IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

SR

18 July 2023

Outline

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

Executive Summary

- Data from scraping Wikipedia and SpaceX REST API
- Data exploration: plotting, SQL queries, plotly dashboard
- Prediction: grid search for logistic regression, SVM, decision tree, k-means

Introduction

- The company SpaceX develops rockets
- Test launches happen at various launch sites, with varying payloads / boosters / etc.
- Goal: Predict success of future launch based on previous launches & accessible parameters

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Webscraping Falcon Wikipedia page + SpaceX REST API
- Perform data wrangling
  - Pandas dataframe / SQL DB, feature generation, standardization
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- ApaceX REST API
  - make request according to api specification
  - get JSON -> parse json -> insert into dataframe

- Webscraping
  - load page -> process with beautiful soup -> find table in DOM
  - parse with custom loop or pandas -> make dataframe

Data Collection – SpaceX API

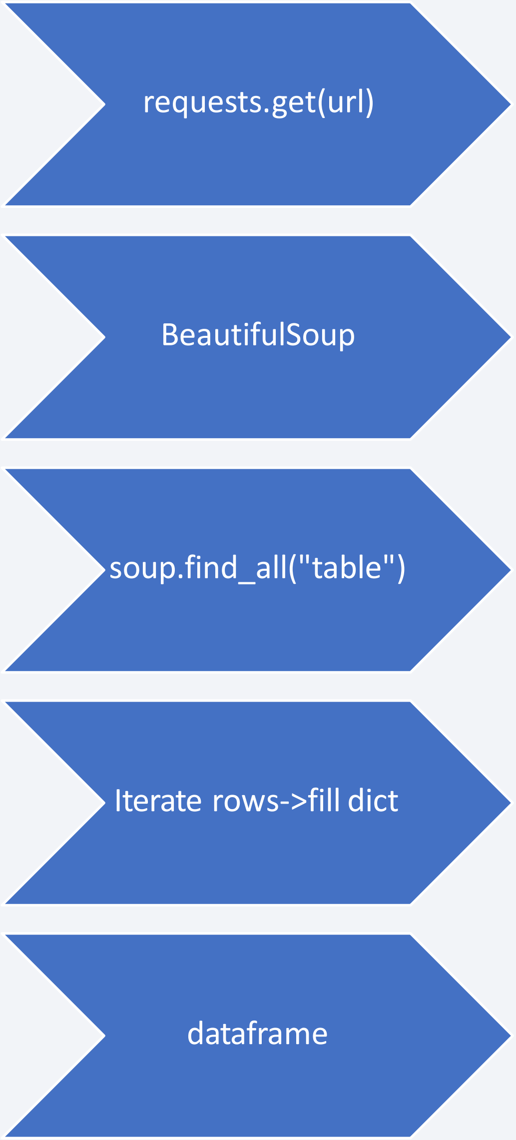
- https://api.spacexdata.com/v4/*
- 1) Get history of launches
- 2) Get further info for various occurring IDs

- https://github.com/s3bru/ibm_capstone/blob/main/1_jupyter-labs-spacex-data-collection-api.ipynb

/launches/past	rocket_id, payload_id, launchpad_id, {core_id},flight_number, date_utc
/rockets/<id>	BoosterVersion
/launchpads/<id>	Longitude, Latitude, LaunchSite
/payloads/<id>	PayloadMass, Orbit
/cores/<id>	Block, ReusedCount, Serial, Outcome, Flights, GridFins, Reused, Legs, LandingPad

Data Collection - Scraping

- Download the page ” List_of_Falcon_9_and_Falcon_Heavy_launches” from wikipedia
- Process with beautiful soup
- Find table in DOM
- Iterate rows, extract info, put in dicts
- Make pandas dataframe
- https://github.com/s3bru/ibm_capstone/blob/main/1_jupyter-labs-webscraping.ipynb



Data Wrangling

- Calculate the number and occurence of mission outcome per orbit type
- Create a landing outcome label from Outcome column
- https://github.com/s3bru/ibm_capstone/blob/main/1_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

- Scatter: FlightNumber - PayloadMass & hue==class
- Scatter: FlightNumber - LaunchSite & hue==class
- Scatter: PayloadMass - LaunchSite & hue==class
- Bar: Sucess_rate – Orbit type
- Scatter: FlightNumber - Orbit type & hue==class
- Scatter: Payload - Orbit type & hue==class
- Line: Year - Sucess_rate

EDA with SQL: Queries 1

- **names of unique launch sites**
- **launch sites beginning with the string 'CCA‘**
- **total payload mass carried by boosters launched by NASA (CRS)**
- **average payload mass carried by booster version F9 v1.1**
- **date when first successful landing in ground pad was achieved**
- **names of boosters with success in drone ship and payload mass between 4000 and 6000**

EDA with SQL: Queries 2

- **total number of successful /failure mission outcomes**
- **names of booster_versions which have carried the maximum payload mass**
- **month names, failure landing_outcomes in drone ship ,booster versions, launch_site for all months in year 2015**
- **Ranking of count of landing outcomes between 2010-06-04 and 2017-03-20**

Build an Interactive Map with Folium

- Markers + labels + circle for nasa JSC + launch sites (relevant locations)
- MarkerClusters containing launches (suc./fail) for each launch site (overview of relevant events)
- Mouse position -> cords (find coords of things in vicinity)
- Marker + line to closest point of coast line, railway, city and highway
- (water -> lower risk of endangerment <-> railway / city / highway -> could be endangered)
- https://github.com/s3bru/ibm_capstone/blob/main/3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Inputs: Dropdown launch sites + Slider payload range
- Graphs:
  - Pie chart: either total succ. per launch site or succ&fail for selected launch site -> comparison of outcomes for all / specific LS
  - Scatter chart: payload – launch success -> shows correlation (for selected inputs)
- https://github.com/s3bru/ibm_capstone/blob/main/3_spacex_dash_app.py

Predictive Analysis (Classification)

- Standardize feature data
- Split data in train / test sets
- Perform parameter grid search for
  - Logistic regression / SVM / decision tree / KNN
- Display confusion matrix / accuracy score for best parameters (test set)
- Pick best performing model
- https://github.com/s3bru/ibm_capstone/blob/main/4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

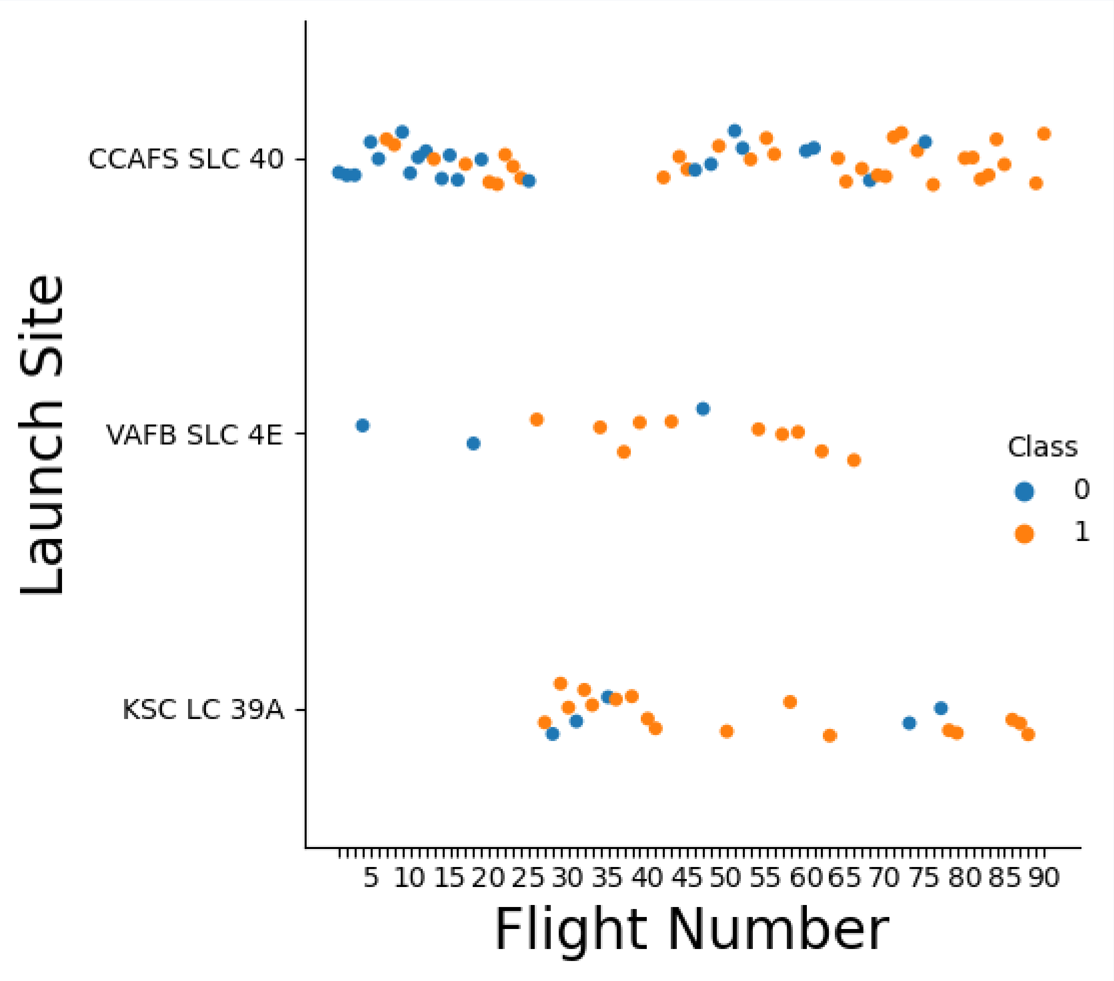
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

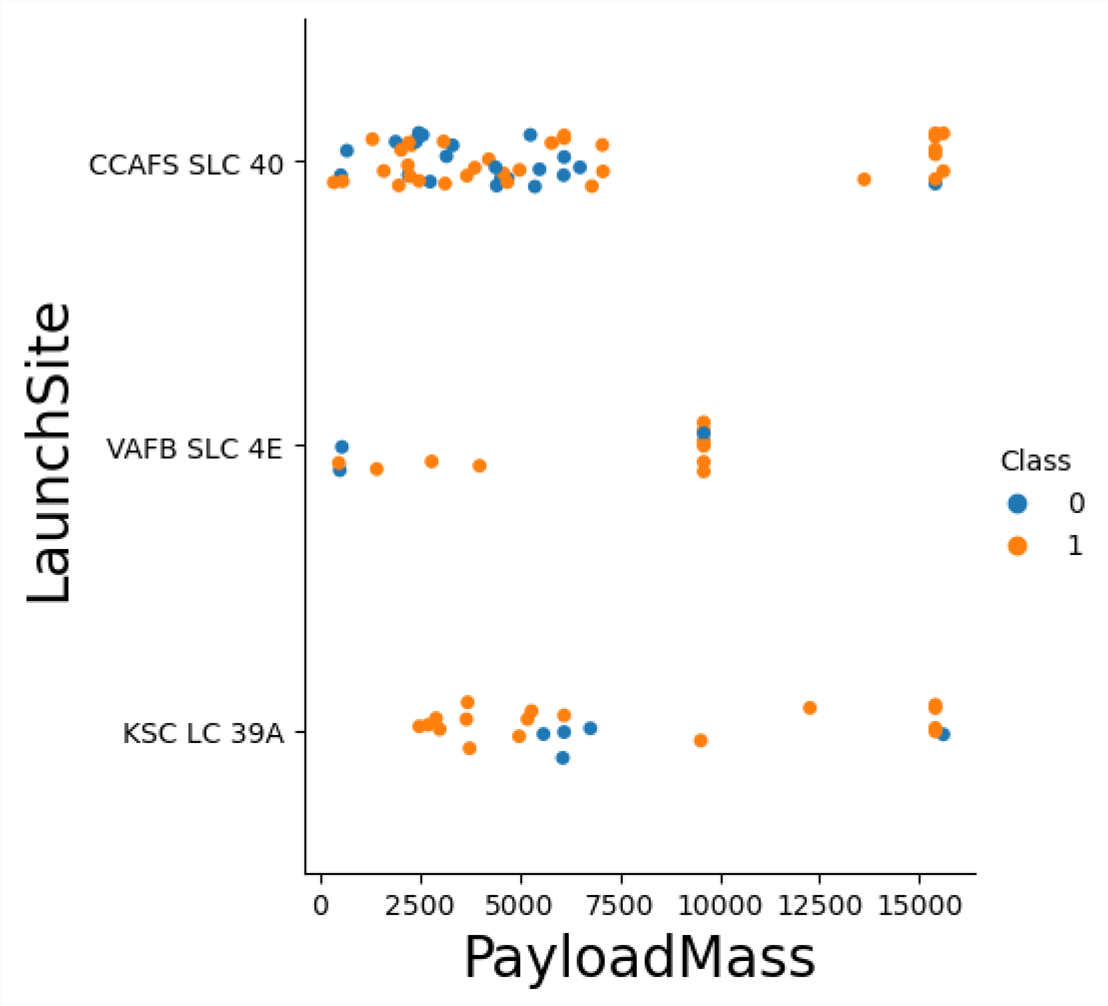
Flight Number vs. Launch Site

- Launches at VAFB mostly successful
- At CCAFS rate increases with flight number



Payload vs. Launch Site

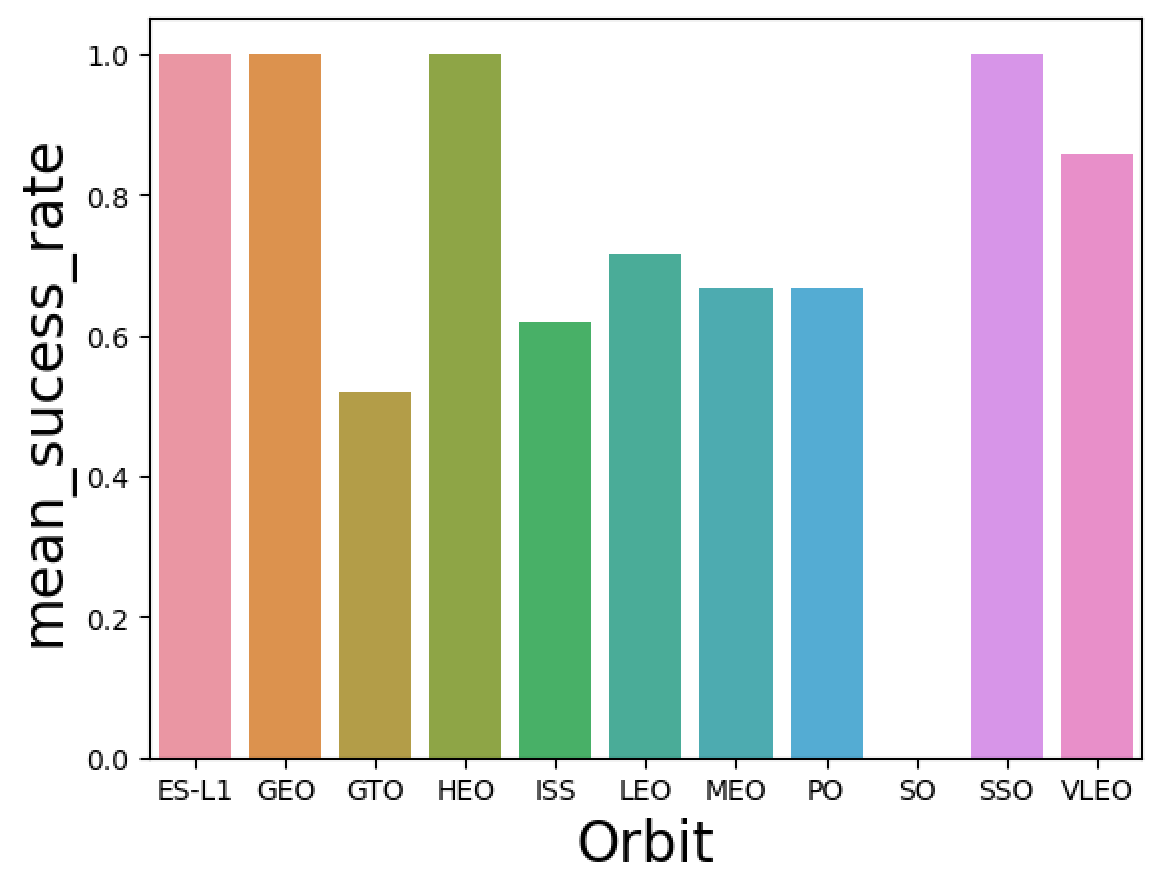
- KSC fail for mid level payload
- CCAFS low correlation, success for high payload
- VAFB fail for low payload



Success Rate vs. Orbit Type

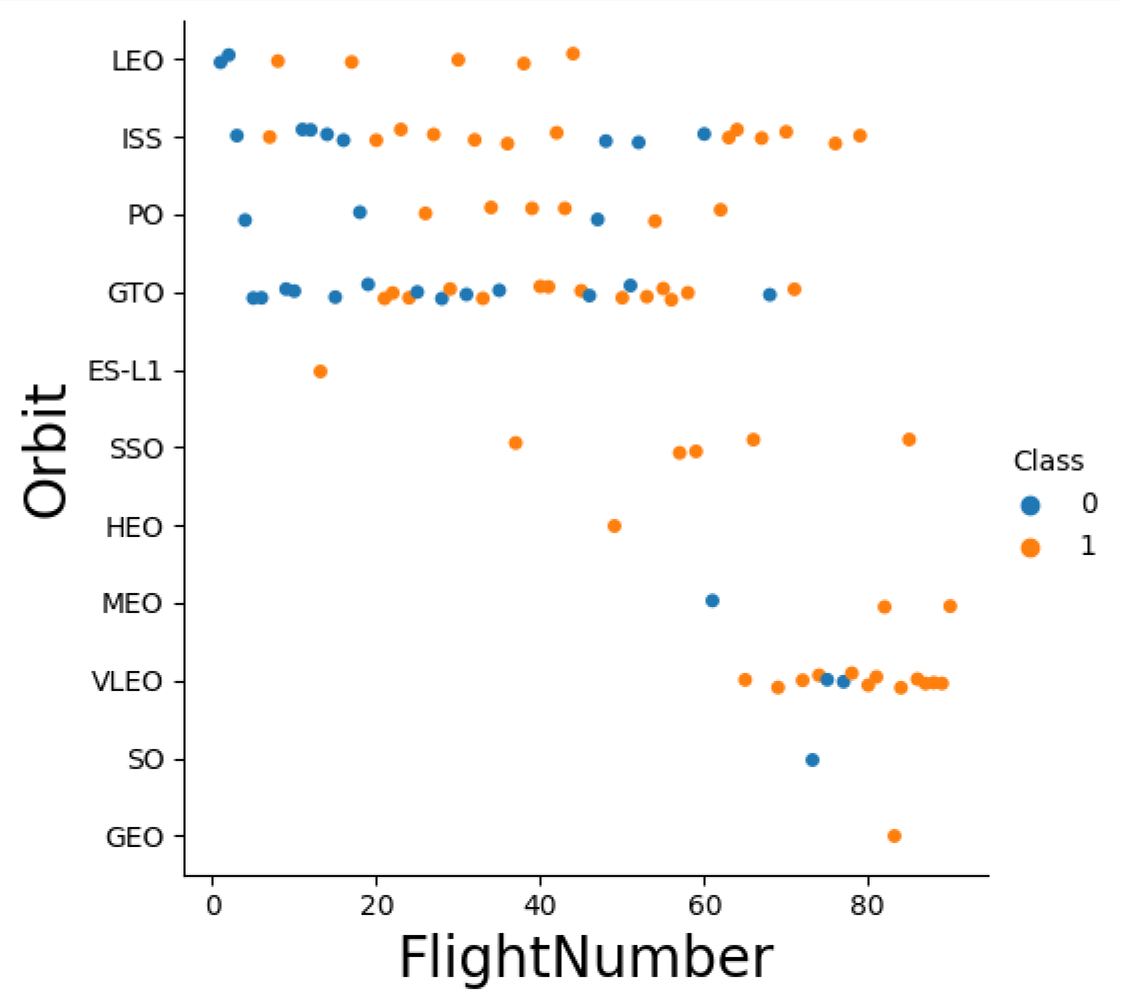
- All launches were successful for ES-L1, GEO, HEO, SSO

- Lowest success rate for GTO



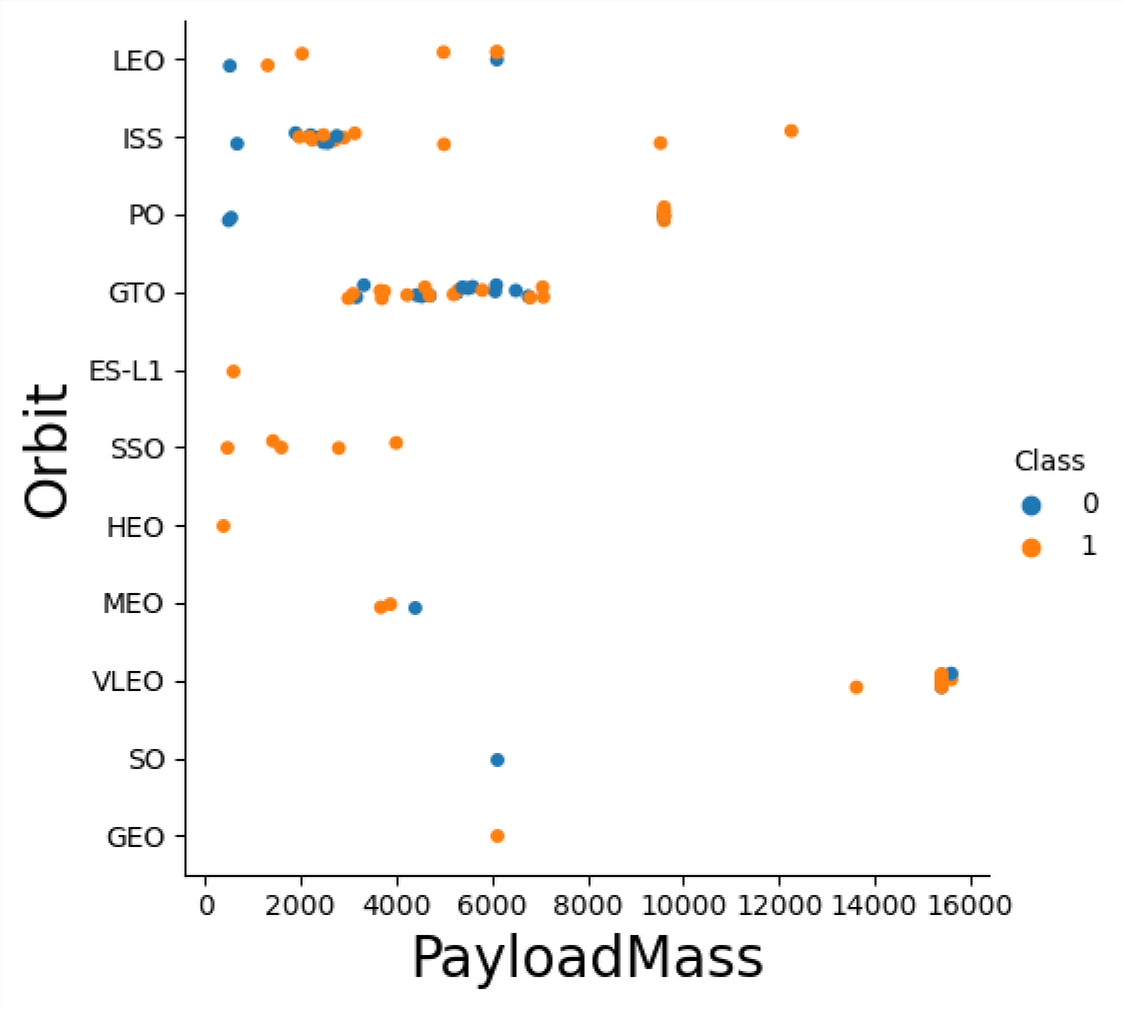
Flight Number vs. Orbit Type

- Low Flight Number : LEO, ISS, PO, GTO
- First fail then more succ.
- SSO always succ.
- Many VLEO attempts, only 2x fail in middle
- Geo succ. at first try



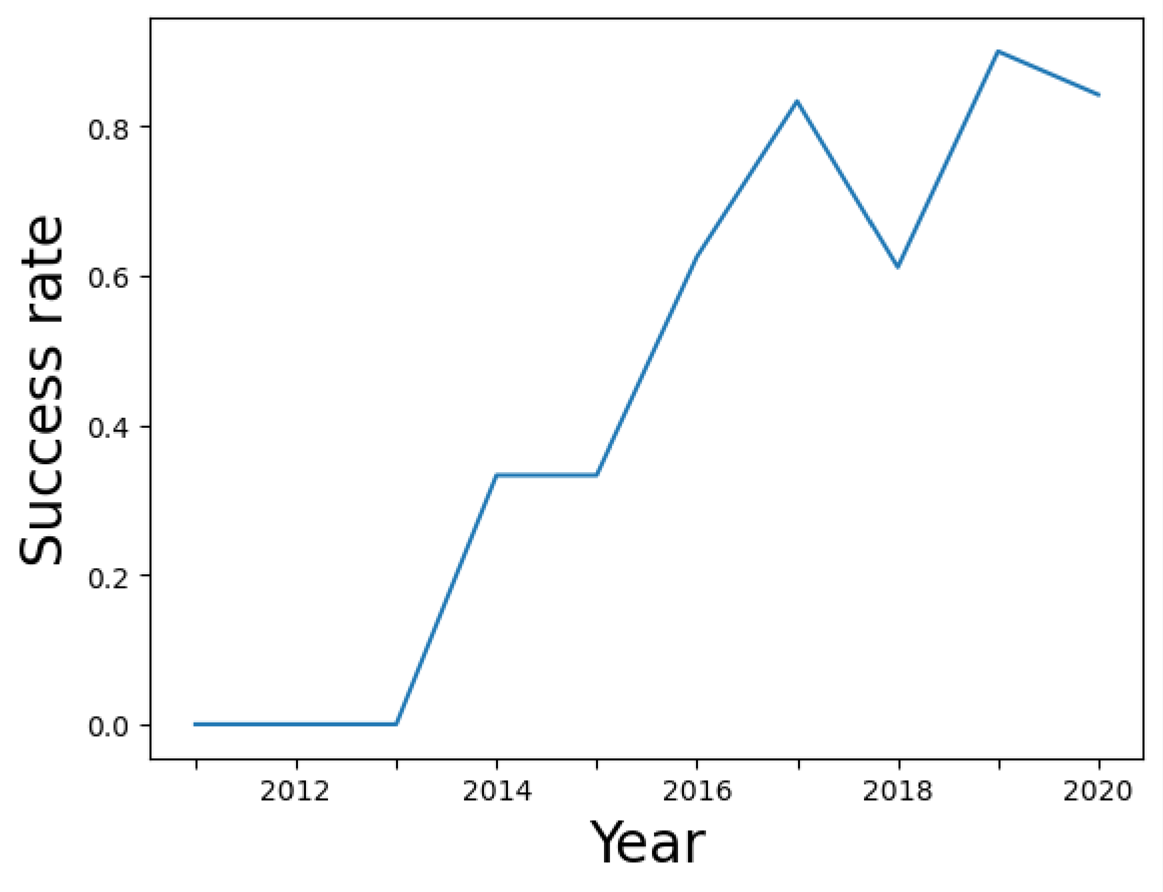
Payload vs. Orbit Type

- LEO, ISS, PO fail with low payload
- GTO uncorrelated



Launch Success Yearly Trend

- Success rate yearly increase except for 2018!



All Launch Site Names

- SQL: Select distinct(Launch_Site) from SPACEXTBL

-> ['CCAFS LC-40', 'VAFB SLC-4E', 'KSC LC-39A', 'CCAFS SLC-40', None]

- None -> incomplete data

Launch Site Names Begin with 'CCA'

- SQL:Select * from SPACEXTBL where Launch_Site like"CCA%" limit 5

-> [('06/04/2010', '18:45:00', 'F9 v1.0 B0003', 'CCAFS LC-40', 'Dragon Spacecraft Qualification Unit', 0.0, 'LEO', 'SpaceX', 'Success', 'Failure (parachute)'),

('12/08/2010', '15:43:00', 'F9 v1.0 B0004', 'CCAFS LC-40', 'Dragon demo flight C1, two CubeSats, barrel of Brouere cheese', 0.0, 'LEO (ISS)', 'NASA (COTS) NRO', 'Success', 'Failure (parachute)'),

('22/05/2012', '7:44:00', 'F9 v1.0 B0005', 'CCAFS LC-40', 'Dragon demo flight C2', 525.0, 'LEO (ISS)', 'NASA (COTS)', 'Success', 'No attempt'),

('10/08/2012', '0:35:00', 'F9 v1.0 B0006', 'CCAFS LC-40', 'SpaceX CRS-1', 500.0, 'LEO (ISS)',

'NASA (CRS)', 'Success', 'No attempt'), ('03/01/2013', '15:10:00', 'F9 v1.0 B0007', 'CCAFS LC-40', 'SpaceX CRS-2', 677.0, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt')]

Total Payload Mass

- SQL: Select PAYLOAD_MASS__KG_ from SPACEXTBL where Customer like "NASA (CRS)%"

-> 48213.0

Average Payload Mass by F9 v1.1

- SQL: Select PAYLOAD_MASS__KG_ from SPACEXTBL where Booster_Version like "F9 v1.0%"

-> 340.4

First Successful Ground Landing Date

- SQL: Select min(Date) from SPACEXTBL where Landing_Outcome = "Success (ground pad)“

->[('01/08/2018',)]

Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL: Select Booster_Version from SPACEXTBL where Landing_Outcome = \"Success (drone ship)\" and PAYLOAD_MASS__KG_ between 4000 and 6000

->[('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT B1021.2',), ('F9 FT B1031.2',)]

Total Number of Successful and Failure Mission Outcomes

- SQL: Select Landing_Outcome from SPACEXTBL where Landing_Outcome like "Success%“

->61

- SQL: Select Landing_Outcome from SPACEXTBL where Landing_Outcome like "Failure%“

->10

Boosters Carried Maximum Payload

- SQL: Select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)

->|F9 B5 B1048.4| |F9 B5 B1049.4| |F9 B5 B1051.3| |F9 B5 B1056.4| |F9 B5 B1048.5| |F9 B5 B1051.4| |F9 B5 B1049.5| |F9 B5 B1060.2 | |F9 B5 B1058.3 | |F9 B5 B1051.6| |F9 B5 B1060.3|
|F9 B5 B1049.7 |

2015 Launch Records

- SQL: Select substr(Date, 4, 2),Landing_Outcome,Booster_Version,Launch_Site from SPACEXTBL where Landing_Outcome= "Failure (drone ship)"and substr(Date,7,4)='2015'

->|('10', 'Failure (drone ship)', 'F9 v1.1 B1012', 'CCAFS LC-40')|
|('04', 'Failure (drone ship)', 'F9 v1.1 B1015', 'CCAFS LC-40')|

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

- SQL: Select count(*),Landing_Outcome from SPACEXTBL group by Landing_Outcome order by count(*) desc

->[(898, None), (38, 'Success'), (21, 'No attempt'), (14, 'Success (drone ship)'), (9, 'Success (ground pad)'), (5, 'Failure (drone ship)'), (5, 'Controlled (ocean)'), (3, 'Failure'), (2, 'Uncontrolled (ocean)'), (2, 'Failure (parachute)'), (1, 'Precluded (drone ship)'), (1, 'No attempt ')]

Section 3

Launch Sites Proximities Analysis

Folium: Launch sites

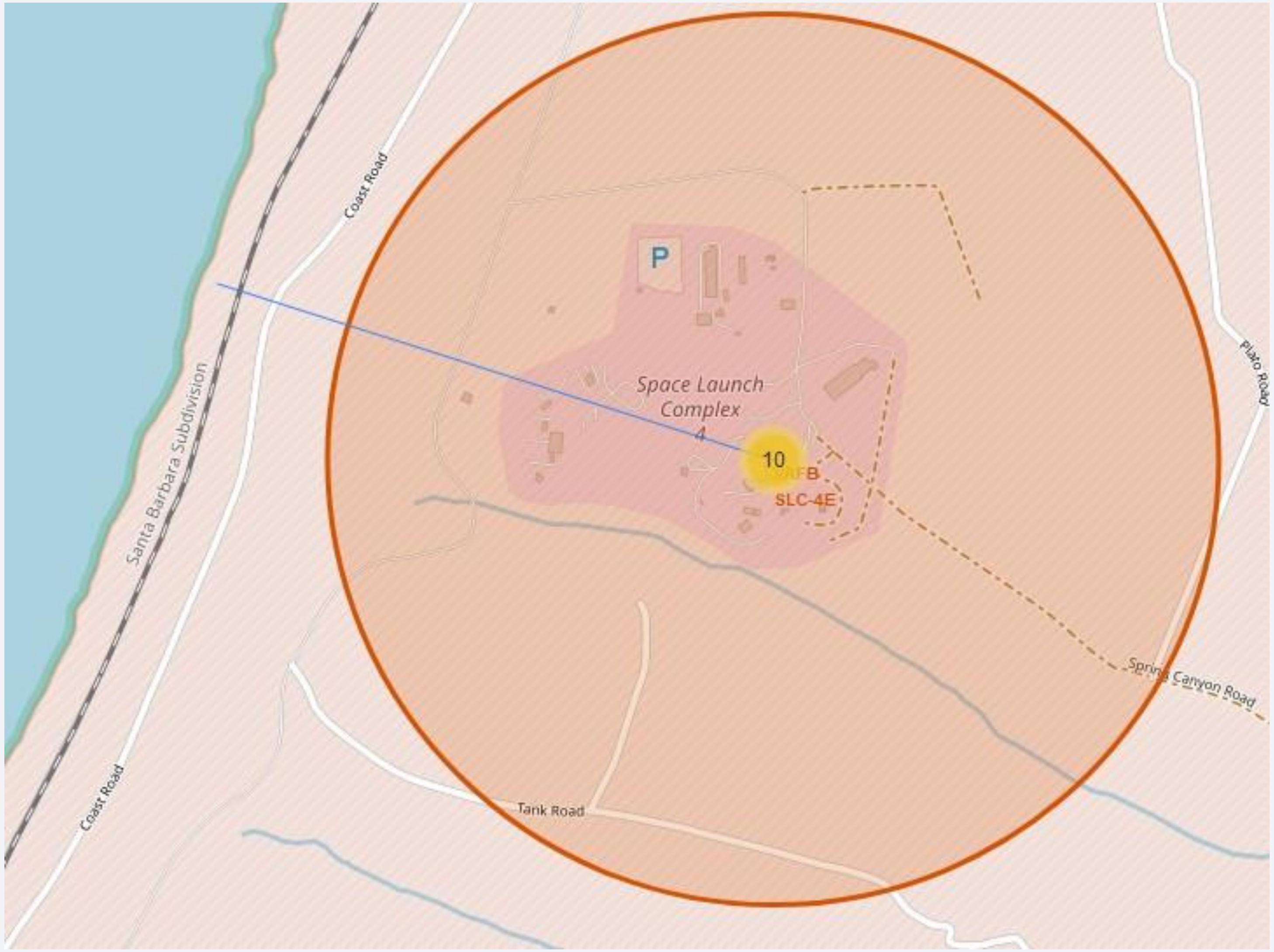
- Markers for JSC + launch sites
- VAFB at west coast, others east coast(grouped very closely)

Folium: Launch events

- For each launch site: show color coded(succ.=green) launch events

Folium: Launch site proximities

- Distance to coastline marked(d=1.31 km)
- Railroad can be seen near coast



Section 4

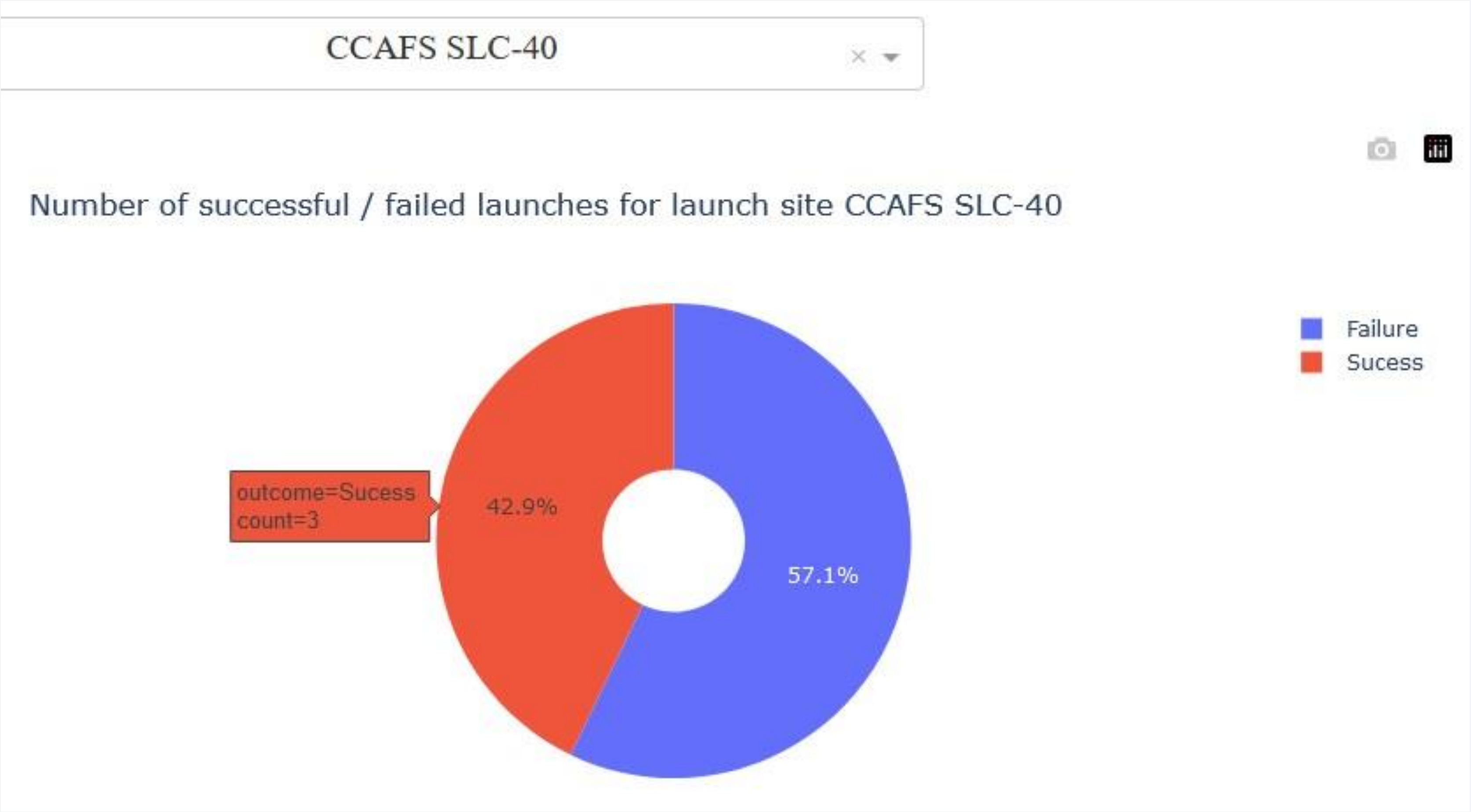
Build a Dashboard with Plotly Dash

Piechart 1

- Values==successful launches for each site

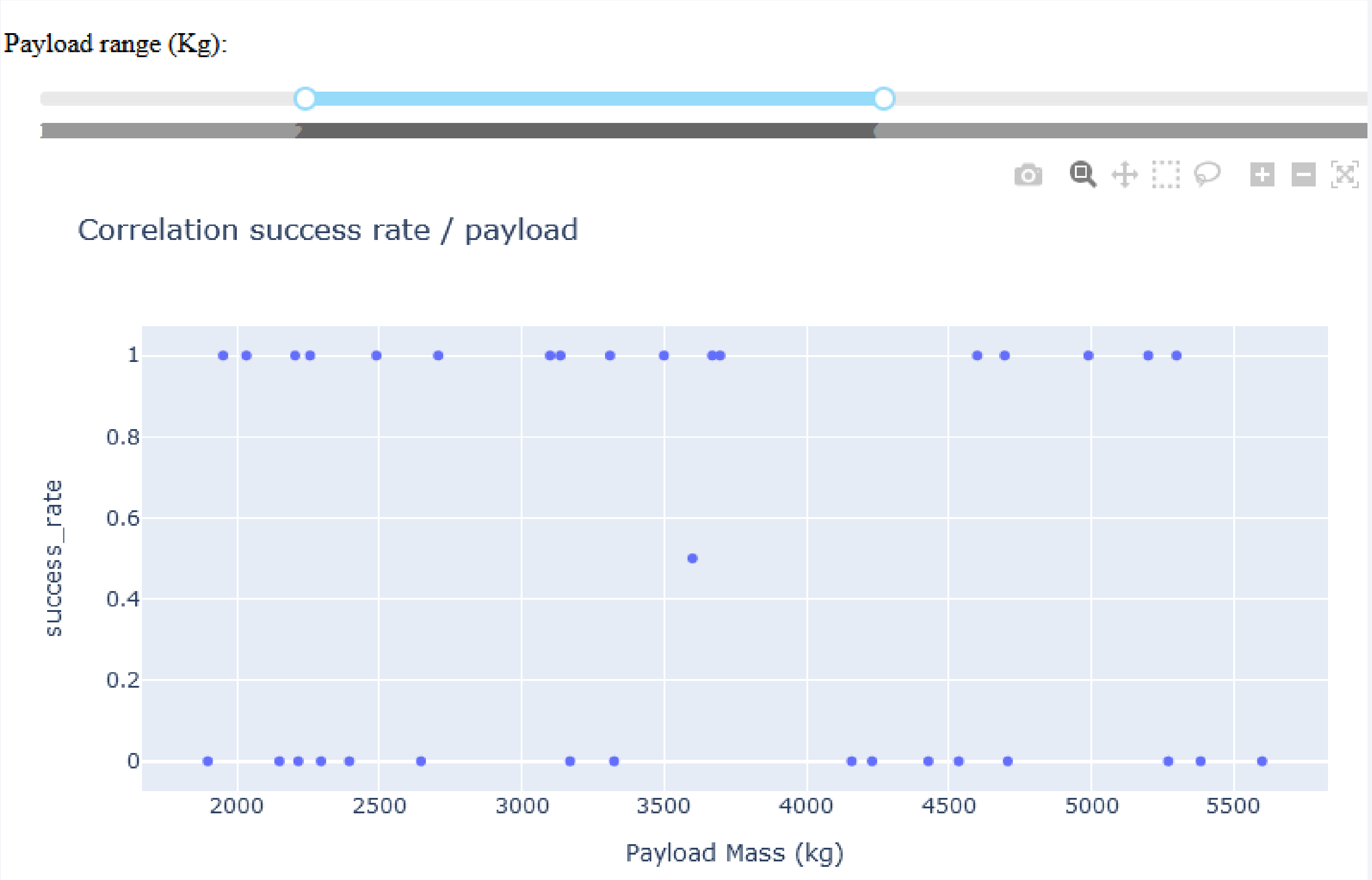
Piechart 2

- Dropdown field selects launch site
- Values==count of succ./ fail for selected launch site



Scatterplot

- Slider selects range for payload mass
- Dropdown selects which launch sites are considered
- Scatter plot for success rate / payload mass

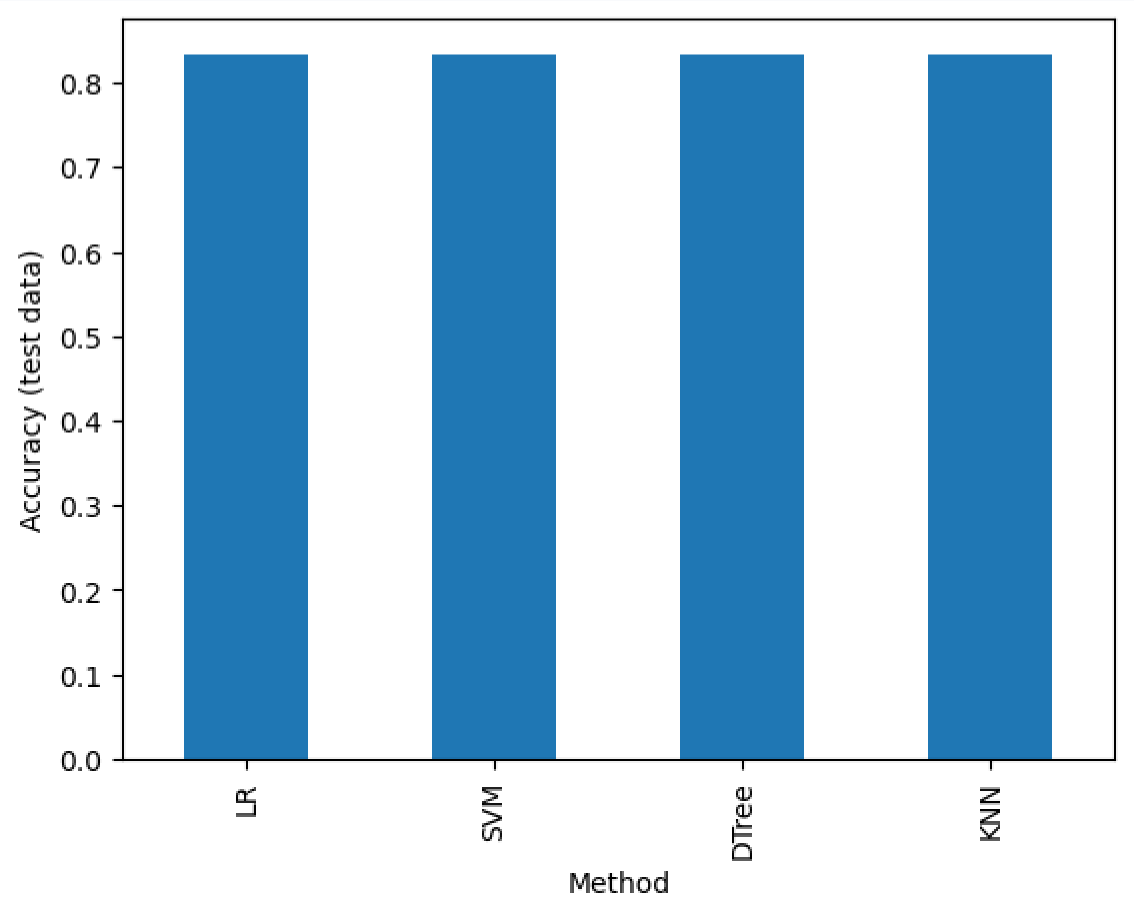


Section 5

Predictive Analysis (Classification)

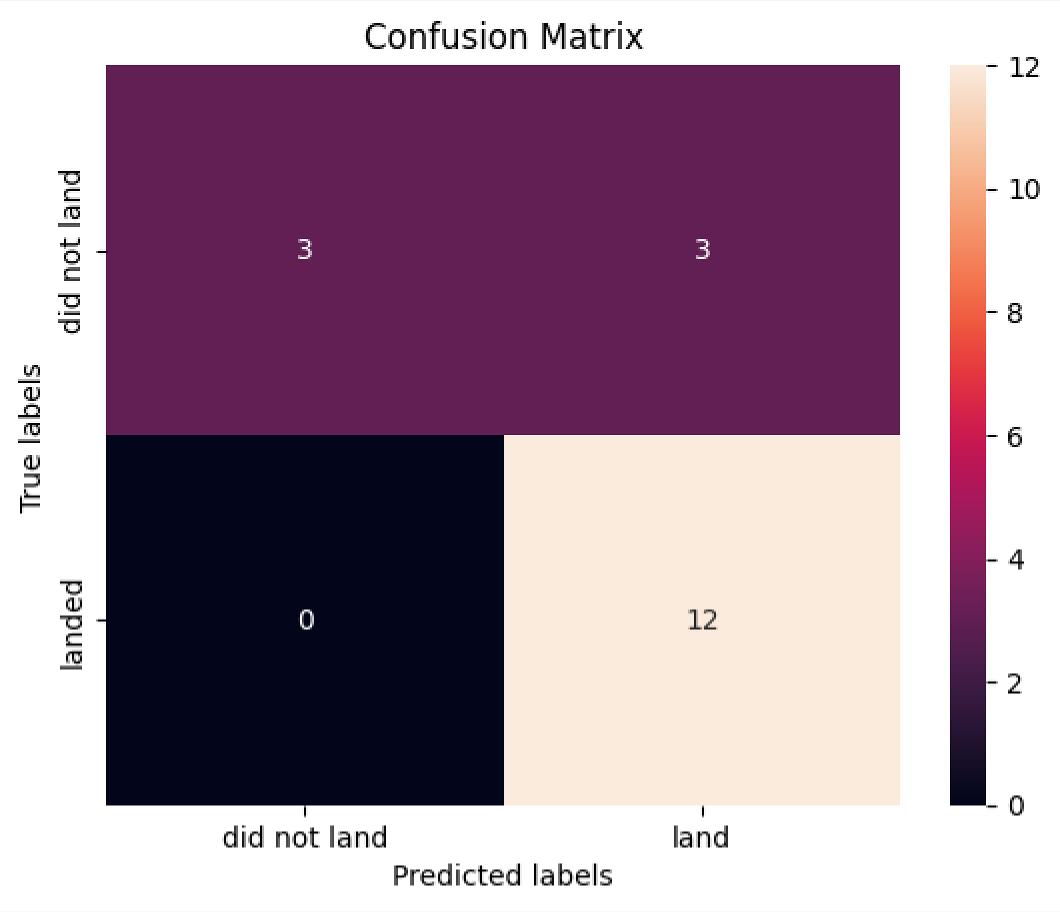
Classification Accuracy

- For the given dataset and train/test split, the optimized models perform with equal accuracy well on the test data



Confusion Matrix

- Shows counts of predicted labels for each category of the true label
- For the test set 3 times a successful landing was predicted falsely



Conclusions

- It’s possible to predict successful landings with 84 % accuracy from data on wiki / SpaceX rest

Appendix

- https://github.com/Xikero/10_IBM_capstone

Thank you!