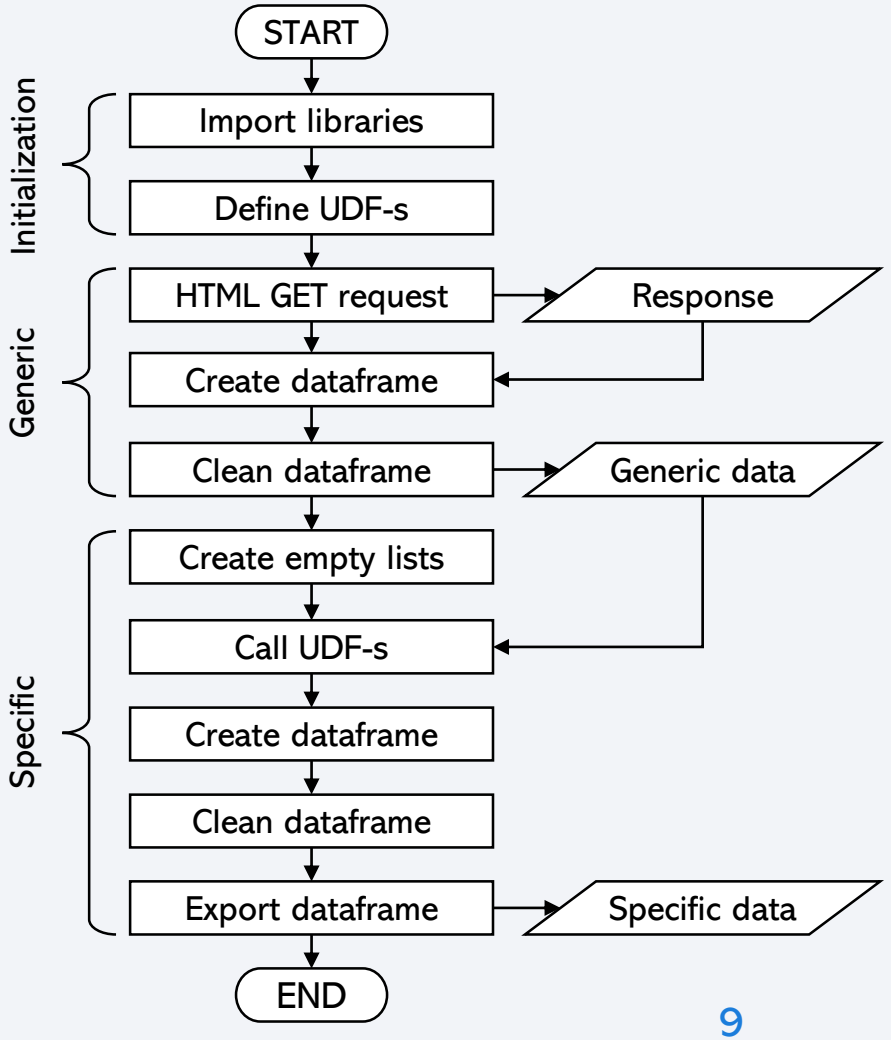
Data Collection

- Goal
  - Collect historical data on Falcon 9 launches
- Means
  - API – SpaceX REST API
  - Web Scraping – Beautiful Soup 4
- Outcome
  - Two datasets have been created: The first one by using the SpaceX REST API, and the second one by performing web scraping of a Wikipedia page using Beautiful Soup 4.
  - Both datasets have been cast into pandas dataframes.
  - They have been cleaned and missing values have been imputed.

Data Collection – API

Collecting data with the SpaceX REST API can be broken down into three main phases:

- Initialization
  - Required libraries are imported.
  - User-defined functions (UDF-s) are defined. These are used later for retrieving specific information, based on generic codes.
- Generic
  - Data is obtained from the 'launches/past' endpoint of the API by making a GET request.
  - JSON part of the response is cast into a pandas dataframe.
  - Dataframe is cleaned by dropping irrelevant columns and properly formatting dates. The resulting dataframe contains mainly codes, specific to the API.
- Specific
  - Empty lists are created. They will be filled up by the UDF-s.
  - UDF-s are called by iterating through the codes in the generic dataframe. They obtain data by sequentially making GET requests to the following API endpoints: 'rockets/', 'launchpads/', 'payloads/' and 'cores/' combined with each generic code.
  - Once the lists are filled up by the UDF-s, they are cast into a dictionary, which in turn gets cast into a pandas dataframe.
  - Dataframe is cleaned by dropping rows containing Falcon 1 launches and by replacing missing payload values by the mean.
  - Dataframe, which now contains mission-specific data that can be interpreted by humans, is exported to CSV.



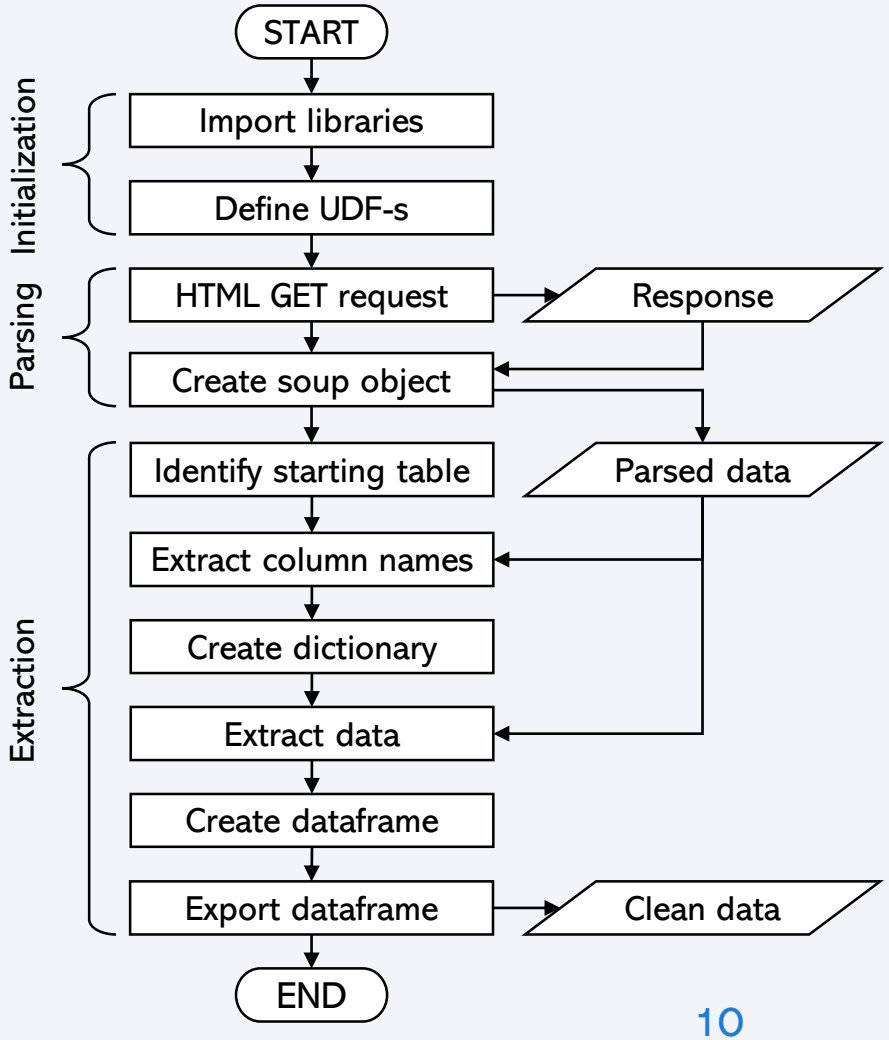
The notebook is shared on GitHub: Module 1 - Data Collection API

Data Collection – Web Scraping

Collecting data with web scraping – using Beautiful Soup 4 – can be broken down into three main phases:

- Initialization
  - Required libraries are imported.
  - User-defined functions (UDF-s) are defined. These are used later for cleaning parsed data.
- Parsing
  - Data is obtained by making a GET request referencing the 'List of Falcon 9 and Falcon Heavy launches' Wikipedia page.
  - Text part of the response is used to create a Beautiful Soup object.
- Extraction
  - Among all tables on the Wikipedia page, the first one is identified, from which data extraction will start.
  - Column names are extracted from this table by using one of the UDF-s.
  - An empty dictionary is created which will be filled up by extracted data.
  - Data is extracted by iterating through all rows of all tables, starting from the table identified above. UDF-s are used to extract cleaned data from the soup object.
  - Dataset is created by casting the filled-up dictionary into a pandas dataframe.
  - Dataframe is exported to CSV.

The notebook is shared on GitHub: Module 1 - Data Collection with Web Scraping

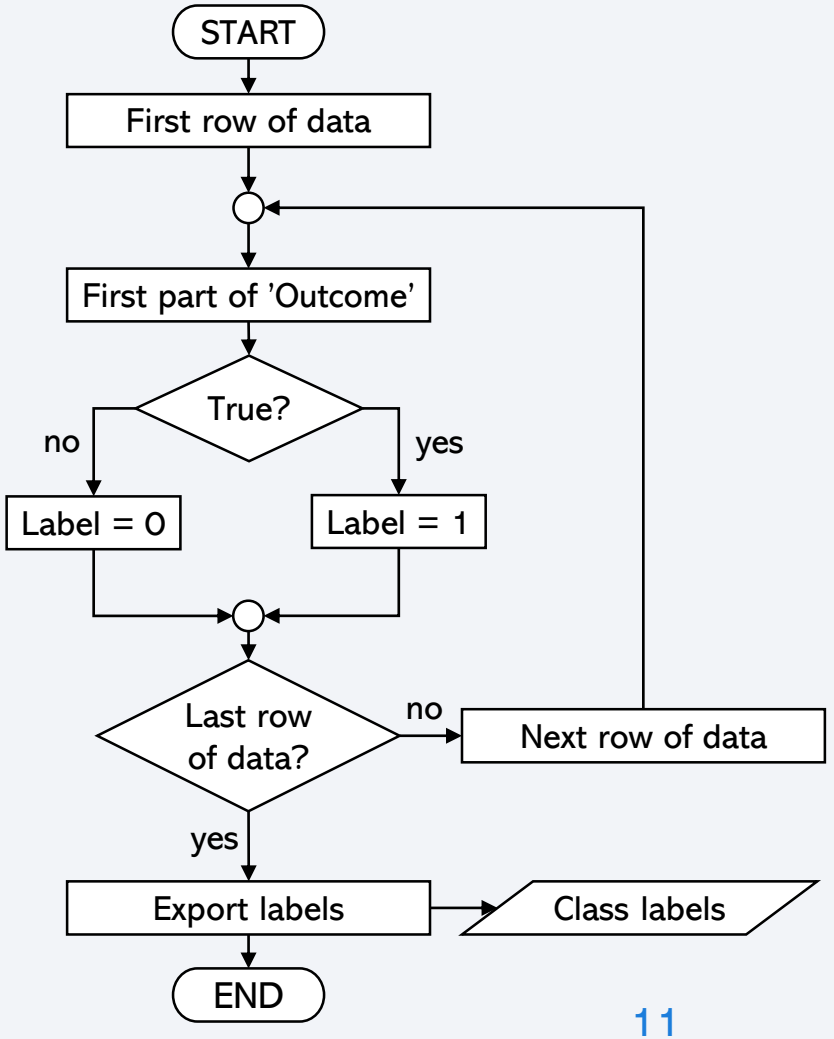


Data Wrangling

Data Wrangling can be summarized as follows:

- Goal
  - Create discrete labels for supervised learning
- Means
  - Notebook – Pandas, Numpy
    - EDA — Exploratory data analysis has been performed to obtain insight into potentially useful patterns in the data, e.g. number of launches per site.
    - Labels — 'Outcome' feature has been identified as relevant for label creation. It is a combination of the values of 'landing_success' and 'landing_type' keys from the 'cores' dictionary, being returned by the SpaceX REST API call. The 'landing_success' part of this feature has been used to determine the class label as follows: 'True' has been associated with a successful booster landing, whilst 'None' and 'False' have been interpreted as unsuccessful booster landings. The logic is shown on the flowchart.
- Outcome
  - Binary labels for each mission:
    - 0 — Unsuccessful booster landing
    - 1 — Successful booster landing

The notebook is shared on GitHub: Module 1 - Data Wrangling



EDA with Data Visualization

'Correlation does not imply causation' is the main principle along which EDA with Data Visualization has been performed:

- Goal
  - Discover potential relationships between features
- Means
  - Visualization – Seaborn
    - Labels — As 'Class' is the target variable, all relationships between selected features have been visualized with the help of it. For some, it has been done directly, using it as one of the two analyzed variables. For others, it has been done indirectly, having it applied to the plots as an additional layer of information, to see what effect do the observed trends have on booster landing outcomes.
    - Features — The following have been selected for visual data exploration: 'FlightNumber', 'Date', 'PayloadMass', 'Orbit' and 'LaunchSite'.
- Outcome
  - Plots
    - 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 Mass vs. Orbit Type
    - Yearly Trend of Launch Success Rate

The notebook is shared on GitHub: Module 2 - EDA with Data Visualization

EDA with SQL

As part of data exploration, information that is not of visual nature, has been investigated as follows:

- Goal
  - Gain insight into data characteristics
- Means
  - Database – SQL
    - Db2 The dataset obtained by web scraping has been uploaded to IBM's 'Db2 on Cloud' service.
    - Queries The cloud-based relational database has been accessed remotely from a notebook, using queries passed on by SQL magic.
- Outcome
  - Tables containing information on:
    - Launch site names
    - Five first launches from Kennedy Space Center's Launch Complex 39A
    - Total payload launched for NASA
    - Average payload launched by booster version F9 v1.1
    - Date of first successful drone ship landing
    - List of boosters successfully landing on the ground, having carried payloads between 4 and 6 metric tons
    - Number of successful and unsuccessful missions
    - List of boosters carrying maximum payload
    - Table of boosters successfully landing on the ground in 2017, including launch site and month of mission
    - Number of successful booster landings between 04.06.2010 and 20.03.2017, in descending order

The notebook is shared on GitHub: Module 2 - EDA with SQL

Build an Interactive Map with Folium

In case of geospatial data, maps have proven to be particularly useful tools for putting the data into context:

- Goal
  - Interactively explore characteristics of launch sites
- Means
  - Spatial Analysis – Folium
    - Launch Sites — All launch sites have been marked.
    - Launches — Launches have been marked at their respective sites. They have been distinguished by landing outcome: Launches resulting in successful booster landings have been marked with green, unsuccessful ones with red color.
    - Distances — Distances have been calculated and marked between launch sites and their following proximities: railways, highways, coastlines and cities.
- Outcome
  - Interactive Map objects with the following children:
    - Circle
    - Marker
    - MarkerCluster
    - MousePosition
    - PolyLine

The notebook is shared on GitHub: Module 3 - Interactive Visual Analytics with Folium

The notebook on GitHub does not show any map objects in the output fields. Therefore, a copy of the notebook with fully functional interactive maps is shared on IBM's Watson Studio and can be accessed via the following link: Module 3 - Interactive Visual Analytics with Folium (IBM Watson Studio)

Build a Dashboard with Plotly Dash

By providing an intuitive experience, direct interaction with the data helps users to better understand it:

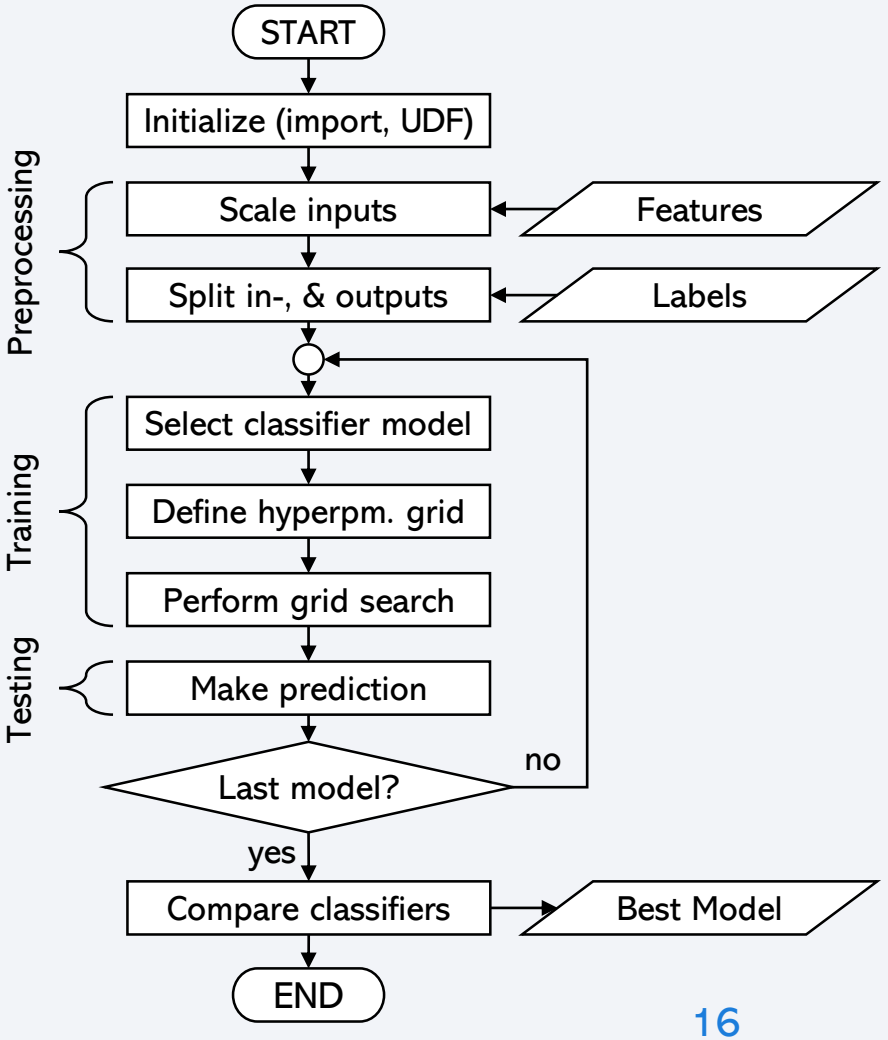
- Goal
  - Interactively explore booster landing success rates
- Means
  - Dashboard – Plotly Dash
    - Plots: Two plots have been created, both as Graph components: The first one is a Pie Chart, the second is a Scatter Plot.
    - Interactions: Two input fields have been created: The first one is a Dropdown list, the second is a RangeSlider. They have been interconnected with the plots as follows: Changing the selection of the Dropdown list updates both the Pie Chart and the Scatter Plot. Manipulating the RangeSlider updates only the Scatter Plot.
- Outcome
  - Interactive Dashboard consisting of:
    - Dropdown: Launch sites: listed both individually and all together.
      - All: Pie Chart: Proportions of successful launches – i.e. launches followed by successful booster landings – per site.
        Scatter Plot: Payload masses of launches from all sites.
      - Selected: Pie Chart: Success rate of the selected launch site.
        Scatter Plot: Payload masses of launches from the selected launch site.
    - Pie Chart: Landing outcomes. Input: Dropdown.
    - RangeSlider: Payload mass: ranging from 0kg to 10000kg, with increments of 1000kg.
      - Range: Scatter Plot: Payload masses of launches from the selected range.
    - Scatter Plot: Payload mass vs. landing outcome, distinguished by booster version. Inputs: Dropdown, RangeSlider.

The source code is shared on GitHub: Module 3 - Interactive Dashboard with Ploty Dash

Predictive Analysis (Classification)

All previous tasks and analyses have been performed in support of the main objective:

- Goal
  - Select a supervised machine learning model to classify booster landing outcomes
- Means
  - Machine Learning – Scikit-learn
    - Preprocessing: Using StandardScaler, each feature has been normalized. Then, both the normalized features and the labels have been split into training and testing sets using train_test_split.
    - Training: Grid Search has been performed on the training set as follows: A grid of hyperparameters has been defined for each of the following classifiers: Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbors. After training on all hyperparameter combinations – a.k.a. grid points – using ten-fold cross-validation, the best-performing parameters and the training accuracy have been output for each classifier model.
    - Testing: After each grid search, the accuracy of the prediction and the confusion matrix have been determined using the testing set.
- Outcome
  - Best classifier model
    - In general: The model yielding the best test performance gets selected.
    - In particular: In this project, all models have shown identical test performances.



The notebook is shared on GitHub: Module 4 - Machine Learning Prediction

Results

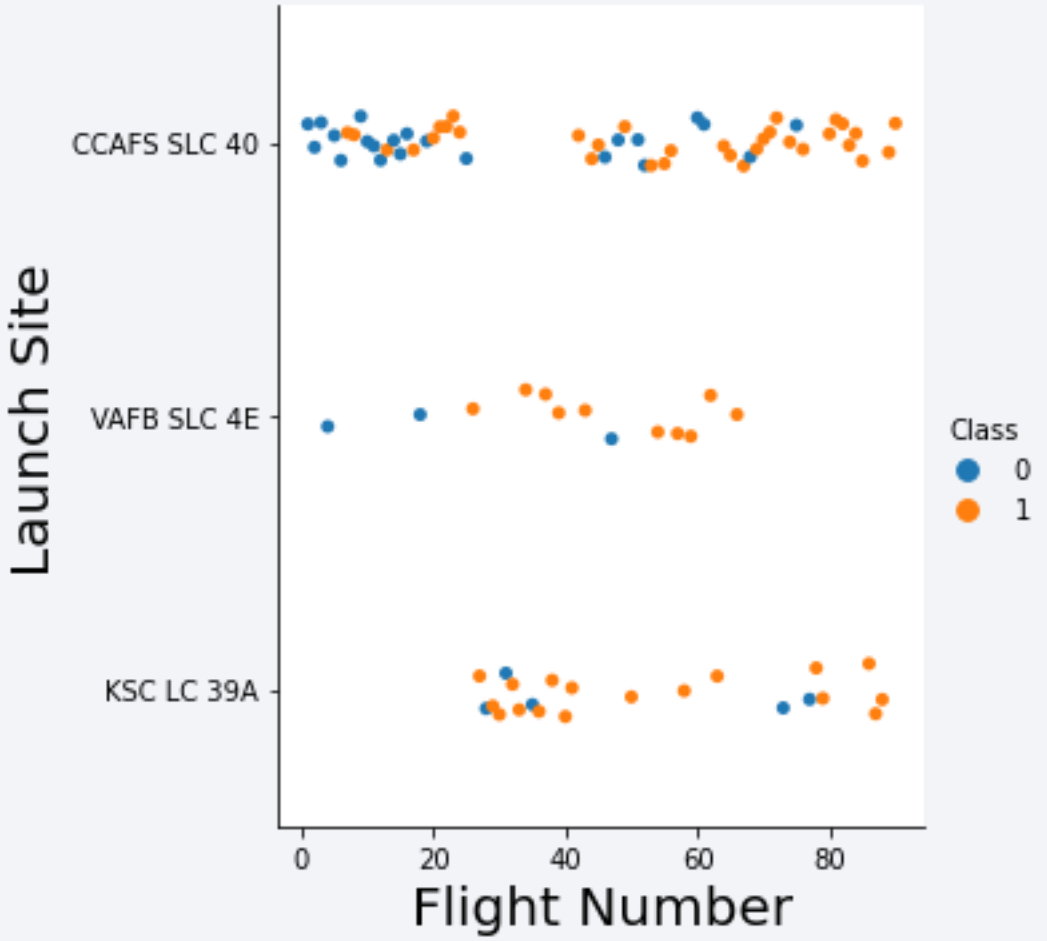
- Exploratory Data Analysis
  - Section 2 — EDA with Data Visualization; EDA with SQL
- Visual Analytics
  - Section 3 — Interactive Map with Folium
  - Section 4 — Dashboard with Plotly Dash
- Predictive Analysis
  - Section 5 — Predictive Analysis (Classification)

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

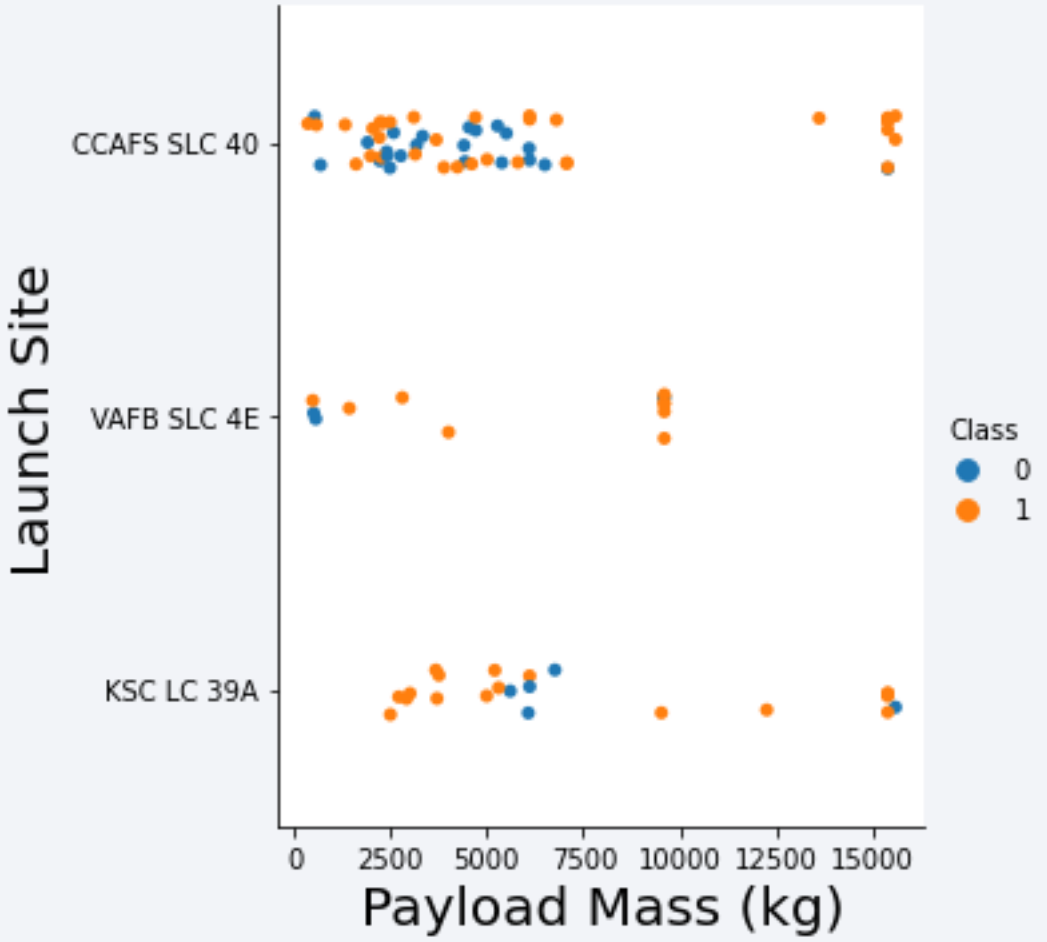
- In the beginning stages of the project, most of the missions have been launched from CCAFS SLC 40.
- Later, the historic KSC LC 39A has also become a frequently used launch site for Falcon 9 rockets.
- From the beginning of the project, VAFB SLC 4E has been used with moderate frequency.
- For a temporary period, no rockets have been launched from CCAFS SLC 40.
- Since a while, VAFB SLC 4E has not been used for any launches.



Launch site codes are explained under 'All Launch Site Names' on slide 25.

Payload Mass vs. Launch Site

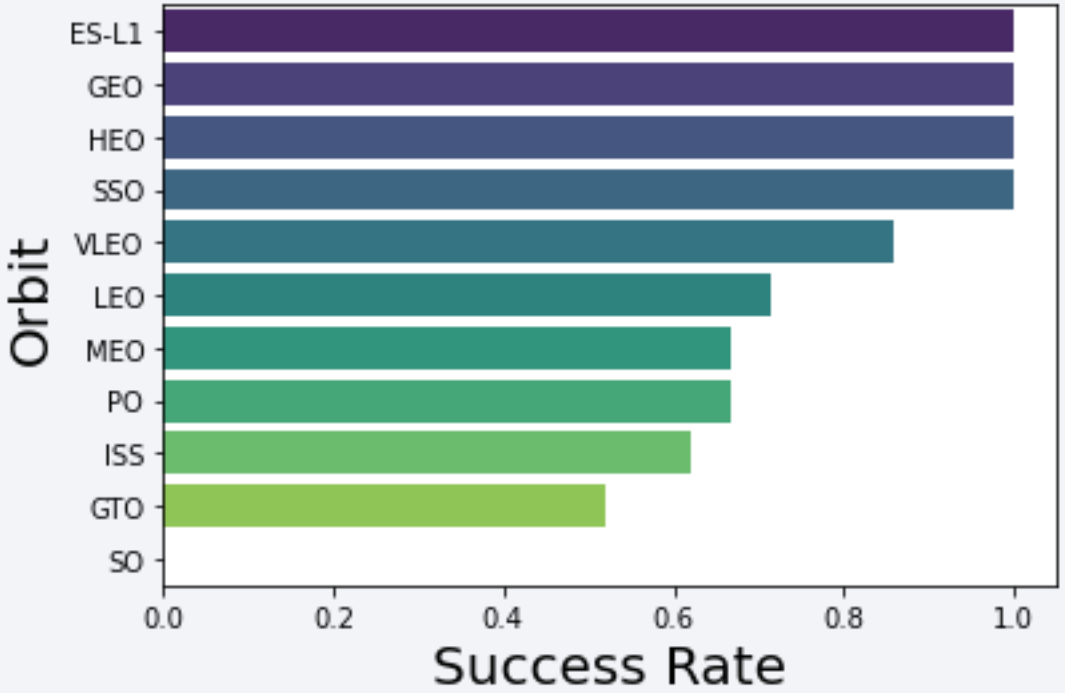
- Most light payloads – mass below 7000kg – have been launched from CCAFS SLC-40.
- Light payload missions launched from KSC LC 39A have ended in more successful booster landings than those launched from CCAFS SLC 40.
- Iridium NEXT satellites – with a total payload of 9600kg per mission – have been launched solely from VAFB SLC 4E.
- The majority of heavy payload missions – mass above 12000kg – have been launched from CCAFS SLC 40, carrying Starlink satellites.



Launch site codes are explained under 'All Launch Site Names' on slide 25.

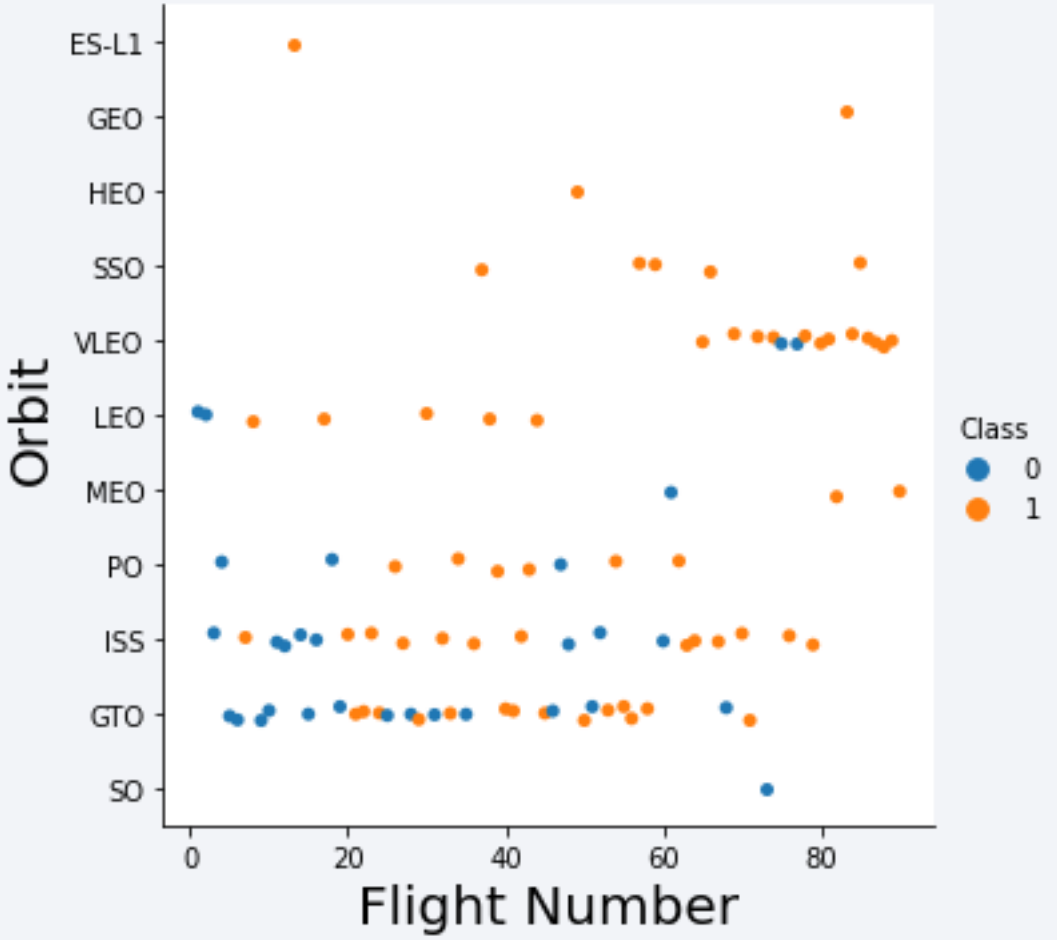
Success Rate vs. Orbit Type

- All boosters have landed successfully, that have launched payloads into the following orbits:
  - ES-L1 Lagrange point 1
  - GEO Geostationary orbit
  - HEO Highly elliptical orbit
  - SSO Sun-synchronous orbit
- Only half of those boosters have landed successfully, that have launched payloads into geostationary transfer orbits (GTO).
- Boosters having launched payloads into other types of orbits have performed successful landings in more than half of all cases – with the exception of SO whose booster hasn't landed.



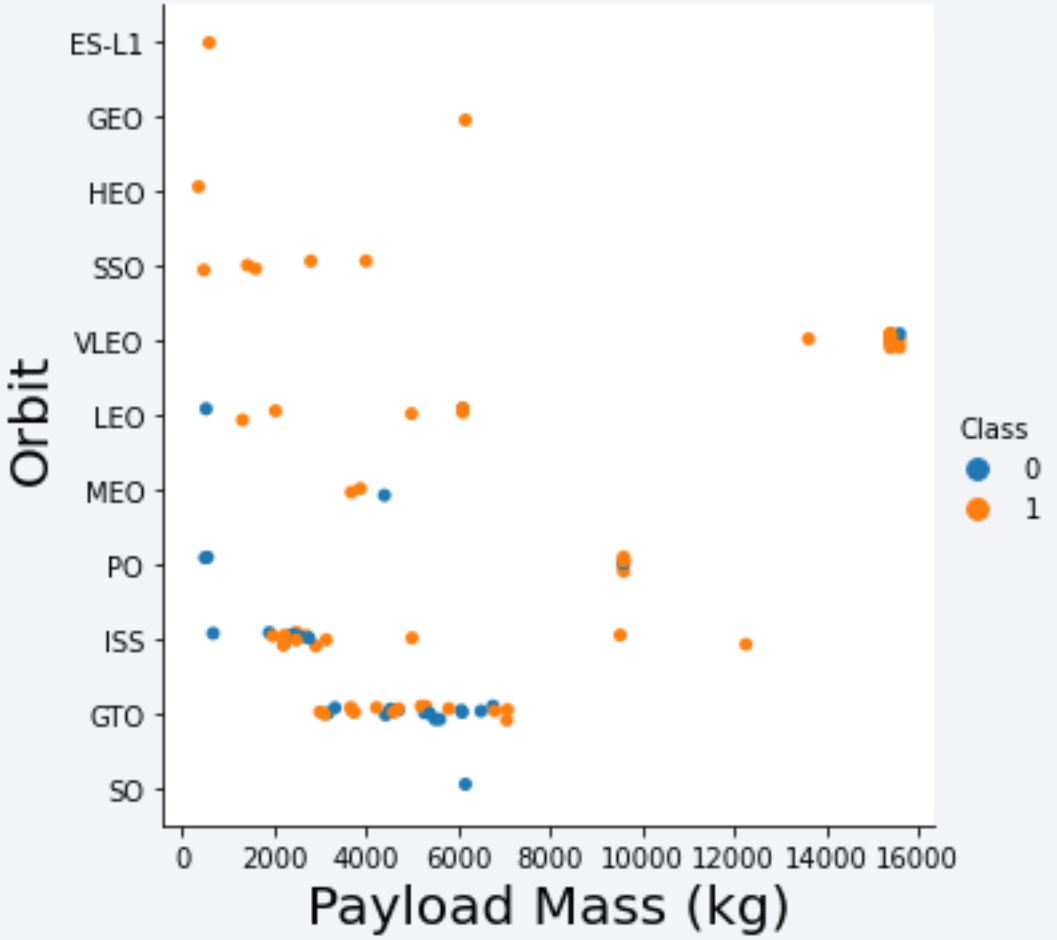
Flight Number vs. Orbit Type

- Increasing flight number did not seem to have any significant effect in improving booster landing success for the most common orbit types:
  - VLEO Very low earth orbit
  - PO Polar orbit
  - ISS International Space Station
  - GTO Geostationary transfer orbit
- Missions to the following two orbits have clearly improved their landing success rates over time:
  - LEO Low earth orbit
  - MEO Medium earth orbit
- For other orbit types, there is no relation between booster landing success and flight number.



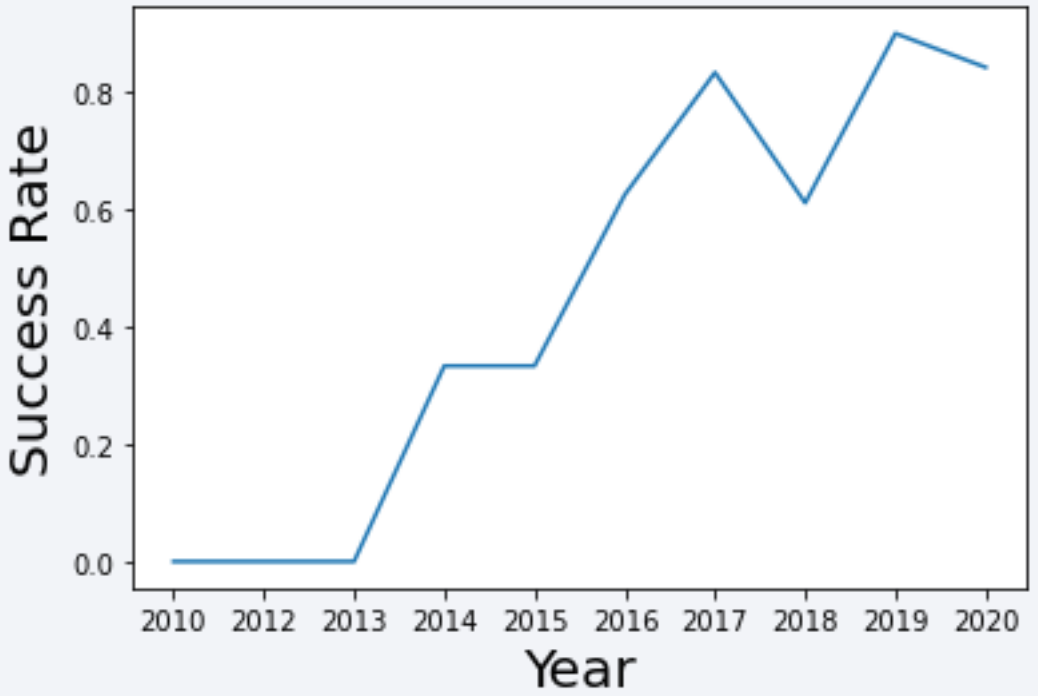
Payload Mass vs. Orbit Type

- Light payloads – mass below 7000kg – have been launched into nearly all types of orbits, with GTO being the most frequently targeted one.
- Almost all medium heavy payloads – mass around 10000kg – have been launched into polar orbits (PO), carrying Iridium NEXT satellites.
- Heavy payloads – mass above 12000kg – have only been launched into the following two orbits, serving the following missions:
  - VLEO Starlink
  - ISS Crew Dragon
- Most unsuccessful booster landings have occurred after launching light payloads.



Launch Success Yearly Trend

- In the early stages of the project, booster landings have either not been attempted, or resulted in failure.
- In 2014, two out of six flights have already ended with successfully controlled soft touchdowns into the ocean.
- In 2015, history was made: For the first time ever, Falcon 9's first stage has successfully landed on the ground.
- In 2016, success continued: The very first drone ship landing has been performed.
- As a general tendency, average yearly booster landing success rates have improved over time.



All Launch Site Names

- Launch site names have been queried as follows:

```
In [5]: %sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL
 * ibm_db_sa://xlz78123:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[5]:

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

- The resulting codes refer to the following launch sites:
  - CCAFS LC-40 — Cape Canaveral Air Force Station Launch Complex 40
  - CCAFS SLC-40 — Cape Canaveral Air Force Station Space Launch Complex 40
  - KSC LC-39A — Kennedy Space Center Launch Complex 39A
  - VAFB SLC-4E — Vandenberg Air Force Base Space Launch Complex 4E

Launch Site Names beginning with 'KSC'

- Five first launches from Kennedy Space Center's Launch Complex 39A have been queried as follows:

In [8]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'KSC%' LIMIT 5
```

```
 * ibm_db_sa://xlz78123:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[8]:

DATE	Time (UTC)	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	Landing _Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

- The mission objectives were the following:
  - SpaceX CRS-10 — Commercial Resupply Service mission to the International Space Station
  - EchoStar 23 — Launch of communications satellite into geosynchronous transfer orbit
  - SES-10 — Launch of communications satellite into geostationary transfer orbit
  - NROL-76 — Launch of classified US national security satellite into low earth orbit
  - Inmarsat-5 F4 — Launch of communications satellite into geostationary transfer orbit

Total Payload Mass

- Total payload launched for NASA has been queried as follows:

In [10]:

```
%%sql
SELECT * FROM
(SELECT SUM(PAYLOAD_MASS__KG_), CUSTOMER FROM SPACEXTBL GROUP BY CUSTOMER)
WHERE CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://xlz78123:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[10]:

1	customer
45596	NASA (CRS)

- In total, 45596kg of payload has been launched for NASA up until the 9th of June 2021.

Average Payload Mass by F9 v1.1

- The average payload launched by booster version F9 v1.1 has been queried as follows:

In [11]:
```
%%sql
SELECT * FROM
(SELECT AVG(PAYLOAD_MASS__KG_), BOOSTER_VERSION FROM SPACEXTBL GROUP BY BOOSTER_VERSION)
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://xlz78123:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[11]:

1	booster_version
2928	F9 v1.1

- On average, Falcon 9 v1.1 rockets have carried 2928kg of payload into orbit per each mission.

First Successful Drone Ship Landing Date

- The date of the first successful drone ship landing has been queried as follows:

In [12]:
```
%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (drone ship)'
```

```
 * ibm_db_sa://xlz78123:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[12]:

1
2016-04-08

- The [first ever booster landing on a floating platform] has been performed on the 8th of April 2016.

Video embedded from YouTube – it might take some time to buffer.

Successful Ground Pad Landing with Payload between 4000kg and 6000kg

- The list of boosters successfully landing on the ground, having carried payloads between 4 and 6 metric tons has been queried as follows:

In [13]:
```
%%sql
SELECT DISTINCT(BOOSTER_VERSION) FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (ground pad)' AND
PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

```
 * ibm_db_sa://xlz78123:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[13]:

booster_version
F9 B4 B1040.1
F9 B4 B1043.1
F9 FT B1032.1

- All three boosters have carried classified payloads, with the following estimated masses:
  - B1032.1 5300kg (USA-276 – NROL-76)
  - B1040.1 4990kg (USA-277 – Boeing X-37B OTV-5)
  - B1043.1 5000kg (USA-280 – Zuma)

Total Number of Successful and Failure Mission Outcomes

- The number of successful and unsuccessful missions has been queried as follows:

In [20]:
```
%%sql

SELECT * FROM

(SELECT SUM(NUM_OUTCOME) AS SUCCESS FROM
(SELECT COUNT(*) AS NUM_OUTCOME, MISSION_OUTCOME FROM SPACEXTBL GROUP BY MISSION_OUTCOME)
WHERE MISSION_OUTCOME LIKE 'Success%')

,

(SELECT SUM(NUM_OUTCOME) AS FAILURE FROM
(SELECT COUNT(*) AS NUM_OUTCOME, MISSION_OUTCOME FROM SPACEXTBL GROUP BY MISSION_OUTCOME)
WHERE MISSION_OUTCOME LIKE 'Failure%')
```

```
 * ibm_db_sa://xlz78123:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[20]:

success	failure
100	1

- From the perspective of mission outcome (i.e. placing the payload into orbit), all but one launches were considered successful.

Boosters Carried Maximum Payload

- The list of boosters carrying maximum payload has been queried as follows:

In [16]:
```
%%sql
SELECT DISTINCT(BOOSTER_VERSION) FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://xlz78123:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[16]:

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

- All boosters have carried multiple payloads at once, adding up to a total mass of 15600kg per each launch.

2017 Launch Records

- The table of boosters successfully landing on the ground in 2017, including launch site and month of mission has been queried as follows:

In [17]:
```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, "Landing _Outcome", BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL
WHERE YEAR(DATE) = 2017 AND "Landing _Outcome" = 'Success (ground pad)'
```

```
 * ibm_db_sa://xlz78123:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[17]:

MONTH	Landing _Outcome	booster_version	launch_site
February	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
May	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
June	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
August	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
September	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
December	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

- The boosters listed above have landed at Landing Zone 1 at Cape Canaveral Space Force Station.

Rank Landing Outcomes Between 04.06.2010 and 20.03.2017

- The number of successful booster landings between 04.06.2010 and 20.03.2017, in descending order, has been queried as follows:

In [18]:
```
%%sql
SELECT COUNT(*) AS COUNT, "Landing _Outcome" FROM SPACEXTBL
WHERE (DATE BETWEEN '2010-06-04' AND '2017-03-20')
AND ("Landing _Outcome" LIKE 'Success%')
GROUP BY "Landing _Outcome"
ORDER BY COUNT DESC
```

```
 * ibm_db_sa://xlz78123:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
Done.
```

Out[18]:

COUNT	Landing _Outcome
5	Success (drone ship)
3	Success (ground pad)

- The above landings have been performed at various locations:
  - Autonomous spaceport drone ship
    - Of Course I Still Love You — 4 landings
    - Just Read The Instructions — 1 landing
  - Landing Zone 1 — 3 landings

Section 3

Launch Sites Proximities Analysis

Launch Site Locations

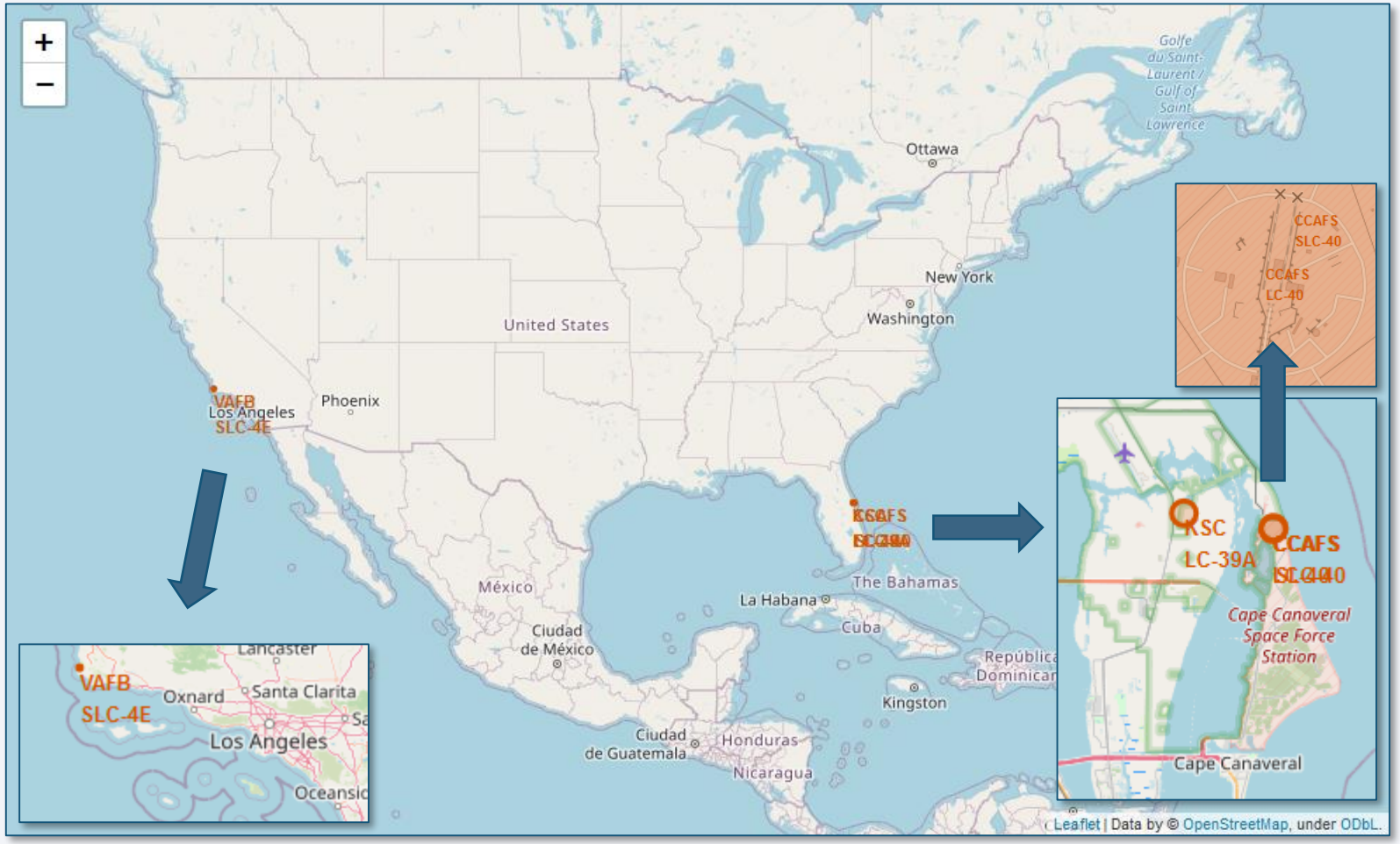
All four launch sites are close to the:

- Equator

  to make it easier for rockets to accelerate to the [first cosmic velocity] – the speed required to reach orbit

- Coast line

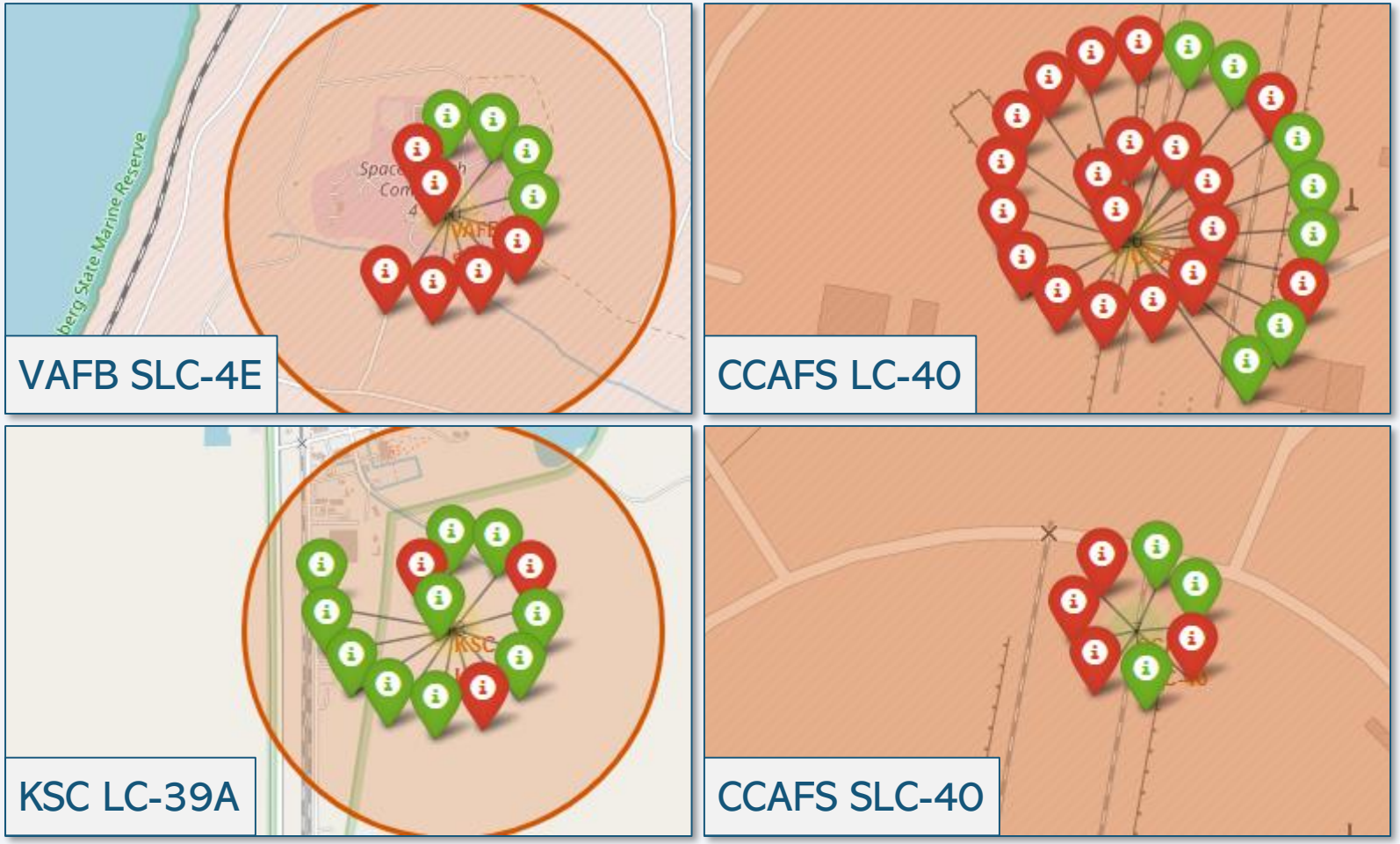
  to make sure that there is a huge unpopulated area – i.e. a safety zone – below the flight path of the rocket, in case an [in-flight accident] happens and debris falls back to Earth



Launch Site Landing Outcomes

Successful booster landings are marked with green, failed ones with red color.

- VAFB SLC-4E
  - 4 successes / 6 failures
- KSC LC-39A
  - 10 successes / 3 failures
- CCAFS LC-40
  - 7 successes / 19 failures
- CCAFS SLC-40
  - 3 successes / 4 failures



The dataset used for this subtask serves illustrative purposes only – it does not include all flights.

Launch Site Proximities

Important aspects of launch site locations are:

- Safety

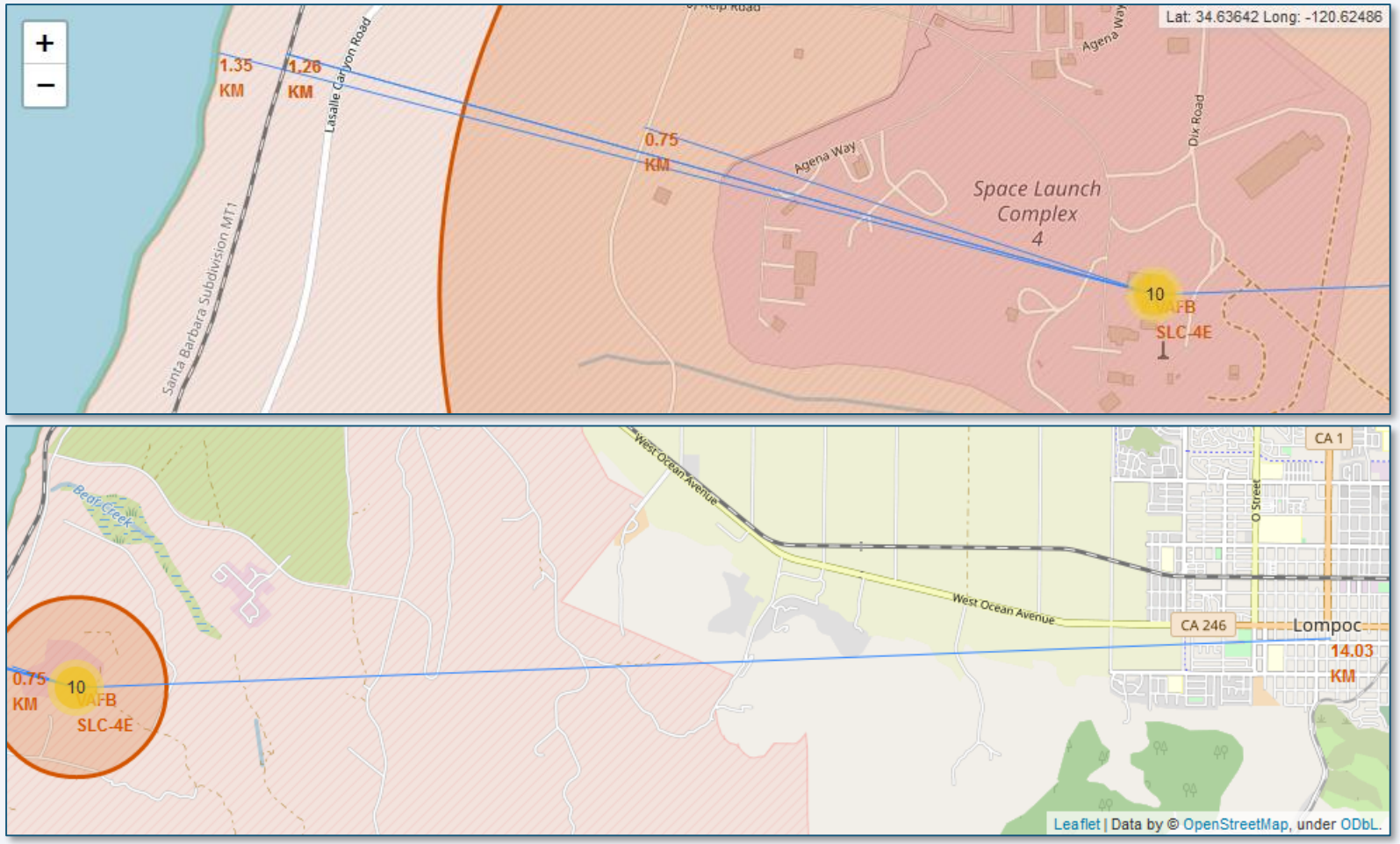
  Protection of human life in case of ground or in-flight accidents.

  - Close to coast line
  - Far from cities

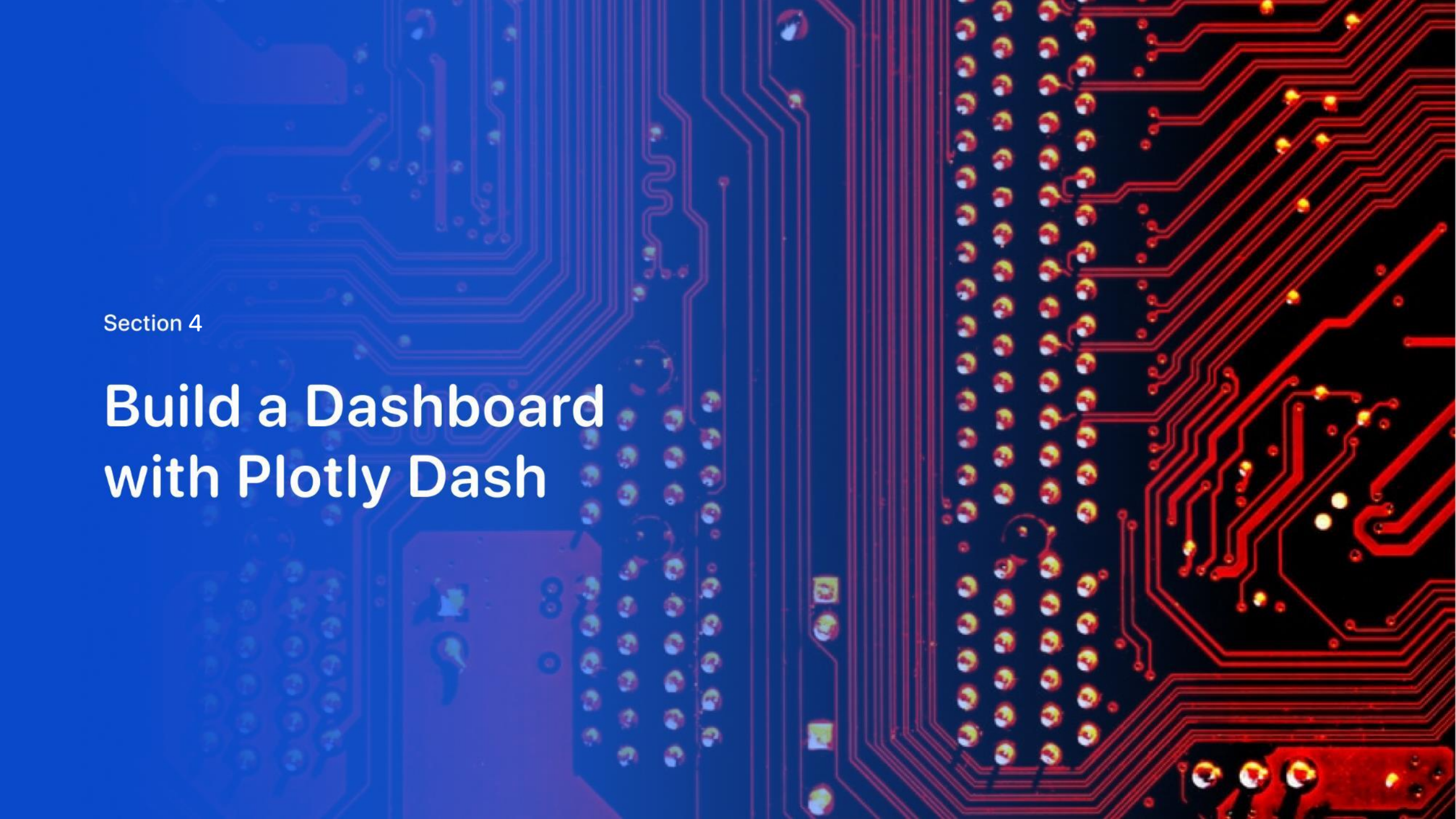
- Transportation

  Good connection to transportation networks for effective logistics and supply chain management.

  - Close to railways
  - Close to highways/roads



Vandenberg Air Force Base Space Launch Complex 4E (VAFB SLC-4E) serves as an example to illustrate the aspects listed above.

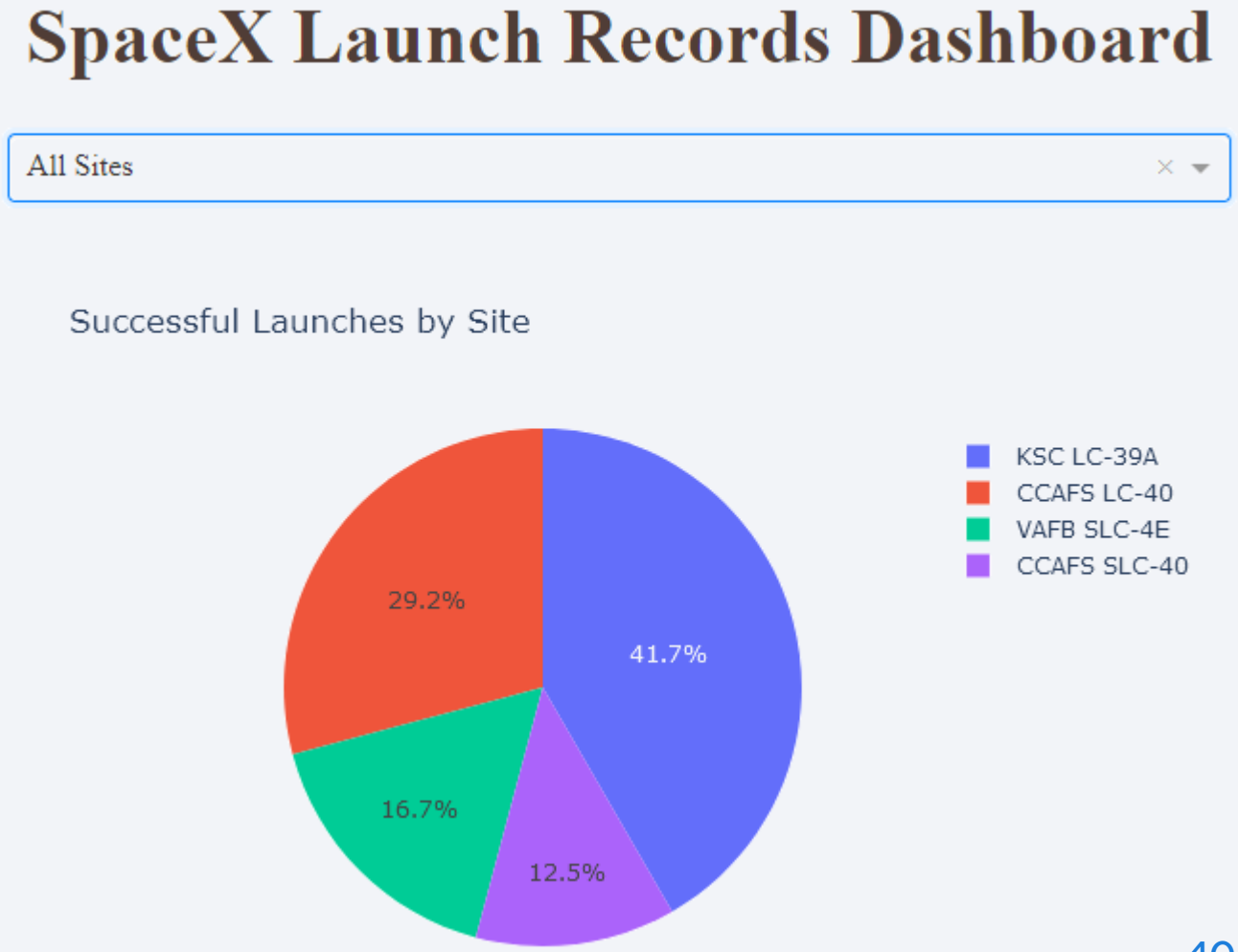

Section 4
Build a Dashboard with Plotly Dash

Successful Launches by Site

- 'Successful Launch' refers to launches ending in successful booster landings.
- The dataset can be summarized as:

Launch Site	Success	All
KSC LC-39A	10	13
CCAFS LC-40	7	26
VAFB SLC-4E	4	10
CCAFS SLC-40	3	7
Sum Total	24	56

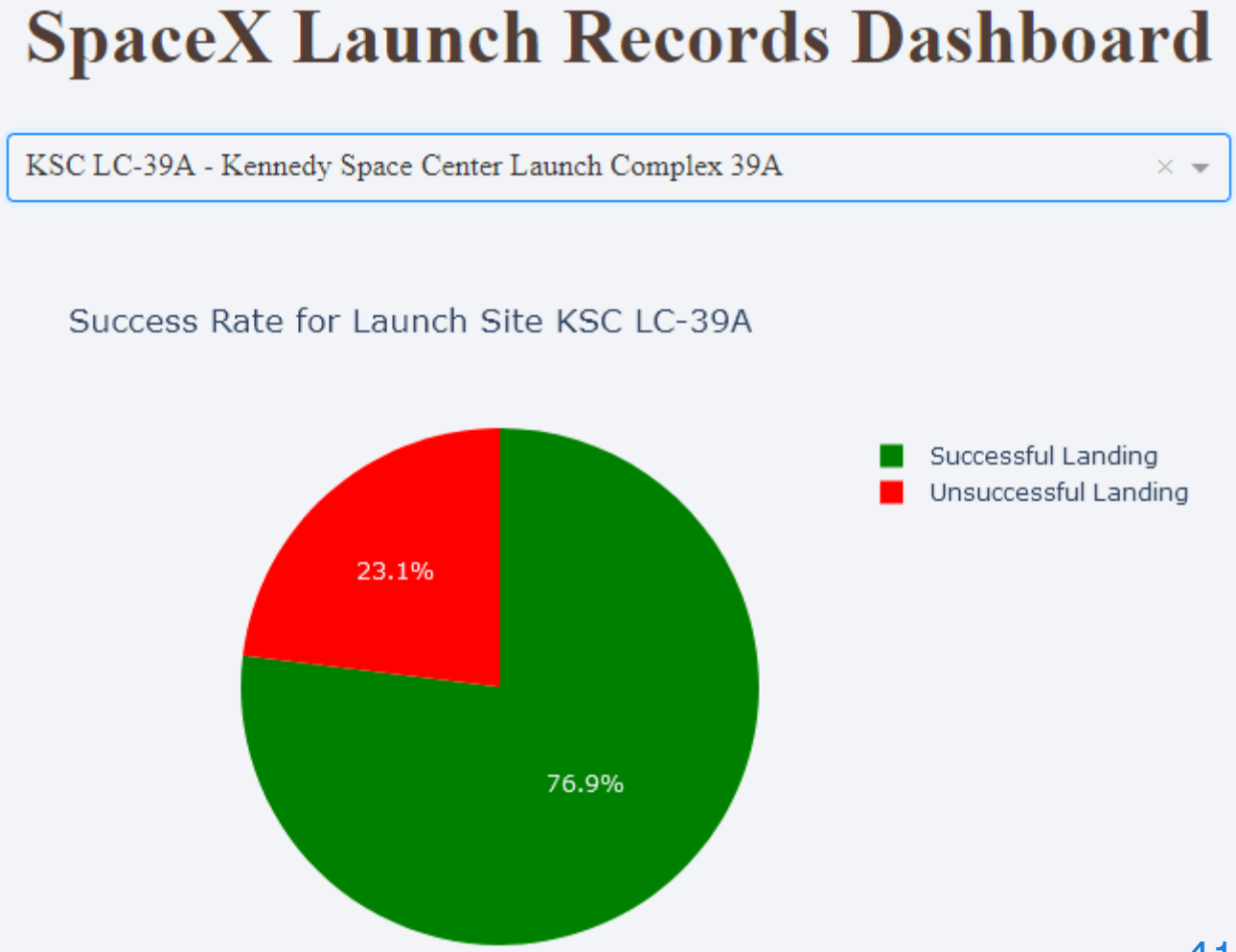
- The pie chart shows the distribution of successful launches among all sites.
- Example: VAFB SLC-4E 4/24 = 16.7%



The dataset used for this subtask serves illustrative purposes only – it does not include all flights.

Most Successful Launch Site

- 'Success Rate' of a selected launch site refers to the ratio of successful launches to all launches.
- Based on the dataset used for this subtask, KSC LC-39A is the launch site with the highest success rate.
- Calculation: 10/13 = 76.9%
  (see summarized table on previous slide)
- Success Rates of other launch sites:
  - CCAFS SLC-40 42.9%
  - VAFB SLC-4E 40%
  - CCAFS LC-40 26.9%



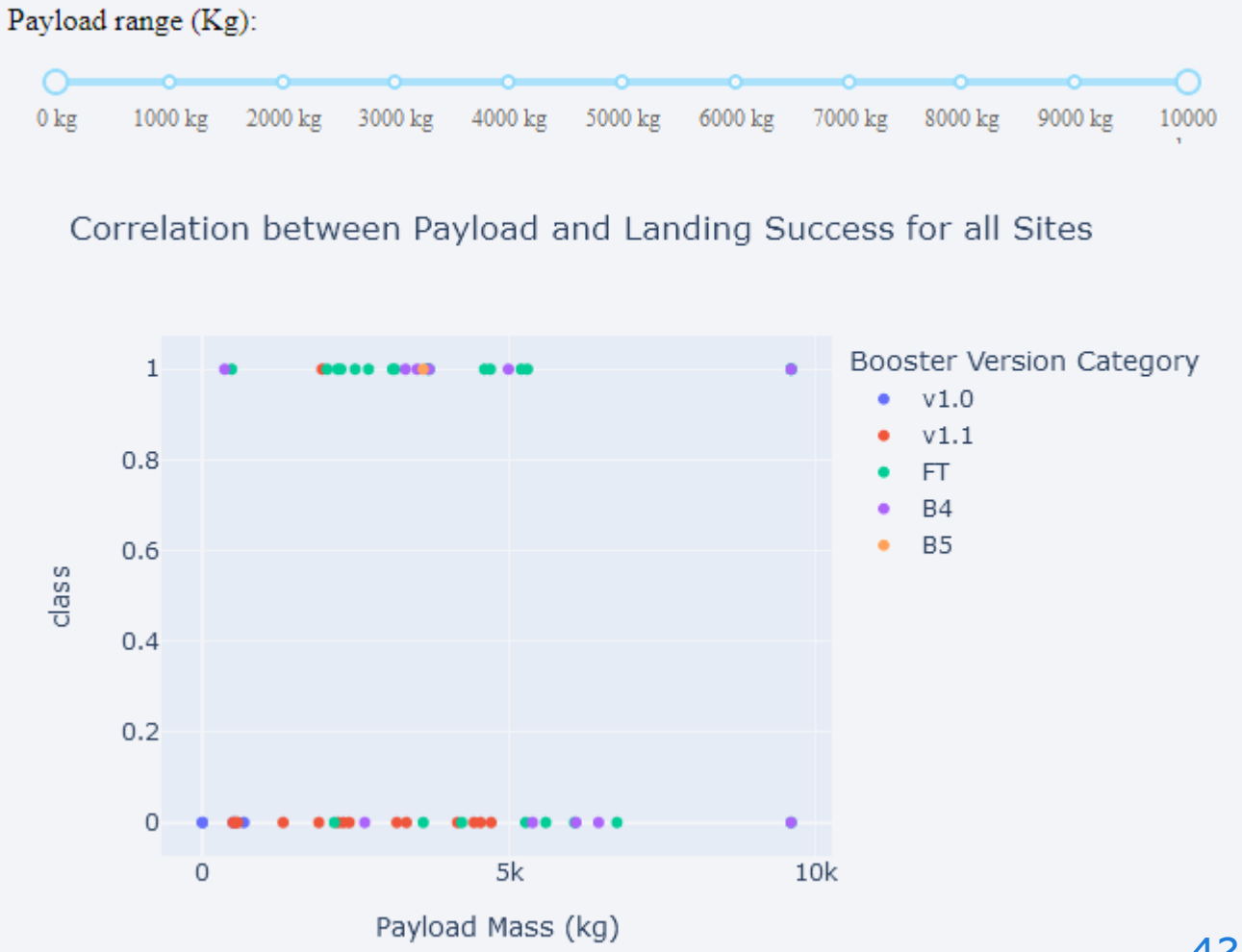
The dataset used for this subtask serves illustrative purposes only – it does not include all flights.

Successful Launches by Payload

- The scatter plot shows booster landing outcomes versus payload masses.
- The dataset can also be summarized as:

Booster Version		Success	Failure
v1.0	First Version	0	5
v1.1	Second Version	1	14
FT	Full Thrust (v1.2)	16	8
B4	Full Thrust Block 4	6	5
B5	Full Thrust Block 5	1	0
Sum Total		24	32

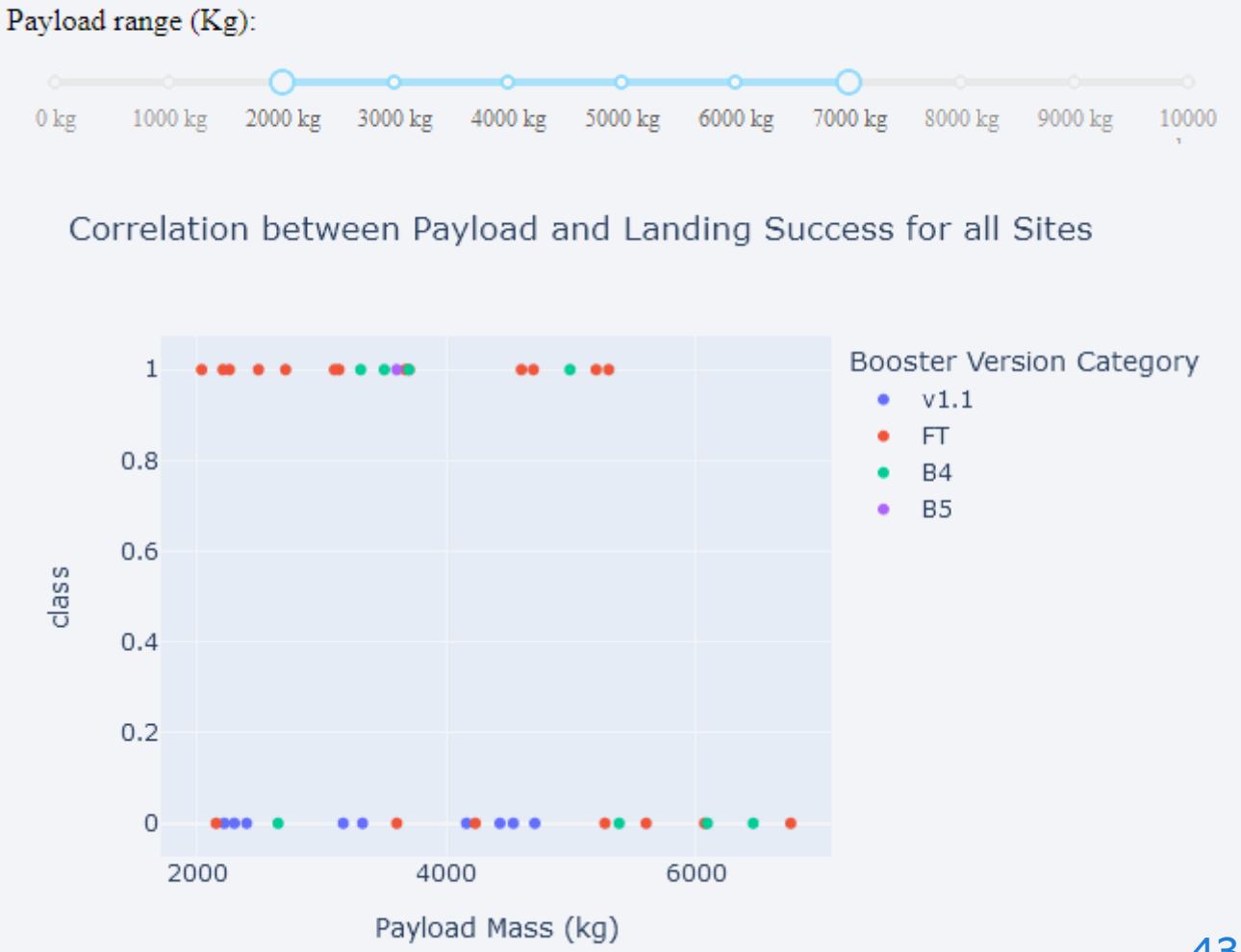
- Most boosters have launched payloads weighing between 2000kg and 7000kg.



The dataset used for this subtask serves illustrative purposes only – it does not include all flights.

Successful Launches by Payload – Reduced Range

- The most successful booster version in the reduced payload range is B5 with a success rate of 100%. However, this rate is based on one single flight only.
- Based on 20 flights, the highest realistic success rate is 65%, achieved by FT.
- The least successful booster version is v1.1 with no successful landings at all.
- Except outliers at 500kg and 9600kg (see scatter plot on previous slide), most successful missions have launched payloads between 2 and 6 metric tons.



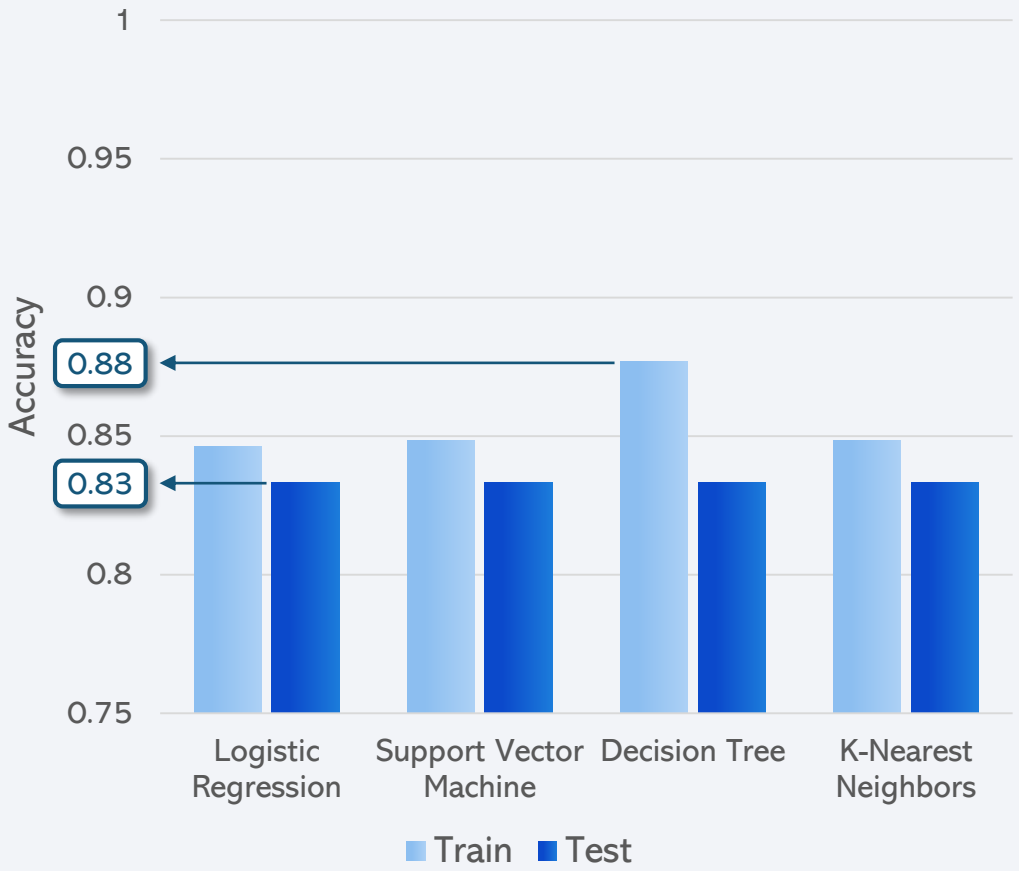
The dataset used for this subtask serves illustrative purposes only – it does not include all flights.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Training
  - Best average accuracies are shown
  - Decision Tree has the highest score
- Testing
  - All models seem to have the same accuracy
  - No clear distinction can be made between them
- Overall
  - It might seem like a good idea to choose Decision Tree as the best performing classifier
  - However, information on this is not clear at all
  - Therefore, further analyses seem to be required



Based on the dataset and the settings, as provided by the course notebook, no distinction could be made between the four classifiers.

Confusion Matrix

- All models have the same confusion matrix
- Evaluation metrics are calculated as follows:
  - Jaccard index 0.80

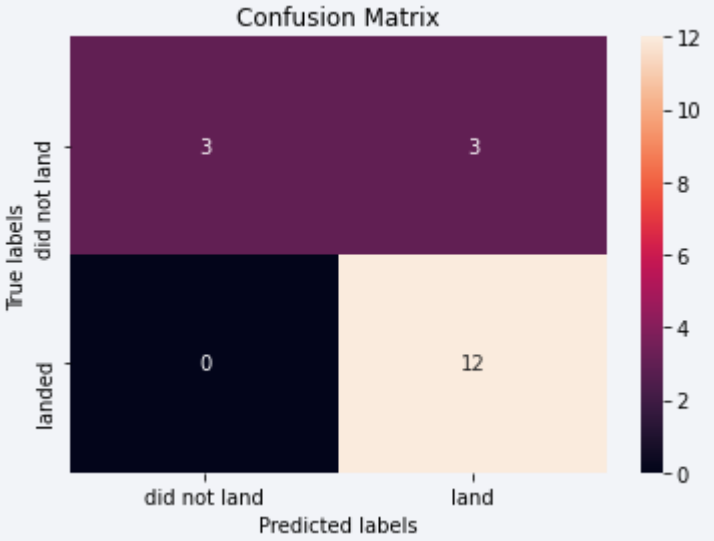
    TP / (TP + FP + FN) = 12 / (12 + 3 + 0)
  - Precision 0.80

    TP / (TP + FP) = 12 / (12 + 3)
  - Recall 1.00

    TP / (TP + FN) = 12 / (12 + 0)
  - F_1 score 0.89

    2 * (Precision * Recall) / (Precision + Recall) =
    = 2 * (0.8 * 1) / (0.8 + 1)
  - Accuracy 0.83

    (TP + TN) / (TP + TN + FP + FN) =
    = (12 + 3) / (12 + 3 + 3 + 0)



TN	FP
FN	TP

Based on the dataset and the settings, as provided by the course notebook, no distinction could be made between the four classifiers.

Model Improvement – Pipeline

- Data leakage
  - In the Labs Exercise, the entire dataset has been scaled before being split into training and testing sets and before performing cross-validation using the testing set.
  - This has carried the risk of data leakage.
- Pipeline

  The above risk has been mitigated by:
  - Performing train-test split first
  - Creating pipelines for each classifier
  - Putting the scaler into each pipeline
  - Performing cross-validation on the pipeline objects

```
In [9]:  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2)

In [10]: X_train.shape, X_test.shape, y_train.shape, y_test.shape

Out[10]: ((72, 83), (18, 83), (72,), (18,))
```

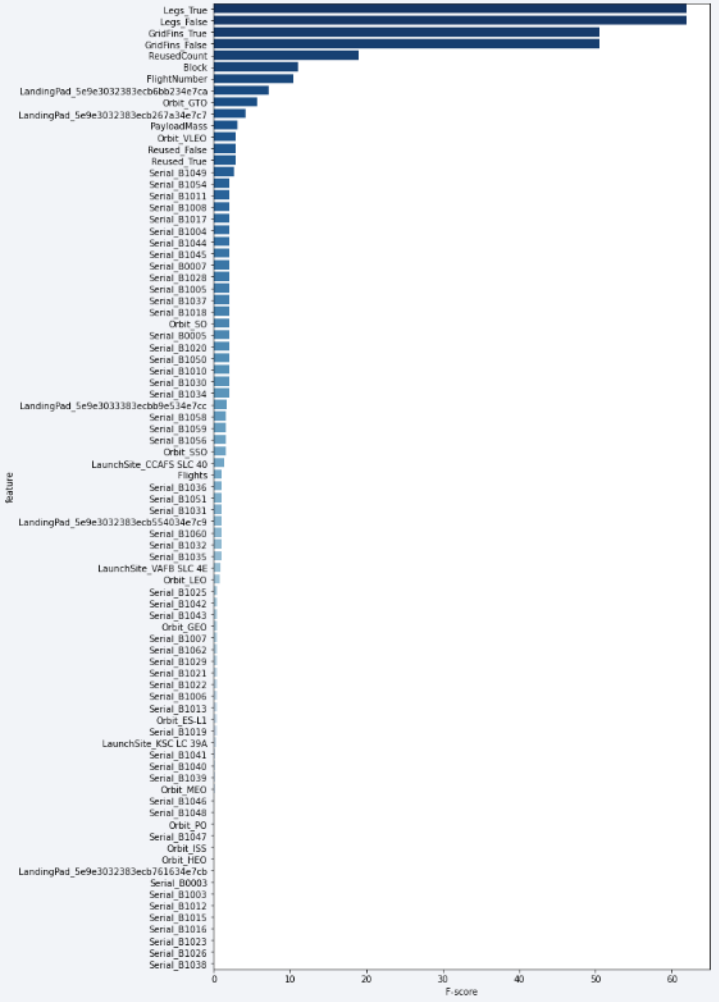
Logistic Regression

```
In [11]: pipe_lr = Pipeline([("scaler", StandardScaler()), ("lr", LogisticRegression())])
         param_lr ={"lr__C":[0.001, 0.01, 0.1, 1, 10, 100],'lr__penalty':['l2'], 'lr__solver':['lbfgs']}
         grid_lr = GridSearchCV(pipe_lr, param_grid=param_lr, cv=10)
         grid_lr.fit(X_train, y_train)
```


The notebook being shared on GitHub has been created in addition to course material: SpaceX_ML_Predict_Part5_pipeline_10foldcv

Model Improvement – Feature Selection

- Feature Selection
  - Because the training set contains more features than samples, it seems to make sense to select only the most relevant features and discard the rest.
- Preliminary Check
  - Has been performed on the entire training set
  - Is based on ANOVA* (f_classif)
  - Most relevant features: those with a p-value less than 0.05
  - Potential reduction from 83 to 10 features
- Pipelines of Classifiers
  - Feature Selection included as first step (SelectKBest)
  - Reduced number of features: 10

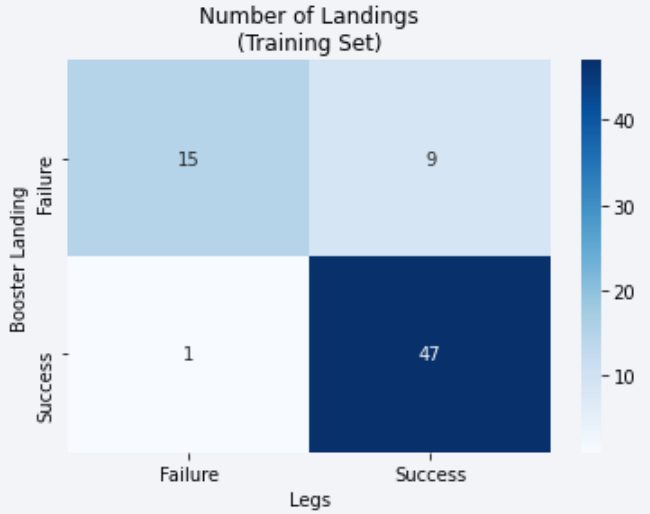


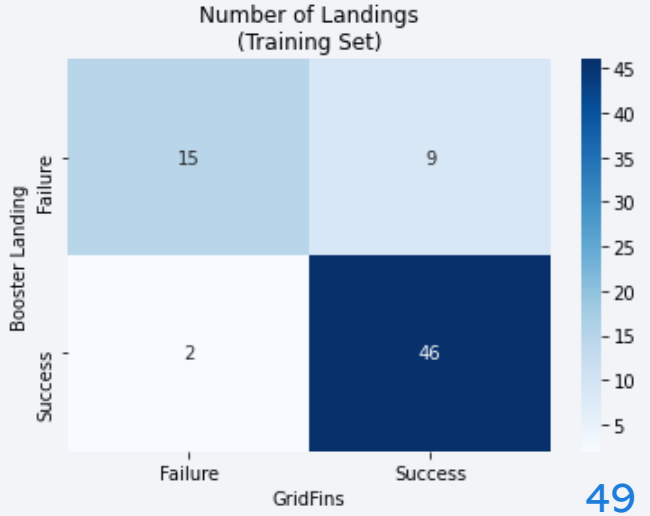
* Even though Chi-squared (chi2) and Mutual Information (mutual_info_classif) would have been better choices for the current dataset – consisting mainly of dummy variables and a binary target variable – they have been dismissed, because during test runs, grid searches on the pipeline containing the SVC algorithm have not returned anything – neither results, nor error messages. Multiple attempts at overcoming this problem – e.g. different scaling, parallel computing, omitting certain kernels – have remained unsuccessful. This suggests that the solving time of the SVC algorithm has been massively increased, possibly due to unstable feature selection in the cross-validation training folds. The other three classifier algorithms have not been impacted by this issue during the above mentioned test runs.

The notebook being shared on GitHub has been created in addition to course material: SpaceX_ML_Predict_Part5_pipeline_with_anova

Model Improvement – Most Useful Features

- Criteria
  - ANOVA has returned both F-, and p-values for each feature.
  - Two features have significantly higher F-values than the rest.
  - These two features also have by far the lowest p-values.
- Most Useful Features
  - Landing Legs
  - Grid Fins
- Intuitive Visualization
  - Confusion Matrices
    - The target variable and the two most useful features are binary.
  - Simply speaking, most of the time when landing legs and grid fins:
    - have not functioned properly, the boosters have failed to land.
    - have worked properly, the boosters have landed successfully.





The notebook being shared on GitHub has been created in addition to course material: SpaceX_ML_Predict_Part5_pipeline_with_anova

Model Improvement – Performance Comparison

- Training
  - Better cross-validation for most classifiers
  - No improvement for K-Nearest Neighbors
- Testing
  - No improvement for most classifiers
  - Worse performance for Decision Tree
- Overall
  - Better training and worse test performance suggests that the Decision Tree has been overfitted
  - Generalization performances have not improved
  - Therefore, more input data seem to be required

Performance Change	Training	Testing
Logistic Regression	0.01	0
Support Vector Machine	0.03	0
Decision Tree	0.04	-0.11
K-Nearest Neighbors	0	0

Based on the dataset, notebook settings and proposed improvements, a clear distinction could still not be made between the four classifiers.

Conclusions

Findings

- Most Useful Features
  - Landing Legs
  - Grid Fins
- Best Test Accuracy
  - 0.83

Implications

- No performance improvement
  - Neither using pipelines
  - Nor performing feature selection
- No clear distinction possible
  - Best performing classifier could not be selected

Recommendation

- More input data seem to be required
  - Number of samples in dataset should be increased

So far, the objective – i.e. training a model that classifies booster landing outcomes with good accuracy – has not yet been achieved.

Appendix A

Increased Dataset (Classification)

Summary

In order to help choosing the optimal classifier model, more data has been collected, with which the analyses have been run multiple times, using different train-test splits.

Steps	Slides
• More input data has been collected	More Input Data
• Classifier performances have been compared to baseline	Model Performance
• Increased dataset has been split up in different ways	Random Seeds
• Performances of different splits have been compiled	Performance Sensitivity
• Statistics of performances have been calculated	Performance Statistics
• Bias and variance have been calculated	Bias and Variance
• Bias-variance tradeoff has been determined	Bias-Variance Tradeoff
• Optimal classifier model has been selected	Optimal Classifier Model

More Input Data

- Increment

Dataset	Baseline*	Increased
Launches considered until	05.11.2020	25.02.2022
Number of launches	90	138

* Baseline refers to the Labs Exercise

  - Number of samples increased by 53%

- Consequence
  - All four classifier algorithms have been run with the increased dataset.
- Question
  - Does increasing the amount of input data lead to an increment in the performance of the models?

Out[61]:

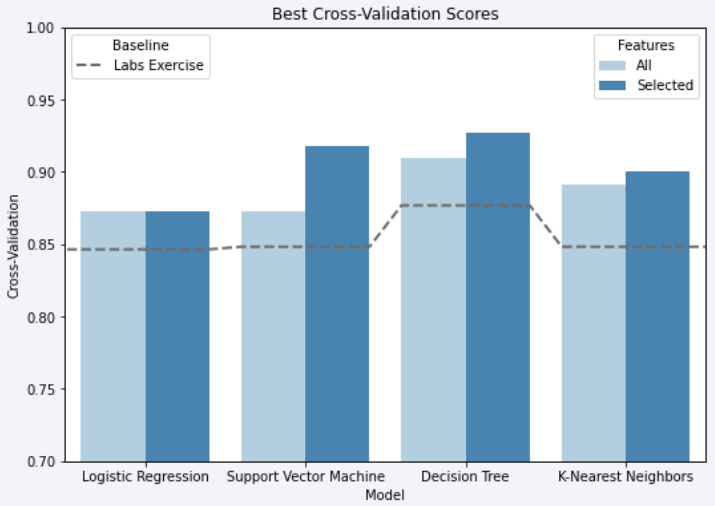
	FlightNumber	Date
0	1	2010-06-04
1	2	2010-12-08
2	3	2012-05-22
3	4	2013-03-01
4	5	2013-09-29
...	...	...
133	134	2022-01-31
134	135	2022-02-02
135	136	2022-02-03
136	137	2022-02-21
137	138	2022-02-25

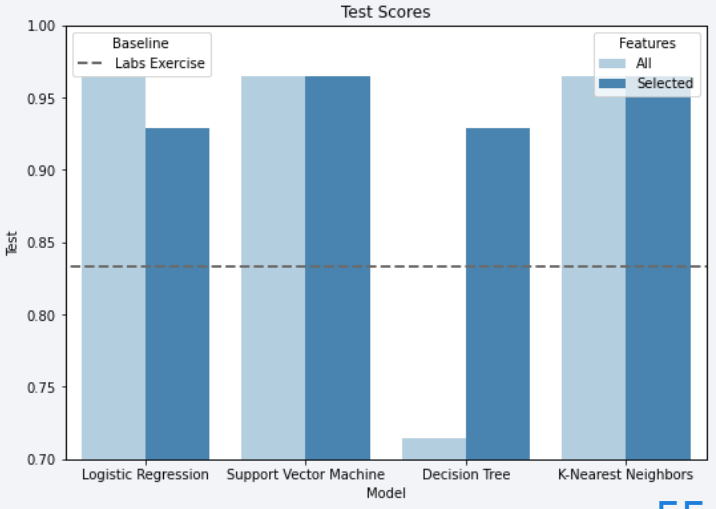
138 rows × 15 columns

The notebook being shared on GitHub has been created in addition to course material: SpaceX_ML_Predict_full

Model Performance

- Training
  - Clear performance improvement compared to baseline.
  - Feature selection seems to improve cross-validation scores.
- Testing
  - Comparison to baseline
    - Most models seem to have far better test scores than the baseline.
  - Unusual result
    - However, most models have also higher test scores than training scores.
  - Unclear performance improvement
    - It is therefore unclear, if more input data alone leads to better overall results.
    - In this particular case, the test split seems to be a lot different – i.e. better, easier – than the training data, resulting in the unexpectedly high test scores.
- Question
  - Does model performance depend heavily on how splits are made?





The notebook being shared on GitHub has been created in addition to course material:

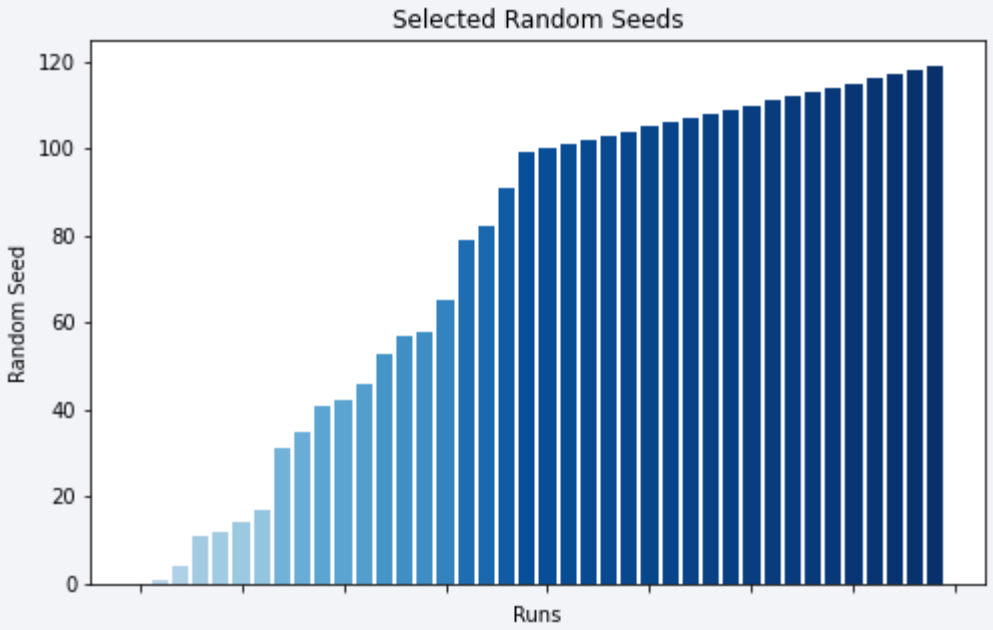
SpaceX_ML_Predict_full

Random Seeds

- In total 40 seeds
  - First half: Uniformly distributed between 0 and 99.
  - Second half: Equally spaced between 100 and 119.
- Analysis
  - All four classifiers, both without and with feature selection, have been run with all the random seeds.
  - The following have been saved for each iteration:
    - Training and Testing Accuracies
    - Indices of Train and Test Splits
    - Best Parameters of all Classifiers
- Question
  - How stable are model performances over different test splits?

```
[12]: for i, rs in enumerate(random_seeds):
```

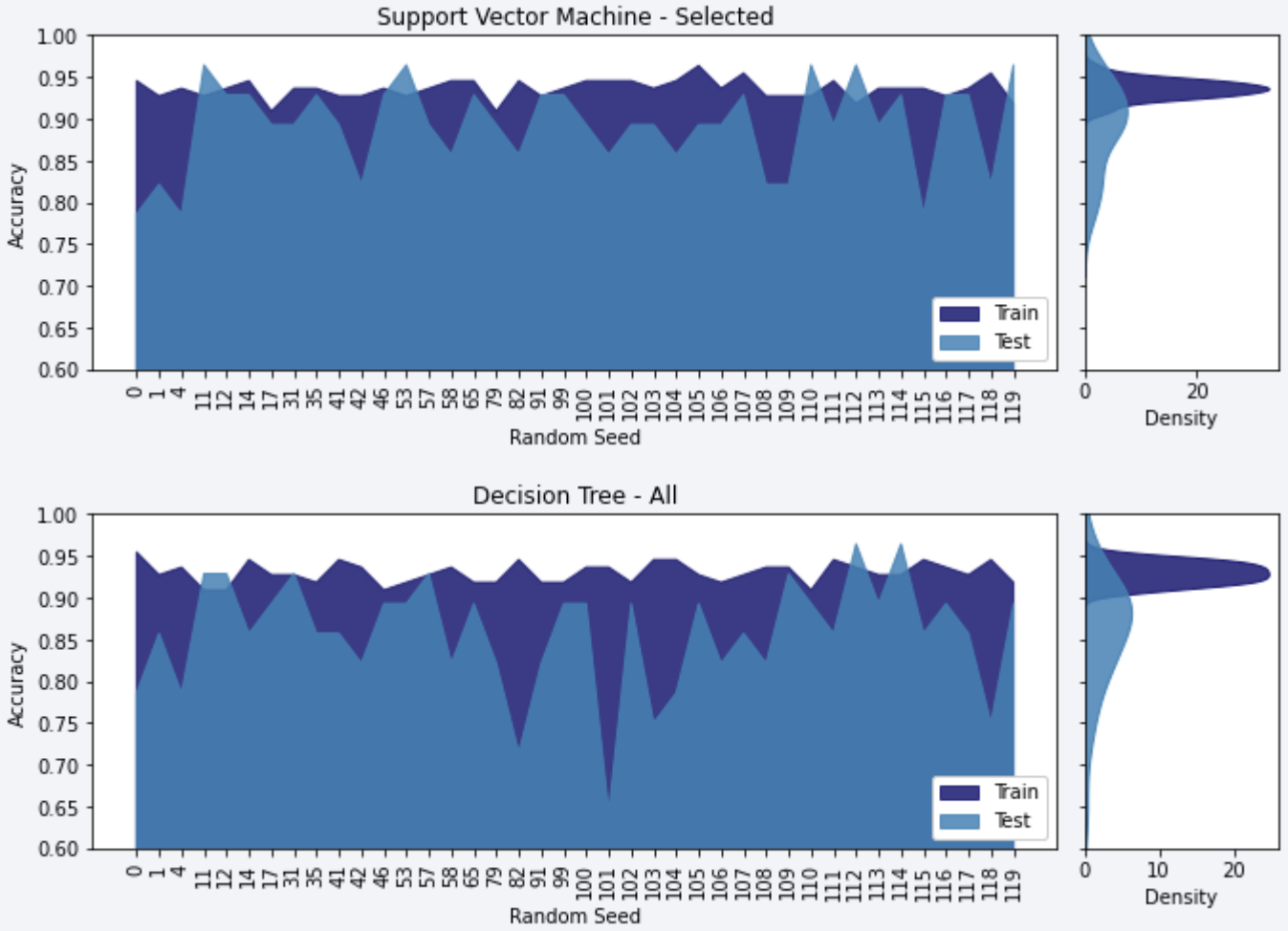
```
train_test_split(df_X, df_y, test_size=0.2, random_state=rs)
```



The notebook being shared on GitHub has been created in addition to course material: SpaceX_ML_Predict_Random_Seeds_Generate

Performance Sensitivity

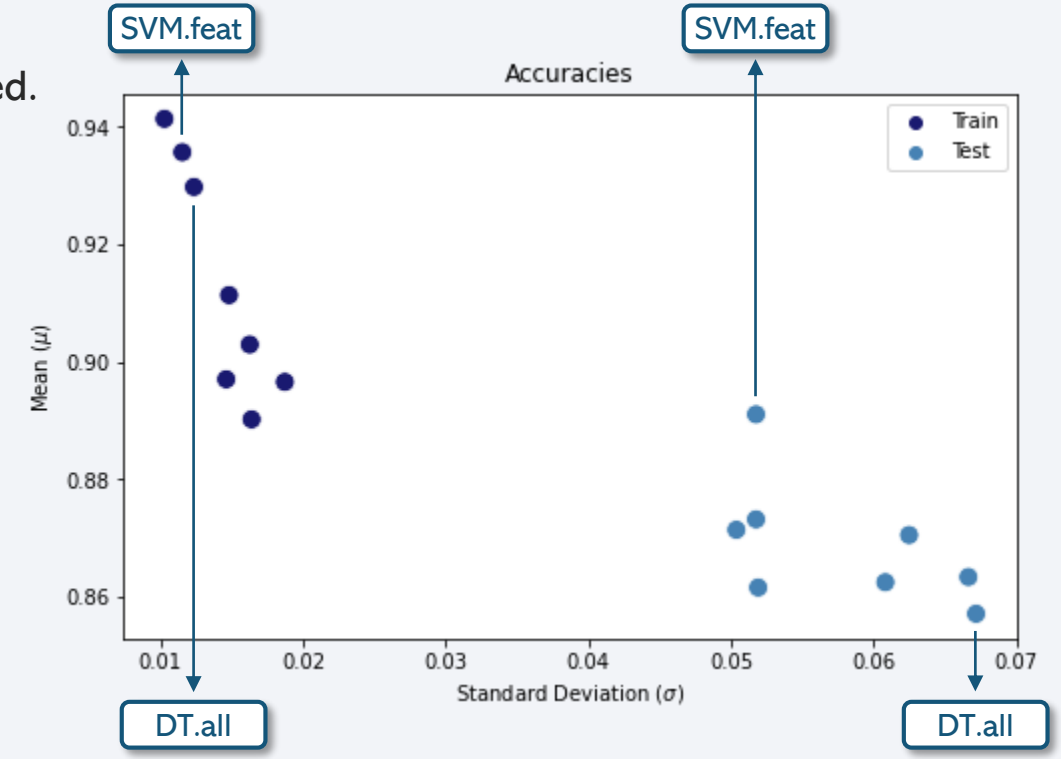
- Different test splits
  - Model performance seems to be sensitive to changes in test split.
- Examples
  - Support Vector Machine – Selected
    - Train performance is quite stable.
    - Test performance has some variance.
  - Decision Tree – All
    - Train performance is less balanced.
    - Test performance has high variance.
- Question
  - What are the statistics of the model performances?



The notebook being shared on GitHub has been created in addition to course material: SpaceX_ML_Predict_Random_Seeds_Analyze

Performance Statistics

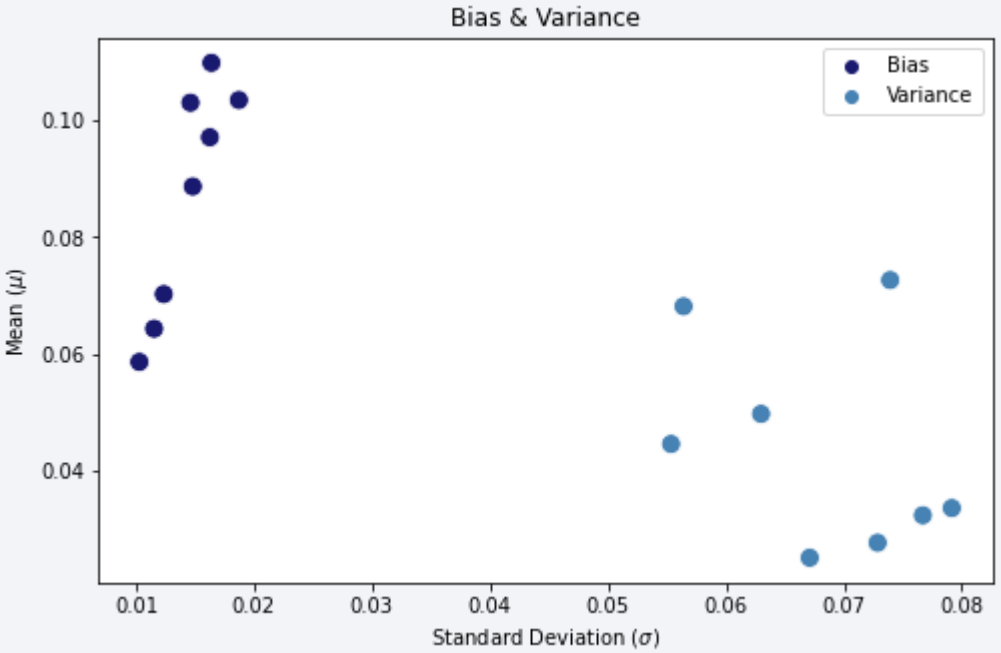
- Calculation
  - Mean and std. deviation of accuracies have been calculated.
- Scatter Plot
  - The difference is clear between train and test accuracies.
  - Interpretation
    - High mean:
      Regardless of test split, model is expected to yield a high accuracy.
    - Low standard deviation:
      Model performance is relatively stable over different test splits.
  - Examples
    - Decision Tree – All:
      Potential overfit: Good training, but poor testing performance.
    - Support Vector Machine – Selected (feat):
      Possibly an optimal fit: Good training and testing performances.
- Question
  - What is the relationship between train and test accuracies?



The notebook being shared on GitHub has been created in addition to course material: SpaceX_ML_Predict_Random_Seeds_Analyze

Bias and Variance

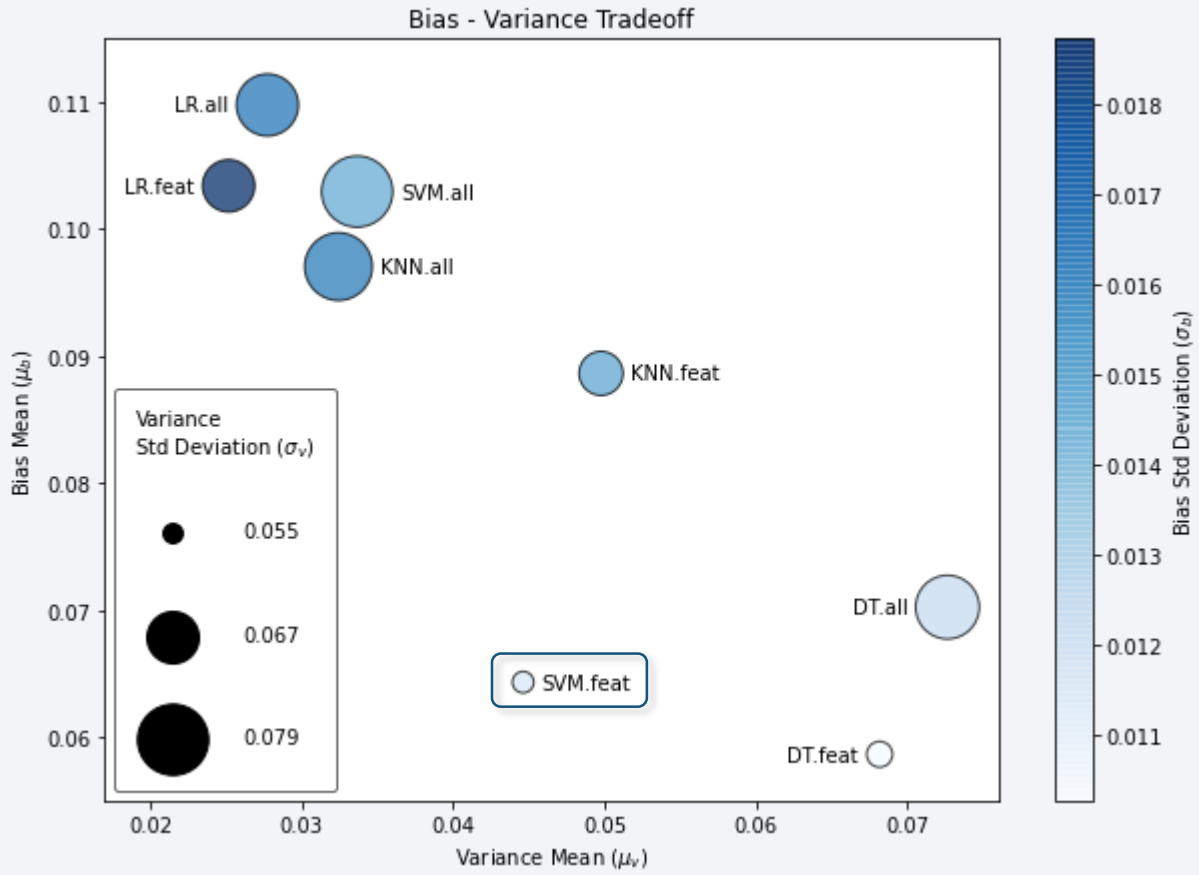
- Calculation
  - Bias = 1 – Training Accuracy
  - Variance = Training Accuracy – Testing Accuracy
  - Statistics : Mean, Standard Deviation
- Scatter Plot
  - The difference is clear between biases and variances.
  - Biases
    - Low standard deviation:
      Due to cross-validation, models are stable over different data splits.
  - Variances
    - High standard deviation:
      Model performances seem to be sensitive to changes in test splits.
- Question
  - What tradeoff can be made between bias and variance?



The notebook being shared on GitHub has been created in addition to course material: SpaceX_ML_Predict_Random_Seeds_Analyze

Bias-Variance Tradeoff

- Bubble Plot
  - Interpretation
    - Coordinates:
      Regardless of test splits, what values are expected?
    - Sizes, Colors:
      How stable are those values over different test splits?
  - Regions
    - Underfit:
      High bias, low variance – upper left corner
    - Overfit:
      Low bias, high variance – lower right corner
- Optimal Model
  - Support Vector Machine with Feature Selection
    - Low bias, low variance – closest to lower left corner.
    - Its performance is stable over different test splits.
- Question
  - What are the hyperparameters of the optimal model?



The notebook being shared on GitHub has been created in addition to course material: SpaceX_ML_Predict_Random_Seeds_Analyze

Optimal Classifier Model

Support Vector Machine with Feature Selection

- Dataframe
  - It contains the results and parameters of each iteration.
- Selection
  - Dataframe has been filtered in several ways:
    - Sorted by test and train accuracies, in descending order
    - Sorted by bias and variance, in ascending order
    - Grouped by hyperparameter combinations, being counted
  - Filtered results have been evaluated against different criteria:
    - Most frequent model having overall highest test score
    - Best model having highest test score which is less than train score
    - Best model among lowest bias models having lowest variance
    - Most frequent model overall
- Hyperparameters*
  - C = 0.001
  - gamma = 10
  - kernel = poly

[16]:

			Accuracies
			Test
(Parameters, C)	(Parameters, gamma)	(Parameters, kernel)	
0.001000	10.00	poly	9
0.031623	1.00	poly	8
1.000000	1.00	rbf	5
0.001000	1.00	poly	4
0.031623	0.10	poly	4
1000.000000	0.10	rbf	4
31.622777	0.10	rbf	3
1.000000	10.00	rbf	1
31.622777	1.00	rbf	1
1000.000000	0.01	rbf	1

* Hyperparameters of Feature Selection: score_func=f_classif, k=10.
See Model Improvement – Feature Selection for further information.

The notebook being shared on GitHub has been created in addition to course material: SpaceX_ML_Predict_Random_Seeds_Analyze

Conclusions

Findings

- Increased Dataset
  - Improves Cross-Validation
- Different random seeds
  - Generalization performances are sensitive to changes in data splits
- Feature Selection
  - Has a tendency to decrease bias
- Decision Tree
  - Seems to be prone to overfitting

Recommendation

- Model deployment
  - Following Pipeline should be trained on the entire dataset:
    - SelectKBest
      - score_func = f_classif
      - k = 10
    - StandardScaler
    - SVC
      - C = 0.001
      - gamma = 10
      - kernel = poly
- Expected Accuracy
  - $0.89\ ^{+0.07}_{-0.10}$

Appendix B

Digital Repository (Links & Datasets)

Links

Source: https://github.com/adamgyonyor/IBM-DS0720EN/tree/master

	Notebooks	Datasets
Course Material	• Module 1 - Data Collection API.ipynb • Module 1 - Data Collection with Web Scraping.ipynb • Module 1 - Data Wrangling.ipynb • Module 2 - EDA with Data Visualization.ipynb • Module 2 - EDA with SQL.ipynb • Module 3 - Interactive Visual Analytics w. Folium.ipynb • spacex_dash_app.py • Module 4 - Machine Learning Prediction.ipynb	Provided by IBM as part of course material: • API_call_spacex_api.json • Wikipedia – List of Falcon 9 launches • dataset_part_1.csv • dataset_part_2.csv • Spacex.csv • spacex_launch_geo.csv • spacex_launch_dash.csv • dataset_part_3.csv
Model Improvement	• SpaceX_ML_Predict_Part5_pipeline_10foldcv.ipynb • SpaceX_ML_Predict_Part5_pipeline_with_anova.ipynb	
Increased Dataset	• SpaceX_ML_Predict_full.ipynb* • SpaceX_ML_Predict_Random_Seeds_Generate.ipynb • SpaceX_ML_Predict_Random_Seeds_Analyze.ipynb	• SpaceX_ML_Predict_X.csv • SpaceX_ML_Predict_y.csv • SpaceX_ML_Predict_Random_Seeds.xlsx

* Hyperparameter grid of SVC has been modified to exclude gamma=1000, because it has not converged.

Thank you!