IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

MILOŠ FEJERČÁK
18/06/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection – sources: SpaceX public API, Wikipedia
  - Data Wrangling – extracting SpaceX launch outcome information

  - Exploratory Data Analysis (EDA) and Data Visualization – using python libraries: matplotlib, plotly, seaborn
  - EDA with SQL – extracting queries from SQL database

  - Building an interactive world map with Folium
  - Building an interactive Dashboard application with Plotly Dash

  - Predictive analysis using Machine Learning models: Linear logistic regression, Support Vector Machine (SVM), Decision Trees and K-Nearest Neighbors (KNN) models.

- Summary of all results
  - Exploratory Data Analysis results
  - Interactive maps and Dashboard
  - Results of predictive analysis

Introduction

- Project background and context

  - SpaceX is the most successful company of the commercial space age (~~and probably the most unresponsible~~), making space travel affordable. The company advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. Based on publicly available data, and utilizing the machine learning models, we are going to predict if SpaceX will reuse the first stage.

- Problems you want to find answers

  - What are the key factors helping us to predict if the SpaceX stage 1 rocket will achieve successful landing?
  - What was the development of Falcon 9 rockets successful landings over recent years?
  - Which is the Machine Learning model giving us the most accurate predictions on landing?

Section 1

Methodology

Methodology

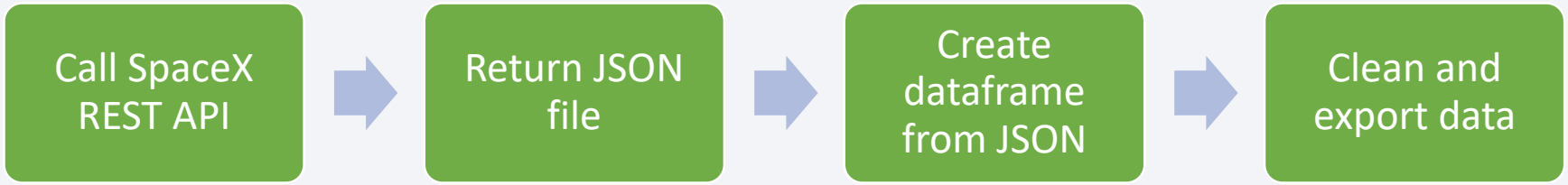
Executive Summary

- Data was collected from SpaceX REST API and using webscraping from Wikipedia
- Subsequently, data were treated for missing values, and categorized for future visualizations and prepared for the machine learning models.
- The exploratory data analysis (EDA) was performed using standard python libraries for plotting based on data gathered from available SQL databases.
- The interactive visual analytics was performed by creating world map visualizations with Folium and interactive dashboard environment Plotly Dash.
- For Machine Learning predictive analysis we used python library – scikit-learn with several classification models to find the best accuracy of prediction.

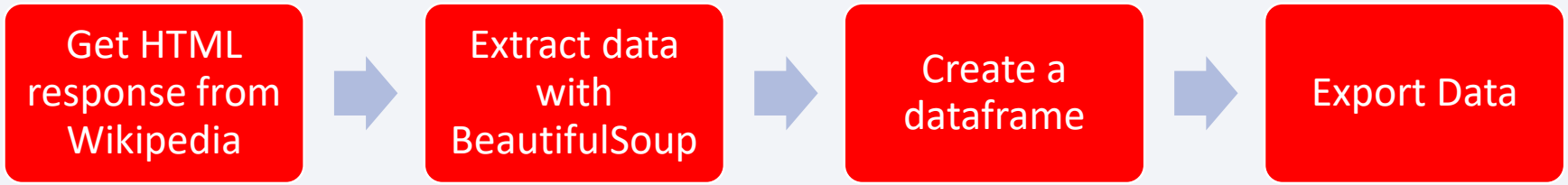
Data Collection

- Data was collected from SpaceX REST API and using webscraping from Wikipedia

From SpaceX REST API we obtained data about: rocket type, rocket launches and payload

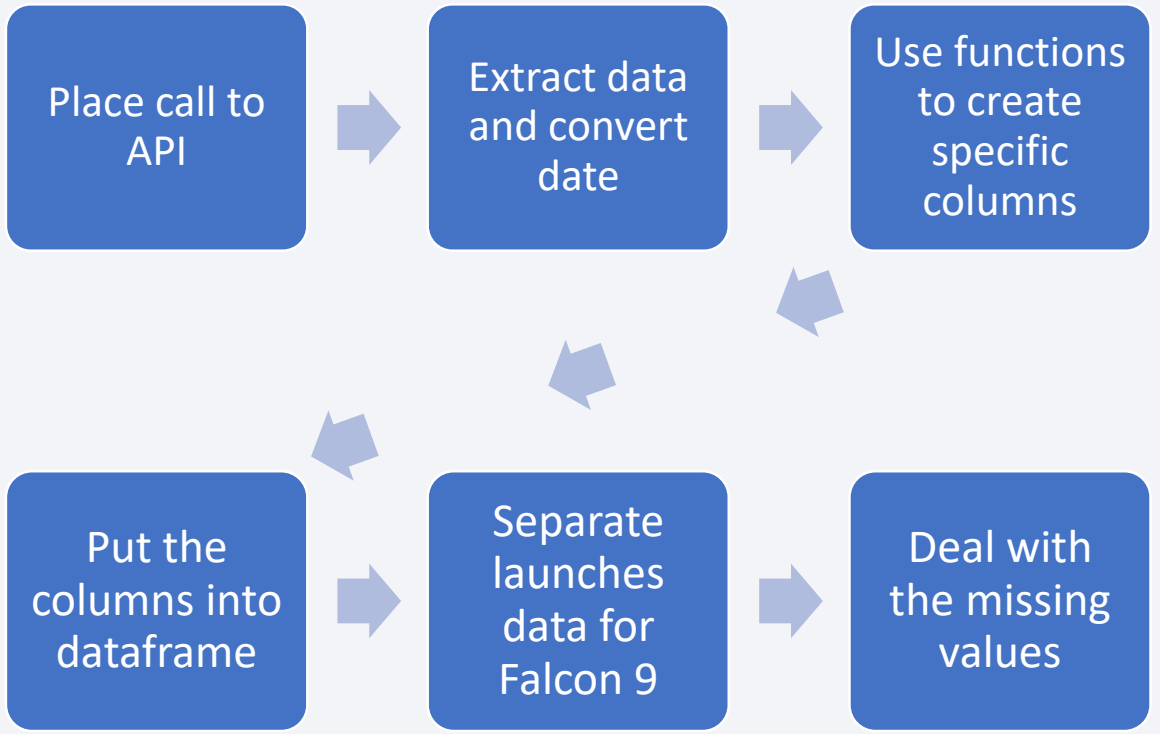


From Wikipedia we obtained data about: rocket launches, landing and payload



Data Collection - SpaceX API

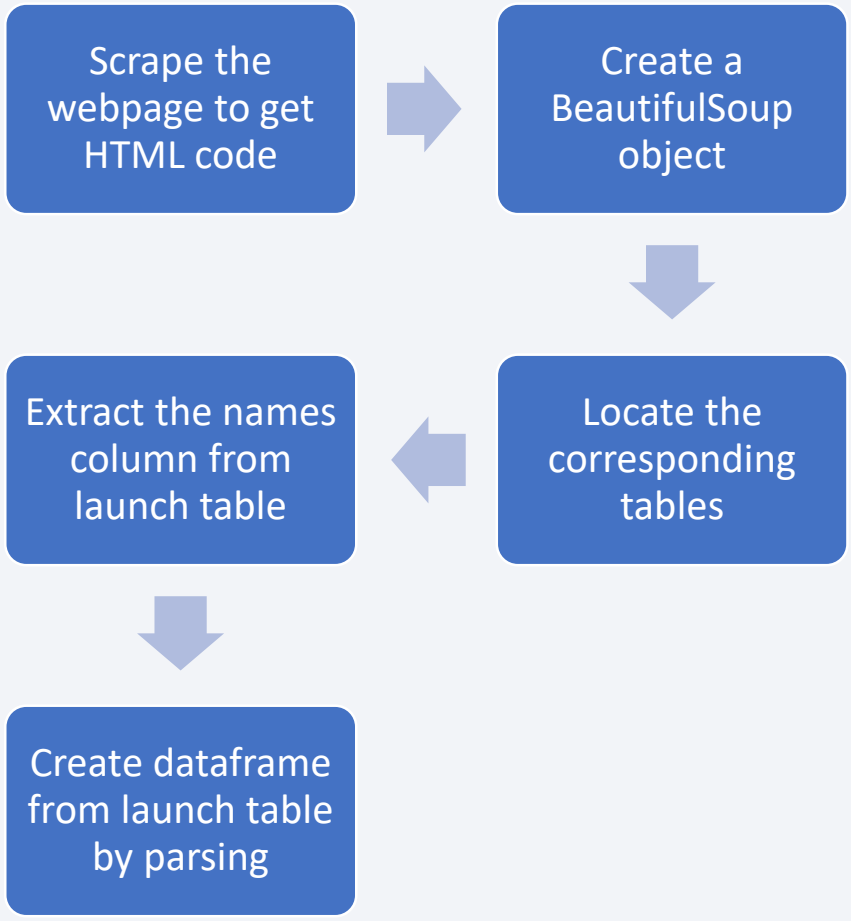
- Data was collected from public SpaceX API.
- From a GET request on a SpaceX API we received data which we stored to pandas dataframe, making it ready for next analysis.



- GitHub URL for Data Collection: https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Data was web-scraped from Wikipedia free encyclopedia.
- We collected the tabulated data with information about rocket launches and this data we stored to pandas dataframe, making it ready for next analysis.

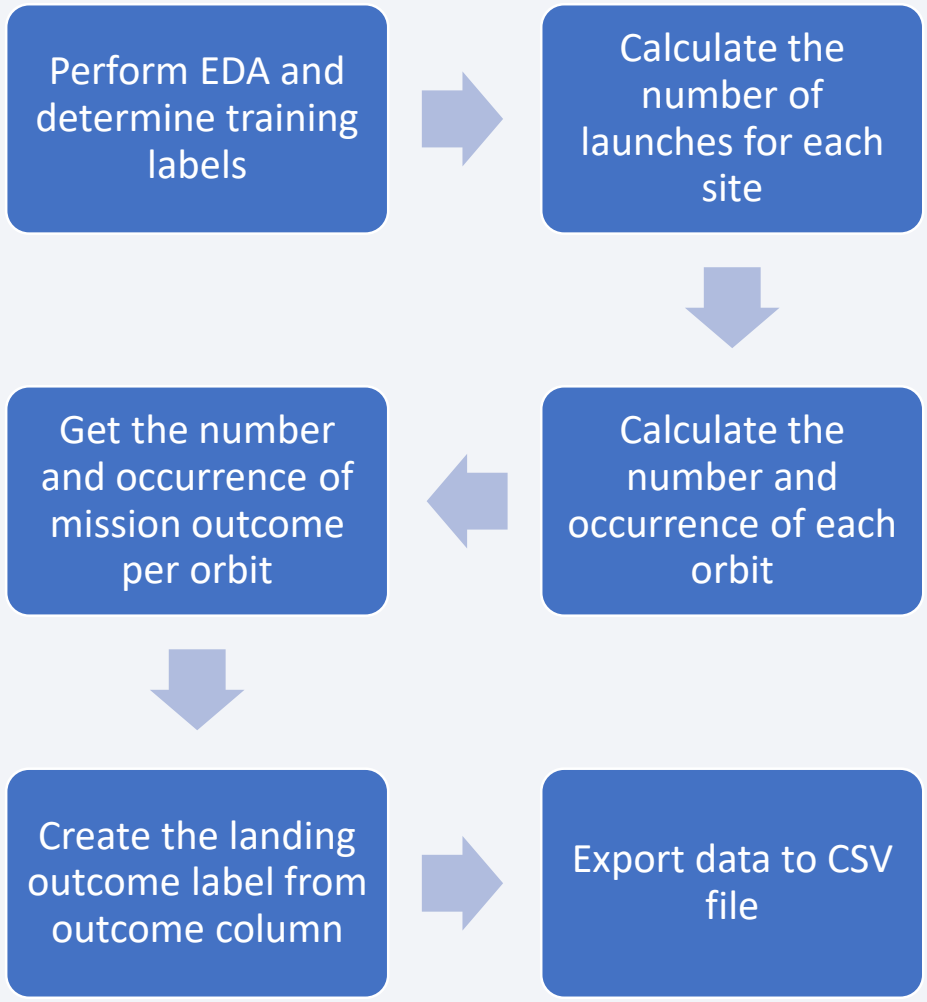


- GitHub URL for Web Scraping: https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- The csv data were taken from previous section, and cleaned to get information about: launch sites, orbit type and mission outcome.
- Mission data outcome were converted to categorical values, for each Falcon 9 first stage successful landing, where we use 1 for successful landing and 0 for failure.
- This data we stored to pandas dataframe, making it ready for next analysis.

- GitHub URL Data Wrangling: https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- Charts used for EDA: Scatter plots, Bar plot and Line plot.

- **Scatter plots:** Flight number vs Payload Mass, Flight Number vs Launch Site, Payload Mass vs Launch Site, Orbit vs Flight Number, Payload Mass vs Orbit type, Orbit type vs Payload Mass
- To visualize relationship between variables, i.e. correlation.

- **Bar Plot**: Success rate vs Orbit Type.
- To visualize relationship between numeric and categoric variables.

- **Line Plot**: Success rate vs Year.
- To visualize trends over time and examine global behavior

- GitHub URL EDA with data visualization: https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/edadataviz.ipynb

EDA with SQL

- Using queries on SQL data we extracted information about:
- Launch Sites,
- Payload Mass
- Dates
- Booster types
- Mission outcomes

- GitHub URL EDA with SQL: https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Markers of all launch sites:
- We added map objects such as markers, circles and lines to describe success or failure for each site on interactive folium map.
- Colored markers describe launch outcome of launch site:
- Based on categorical values of successful/failure launch assigned with 1/0 respectively, we plotted markers of this events using green/red colors.
- Lines describe proximities from launch site to points of interest:
- Here using lines showing distances from launch sites we try to examine importance of position for a launch site. We focused on distances from: railways, highways, cities, coastline and proximity to equator.

- GitHub URL Folium map: https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/lab_jupyter_launch_site_location.ipynb

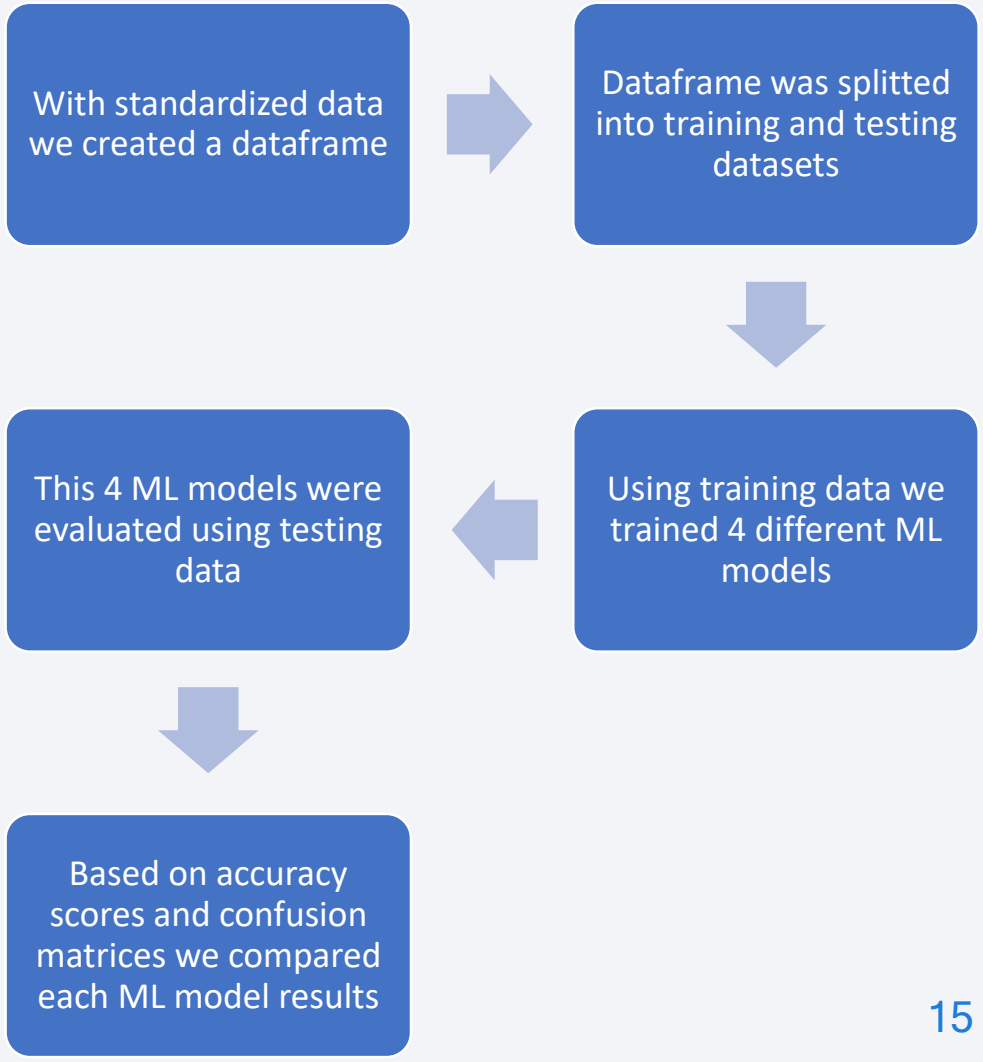
Build a Dashboard with Plotly Dash

- The Dashboard app consists of a dropdown menu, pie chart, payload range slider and a scatter plot.
- In Dropdown menu you can choose one launch site or all of them.
- In respect to the Dropdown selection, the Pie plot will render to visualize success vs failure percentage result of the launch site/s.
- Using the range slider you can choose range of a payload in 1000 kg metrics.
- Finally, the scatter plot will render according to the previously selected payload mass range and launch site, displaying the booster version category vs payload mass.

- GitHub URL Plotly Dash: https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/spacex_dash_app.py

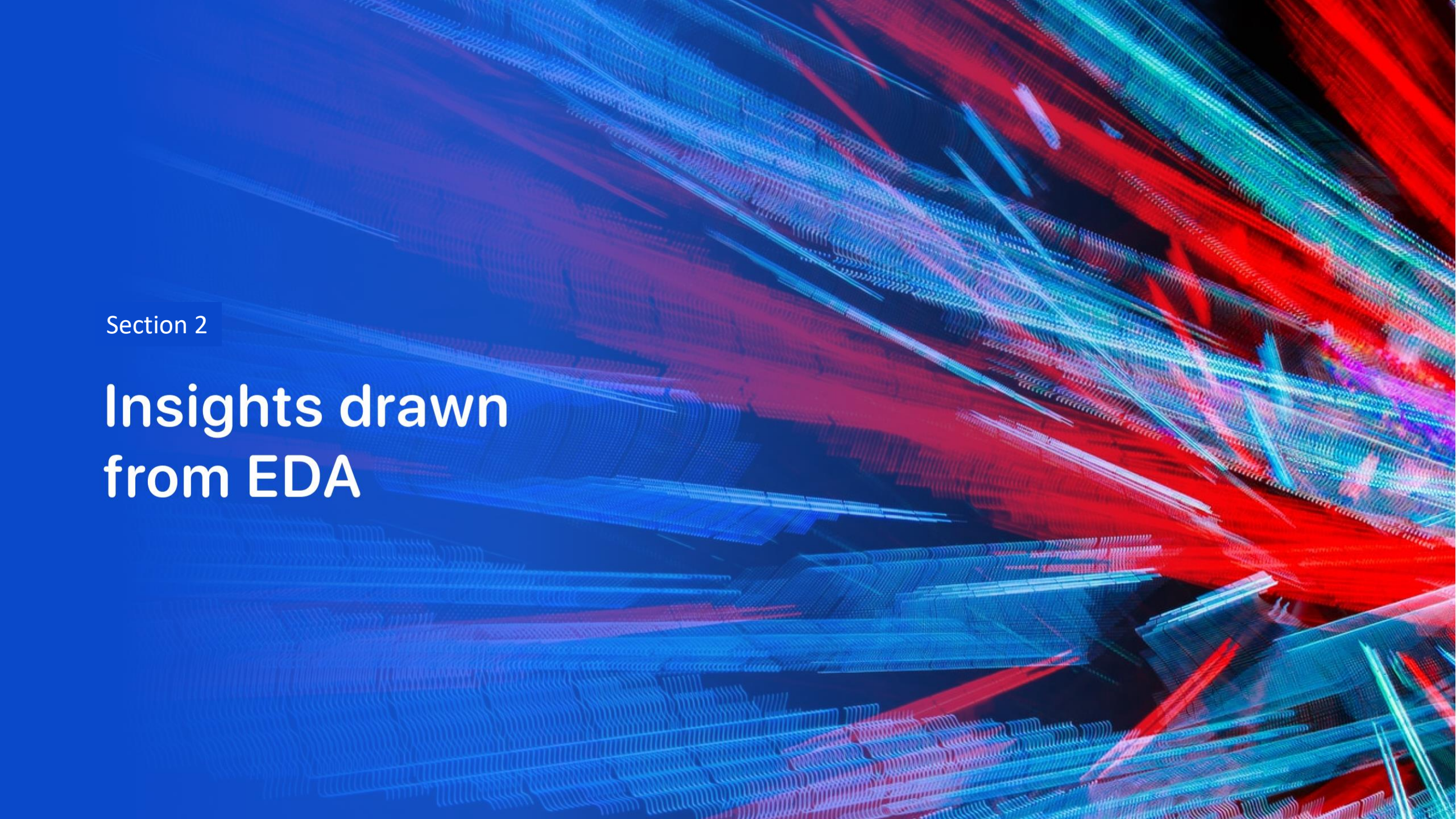
Predictive Analysis (Classification)

- Machine Learning (ML) predictive analysis was conducted utilizing models of: Logistic regression, Support Vector Machine (SVM), Decision trees, K-Nearest Neighbors (KNN). Which were trained on training data, and tested on testing dataset.
- We evaluated the hyperparameters using GridSearchCV function from scikit-learn python library for ML.
- Based on selected best parameters we compared the accuracy of each model deployed on testing data, and examined the calculated confusion matrix.
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose



Results

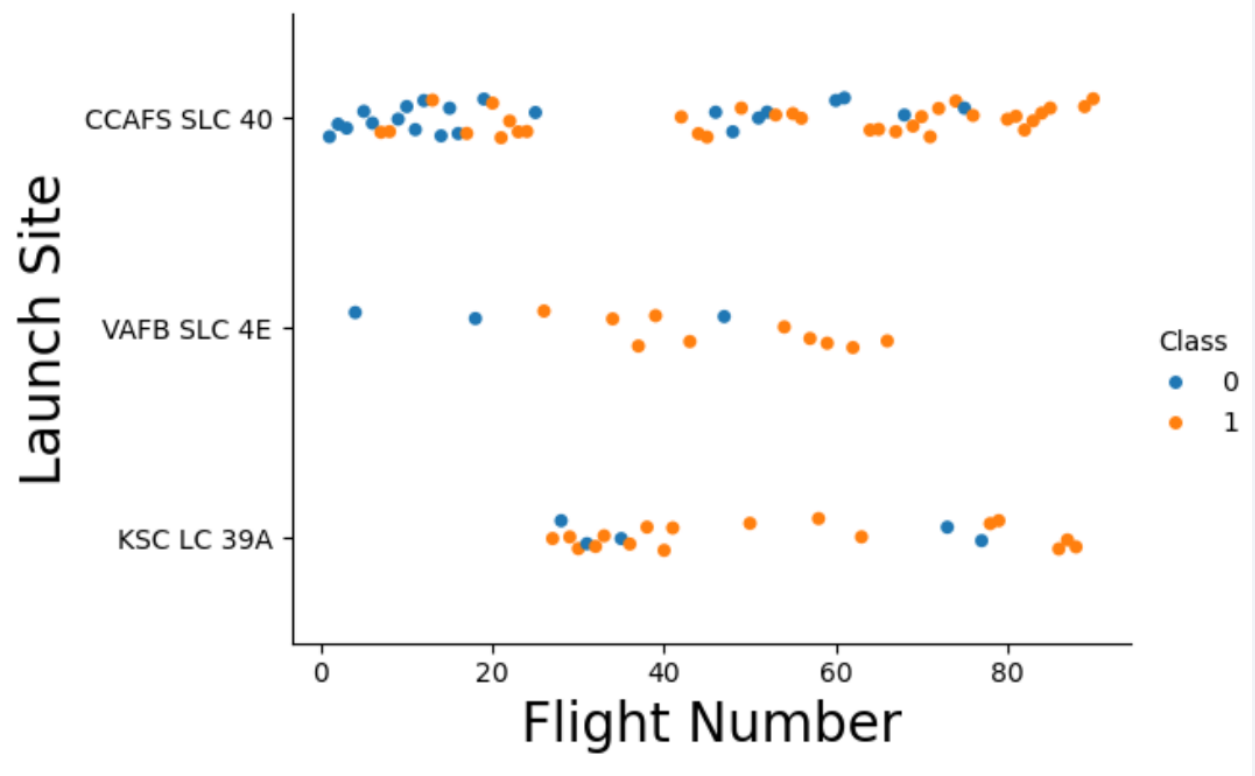
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

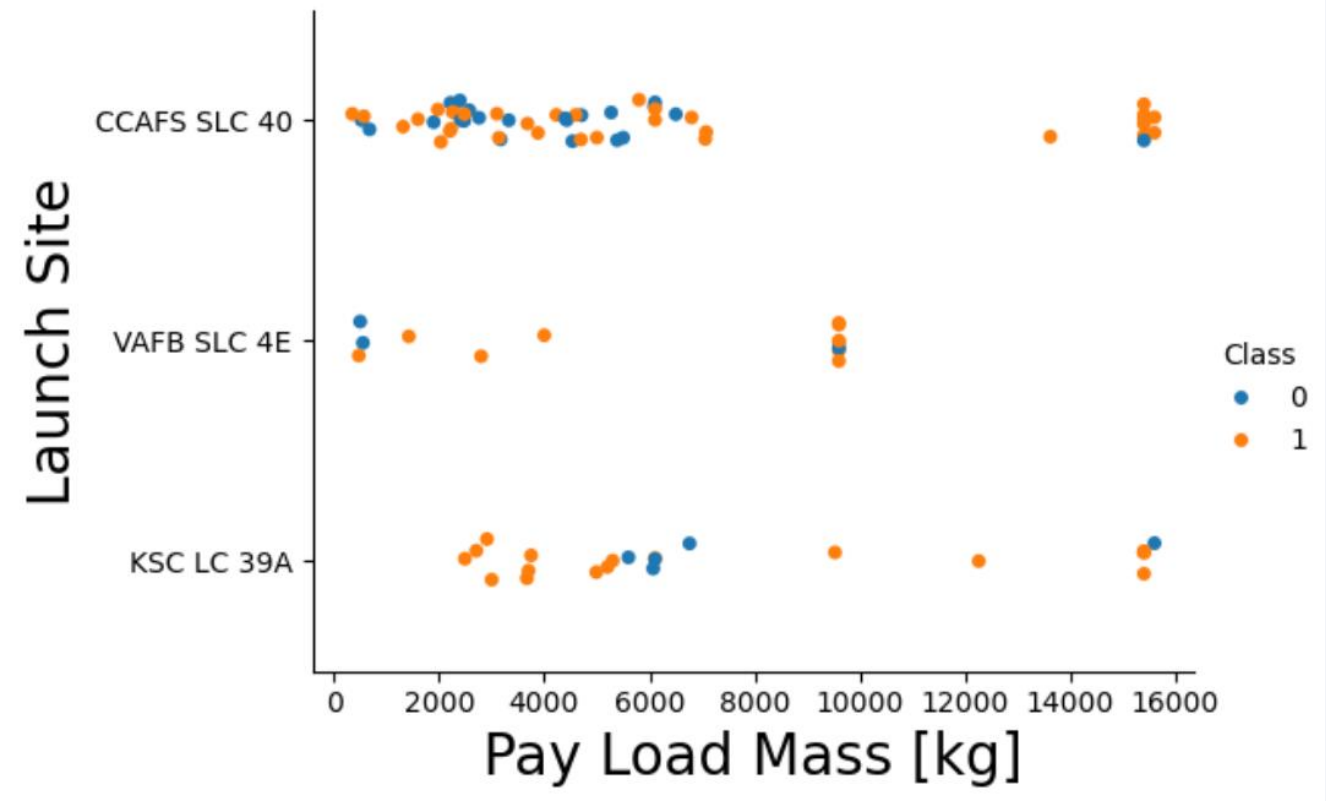
- While class 0 – blue represent failed landing, class 1 – orange stands for successful landing. We can observe how data varies for each Launch Site.

- Although eventually with higher flight number i.e. more launch attempts the successful Falcon 9 first stage landings increases.



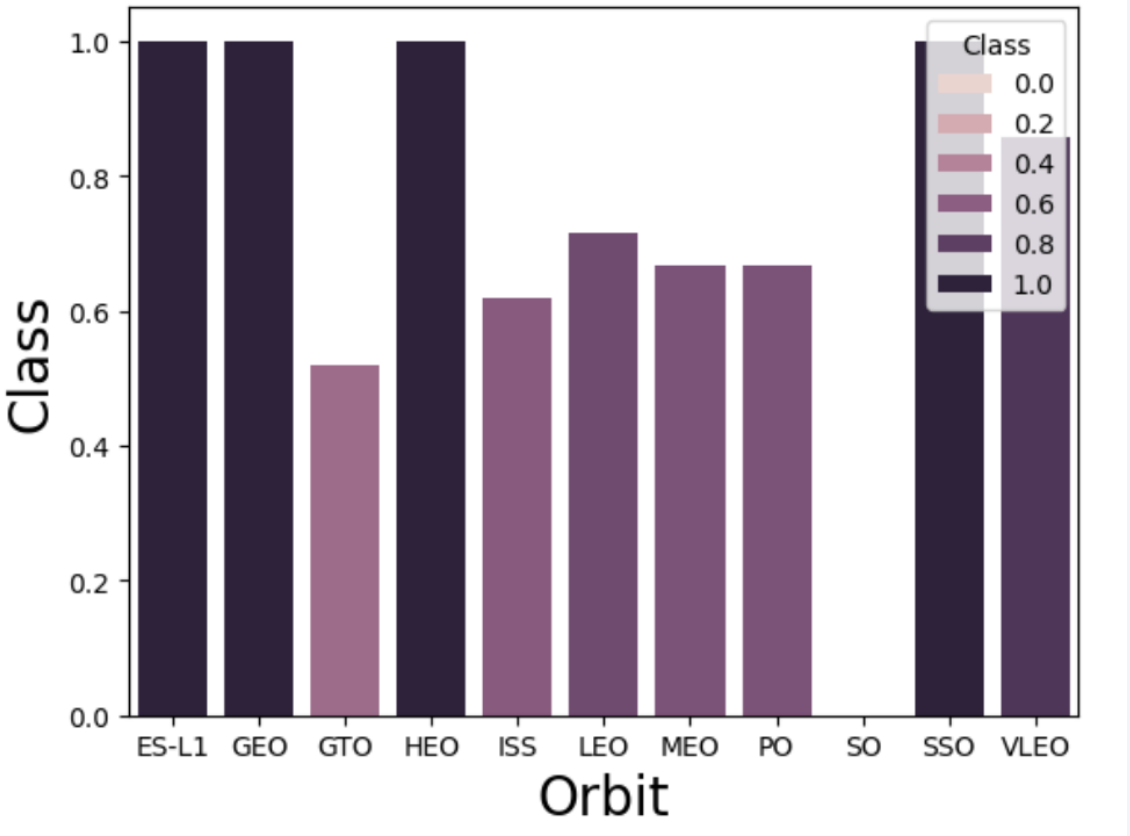
Payload vs. Launch Site

- Generally, we can see higher success in landing with higher payload masses, with some „statistical noise“ data.
- Low payload mass at CCAFS site doesn't seem to be corelated with success of landing.
- For KSC LC we observe some correlation with success results at low payload masses up to 5500 kg, and subsequently and failure area between 5500 to 7000 kg of payload.
- VAFB data are scarce to state any significant conclusions.



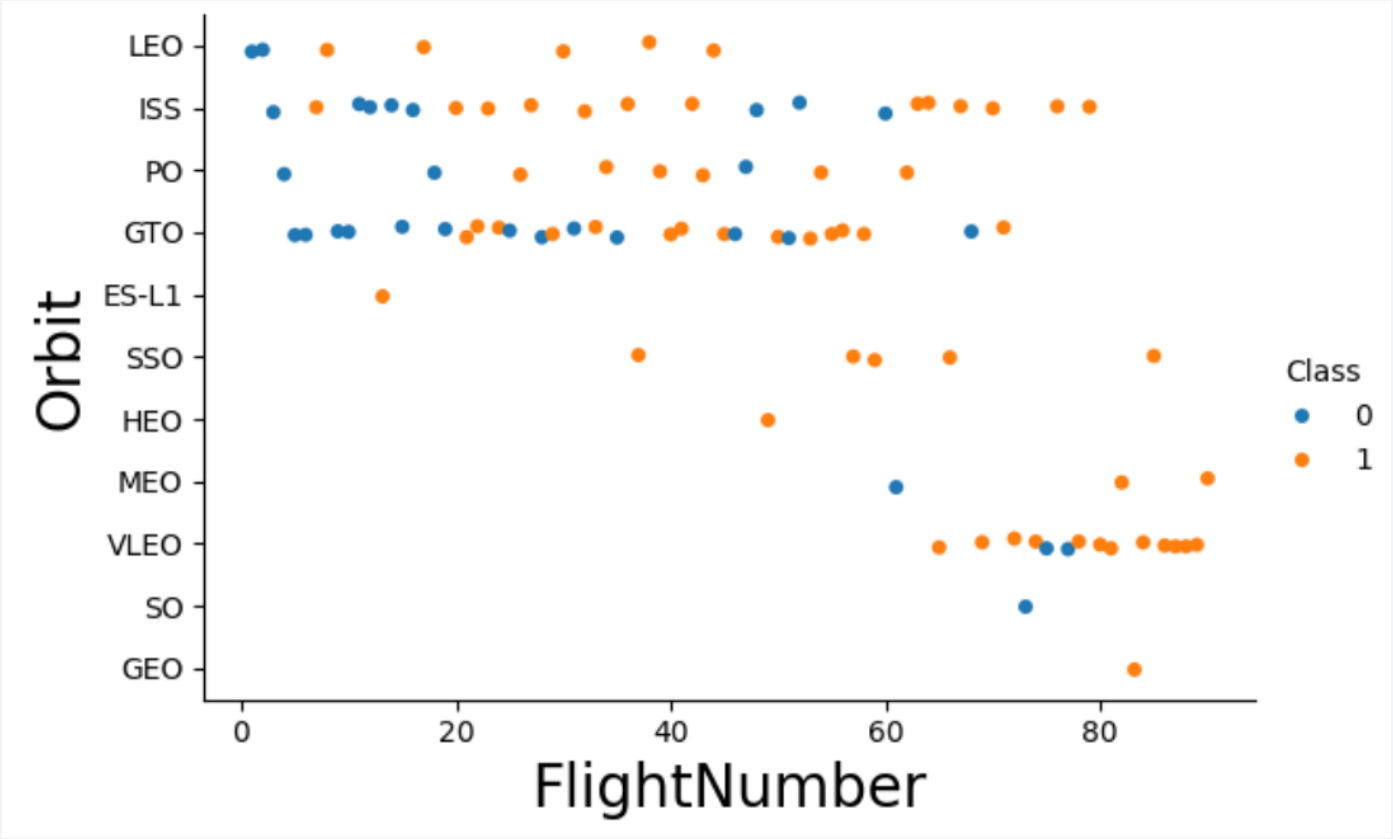
Success Rate vs. Orbit Type

- We can see that Orbits: ES-L1, GEO, HEO and SSO have the 100% success rate.
- Orbit SO, success rate is 0%.
- Moderate success rate we observe for GTO, ISS, LEO, MEO, PO and VLEO Orbits.



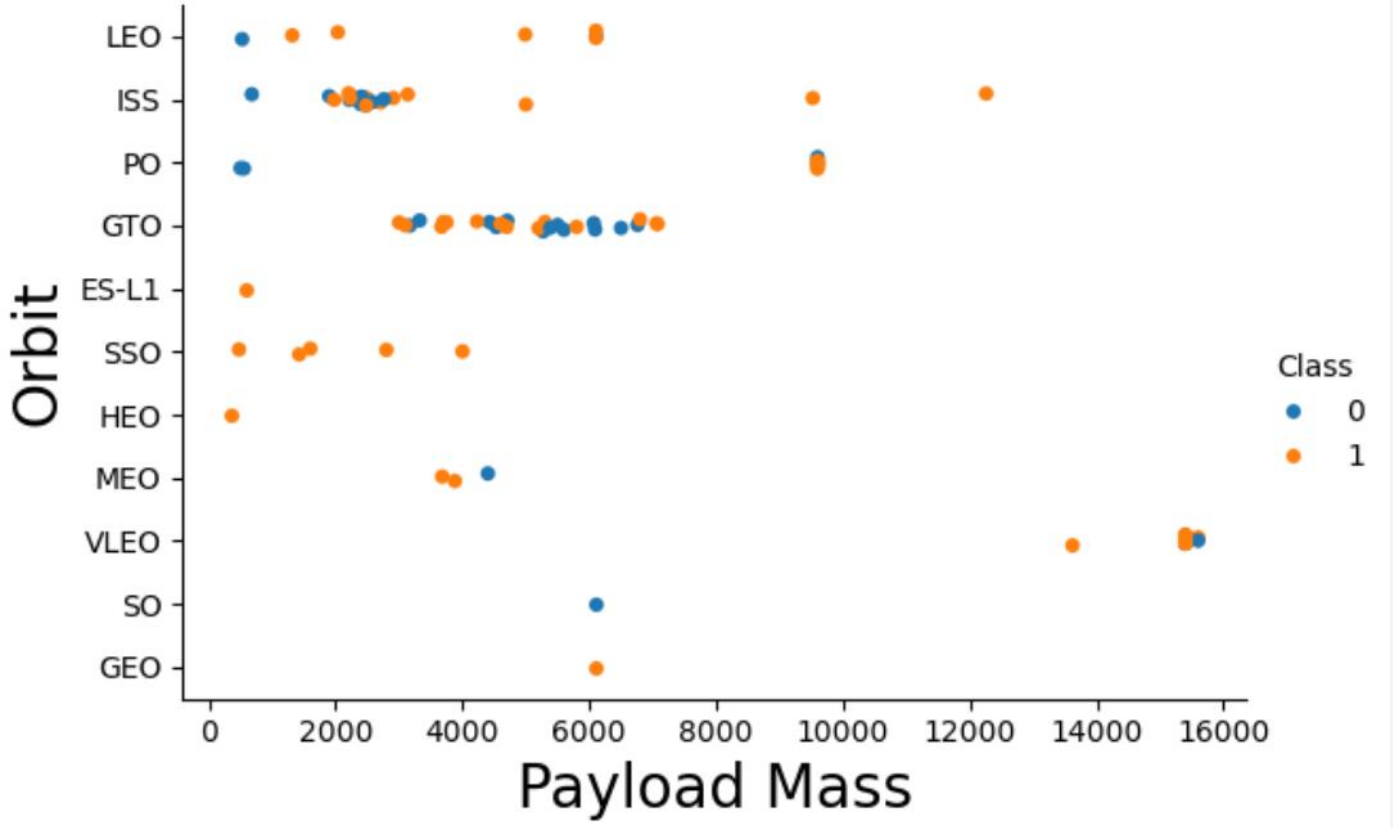
Flight Number vs. Orbit Type

- For LEO Orbit we observe higher success rate with higher Flight Number.
- Also, SSO Orbit appears to be100% successful.
- Examining data of e.g. GTO Orbit we cannot draw any conclusion between Flight Number and Orbit relationship, while the same conclusion we expect if we look at this plot in general.
- Again, generally we can see that to get a space rocket on one of the Earths orbit could be a tricky business. Data collection like this forbid us to make any solid conclusions/predictions. Simply more data is required.



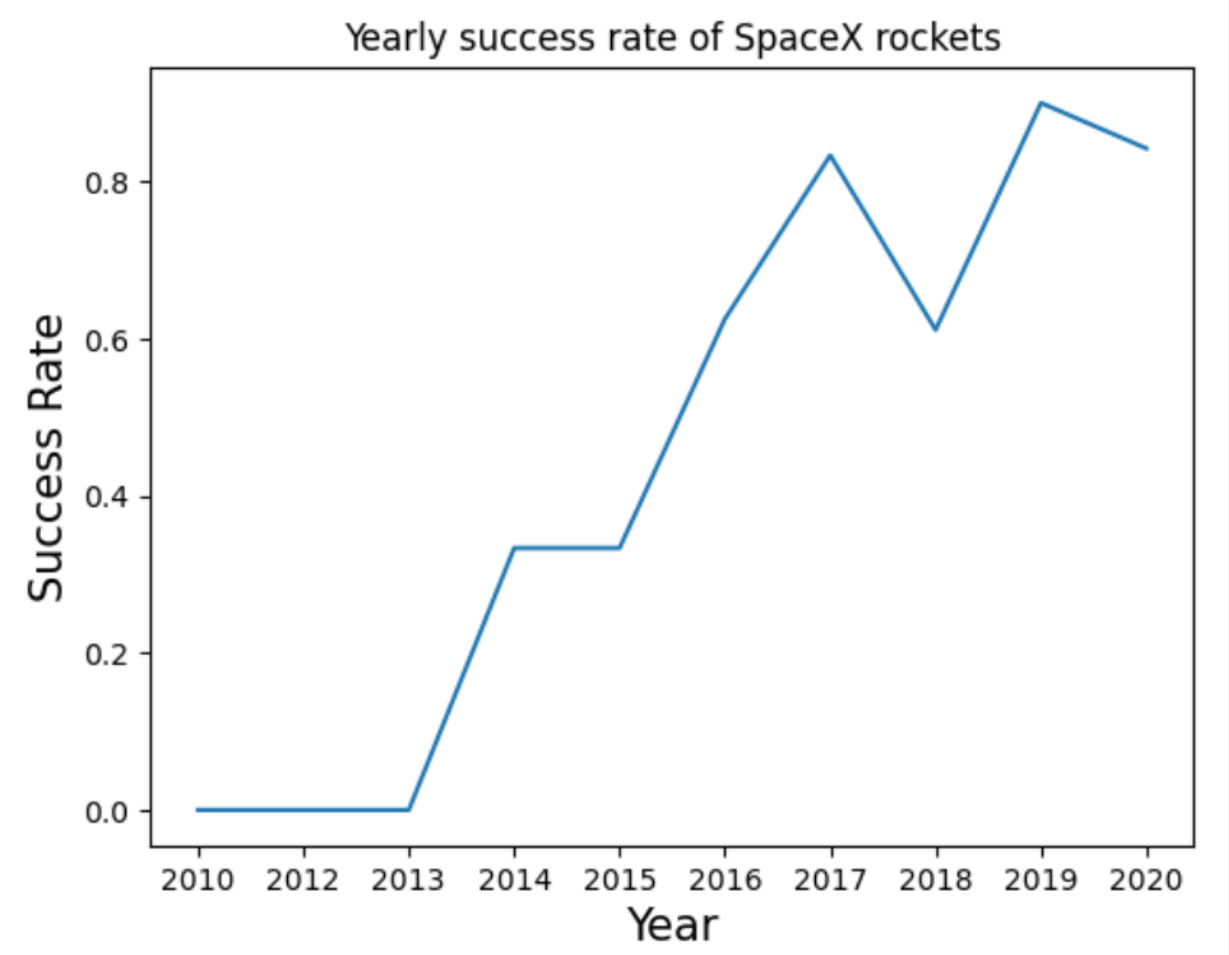
Payload vs. Orbit Type

- Heavier payload masses appears to have higher success rates for LEO, ISS and PO Orbits.
- How unpredictable this results are we can observe on the GTO Orbit, with span from 3000 to 7000 kg payload.
- Again, if this should be some explanatory statistical correlation measure, then God bless this space agency...



Launch Success Yearly Trend

- Apparently, SpaceX started rocket launches in 2013, since then we observe rising trend in success rate of this launches.



All Launch Site Names

- How to display distinct names of Launch sites from some tabulated values, using sql query.

```
[9]: %sql SELECT DISTINCT(LAUNCH_SITE) from SPACEXTBL
 * sqlite:///my_data1.db
Done.
```

[9]:

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

Launch Site Names Begin with 'CCA'

- How to find 5 records where launch sites begin with `CCA` using sql query.

```
[10]: %sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5
 * sqlite:///my_data1.db
Done.
```

[10]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

```
[15]: %sql SELECT sum(PAYLOAD_MASS__KG_) AS payloadmass FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)'
 * sqlite:///my_data1.db
Done.
```

[15]:

payloadmass
45596

- A sql language query to display total payload mass carried by boosters launched by NASA (CRS).

Average Payload Mass by F9 v1.1

```
[17]: %sql SELECT avg(PAYLOAD_MASS__KG_) AS payloadmass FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
 * sqlite:///my_data1.db
Done.
```

[17]:

payloadmass
2928.4

- How to calculate the average payload mass carried by booster version F9 v1.1 using sql query.

First Successful Ground Landing Date

```
[19]: %sql SELECT MIN(Date) AS First_SC_LD_Date FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success (ground pad)'
 * sqlite:///my_data1.db
Done.
```

[19]:

First_SC_LD_Date
2015-12-22

- Here an example how to find the date of the first successful landing outcome on ground pad using sql query.

Successful Drone Ship Landing with Payload between 4000 and 6000

[21]:
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

```
 * sqlite:///my_data1.db
Done.
```

[21]:

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

- Here we have the sql query and results of how to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000, for which we used WHERE function.

Total Number of Successful and Failure Mission Outcomes

```
[22]: %sql SELECT COUNT(Mission_Outcome) AS Succesful FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Success%'
```

```
 * sqlite:///my_data1.db
Done.
```

[22]:

Succesful
100

```
[23]: %sql SELECT COUNT(Mission_Outcome) AS Failure FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Failure%'
```

```
 * sqlite:///my_data1.db
Done.
```

[23]:

Failure
1

- Here is presented how to calculate the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

```
[25]: %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) ORDER BY Booster_Version
 * sqlite:///my_data1.db
Done.
```

[25]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

- Here we have name list of the boosters which have carried the maximum payload mass, alphabetically ordered.

2015 Launch Records

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[29]:
```
%sql SELECT substr(Date, 6,2) AS Month, Launch_Site, Booster_Version, Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome = 'Failure (drone ship)' AND substr(Date,0,5)='2015'
```

```
 * sqlite:///my_data1.db
Done.
```

[29]:

Month	Launch_Site	Booster_Version	Landing_Outcome
01	CCAFS LC-40	F9 v1.1 B1012	Failure (drone ship)
04	CCAFS LC-40	F9 v1.1 B1015	Failure (drone ship)

- In above presented code we get the list the failed landing_outcomes on drone ship, their booster versions, and launch site names for in year 2015. Please note the SQLLite note if you are using it to get proper month and date.

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

```
[33]: %sql SELECT [Landing_Outcome], COUNT(*) AS Number FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY [Landing_Outcome] ORDER BY Number DESC
 * sqlite:///my_data1.db
Done.
```

[33]:

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

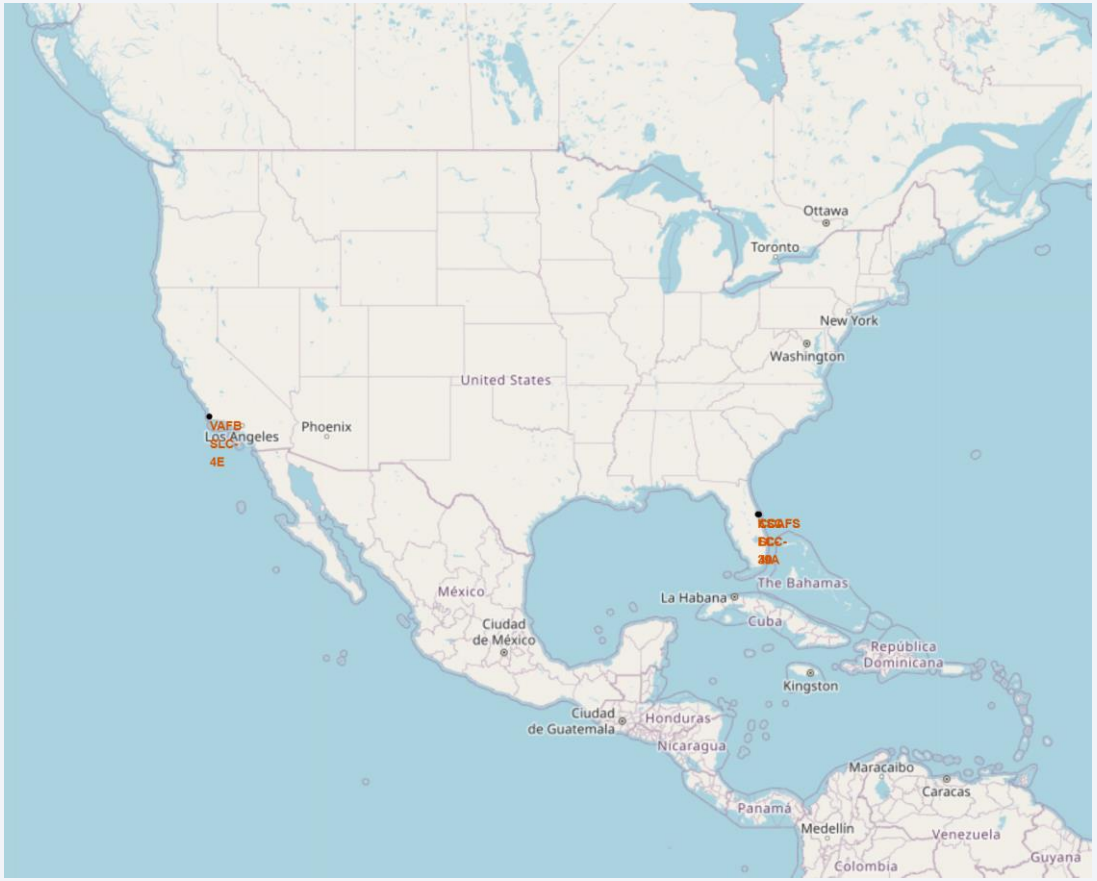
- Above presented sql code is ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

Section 3

Launch Sites Proximities Analysis

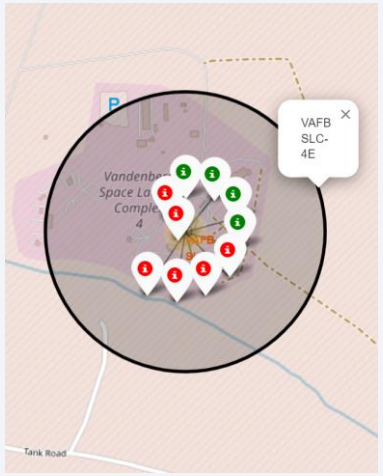
Falcon 9 launch site locations on Folium world map

- All the Space X launch sites are located at the coastal line of USA.
- The VAFB SLC-4E is in California, close to Los Angeles.
- The other three: KSC LC-39A, CCAFS LC-40 and CCAFS SLC-40 are in Florida.
- The reasons for this locations are: at first to use higher radial velocity of Earth's rotation which is highest closer to the equator. Secondly to prevent damage in case of rocket explosion, so the debris can fell into the ocean.

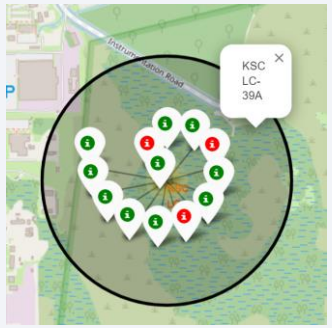


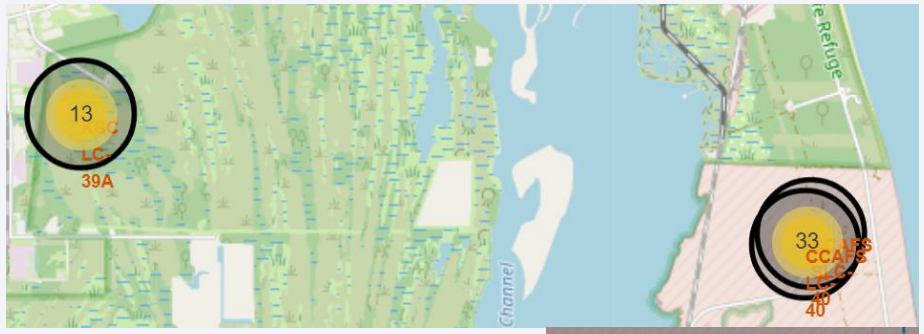
Color labeled marks showing successful/failure launches

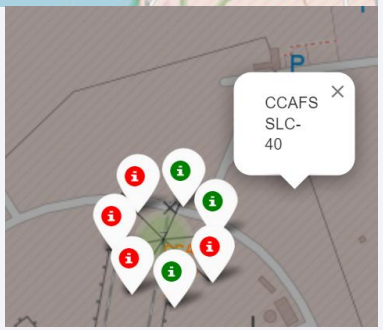
California



Florida



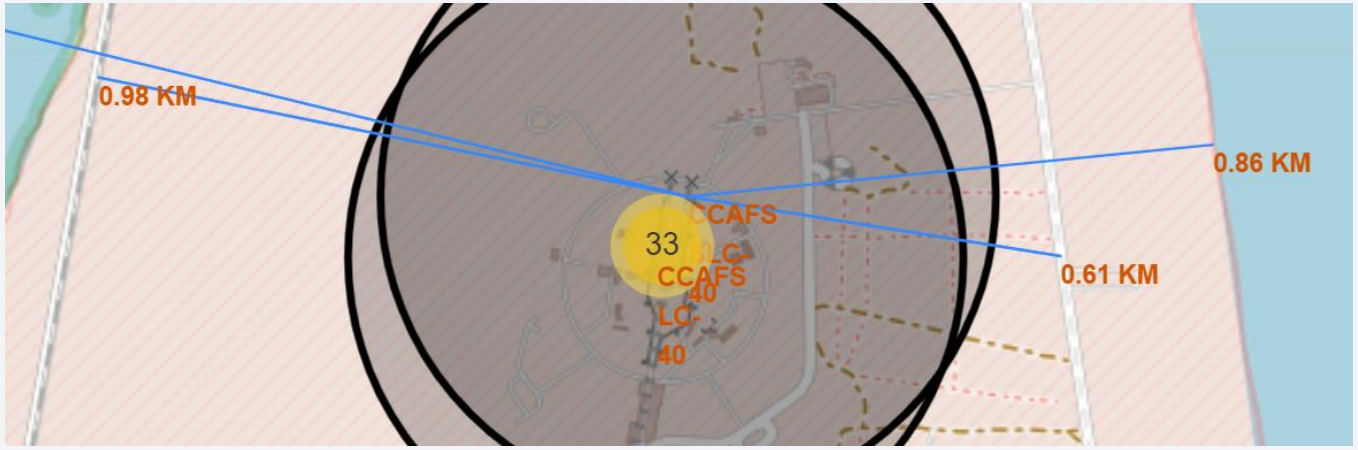


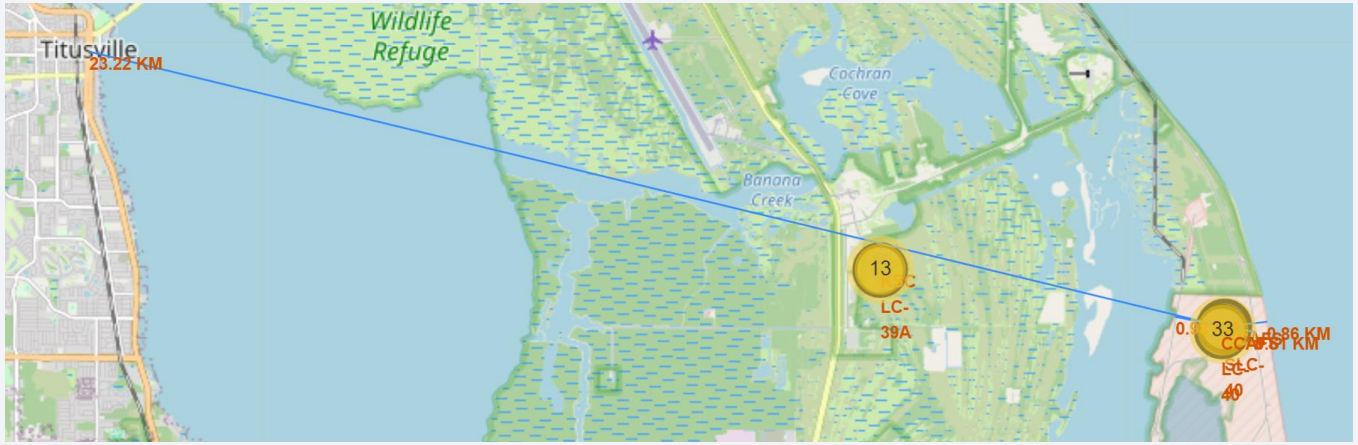


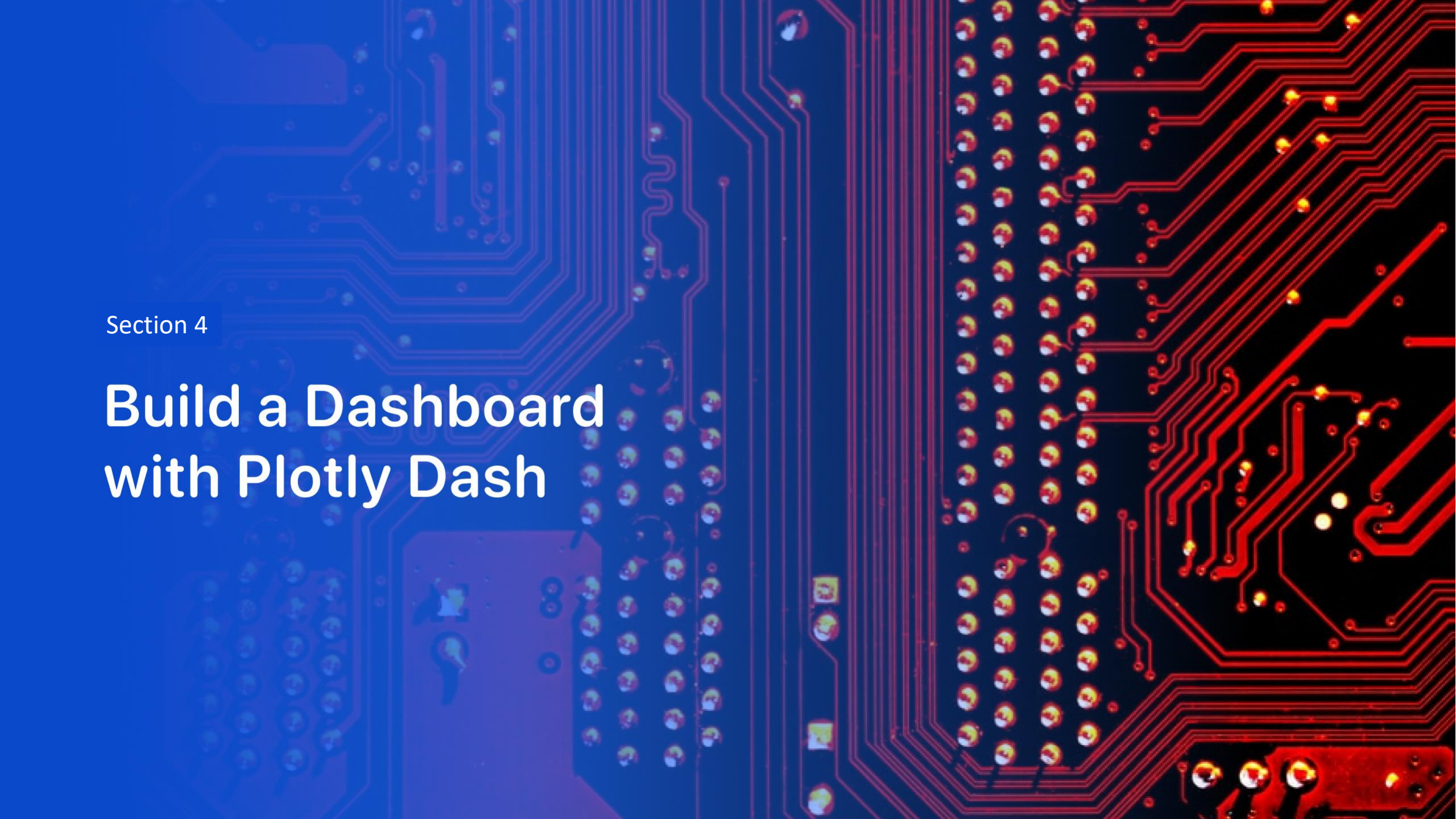
- Successful launches are marked with green color
- Failures are marked with red

Distances from CCAFS SLC-40 launch site to its proximities

- The top figure is showing distances from CCAFS SLC-40 launch site to the coastline (0.86 km), to nearest railway (0.98 km) and nearest highway (0.61 km).
- Bottom figure show distance to nearest city Titusville which is about 23 km.
- Railways and highways are relatively not so far, as one need to transport various construction parts, fuels, payloads etc. to the rockets.
- Distance to the city is significant, compared to the coastal line.

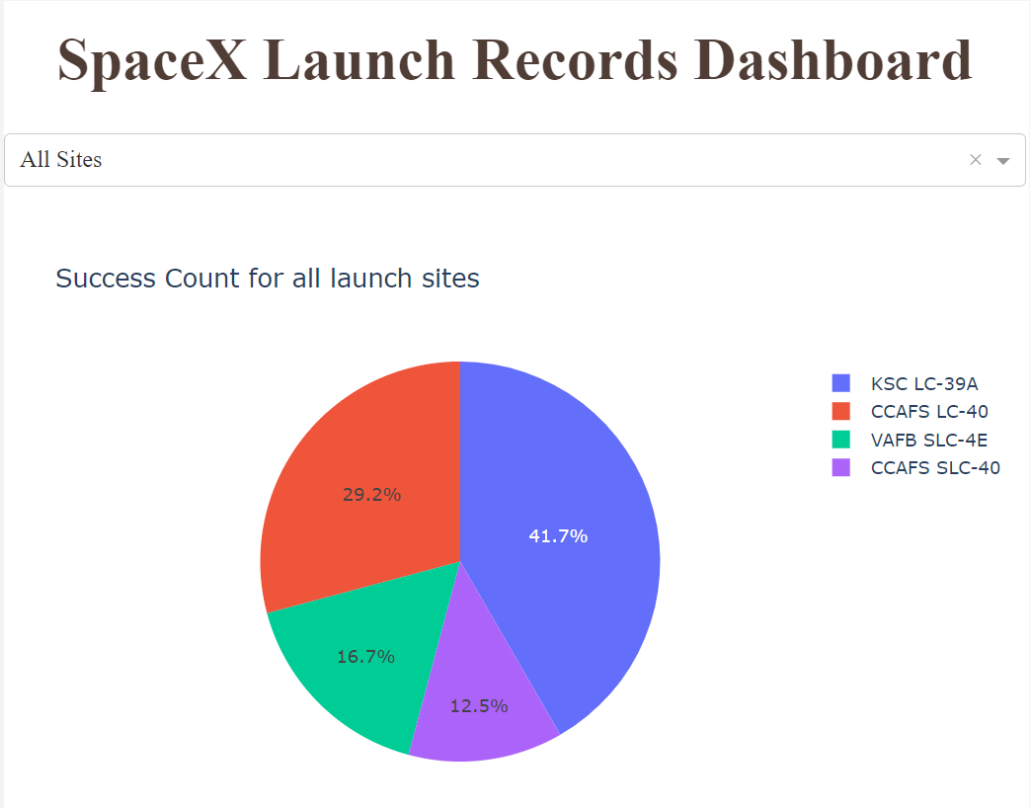





Section 4
Build a Dashboard with Plotly Dash

Pie chart of launch success count for all sites

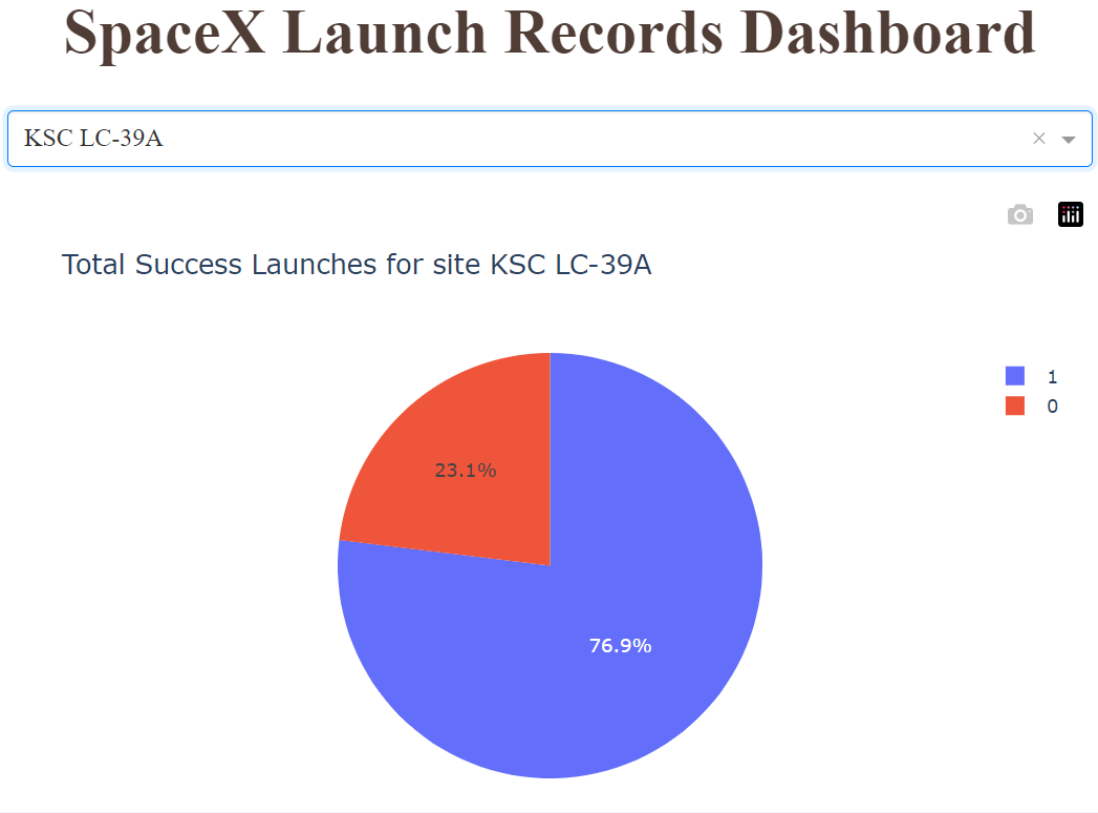
- From the dropdown menu we can select each or all landings sites disclosed with the colored legend next to the pie chart.
- Selecting all sites we can compare the percentage outcome of successful launch between all sites, please note here we are comparing only the successful counts relative to each site.
- As we can observe on the pie chart, highest count of successful launches happened at the KSC LC-39A launch site, with relative count of 41.7%. We will have a closer look at this statistics on the next slide.



Pie chart of launch site with highest launch success ratio

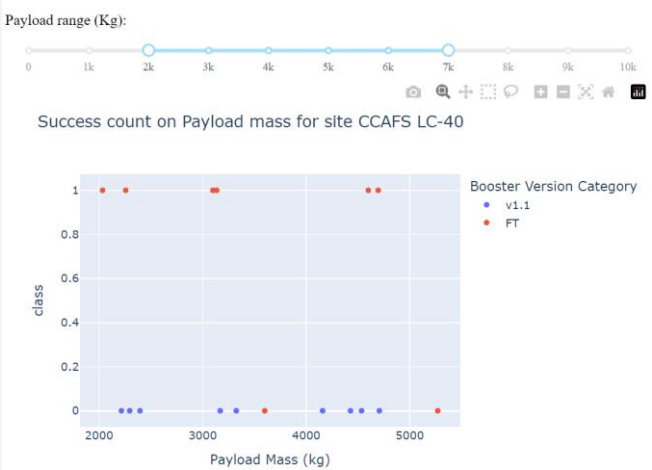
- On right we can see pie chart visualization of successful (legend = 1 blue color) vs failed (legend = 0 red color) launches for the SpaceX Falcon 9 rocket at the Kennedy Space Center Launch Complex 39A (KSC LC-39A).

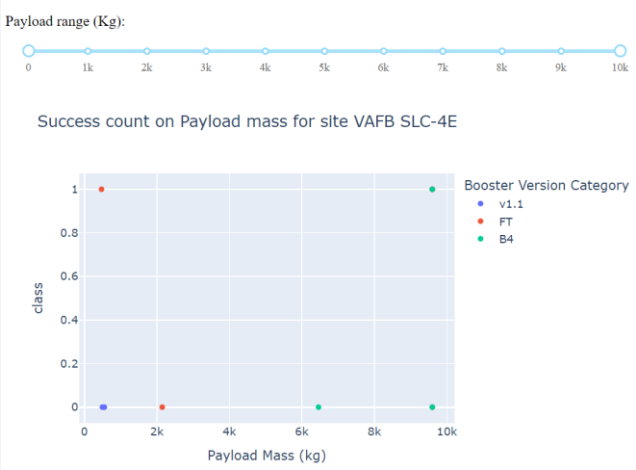
- KSC LC-39A has the highest total successful launches of 76.9% from all the analyzed launch sites.

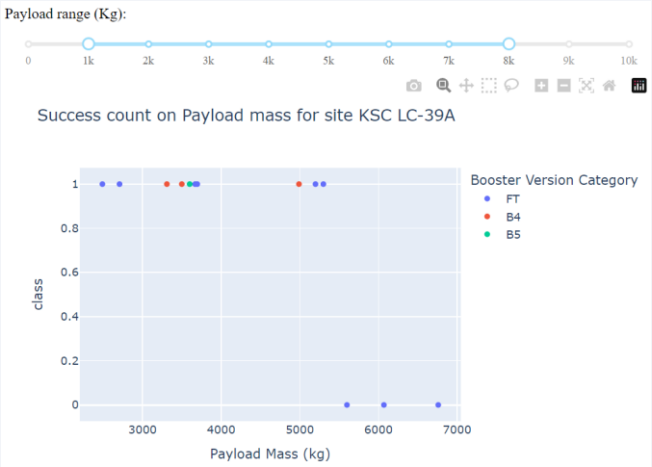


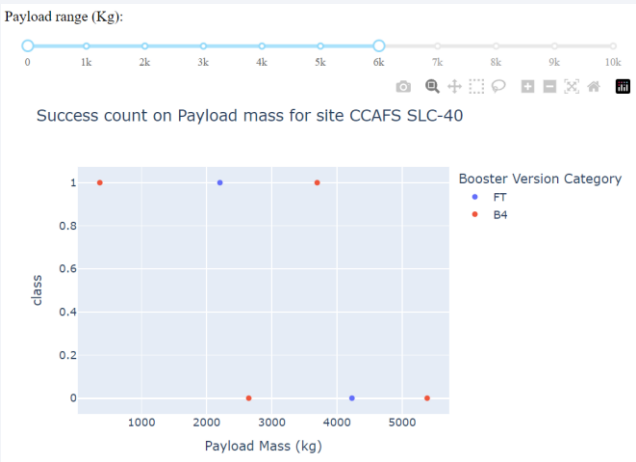
Payload Mass vs Launch outcome comparison

- On the right side we can see four categorical scatter plots visualizations of each landing site showing the success rate of launch with selected payload mass. The payload mass of each site is displayed only in significant ranges of payload mass for such launch site, which was adjusted with the range slide above.
- As we can see, for the bottom left site KSC LC-39A with most successful launches, the rocket launches were quite successful for the payloads up to 5500 kg.
- To draw any further opinion on success of the other launch sites or the booster version categories, is simply too difficult with provided data.









Section 5

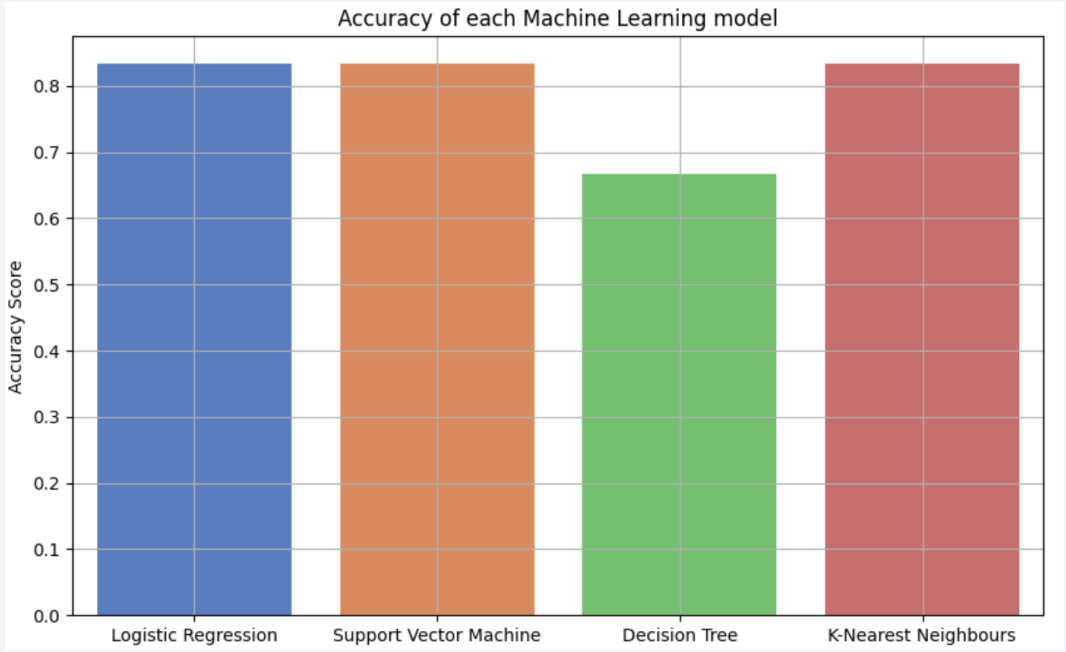
Predictive Analysis (Classification)

Classification Accuracy

- Displaying the classification accuracy of each ML model we can state that Logistic Regression, Support Vector Machine and K-Nearest Neighbors models are performing equally well, with accuracy score of 0.83. Slightly behind is the Decision Tree ML model with accuracy of 0.66. All four models are compared in the bar plot on right.

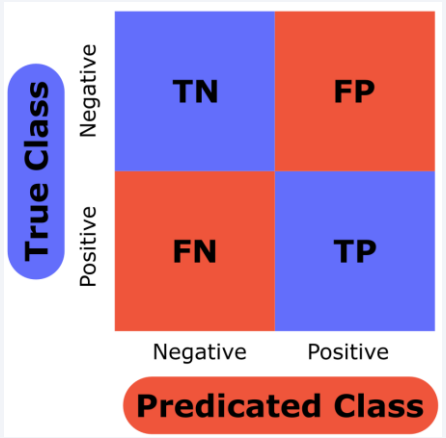
- However, this is only the interpretation of a one single run of models, with more runs we observe different results for the Decision Tree model, as some implicating randomness. Sometimes the Decision tree accuracy is below other three models, sometimes above, and sometime equal to 0.83. Anyway, from stability point of view, is better to use Logistic Regression, Support Vector Machine and K-Nearest Neighbors models.

	Algorithm	Accuracy Score	Best Score
0	Logistic Regression	0.833333	0.846429
1	Support Vector Machine	0.833333	0.848214
2	Decision Tree	0.666667	0.860714
3	K-Nearest Neighbours	0.833333	0.848214



Confusion Matrix

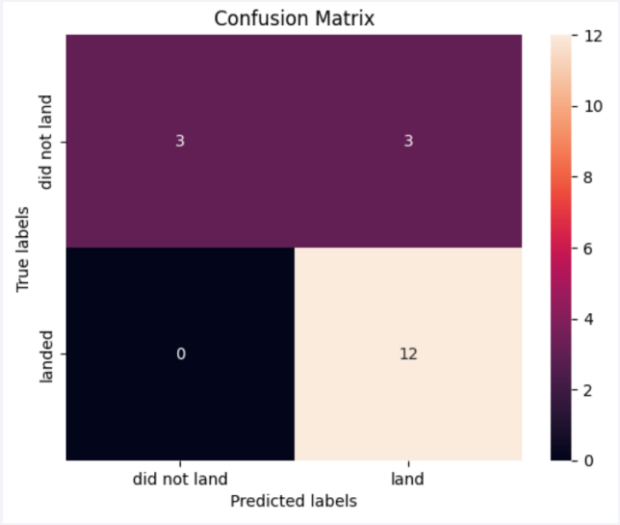
- The confusion matrices for these models could be read as following chart:



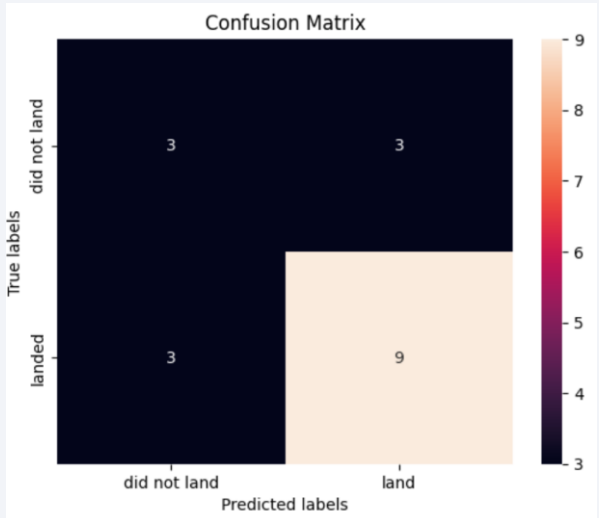
- From results shown on the left we have the same confusion matrix, i.e. results for Lin Reg, SVM and KNN models, with 12 True Positive values, the main problem still remain the False Positive class – unsuccessful landing marked as successful.

- Looking at confusion matrix of Decision Trees ML model, we observe also 3 False Negative values, which handicaps this model to the favor of other three ML models.

Lin Reg, SVM, KNN



Decision Trees



Conclusions

- The more rockets you send up, the more of them will successfully launch and eventually land back on Earth, which is not a surprise, as you can learn from your previous mistakes. Definitely the strategy used in such a rich company as SpaceX.
- In general, from examined datasets, rockets with lower payload are showing better results of successful launches than rockets with higher payload masses.
- Most successful launches happened at the KSC LC-39A launch site with payload mass up to 5500 kg. This I found as the most interesting result from all analysis, perhaps some very responsible team was working at that launch base...
- Orbits with 100% success rate are ES-L1, HEO, SSO and GEO, although with such limited data this is just mere constatation.
- ML models can predict – in theory – a successful landing, however examined datasets were quite small to trust these predictions. On the other hand, as a landing of a space rocket is incredibly complex process depended on many variables, I hope that such predictive ML analysis was meant only for learning purposes.

Appendix

Git hub URLs to jupyter notebooks and dash.py project

- Data Collection:
  https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb
- Web Scraping:
  https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/jupyter-labs-webscraping.ipynb
- Data Wrangling:
  https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb
- EDA with Data Visualization:
  https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/edadataviz.ipynb
- EDA with SQL:
  https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb
- Folium Map:
  https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/lab_jupyter_launch_site_location.ipynb
- Plotly Dash:
  https://github.com/MilosFejercak/Coursera-IBM-DSP-Final-Project/blob/main/spacex_dash_app.py

Thank you!