IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Zsolt Miklos Takacs
29/09/2022

Outline

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

- GitHub: https://github.com/jonjehu/Zsolt-Takacs-Coursera/tree/week-5

Executive Summary

- Our task is to predict successful missions using data from completed Space X missions, such as payload mass, launch site, landing site.
- I used the following methodologies to analyze data:
  - The Data Collection with Web Scraping, Data Wrangling.
  - Extraction and Transformation with Exploratory Data Analysis (EDA) and with SQL.
  - The Visualization of EDA with using Pandas and Matplotlib library.
  - Interactive Visual Analytics with Folium and Dashboards with Ploty Dash.
  - Predictive Analysis with the Machine Learning classification models: Logistics Regression, Support Vector Machine, Decision tree, K nearest neighbors method.
- Most of the data was collected from public sources. The EDA has defined the usable features for the predict. Finally the machine learning determined the most effective models for a successful mission.

Introduction

- *In this capstone, we will predict if the Falcon 9 first stage will land successfully. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.*

- *Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.*

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected from https://api.spacexdata.com/v4/ Space X API and https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches web scraping.
- Perform data wrangling
  - Summarized and analyzed the values and determined the Outcome Labels.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash

Methodology

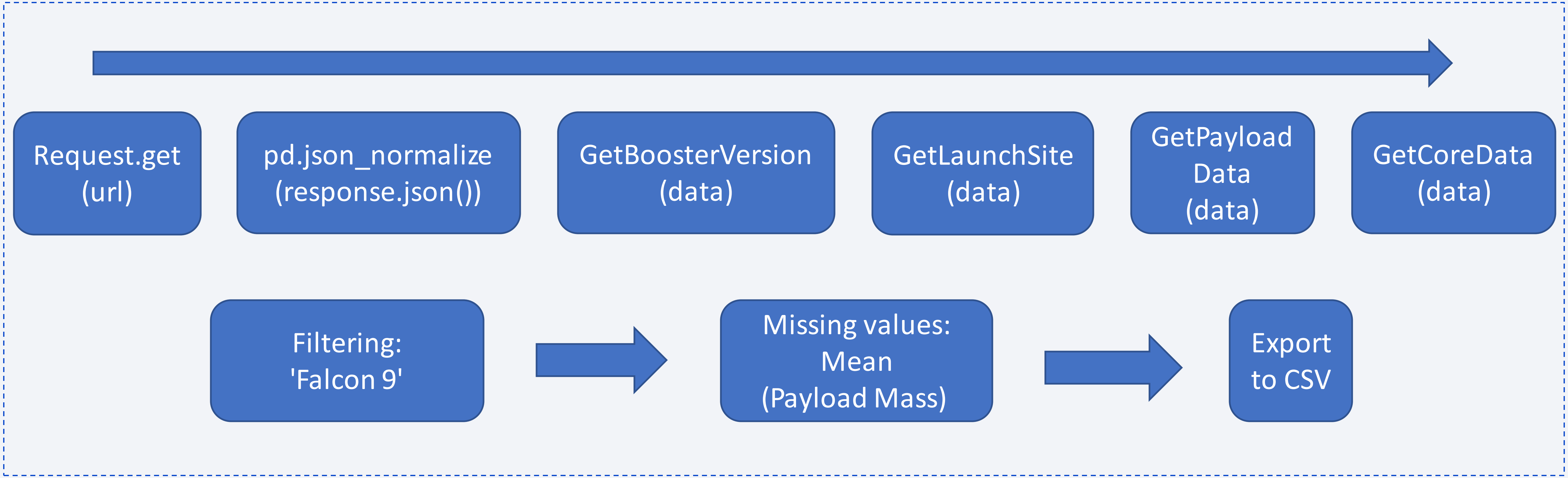
Executive Summary

- Perform predictive analysis using classification models
  - Performed exploratory Data Analysis and determined Training Labels
    - Created a column for the class
    - Standardized the data
    - Split into training data and test data
    - Best Hyperparameter for SVM, Classification Trees and Logistic Regression
    - The method performs best using test data

Data Collection

- Requested to the SpaceX API from:
  - Rocket: https://api.spacexdata.com/v4/rockets/
  - Launchpad: https://api.spacexdata.com/v4/launchpads/
  - Payload: https://api.spacexdata.com/v4/payloads/
  - Cores: https://api.spacexdata.com/v4/cores/
  - Rocket launch data: https://api.spacexdata.com/v4/launches/past
- Filtered the dataframe to only include Falcon 9 launches
- Missing values were replaced by mean value of Payload Mass
- Dataset was exported to CSV file

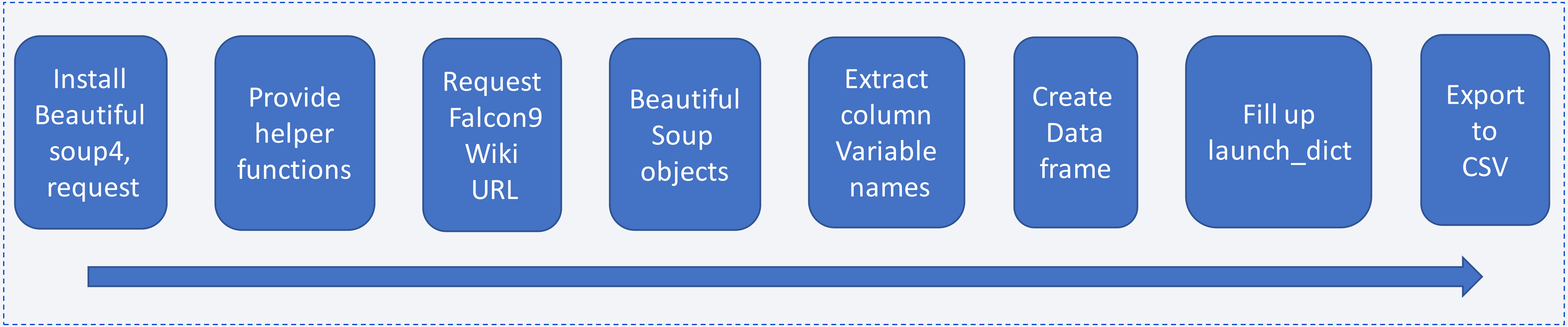
Data Collection – SpaceX API flowchart



- https://github.com/jonjehu/Zsolt-Takacs-Coursera/tree/week-1

Data Collection - Scraping

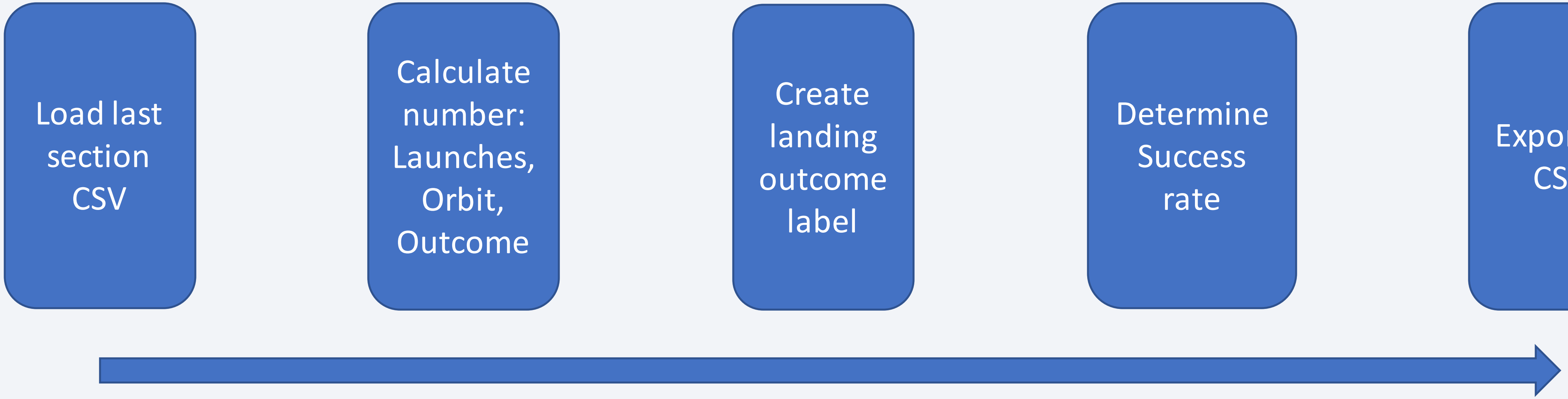
- Scraped the data from: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



- GitHub URL: https://github.com/jonjehu/Zsolt-Takacs-Coursera/blob/week-1/Complete%20the%20Data%20Collection%20with%20Web%20Scraping%20lab.ipynb

Data Wrangling

- Performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.



- GitHub URL: https://github.com/jonjehu/Zsolt-Takacs-Coursera/blob/week-1/Complete%20the%20EDA%20lab%20Lab%202%20Data%20wrangling.ipynb

EDA with SQL

SQL queries to solve the assignment tasks:

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first succesful landing outcome in ground pad was acheived.
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
9. List the records which will display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015.
10. Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

- GitHub URL: https://github.com/jonjehu/Zsolt-Takacs-Coursera/blob/week-2/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with Data Visualization

- Visualized with scatter point chart the relationship between:
  - Flight Number and Launch Site
  - Payload and Launch Site
  - FlightNumber and Orbit type
  - Payload and Orbit type
- Visualized the relationship between success rate of each orbit type with bar chart
- Visualized the launch success yearly trend with a line chart

- GitHub URL: https://github.com/jonjehu/Zsolt-Takacs-Coursera/blob/week-2/jupyter-labs-eda-dataviz_finished.ipynb

Build an Interactive Map with Folium

- Marked all launch sites on a map with folium.Circle and and folium.Marker to add a highlighted circle area with a text label and marker.
- Marked the success/failed launches for each site on the map as added a folium.Marker to marker_cluster.
- Calculated the distances by drawn a line between a launch site to its closest city, railway, highway.
- After plotted distance lines to the proximities, I could answer a few questions easily about location of launch sites.

- GitHub URL: https://github.com/jonjehu/Zsolt-Takacs-Coursera/blob/week-3/lab_jupyter_launch_site_location-Finished.ipynb

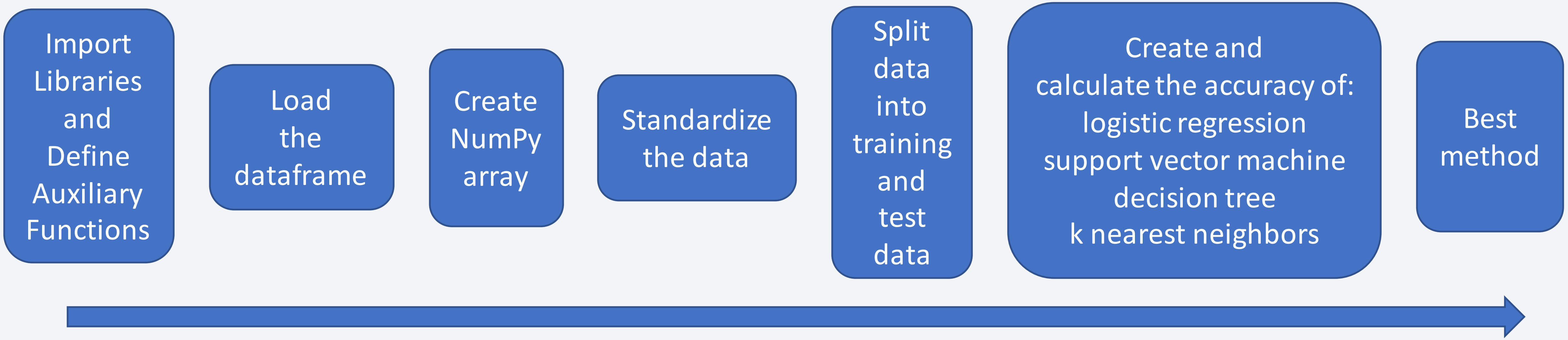
Build a Dashboard with Plotly Dash

- Added to a dashboard:
  - Launch Site Drop-down Input Component
  - Callback function to render success-pie-chart based on selected site dropdown
  - Range Slider to Select Payload
  - Callback function to render the success-payload-scatter-chart scatter plot

- GitHub URL: https://github.com/jonjehu/Zsolt-Takacs-Coursera/blob/week-3/Zsolt%20Takacs%20-%20lab_theia_plotly_dash%20finished.ipynb

Predictive Analysis (Classification)

- Performed exploratory Data Analysis and determine Training Labels
  - create a column for the class
  - Standardize the data
- Split into training data and test data to find best Hyperparameter for SVM, Classification Trees Logistic Regression and K nearest neighbors object.
- The best method is Decision tree method with 94% accuracy.



- GitHub URL: https://github.com/jonjehu/Zsolt-Takacs-Coursera/blob/week-4/SpaceX_Machine%20Learning%20Prediction_Part_5_finished.ipynb

Exploratory data analysis results 1

- the number of launches on each site:
  - CCAFS SLC 40: 55, KSC LC 39A: 22, VAFB SLC 4E: 13
- the number and occurrence of each orbit:
  - GTO: 27, ISS: 21, VLEO: 14, PO: 9, LEO: 7, SSO: 5, MEO: 3, ES-L1: 1, HEO: 1, SO: 1, GEO: 1
- the number and occurence of mission outcome per orbit type:
  - True ASDS: 41, None None: 19, True RTLS: 14, False ASDS: 6, True Ocean: 5, False Ocean: 2, None ASDS: 2, False RTLS: 1
- the names of the unique launch sites in the space mission:

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

Exploratory data analysis results 2

- the total payload mass carried by boosters launched by NASA (CRS):
  - NASA_CRS_total_payload_mass: 45596
- average payload mass carried by booster version F9 v1.1
  - average_payload_mass_F9_v1_1: 2928.4
- the date when the first succesful landing outcome in ground pad was achieved:

first_succes_landing
22-12-2015

- the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - bvF9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2

Exploratory data analysis results 3

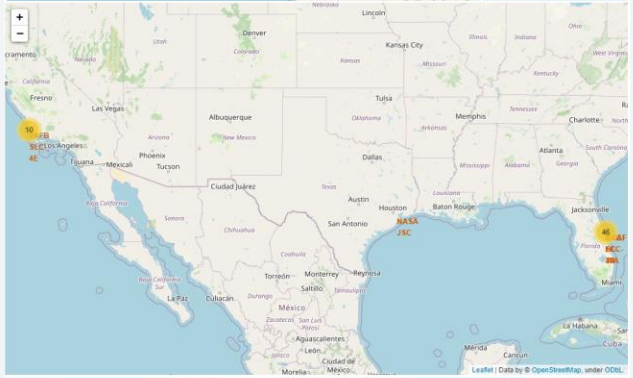
- the total number of successful and failure mission outcomes:
  - Mission_Outcome Count_MO
  - Failure (in flight): 1
  - Success (payload status unclear): 1
  - Success: 99
- the names of the booster_versions which have carried the maximum payload mass
  - 15600 kg F9 B5 B1048.41, F9 B5 B1048.5, F9 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
- the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - Months: 01 Failure (drone ship) F9 v1.1 B1012CCAFS LC-40 2015
  - Months: 04 Failure (drone ship) F9 v1.1 B1015CCAFS LC-40 2015

Exploratory data analysis results 4

- the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order:
Landing_Outcome	Success_Landing_outcomes_between_04062010_20032017
Success (drone ship)	5
Success (ground pad)	3

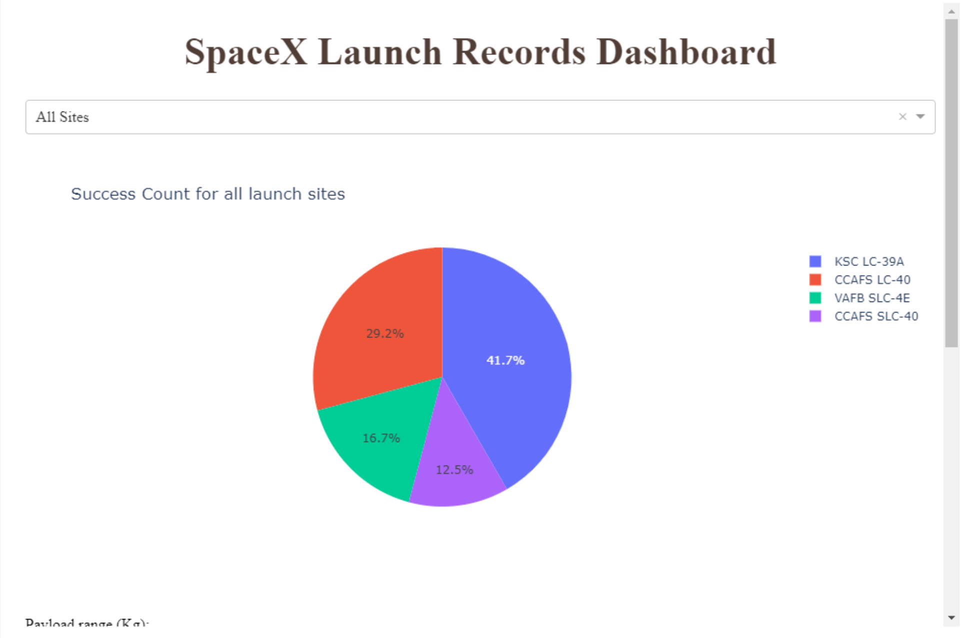
Interactive analytics demo in screenshots results 1

- all launch sites on a map:
  - Launch Site Lat Long
  - 0 CCAFS LC-40 28.562302 -80.577356
  - 1 CCAFS SLC-40 28.563197 -80.576820
  - 2 KSC LC-39A 28.573255 -80.646895
  - 3 VAFB SLC-4E 34.632834 -120.610745
- the success/failed launches for each site on the map
- the distances between a launch site to its proximities:
  - distance_railroad = 1.2860143661039483 km
    distance_highway = 0.5868025820947117 km
    distance_city = 51.21971215802352 km



Interactive analytics demo in screenshots results 2

- SpaceX Launch Records Dashboard by plotly_dash:



Predictive analysis results 1

- NumPy array from the column Class in data:
  - array([0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
  - 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  - 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
  - 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  - 1, 1], dtype=int64)
- Standardize the data in X:

```
array([[-1.71291154e+00, -1.94814463e-16, -6.53912840e-01,
        -1.57589457e+00, -9.73440458e-01, -1.05999788e-01,
        -1.05999788e-01, -6.54653671e-01, -1.05999788e-01,
        -5.51677284e-01,  3.44342023e+00, -1.85695338e-01,
        -3.33333333e-01, -1.05999788e-01, -2.42535625e-01,
        -4.29197538e-01,  7.97724035e-01, -5.68796459e-01,
        -4.10890702e-01, -4.10890702e-01, -1.50755672e-01,
        -7.97724035e-01, -1.50755672e-01, -3.92232270e-01,
         9.43398113e+00, -1.05999788e-01, -1.05999788e-01,
        -1.05999788e-01, -1.05999788e-01, -1.05999788e-01,
        -1.05999788e-01, -1.05999788e-01, -1.05999788e-01,
        -1.05999788e-01, -1.05999788e-01, -1.05999788e-01,
        -1.05999788e-01, -1.05999788e-01, -1.05999788e-01,
        -1.05999788e-01, -1.05999788e-01, -1.05999788e-01,
        -1.05999788e-01, -1.50755672e-01, -1.05999788e-01,
        -1.05999788e-01, -1.05999788e-01, -1.05999788e-01,
        -1.05999788e-01, -1.50755672e-01, -1.05999788e-01,
        -1.50755672e-01, -1.50755672e-01, -1.05999788e-01,
        -1.50755672e-01, -1.50755672e-01, -1.05999788e-01,
```

Predictive analysis results 2

- train_test_split to split the data X and Y into training and test data:
  - Train set: (72, 83) (72,), Test set: (18, 83) (18,)
- logistic regression object:
  - tuned hpyerparameters :(best parameters) {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
  - accuracy : 0.8464285714285713
- support vector machine object:
  - tuned hpyerparameters :(best parameters) {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
  - accuracy : 0.8482142857142856
- decision tree classifier object:
  - tuned hpyerparameters :(best parameters) {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'}
  - accuracy : 0.8607142857142858

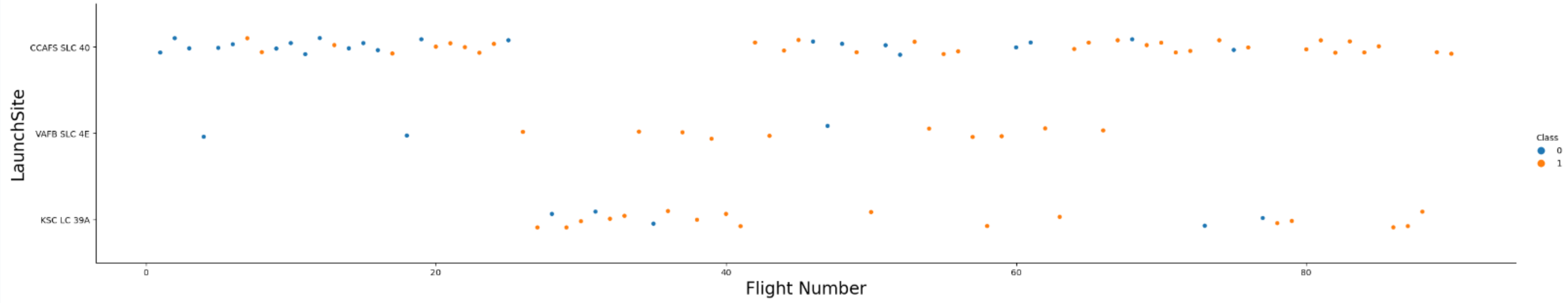
Predictive analysis results 3

- k nearest neighbors object:
  - tuned hpyerparameters :(best parameters) {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
  - accuracy : 0.8482142857142858
- the method performs best:
  - Accuracy for Logistics Regression method: 0.8333333333333334
  - Accuracy for Support Vector Machine method: 0.8333333333333334
  - Accuracy for Decision tree method: 0.9444444444444444
  - Accuracy for K nearest neighbors method: 0.8333333333333334
  -
  - The best method is Decision tree method with 94% accuracy.

Section 2

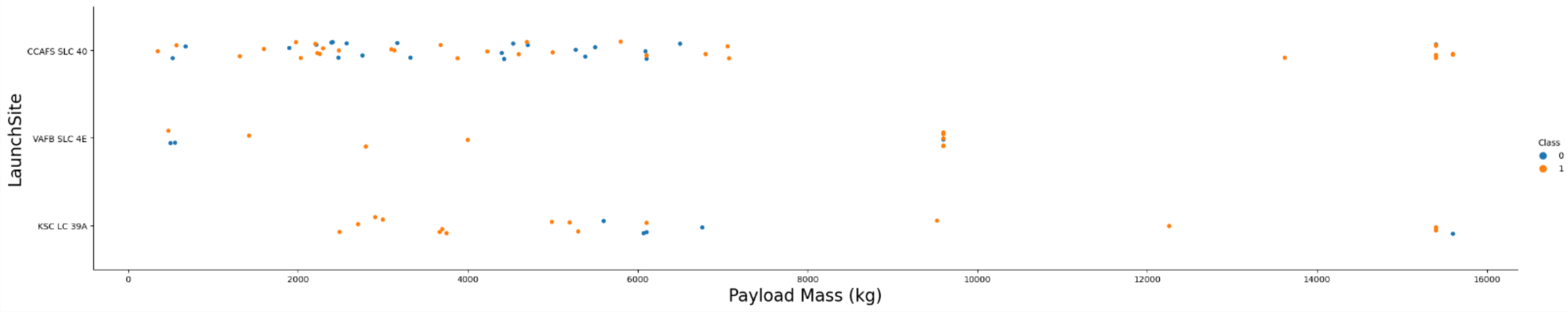
Insights drawn from EDA

Flight Number vs. Launch Site



- The last 13 flight were successful.
- The number of unsuccessful launches has obviously decreased.
- The CCAFS SLC 40 is the most used platform and it has got very high success rate in last quarter of flights.

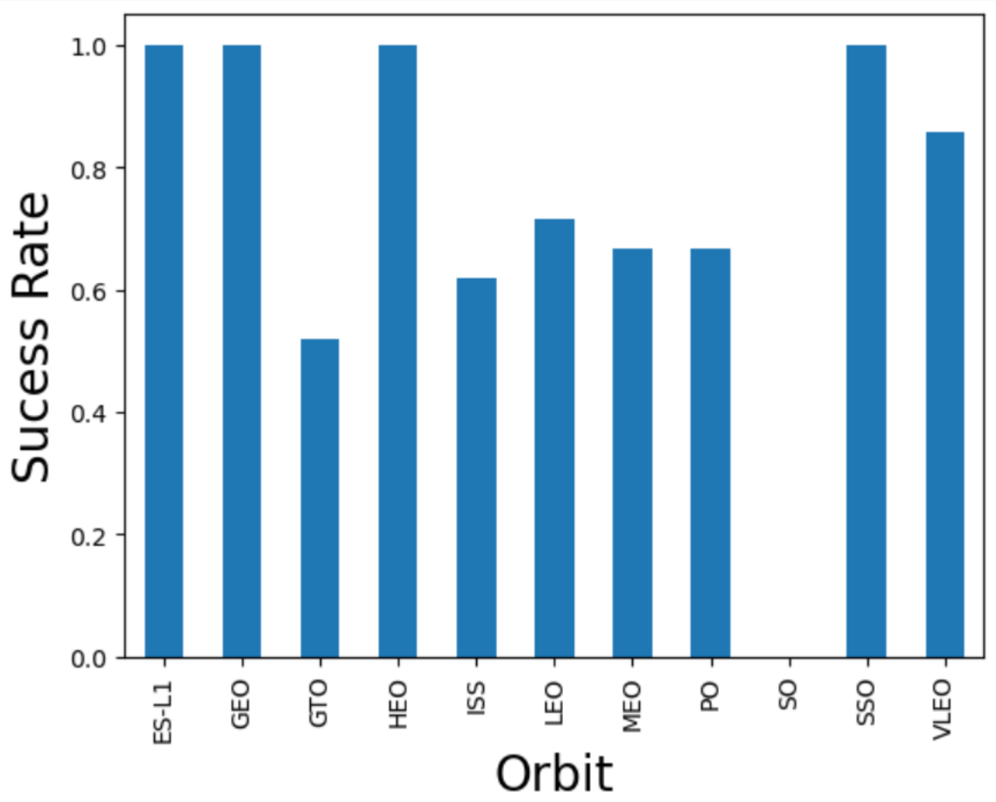
Payload vs. Launch Site



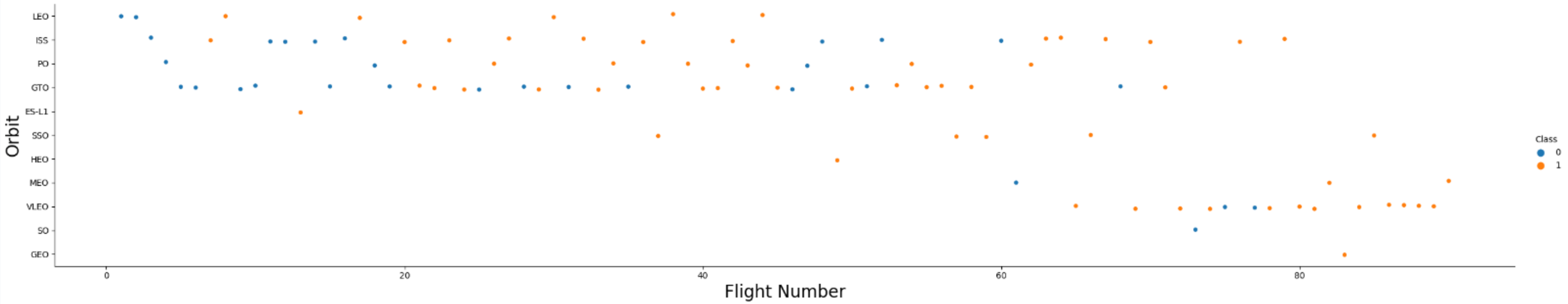
- There is almost 100% success rate above 9000 kg payload mass.
- Most used launch site above 12000kg is the CCAFS SLC.
- Probably, the early practice launches received a lower payload mass.

Success Rate vs. Orbit Type

- Perfect success rate happened in orbit:
  - ES-11, GEO, HEO.
- Worst happened in GTO and ISS orbit.
- SO is the same orbit as SSO. SO has got 1 unsuccess launch.

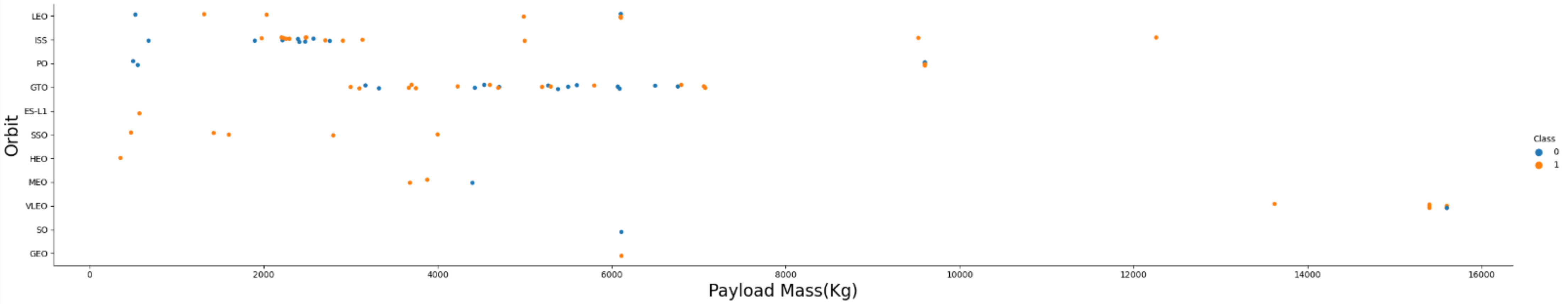


Flight Number vs. Orbit Type



- It can be clearly seen that the success rate has continuously increased in all orbits.
- VLEO and ISS were the most used orbit in the last segment.

Payload vs. Orbit Type

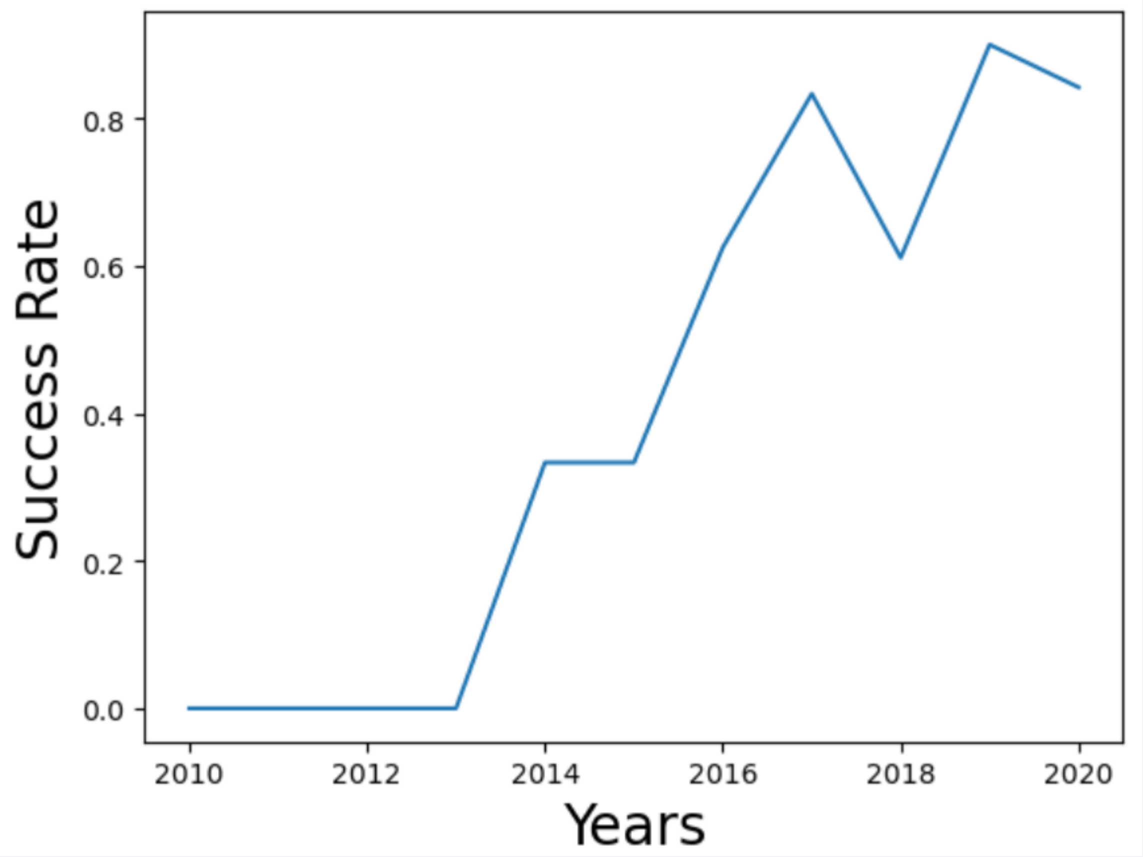


- For GTO orbit we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.
- ISS orbit has got widest range of payload mass.

Launch Success Yearly Trend

- The first 3 years was the technology improvement.

- The success rate has been steadily increasing since 2013, but in 2018 there was a sharp decline.



All Launch Site Names

- The names of the unique launch sites:
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40
- %sql select distinct(LAUNCH_SITE) from SPACEXDATASET

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- %sql select * from SPACEXDATASET where LAUNCH_SITE like 'CCA%' limit 5;

Total Payload Mass

- The total payload mass carried by boosters launched by NASA (CRS):
  - NASA_CRS_total_payload_mass: 45596 kg

- %sql select sum(PAYLOAD_MASS__KG_) as NASA_CRS_total_payload_mass from SPACEXDATASET where customer = 'NASA (CRS)'

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1
  - average_payload_mass_F9_v1_1: 2928.4

- %sql select avg(PAYLOAD_MASS__KG_) as average_payload_mass_F9_v1_1 from SPACEXDATASET where booster_version = 'F9 v1.1'

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad
  - first_succes_landing: 22-12-2015

- %sql select DATE as first_succes_landing from SPACEXDATASET where "LANDING _OUTCOME" like "Success (ground pad)" order by date desc limit 1;

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
  - Booster Version: F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2

- %sql select booster_version as bv from SPACEXDATASET where PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 and "LANDING _OUTCOME" like "Success (drone ship)"

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes
  - <u>Mission_Outcome Count_MO</u>
  - Failure (in flight) 1
  - Success (payload status unclear) 1
  - Success 99
- %sql select mission_outcome,count(mission_outcome) as Count_MO from SPACEXDATASET group BY mission_outcome order BY count("LANDING _OUTCOME");

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass:

- %sql select PAYLOAD_MASS__KG_,booster_version as BV from SPACEXDATASET where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXDATASET) order by booster_version;

PAYLOAD_MASS__KG_	BV
15600	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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
Month	Landing Outcome	Booster Version	Launch Site	Year
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015

- %sql SELECT substr(Date, 4, 2) as Month, "landing _outcome", BOOSTER_VERSION, LAUNCH_SITE, substr(Date, 7, 4) as Year FROM SPACEXDATASET where substr(Date,7,4)='2015' and "landing _outcome" = "Failure (drone ship)";

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

- 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:

Landing_Outcome	Success_Landing_outcomes_between_04062010_20032017
Success (drone ship)	5
Success (ground pad)	3

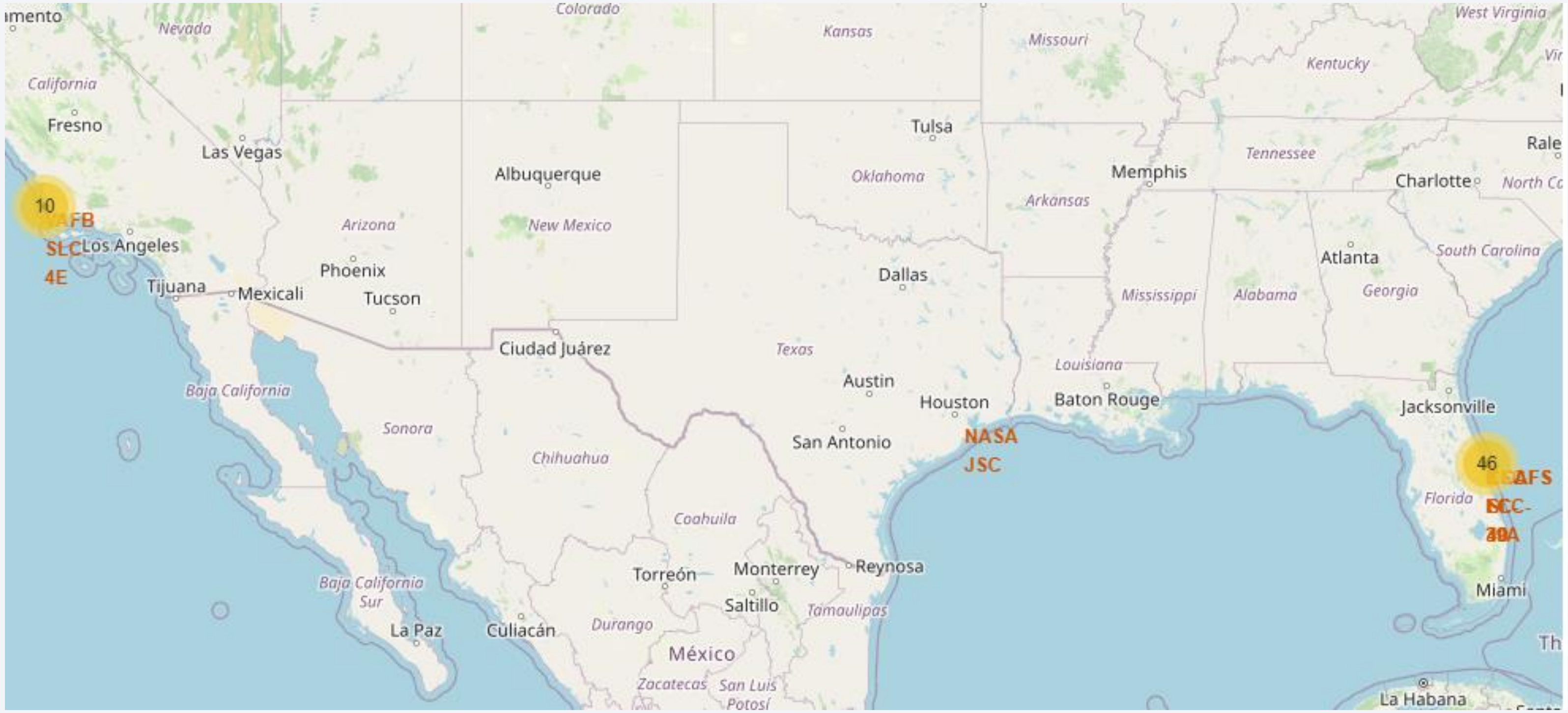
- %sql select landing_outcome, count(landing_outcome) as Success_Landing_outcomes_between_04062010_20032017 from SPACEXDATASET where landing_outcome like 'Success%' and Date between '2010-06-04' and '2017-03-20' group by landing_outcome order by Date desc;

- Reformated the date before the query: %sql UPDATE SPACEXDATASET SET Date = substr(Date, 7, 4) || '-' || substr(Date, 4,2) || '-' || substr(Date, 1,2);

Section 3

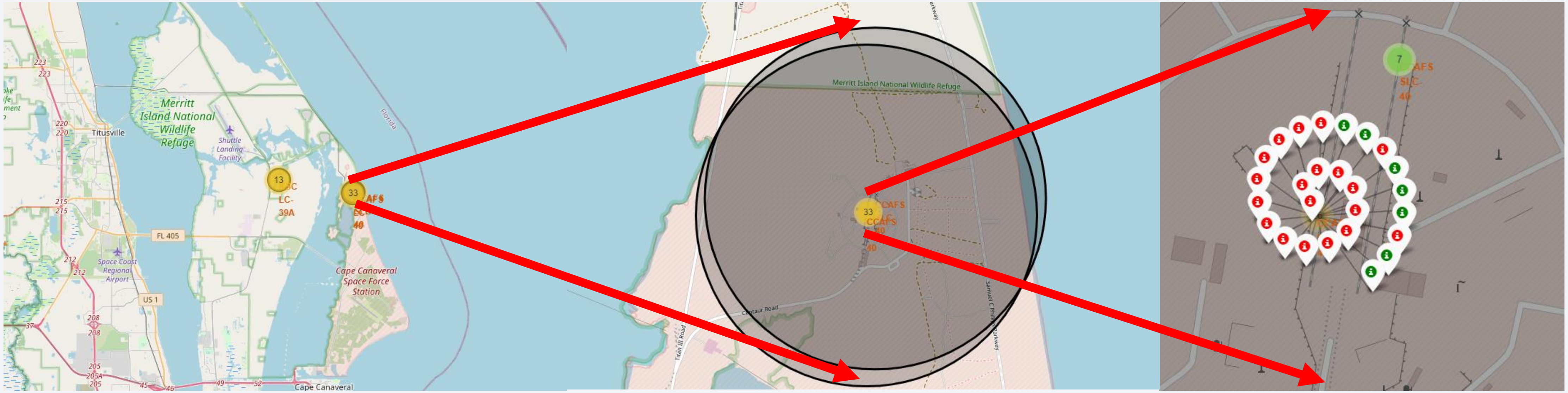
Launch Sites Proximities Analysis

The success/failed launches for each site location markers on a global map



- All launch sites on the coast in California and Florida, and the NASA space center in Houston

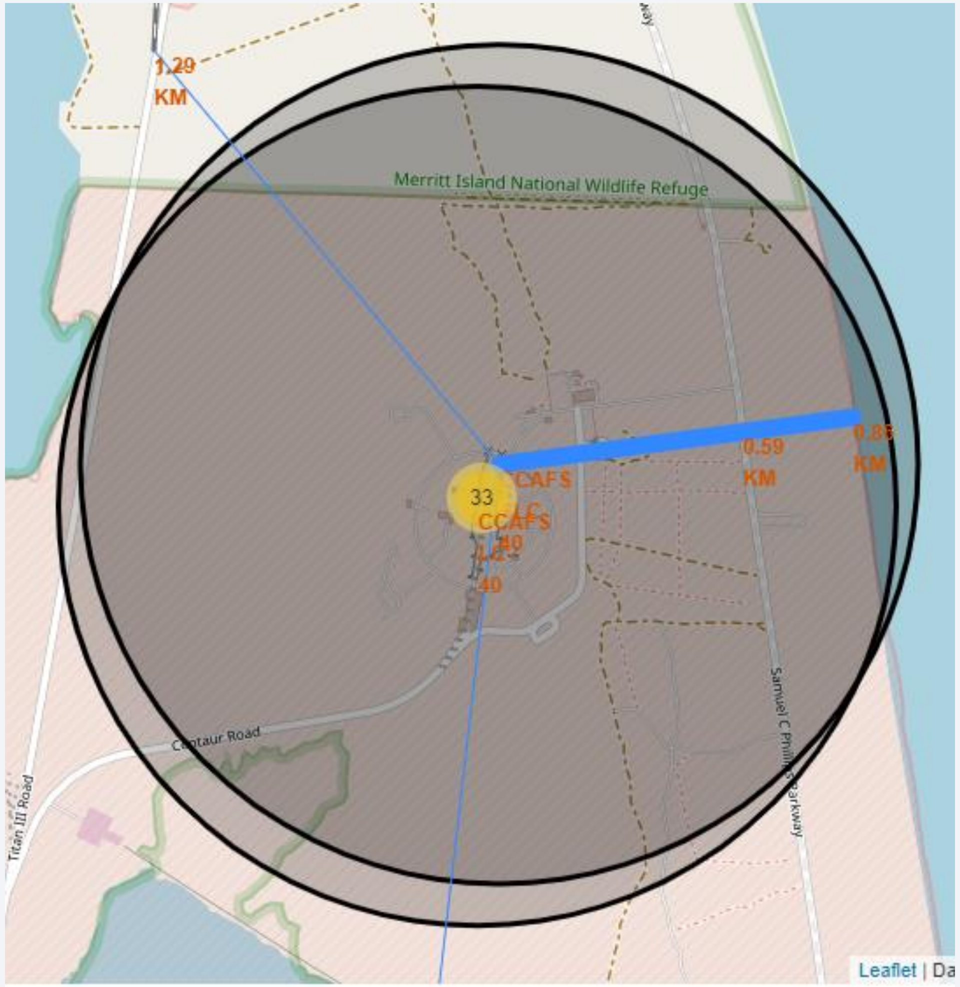
Launch outcomes on the map



- The color-labeled launch outcomes on CCASF SLC-40 site

The distances between a launch site to its proximities

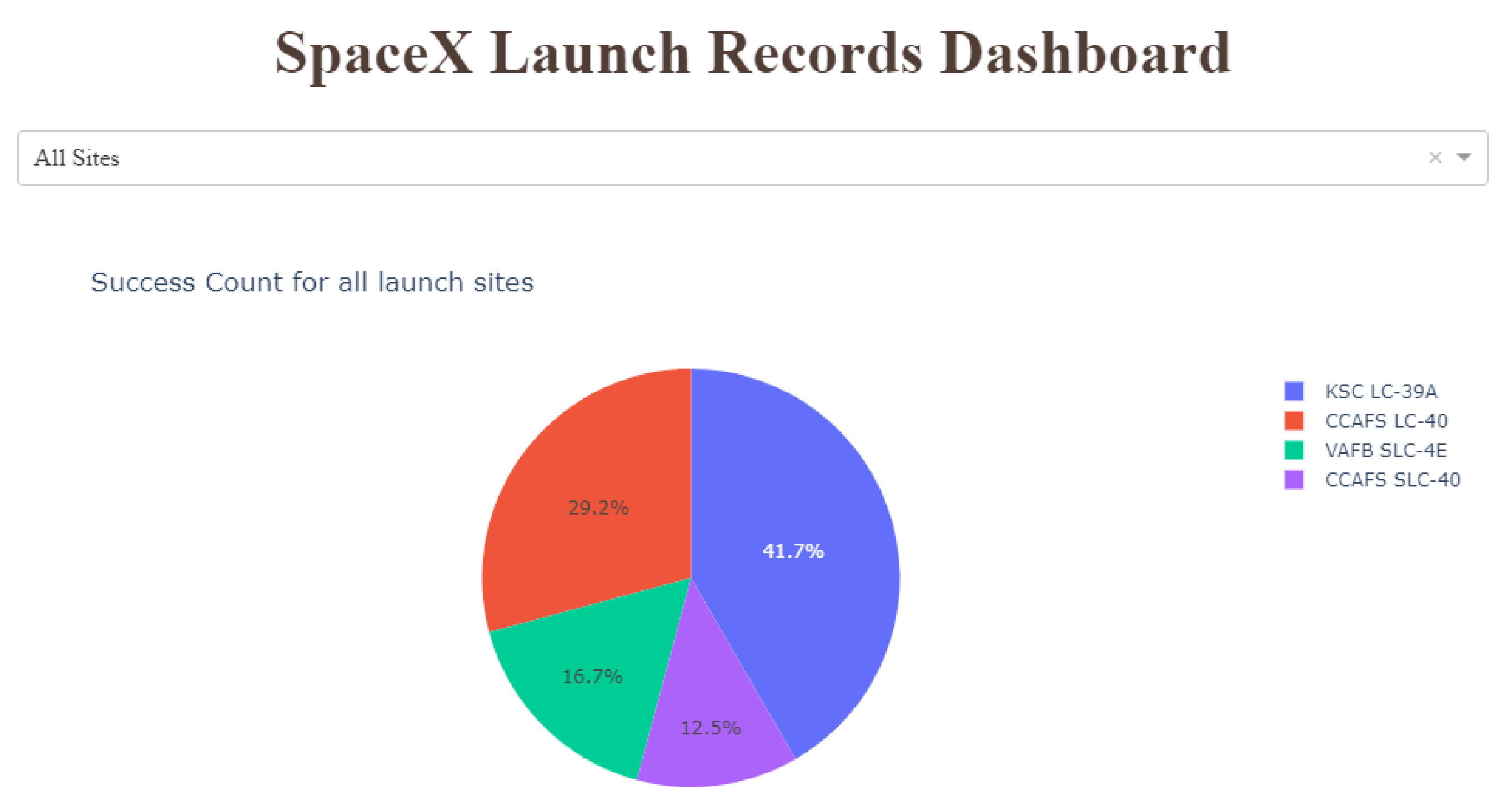
- distance_railroad = 1.2860143661039483 km
- distance_highway = 0.5868025820947117 km
- distance_city = 51.21971215802352 km
- distance_coastline = 0.8555636865674044 km



Section 4

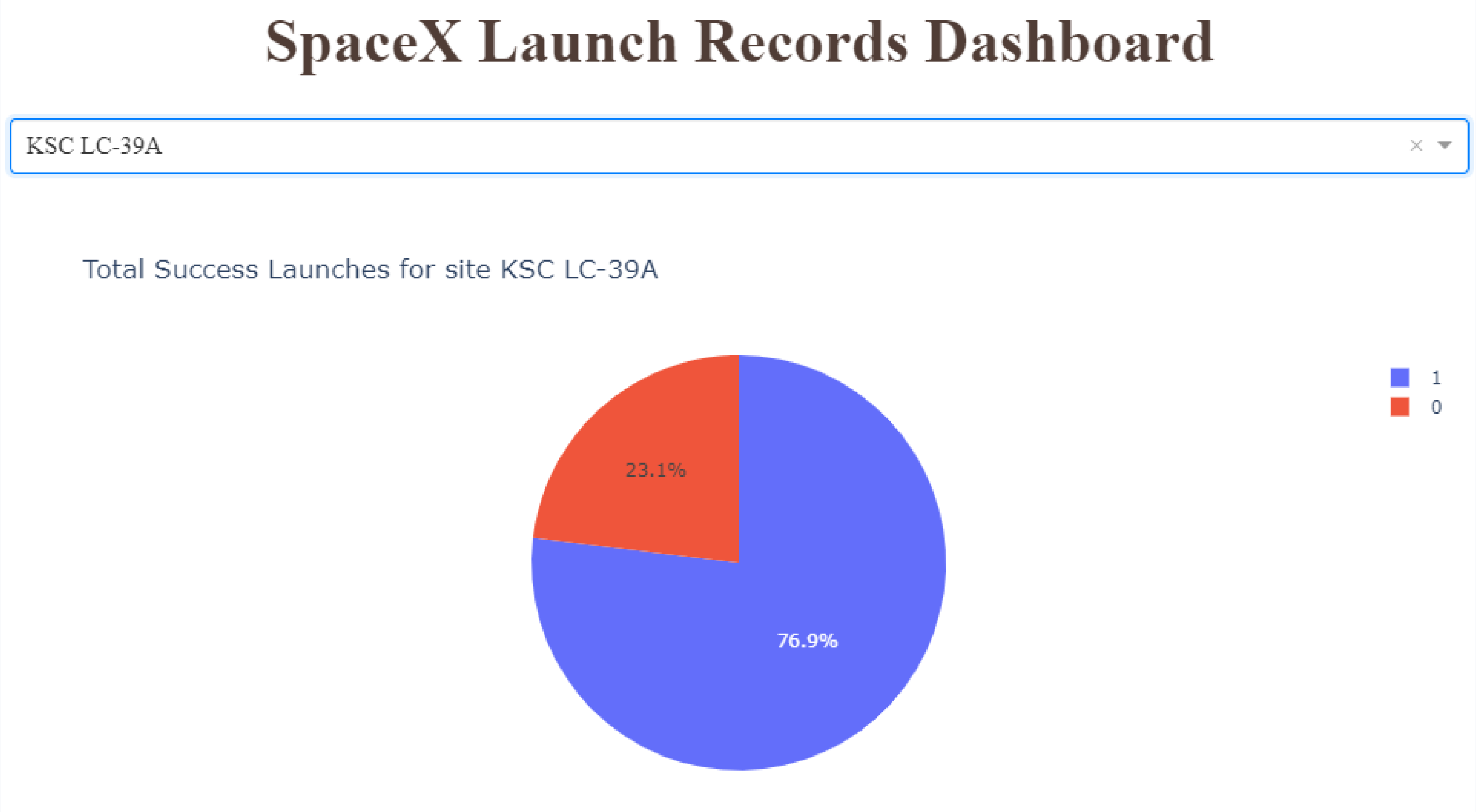
Build a Dashboard with Plotly Dash

All sites success launches



- The most successful launches were on the KSC LC-39A site

Most successful launch site



- KSC LC-39A is the most successful site 76.9% success rate

Payload range with the most evaluable data

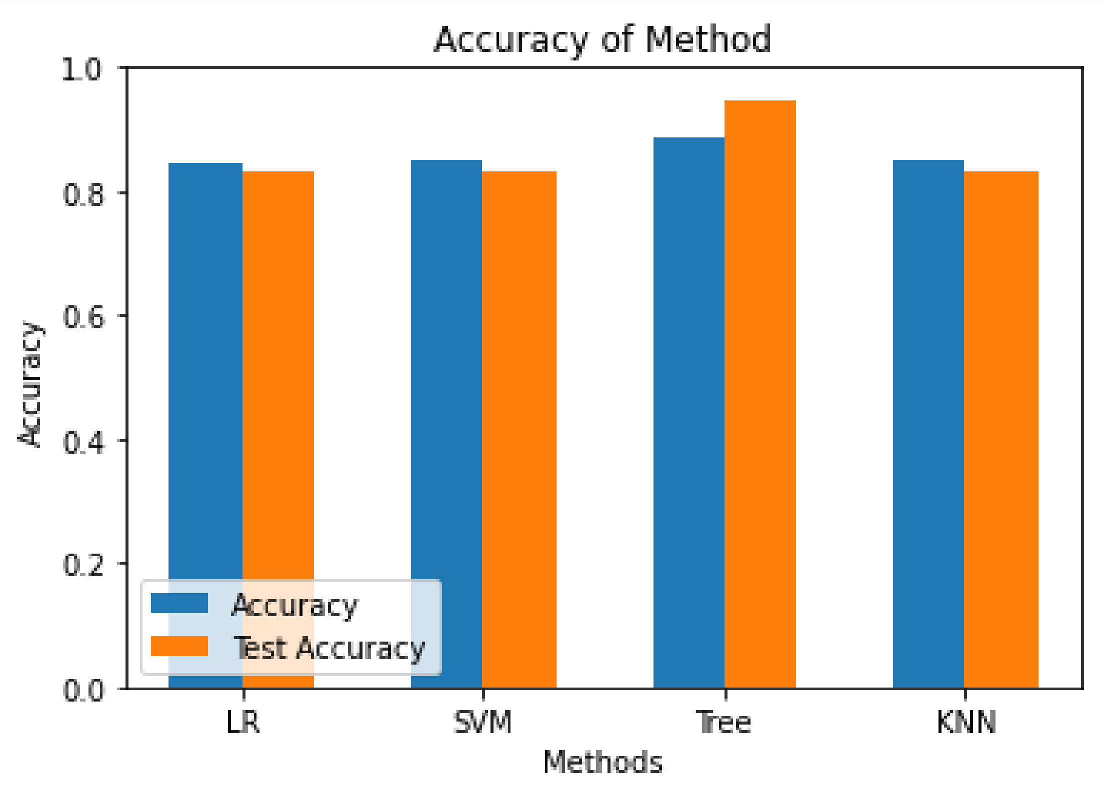
- The range with most evaluable data is between 2000 and 7000 kg of Payload Mass
- The most success launches were with the FT booster version in this Payload Mass range

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The best method is Decision tree method with 94% accuracy.
  - Train Accuracy: 0.8875,
  - Test Accuracy: 0.94444
- tuned hpyerparameters :(best parameters)
  - {'criterion': 'gini',
  - 'max_depth': 6,
  - 'max_features': 'sqrt',
  - 'min_samples_leaf': 2,
  - 'min_samples_split': 10,
  - 'splitter': 'random'}



Confusion Matrix

- The confusion matrix of the Decision Tree method contains the most biggest true positive value and small false values

Conclusions

- Success rate of launching is increasing time by time
- Above 9000 kg payload mass is very low risk.
- The best launch site is KSC LC-39A with highest success rate
- The Decision tree method is the best method.

Appendix

- On the Surface of Skills Network Labs, I was not able to make the Dashboard with Plotly Dash. Finally I made it in Jupyter Notebook. That was the hardest time when I try to find the good position of the characters on the code of Plotly. I will never use it again for dashboard.
- Ridiculous, I had to get an IBM Cloud Feature Code four times to complete the course. I don't really understand why IBM reduced the amount of usable data and processing time on free accounts.
- IBM should create an unlimited website where students can learn IBM programs absolutely free.
- All in all, it was a very informative and thorough course, which I will have to revisit for a very long time in order to fully master Data Science.
- So Long, and Thanks for All the Fish

Thank you!