IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

SpaceX Booster Landing Prediction for Competitive Launch Bidding

Daniel Nezich

September 15, 2021

Outline

Executive Summary

- Data summarizing SpaceX launch history and booster landing outcomes is visualized and used to train supervised learning models
- A Support Vector Machine model provides highest prediction accuracy of booster landing outcome (83.3%)

Introduction

- Commercial space launch companies are pursuing landing and re-use of first stage boosters (the largest and easiest to recover component in an orbital rocket) in order to decrease equipment cost and rocket construction/qualification lead time. These are two critical measures of efficiency and predictors of profitability in the space launch sector.
- In order to make accurate predictions of booster recovery for optimization of launch pricing we want to discover:
  - What details of the launch are predictive of booster recovery?
  - With what model can we best predict booster recovery?
  - With what accuracy can we predict booster recovery?

Business Case

- Launch pricing can be tailored to the mission for competitive pricing
  - Assume all launches are equivalent except for recovery
  - Let C_R be the launch cost with recovery and C_N be the launch cost without recovery
    - The savings due to recovery is $D = C_N - C_R$
  - Let R be the naïve rate of recovery
    - The average cost per launch is $C_{Avg} = C_N - RD = C_R + (1-R)D$
  - Let pricing be C_N if non-recovery is predicted and C_{PR} if recovery is predicted
    - Assume prediction with accuracy A and worst-case errors (all errors are false recoveries)
    - Cost penalty due to accuracy must be included in predicted recovery cost: $C_{PR} = C_R + D(1-A)$
    - The average cost with savings with predictive pricing for recovery is $E = C_{Avg} - C_{PR} \geq (A-R)D$
  - This study determines $R = 0.67$ and $A = 0.83$ such that $E \geq 0.16\ D$
    - Launch pricing for predicted recovery can be priced below the average launch cost by at least 16% of the recovery savings, which maximizes competitiveness for recovered launches at the cost of full price for predicted nonrecovered launches
  - The booster comprises ~60% of the launch cost, or ~$37 million [1], so $D = 0.60\ C_N$, C_N = $62 million
    - The cost for a predicted recovered launch can be lowered below average by a significant $E \geq 0.10\ C_N$ = $6.0 million
- The average launch cost for upcoming missions C_{Avg} can be accurately estimated using the booster recovery rate R predicted for a set of upcoming launches

[1] https://www.cnbc.com/2020/04/16/elon-musk-spacex-falcon-9-rocket-over-a-million-dollars-less-to-insure.html

Section 1

Methodology

Methodology

Executive Summary

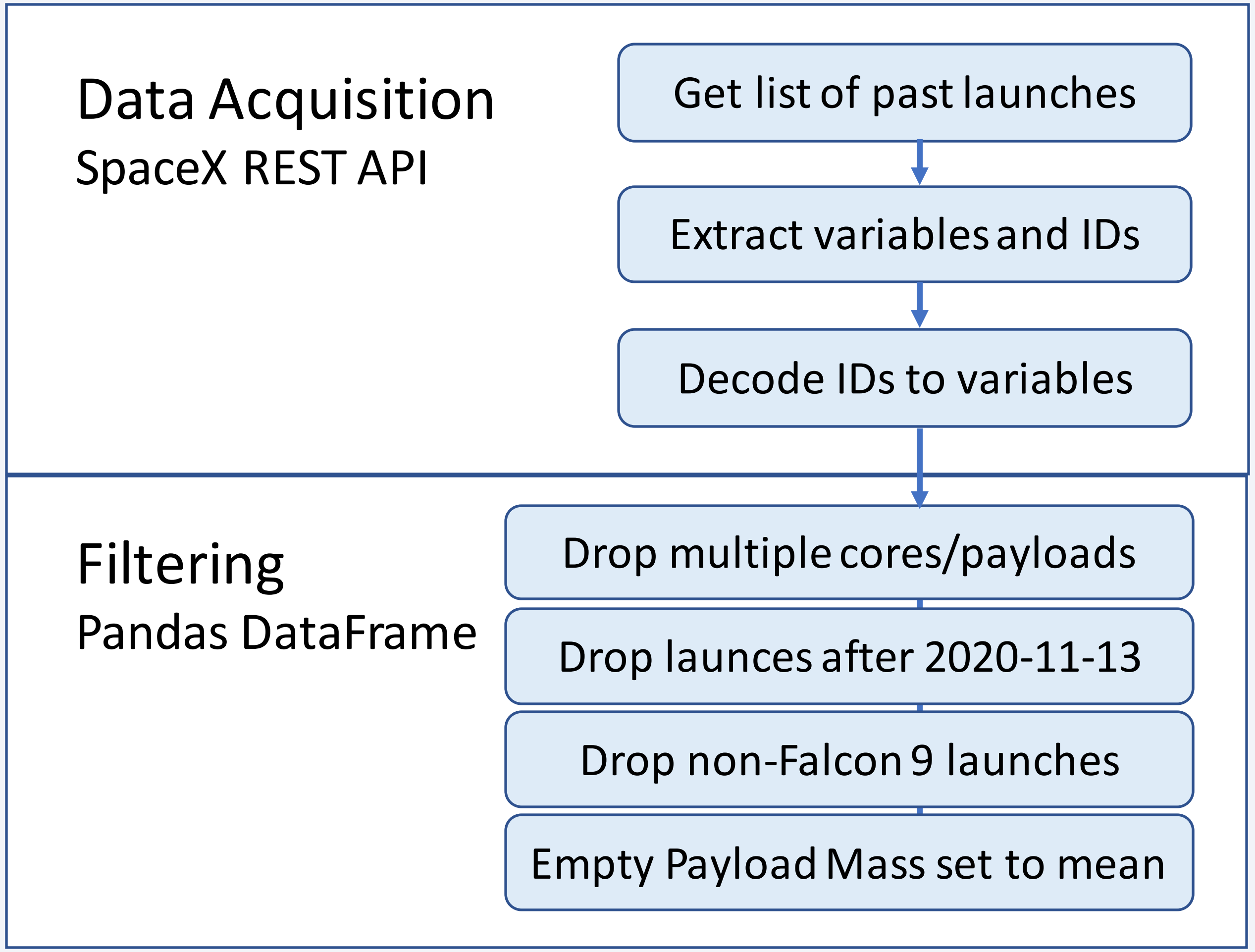
- Data collection methodology:
  - Use the SpaceX REST API and webscraping to collect a list of launches and launch parameters including booster landing outcomes
- Perform data wrangling
  - Missing values were replaced with the average, and only single-core single-payload Falcon 9 launches prior to 2020-11-13 are considered
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Use scikit-learn GridSearchCV to tune hyperparameters for Logistic Regression, Support Vector Machine, Decition Tree, and K-Nearest Neighbors models, optimizing for prediction accuracy

Data Collection

- Two datasets were collected, using different collection techniques:
  - Direct requests to the SpaceX REST API
  - Webscraping a Wikipedia page
- Collected data consists of:
  - A list of rocket flights and relevant flight parameters like date, rocket type, payload mass, etc.
  - Only single-core, single-payload Falcon 9 launches before mid-November 2020 were retained
- Data was sasved in a pandas DataFrame (and to file for reference)

Data Collection – SpaceX API

- Used *requests* to get data from SpaceX REST API endpoints including a list of past launches and launch parameters
    - https://api.spacexdata.com/v4/
        - launches/past
        - rockets/<Rocket_ID>
        - launchpads/<Launchpad_ID>
        - payloads/<Payload_ID>
        - cores/<Core_ID>
- Responses parsed into a *pandas* DataFrame (for all variables, see Appendix)



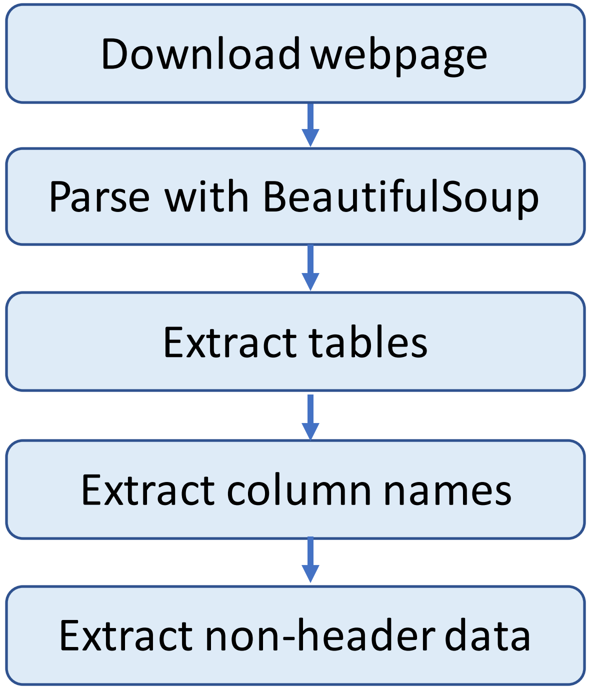
https://github.com/Arkadiatri/SpaceX-Prediction/blob/master/SpaceX_Data_Collection_API.ipynb

Data Collection - Scraping

- Used *requests* to download the Wikipedia page List of Falcon 9 and Falcon Heavy Launches
- Parsed webpage using BeautifulSoup
  - Extracted all tables detailing past launches
  - Extracted cell data with custom functions
- Data assembled into a *pandas* DataFrame (for all variables, see Appendix)

Data Acquisition

Webscraping



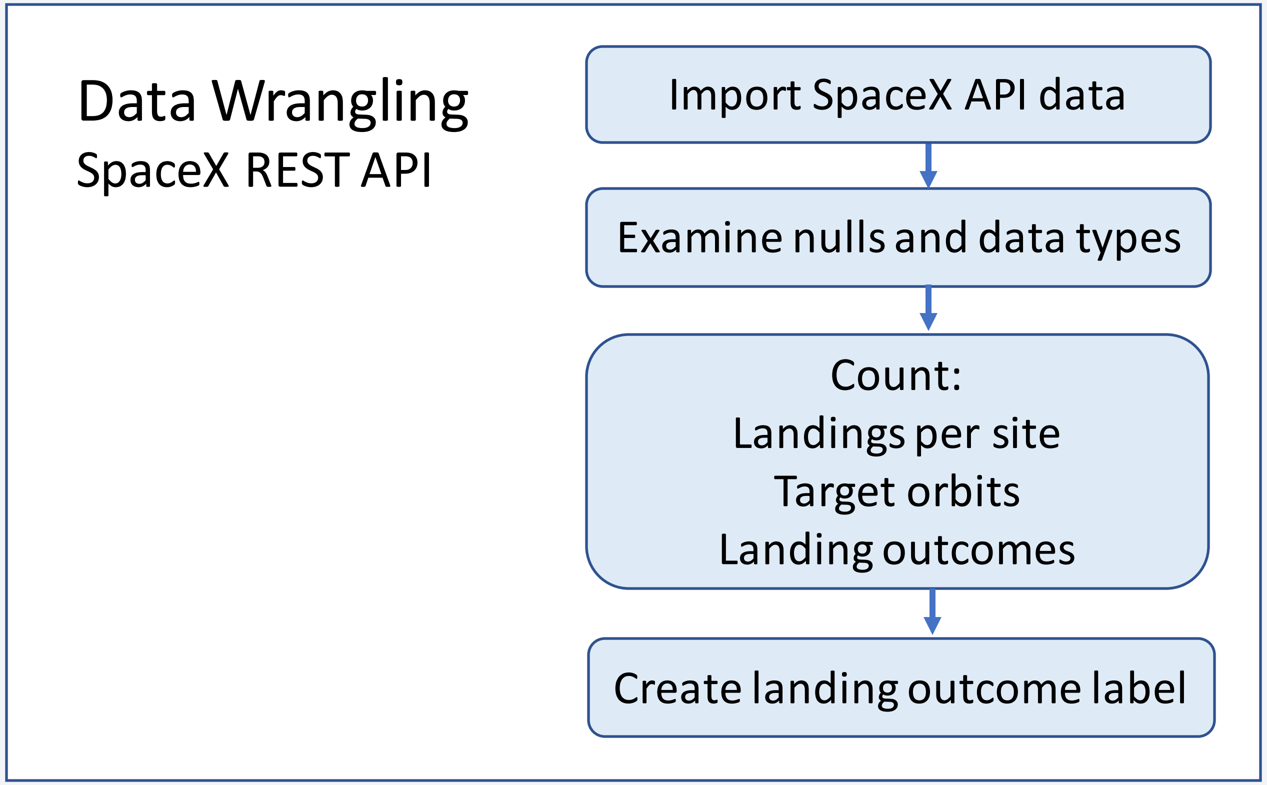
Filtering

None

https://github.com/Arkadiatri/SpaceX-Prediction/blob/master/SpaceX Web Scraping.ipynb

Data Wrangling

- Data from the SpaceX API was read into a *pandas* DataFrame and examined
- Successful landings were identified and used to create a label column
- Successful landing rate is 2/3



https://github.com/Arkadiatri/SpaceX-Prediction/blob/master/SpaceX Data Wrangling.ipynb

EDA with Data Visualization

- Examined categorical plots (*seaborn* catplot) labeled by landing outcome to visualize trends in data
  - Payload Mass vs. Flight Number: later flights with less mass are more likely to land
  - Launch Site vs. Flight Number: sites are preferenced during periods of time causing failure clustering
  - Launch Site vs. Payload Mass: payload mass differs between sites indicating biased mission profiles
  - Orbit Type vs. Flight Number: some orbits are biased toward later launch dates (successful landing)
  - Orbit Type vs. Payload Mass: heavier payloads increase low orbit success and decrease high orbit success
- Examined Landing Success Rate vs. Target Orbit: mass affects success differently by orbit
- Examined Landing Success Rate vs. Year: increase from 0 in 2013 to ~85% in 2020
- Created dummy-encoded feature dataframe using 12 relevant features (see Appendix for list)

https://github.com/Arkadiatri/SpaceX-Prediction/blob/master/SpaceX EDA with Visualization.ipynb

EDA with SQL

- Loaded Web Scraping results into an IBM Db2 database
- Ran SQL queries to the Db2 database in JupyterLab
  - Queries demonstrating basic SQL interrogation
    - E.g. Rank the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order

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

- Informational only, no results saved to file

https://github.com/Arkadiatri/SpaceX-Prediction/blob/master/SpaceX Exploratory Data Analysis.ipynb

Build an Interactive Map with Folium

- Created an interactive map with *folium* to display launch sites using the following elements:
  - Circle – For indicating a region of interest (launch sites)
  - Marker - For placing an item on the map:
    - DivIcon – For displaying arbitrary text/HTML, such as labels like location name or distance
    - Icon – For displaying a simple symbol (map pointer with circle of customizable color inside), like landing success coded by color
  - MarkerCluster – Plugin element for containing other elements (Markers), grouping them together when zoomed out for easy viewing
  - MousePosition – Plugin element for displaying the mouse coordinates in latitude and longitude, for locating regions of interest
  - PolyLine – Draws a line between coordinates, e.g. between launch site and region of interest
- End result: a map with circles and labels for each launch site, a MarkerCluster containing a Marker for each launch that is color coded to the landing success, lines and labels for the map from a launch site to the nearest railway, ocean, and city labeled with distances in kilometers, and a display for cursor latitude and longitude

https://github.com/Arkadiatri/SpaceX-Prediction/blob/master/SpaceX Interactive Visual Analytics.ipynb

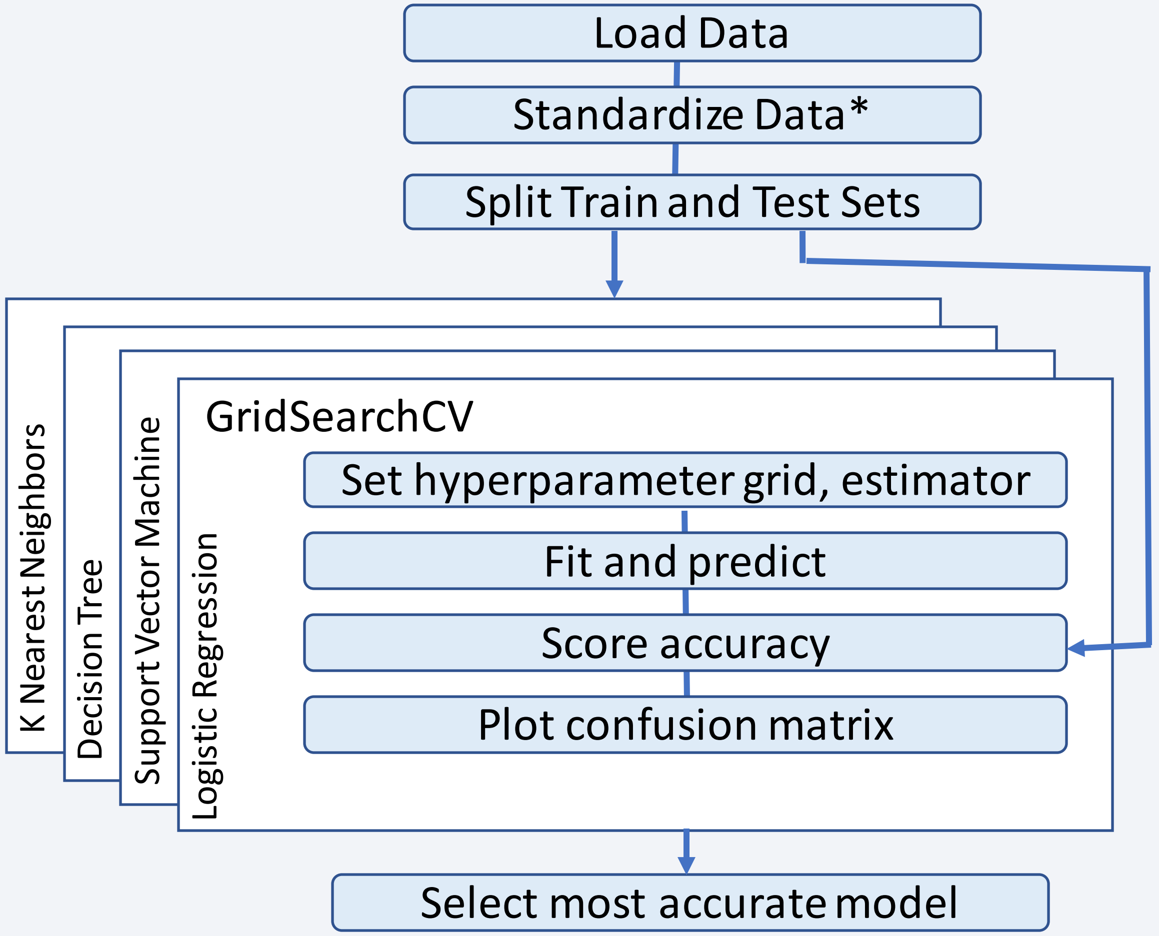
Build a Dashboard with Plotly Dash

- An interactive dashboard with the following elements was made in Plotly Dash:
  - Dropdown menu for selecting launch site for which to display data in the pie and scatter charts
  - Pie chart of landing successes by launch site, or landing outcomes if one launch site is selected
  - Range slider to select payload mass range for which to display data in the scatter chart
  - Scatter chart of landing outcomes vs. payload mass by booster version
  - Text for landing success rates and counts for all selected payload masses and breakdowns by booster versions
- These plots allow fast and intuitive exploration of the data and relations among variables

https://github.com/Arkadiatri/SpaceX-Prediction/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- Use features from EDA With Data Visualization section and class labels from Data Wrangling section, split into train and test sets (80-20 split)
- Run GridSearchCV using the training set and various estimators (logistic regression, support vector machine, decision tree, k-nearest neighbors) with appropriate hyperparameter grids
- Using tuned estimators with the highest training accuracy, compute the accuracy and plot the confusion matrix for the test set
- Select the tuned estimator with the highest test set accuracy as the final model



https://github.com/Arkadiatri/SpaceX-Prediction/blob/master/SpaceX Machine Learning Prediction.ipynb

* Standardization should properly be done after test/train splitting

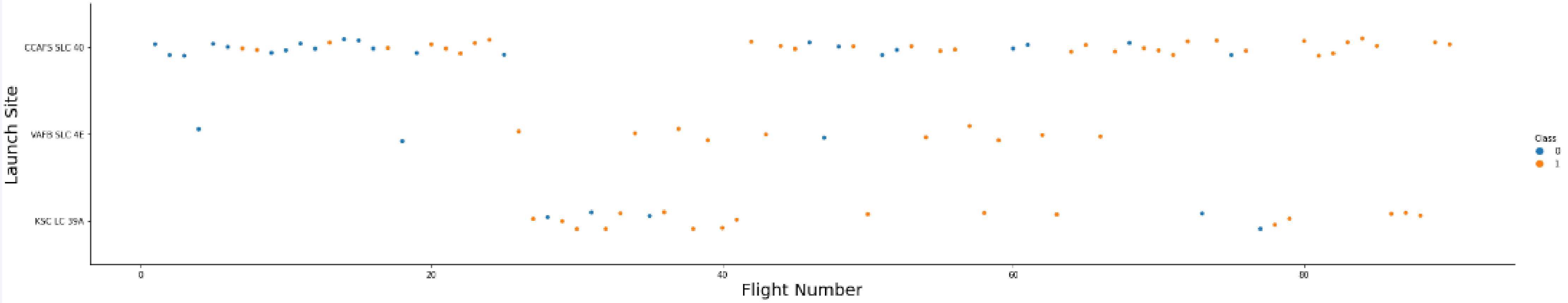
Results

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

Section 2

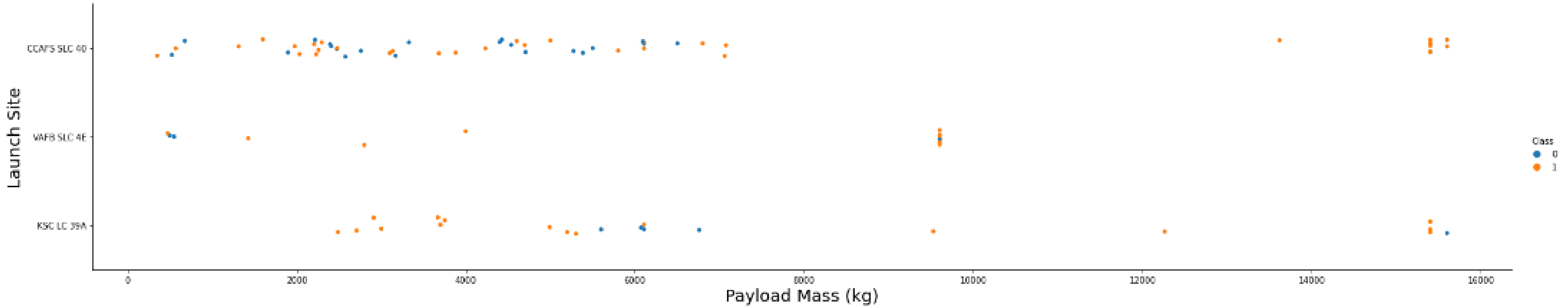
Insights drawn from EDA

Flight Number vs. Launch Site



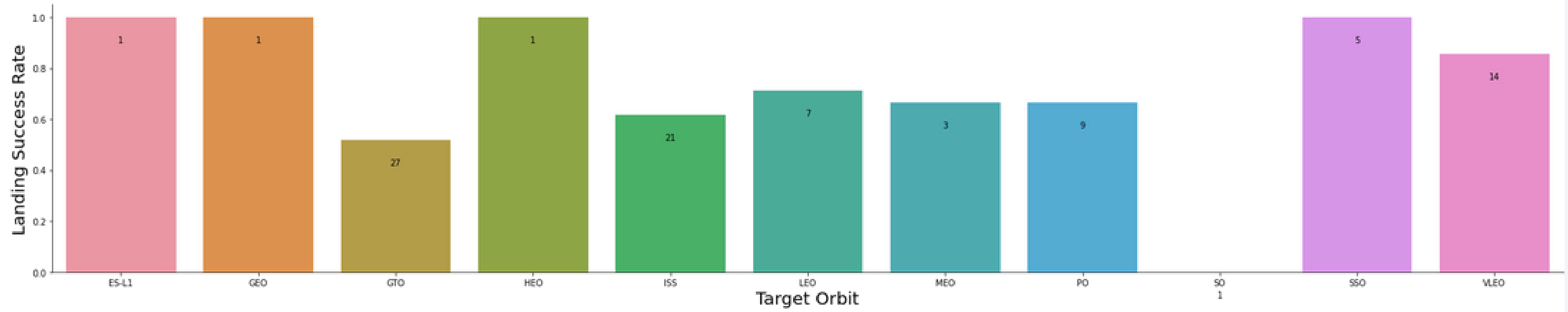
Missions are distributed among the launch sites non-uniformly in time; early launches, which are more likely to fail, occurred primarily at CCAFS SLC 40, hence the larger failure rate. However, this does not explain the root cause.

Payload vs. Launch Site



High mass payloads are equally likely to fail at each of CCAFS SCL 40 and KSC CL 39A from which they are launched. Medium mass payloads are relatively successful and launched primarily from WAFB SLC 4E. Low mass payloads, which we suspect comprise the bulk of test/development flights, are launched primarily from CCAFS SLC 40, with VAFB SLC 4E as an initial secondary site (probably mission-selected) before it was relegated to a medium-mass launch site as KSC LC 39A was developed as a secondary launch site for low and high mass payloads.

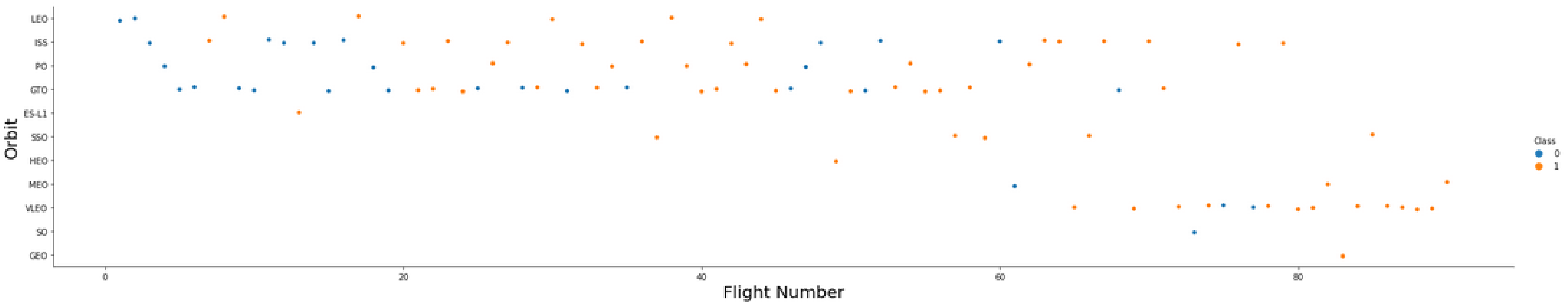
Success Rate vs. Orbit Type



Very low and very high orbits appear most successful, with the exception of SO and GTO, though statistics are too low for the highest orbits. More significant is the trand with target height: GTO is highest and has lowest success probability, followed by MEO then LEO then VLEO for which the probabilities of success increase. The ISS is in LEO, though, so this counteracts the trend somewhat. Having access to a database of orbital parameters would allow better evaluation of this hypothesis.

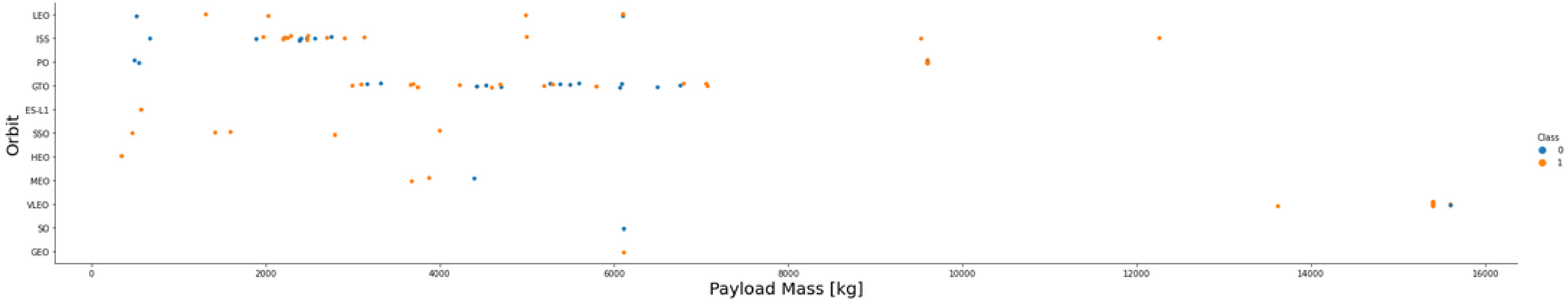
Numbers on bars indicate support

Flight Number vs. Orbit Type



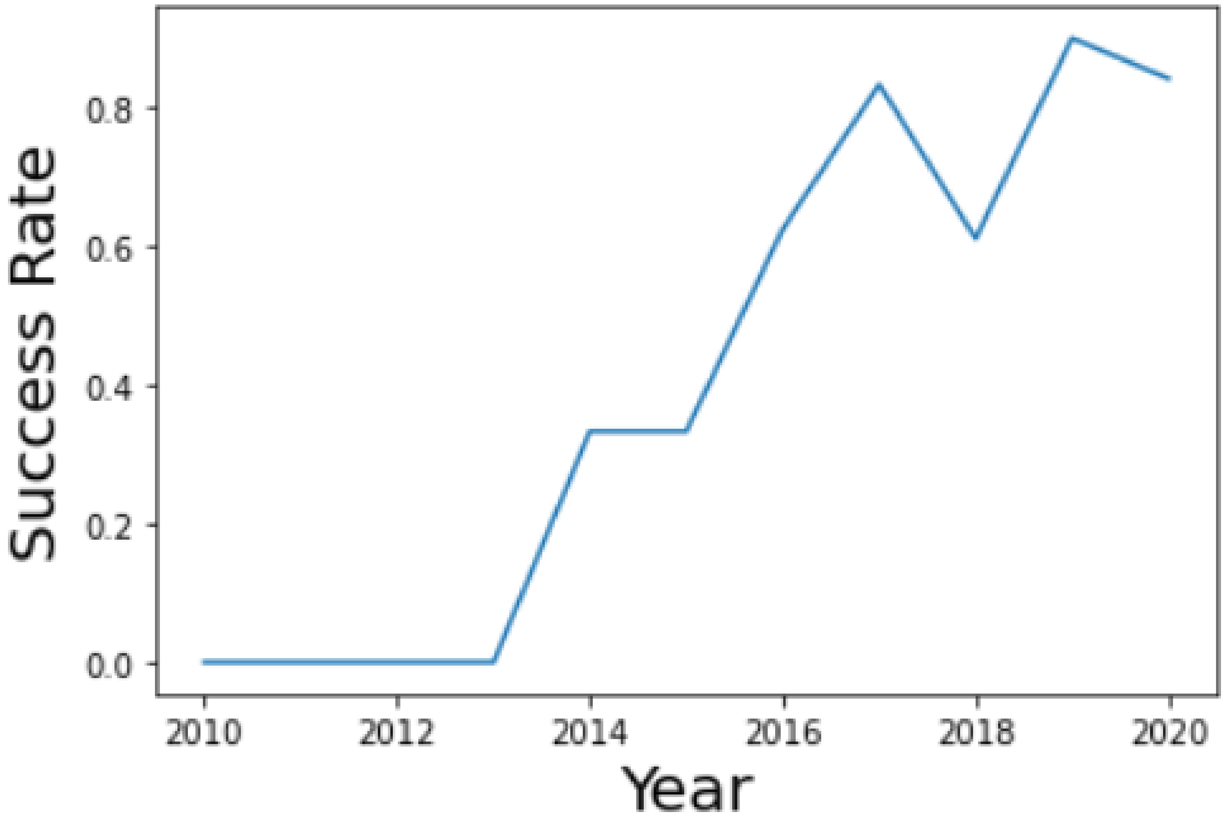
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



You should observe that Heavy payloads have a negative influence on GTO orbits and positive on LEO (ISS), Polar orbits.

Launch Success Yearly Trend



You can observe that the sucess rate since 2013 kept increasing till 2020.

All Launch Site Names

```
%sql SELECT DISTINCT launch_site from spacextbl
```

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

- There are four unique launch site names

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM spacextbl WHERE launch_site LIKE 'CCA%' LIMIT 5
```

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

- We can retrieve only entries for launches at Cape Canaveral Air Force Station (now Space Force Station)

Total Payload Mass

```
%sql SELECT SUM(payload_mass__kg_) FROM spacextbl WHERE customer='NASA (CRS)'
```

1
45596

- SpaceX has launched over 45 metric tons to the ISS for NASA

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(payload_mass__kg_) FROM spacextbl WHERE booster_version LIKE 'F9 v1.1%'
```

1
2534

- The average Falcon 9 v1.1 launch carries 2.5 metric tons to orbit

First Successful Ground Landing Date

```
%sql SELECT MIN(date) FROM spacextbl WHERE landing__outcome LIKE 'Success%(ground pad)%'
```

1
2015-12-22

- The first successful ground landing of a booster was on December 22nd, 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT DISTINCT booster_version FROM spacextbl WHERE landing__outcome LIKE 'Success%drone ship%' AND payload_mass__kg_ BETWEEN 4000 AND 6000
```

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

- Four boosters have successfully landed on a drone ship after carrying a 4-6 metric ton payload

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT SUM(CASE WHEN mission_outcome LIKE 'Success%' THEN 1 ELSE 0 END) AS Success, SUM(CASE WHEN mission_outcome LIKE 'Failure%' THEN 1 ELSE 0 END) AS Failure FROM spacextbl
```

success	failure
100	1

- Falcon 9 boosters have a high mission success rate of 99%

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT booster_version FROM spacextbl WHERE payload_mass__kg_ IN (SELECT MAX(payload_mass__kg_) FROM spacextbl)
```

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

- Six boosters have carried the maximum payload on a total of 12 launches

2015 Failed Drone Ship Landings

```
%sql SELECT landing__outcome, booster_version, launch_site FROM spacextbl WHERE landing__outcome='Failure (drone ship)' AND YEAR(date)=2015
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- There were only two failed drone ship landings in 2015

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

```
%sql SELECT landing__outcome, COUNT(*) AS count FROM spacextbl WHERE date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing__outcome ORDER BY count DESC
```

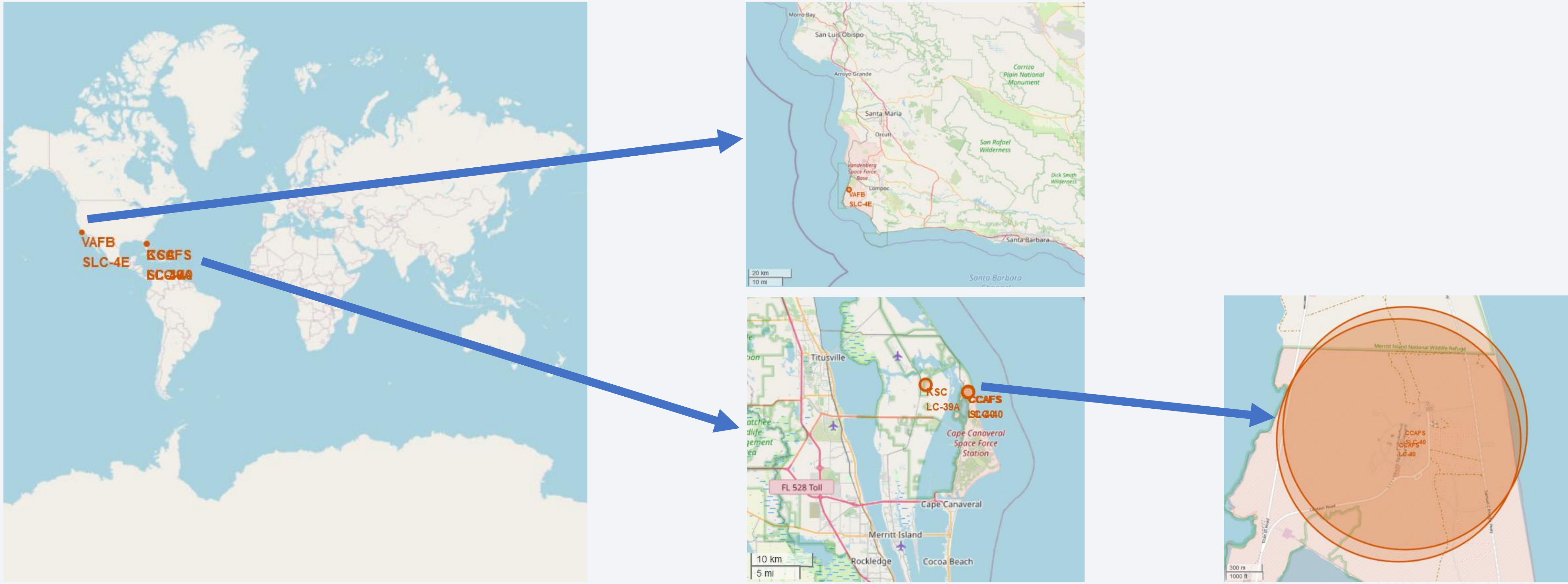
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- The most frequent landing outcome was no attempt (likely due to mission parameters), and drone ship landing were 50% successful in this time period

Section 4

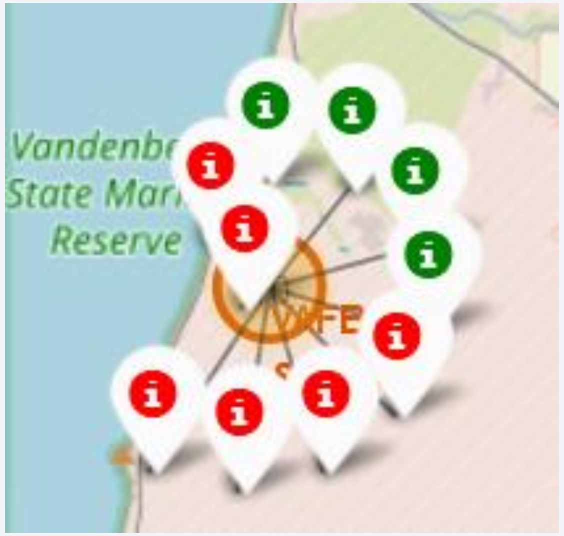
Launch Sites Proximities Analysis

Launch Site Locations

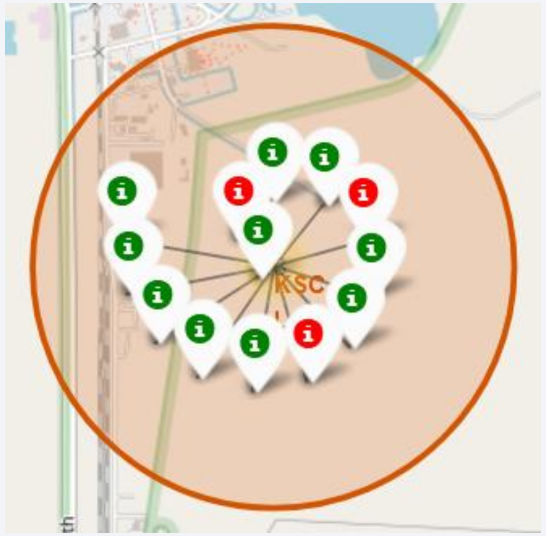


- There are four launch sites total close to the equator on either coast of the continental United States; one in California and three in Cape Canaveral, Florida

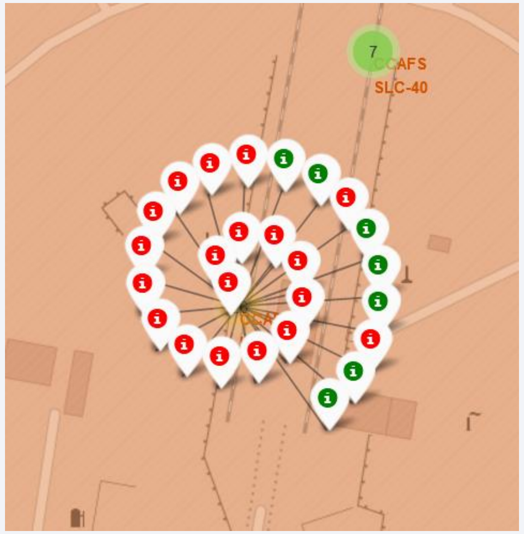
Landing Outcome Visualization



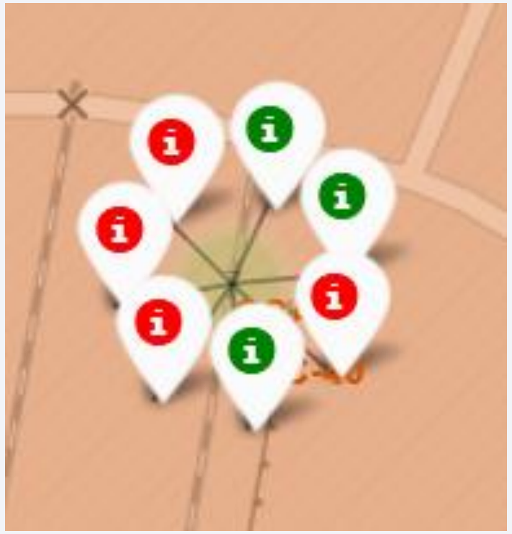
VAFB SLC-4E



KSC LC-39A



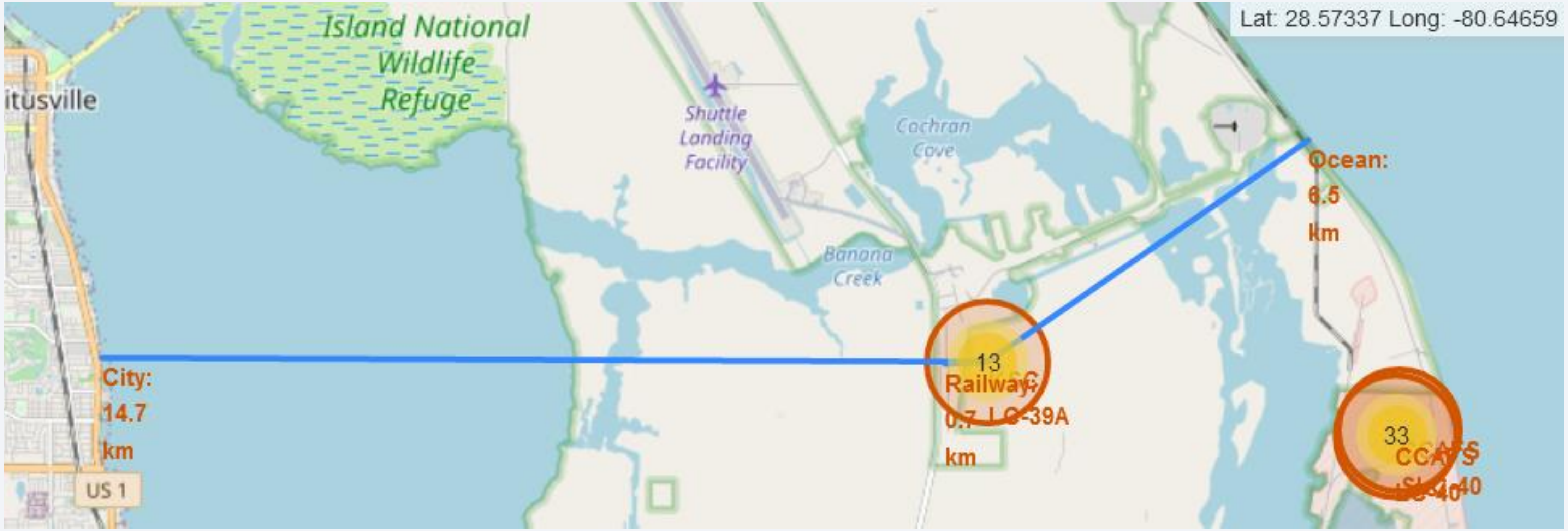
CCAFS LC-40



CCAFS SLC-40

- One marker per launch, green (red) is an (un)successful booseter landing
- Highest booster recovery rate at KSC LC-39A, lowest at CCAFS LC-40

Launch Site Proximities

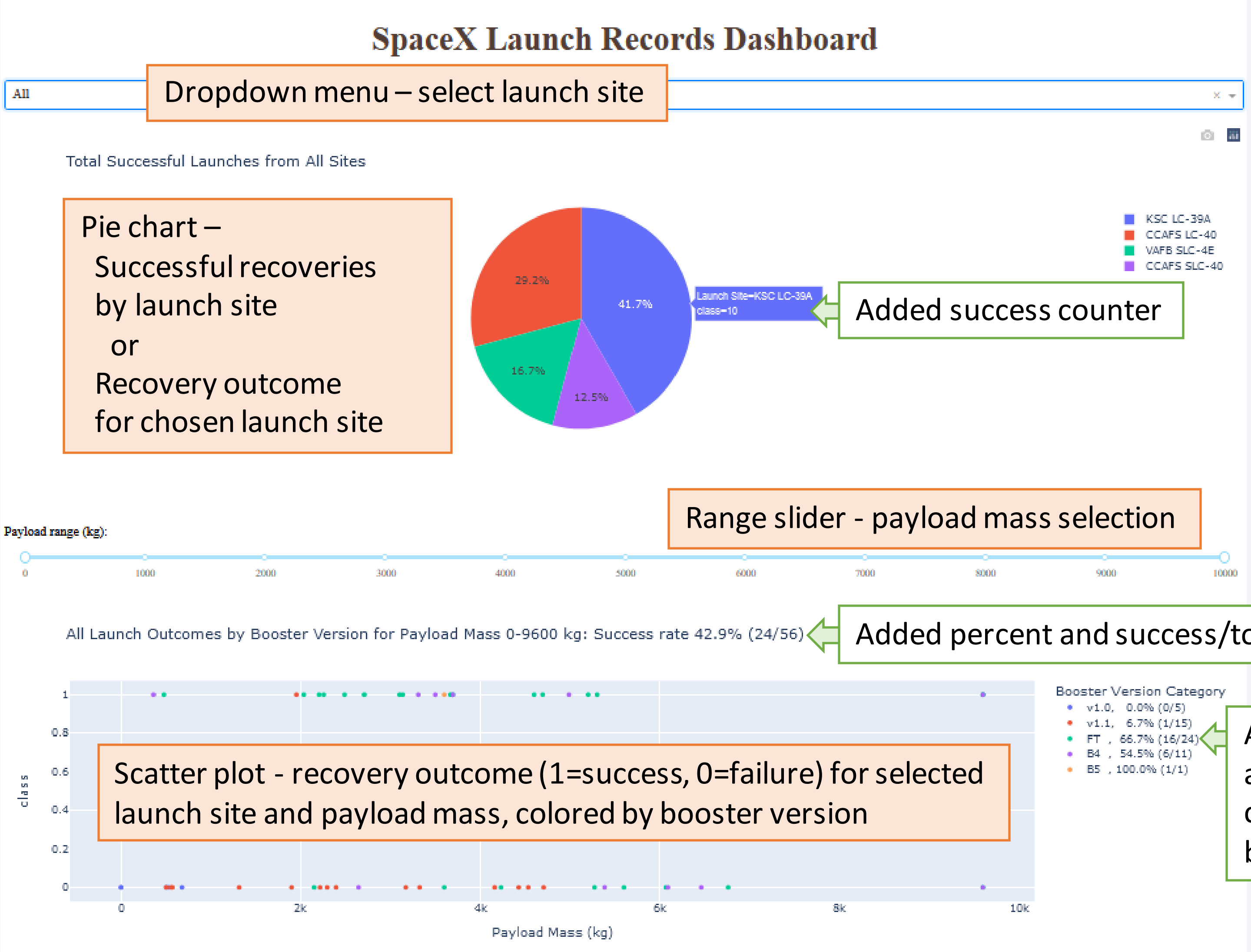


- Proximal features for KSC LC-39A:
  - Railroad 0.7 km, Ocean 6.5 km, City 14.7 km

Section 5

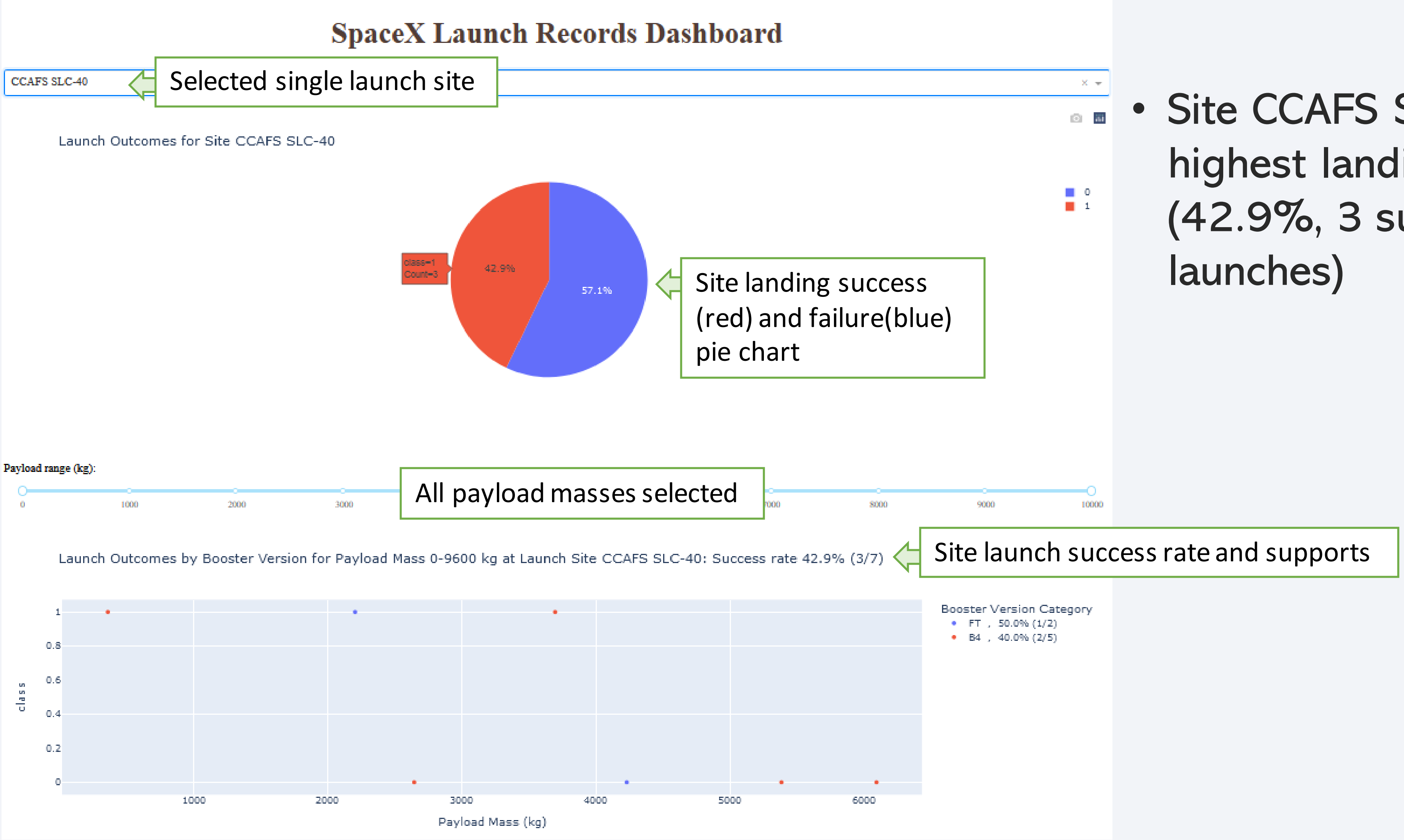
Build a Dashboard with Plotly Dash

Dashboard – All Launch Sites



- KSC LC-39A has the highest number of successful booster recoveries (10, highlighted)
- FT (v1.2) boosters have the highest successful landing rate (66.7%) of versions with more than one launch (B5 has 100% successful landings, but only one launch)

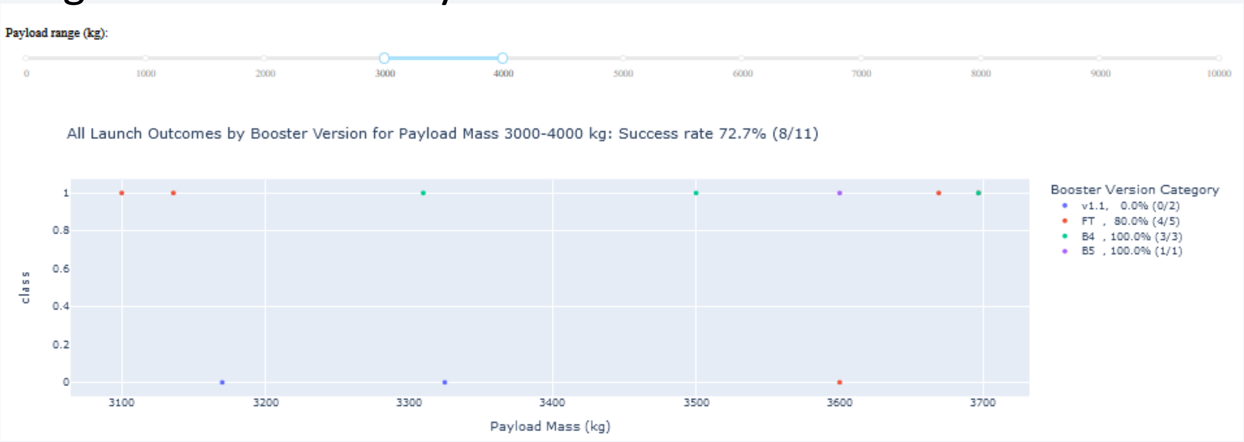
Dashboard – Most Successful Launch Site



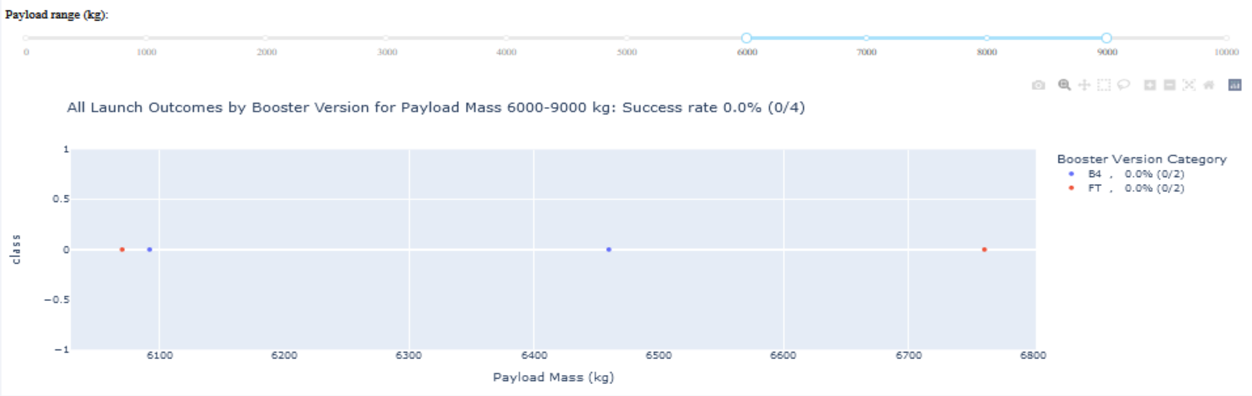
- Site CCAFS SLC-40 has the highest landing success rate (42.9%, 3 successes of 7 launches)

Dashboard – Payload Mass Effect

Highest: 72.7% for Payload Mass 3-4 tons



Lowest: 0% for Payload Mass 6-9 tons



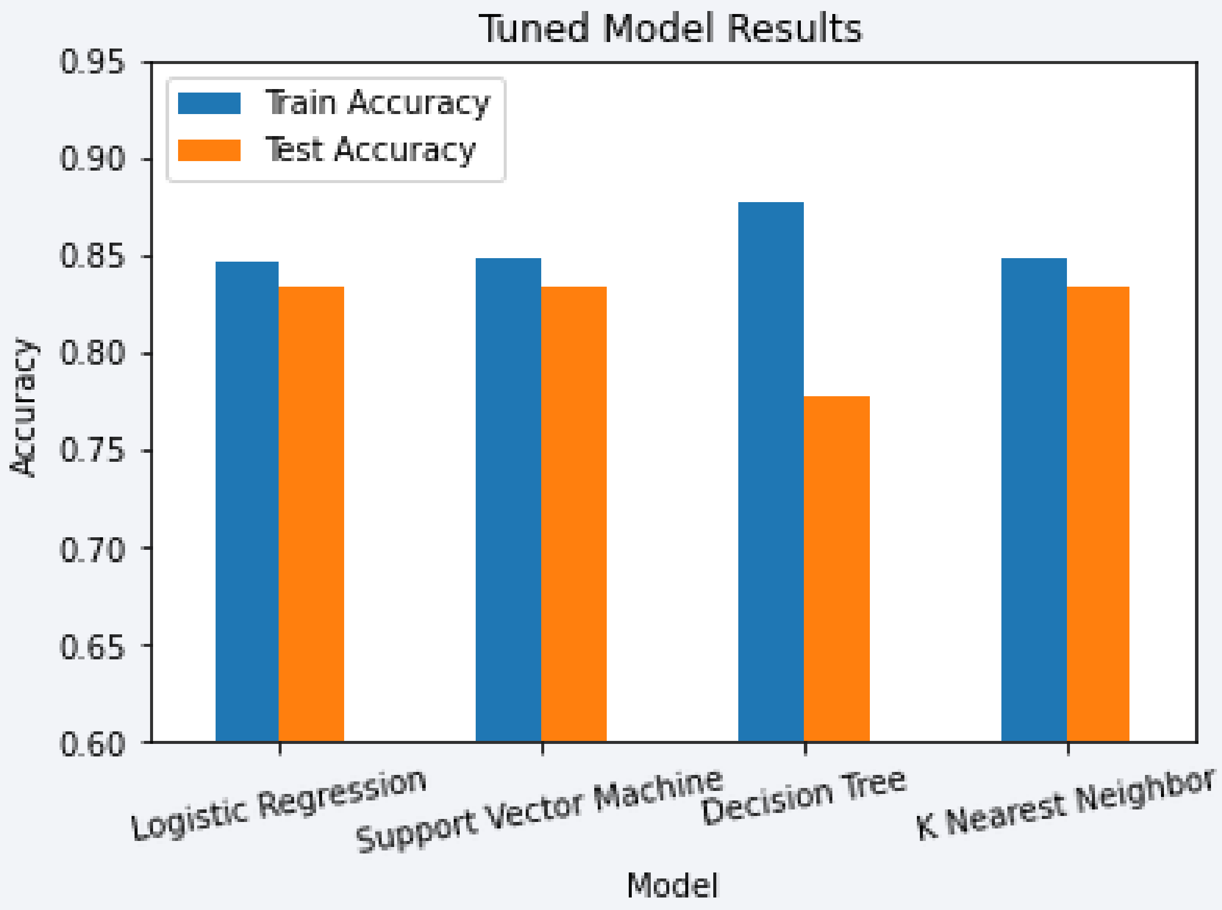
- Among all sites, in payload mass ranges of 1 metric ton, 3-4 ton payloads have the highest landing success rate and 6-9 ton payloads have the lowest landing success rate
- Note there are no launches in the 7-9 ton payload mass window
- These results can be seen by visually windowing the All Launch Sites scatter plot, but using the slider computes the actual success/total counts and success rate

Section 6

Predictive Analysis (Classification)

Classification Accuracy

- Surprisingly, three models have the same test accuracy and are equally optimal estimators:
  - Logistic Regression
  - Support Vector Machine
  - K Nearest Neighbor

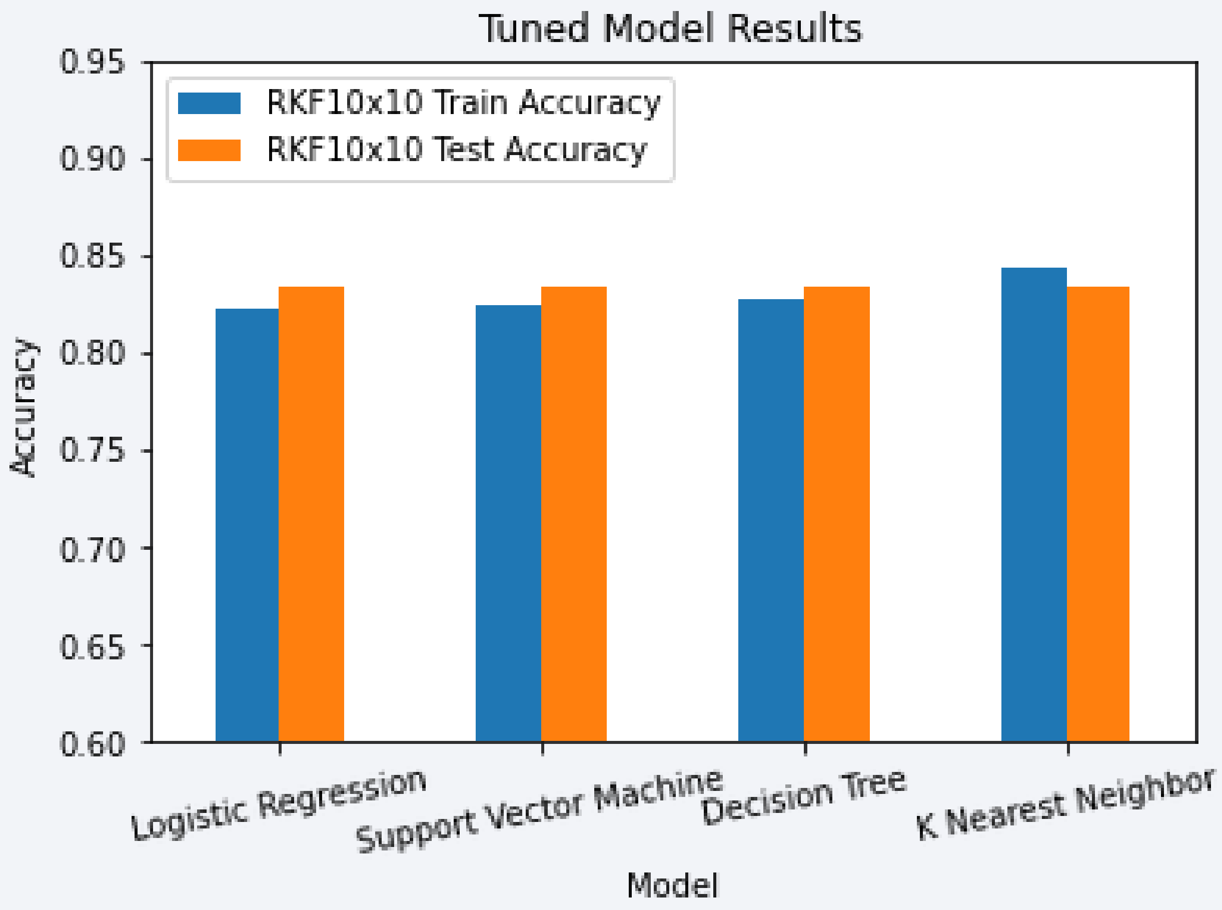


Confusion Matrix

- All three identically predictive models have the same confusion matrix
- False positive predictions are the primary error

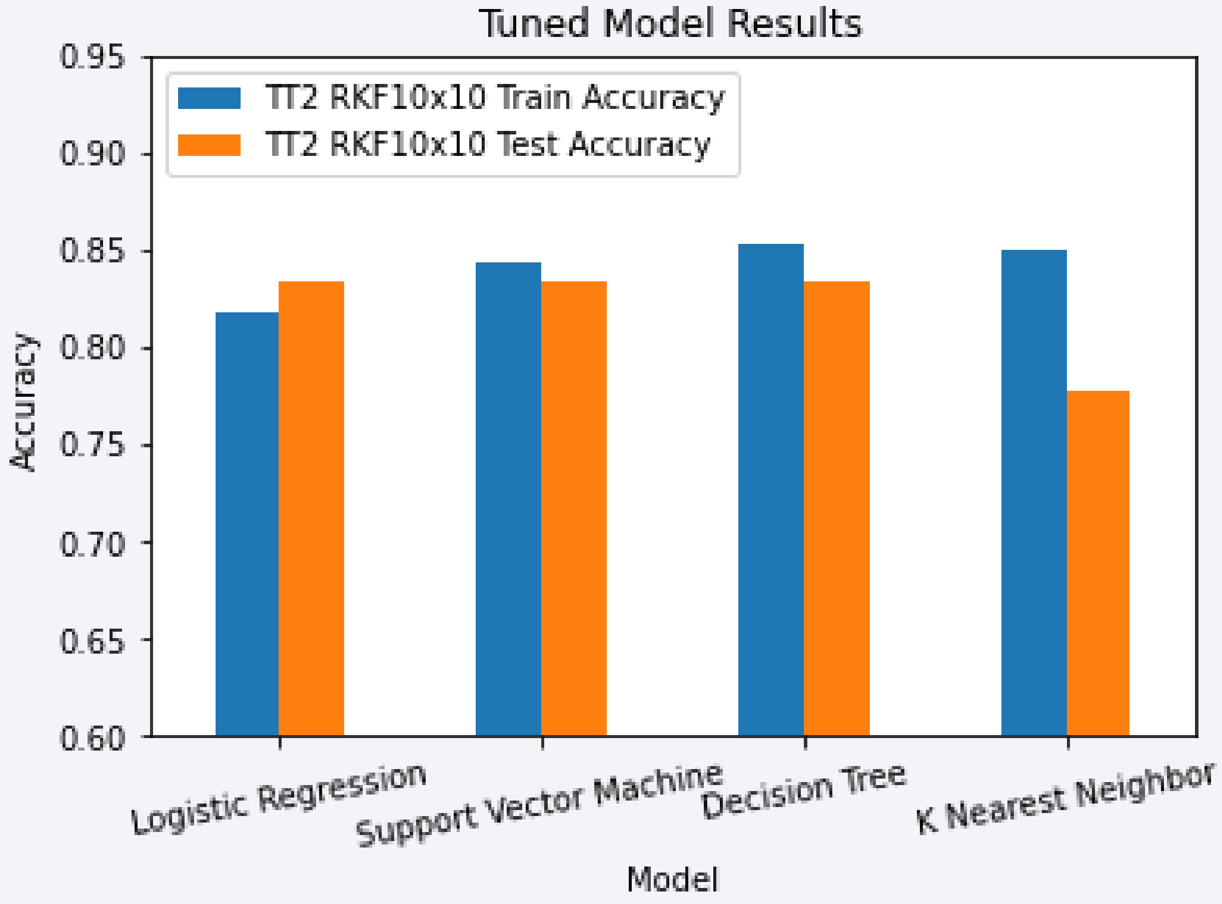
Extension – Repeated K-Fold Cross-Validation

- Taking the best models from 10 repeats of the previous 10-fold cross-validation approach results in identical test accuracy for all estimators
- Confusion matrix for all models is now identical (as shown previously)



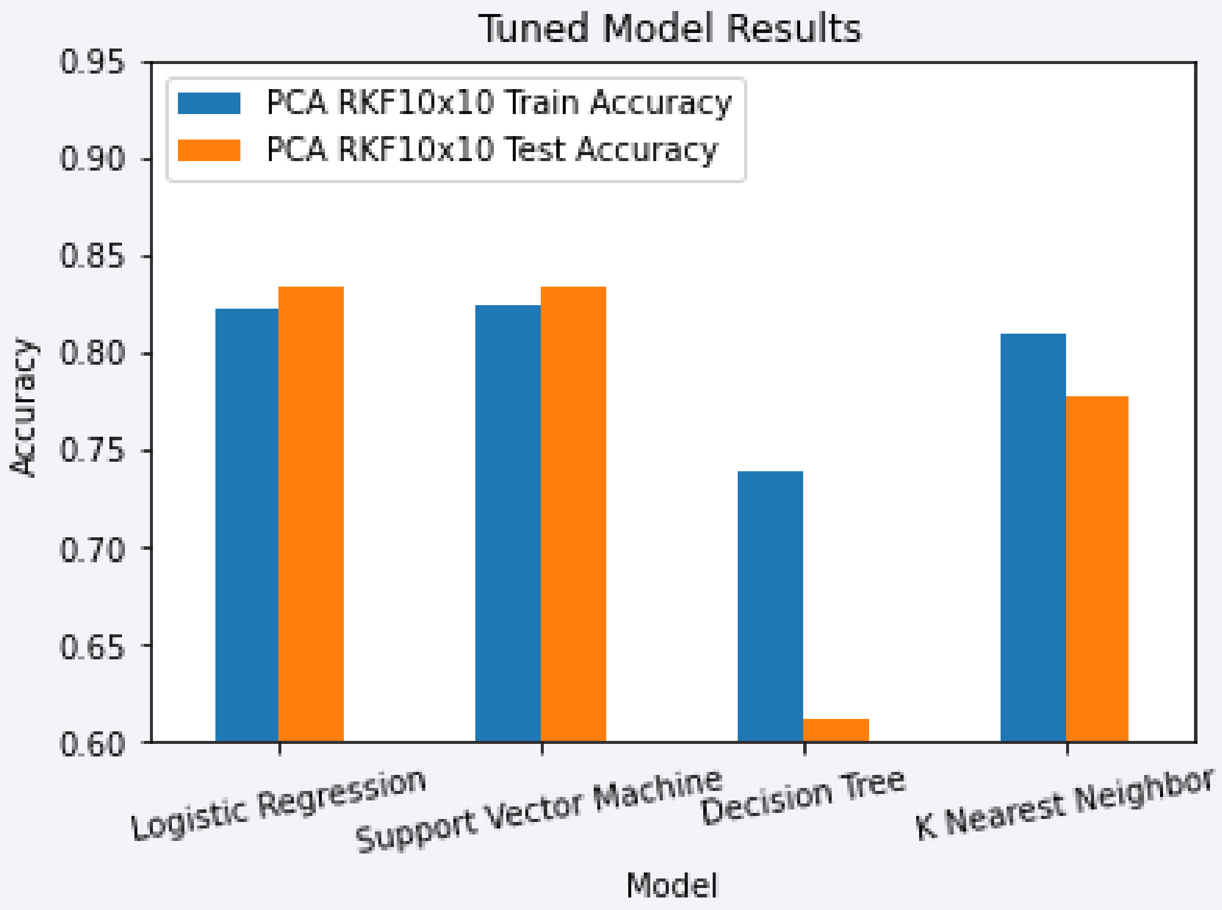
Extension – Alternate Train/Test Split

- An alternate train/test split randomization leads to different fitting results
- Confusion matrices are altered in distribution of errors among false positives and false negatives



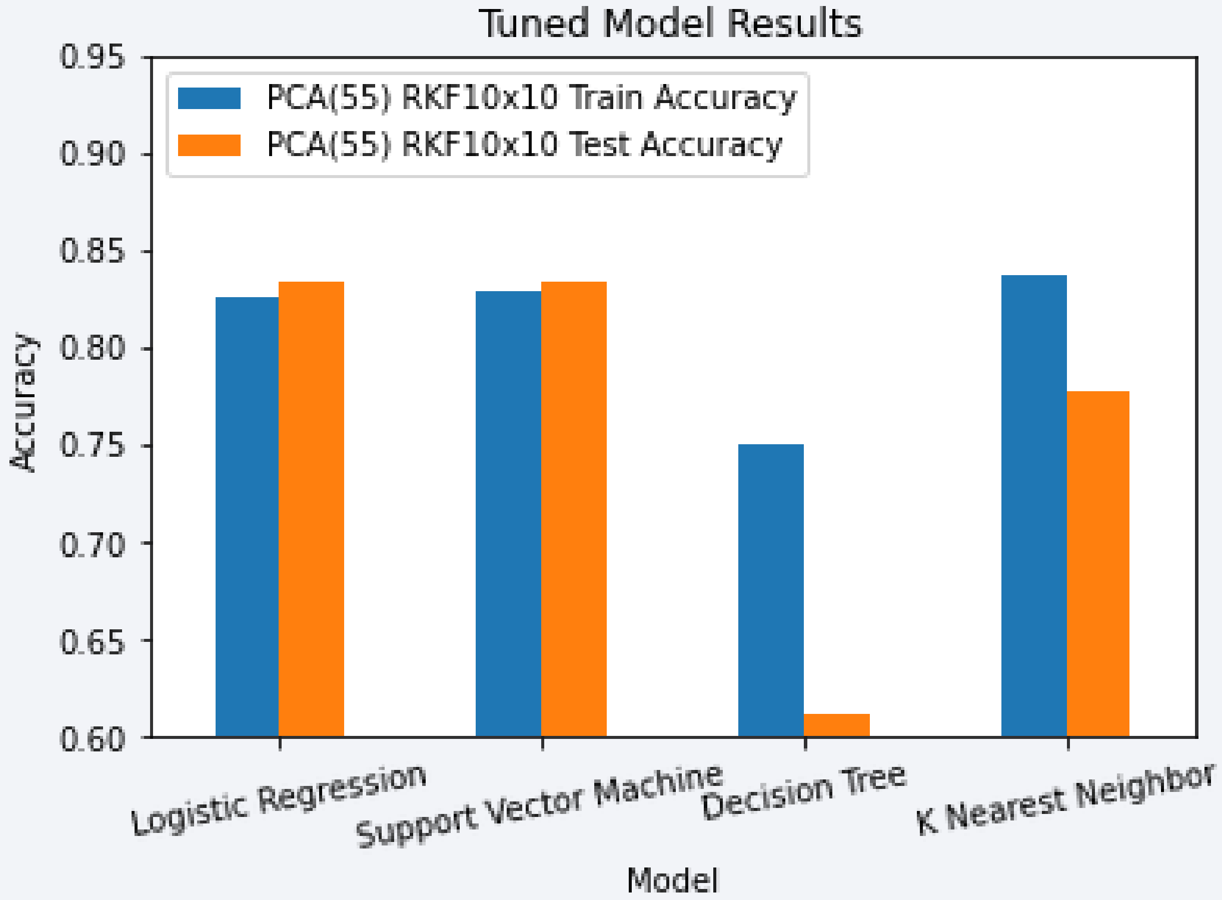
Extension – Feature Importance by PCA (83)

- Principal Component Analysis with all 83 components produces very similar results, with no improvement in maximum test accuracy
- Decision Tree performance is degraded since information is clustered in fewer features and overfitting is likely to low-information features



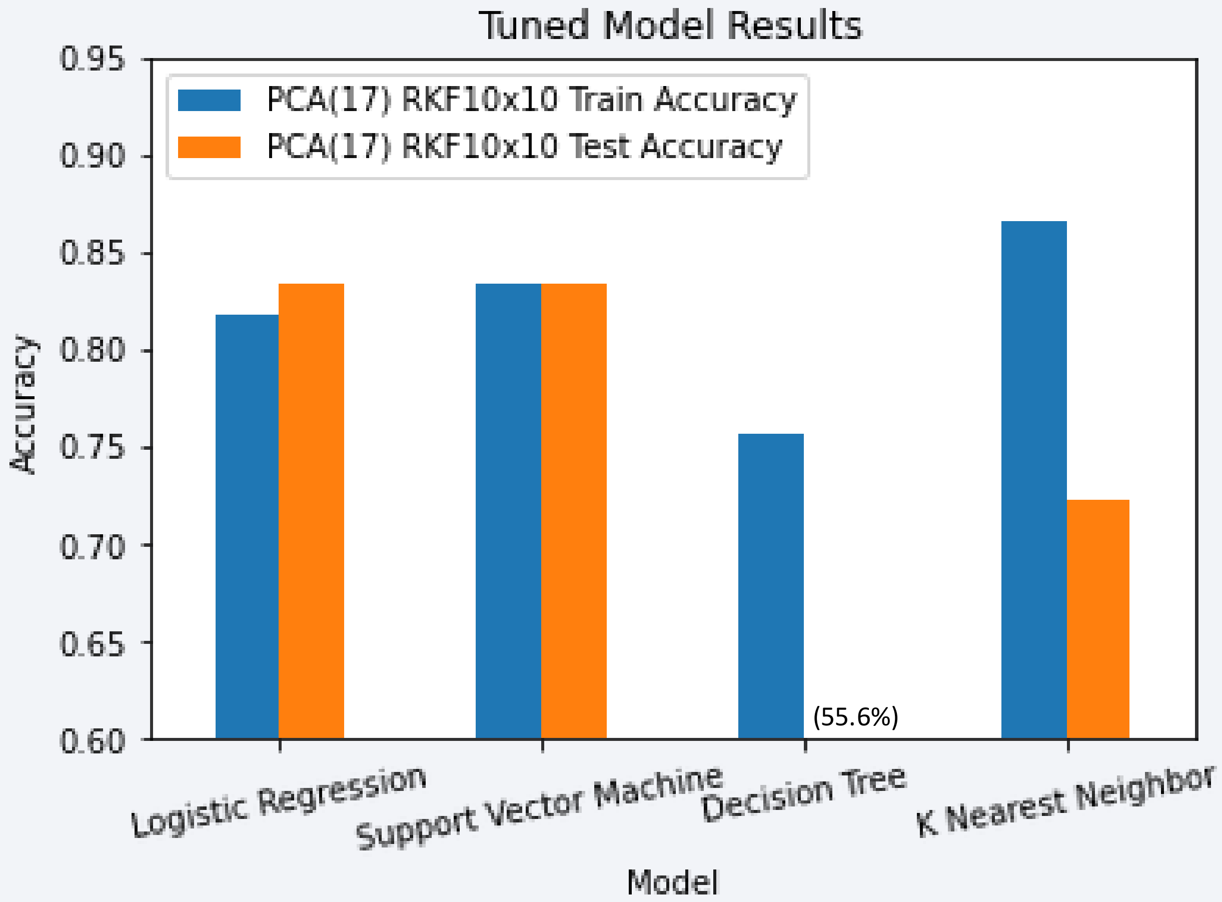
Extension – Feature Importance by PCA (55)

- Principal Component Analysis with top 55 retained components produces very similar results, with no improvement in maximum test accuracy
- K Nearest Neighbor model now suffers an accuracy decrease in addition to Decision Tree



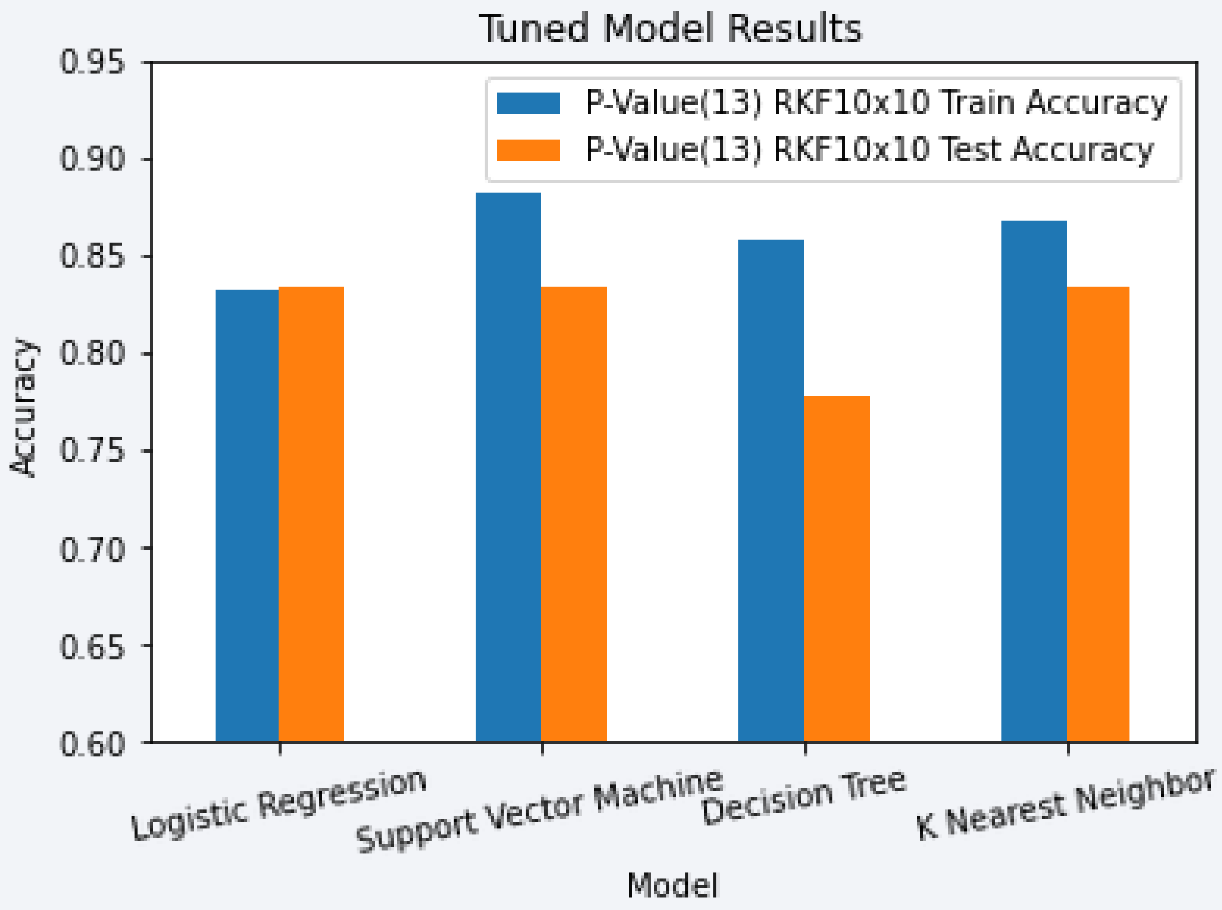
Extension – Feature Importance by PCA (17)

- Principal Component Analysis with top 17 components produces very similar results, with no improvement in maximum test accuracy
- K Nearest Neighbor model now suffers an accuracy decrease
- Curiously, Decision Tree performs even worse



Extension – Feature Importance by P-Value (13)

- Restricting features to those with p-values above 0.05 (13 total) produces improved training accuracy, with no improvement in maximum test accuracy
- Decision Tree performs worse compared to the repeated k-fold cross-validation seen previously



Conclusions

- A support vector machine is the recommended booster recovery prediction model
    - Consistently achieves highest test accuracy, supported by highest train accuracy
    - A logistic regression model is a test accuracy peer with lower train accuracy but faster training
- Booster recovery can be predicted with 83.3% accuracy
    - Errors are primarily false positives
    - Improvement over:
        - Random guessing 50%
        - Always predict majority class (recovery) 66.7%
    - SpaceX can reduce predicted recovery launch prices by 16% of the recovery savings below the average launch cost in order to provide an extremely competitive bid

Acknowledgements

- Contributors to StackOverflow and forums for python libraries used in this project
- IBM Data Science Professional Certificate Coursera course contributors
  - Primary Instructors: Joseph Santarcangelo, Yan Luo
  - Other Contributors & Staff
    - Project Lead: Rav Ahuja
    - Instructional Designer: Lakshmi Holla
    - Lab Authors: Joseph Santarcangelo, Yan Luo, Azim Hirjani, Lakshmi Holla
    - Technical Advisor: Yan Luo
  - Production Team
    - Publishing: Grace Barker, Rachael Jones
    - Project Coordinators: Kathleen Bergner
    - Narration: Bella West
    - Video Production: Simer Preet, Lauren Hall, Hunter Bay, Tanya Singh, Om Singh
  - Teaching Assistants and Forum Moderators
    - Malika Singla
    - Duvvana Mrutyunjaya Naidu

Appendix

- Data Collection – SpaceX REST API:
  - https://api.spacexdata.com/v4/launches/past for Flight Number, Date, Core, Rocket ID, Payload ID, Launch Pad ID
  - Core dict extracted to obtain: Core ID, Outcome, Flights, Grid Fins, Reused, Legs, Landing Pad
  - https://api.spacexdata.com/v4/rockets/ for Rocket ID to Booser Version
  - https://api.spacexdata.com/v4/launchpads/ for Launch Pad ID to Latitude, Longitude, Launch Site
  - https://api.spacexdata.com/v4/payloads/ for Payload ID to Payload Mass, Orbit
  - https://api.spacexdata.com/v4/cores/ for Core ID to Block, Reused Count, Serial
  - Saved to dataset_part_1.csv
- Data Collection – Web Scraping:
  - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
  - Parsed to obtain: Flight Number, Launch Site, Payload, Payload Mass, Orbit, Customer, Launch Outcome, Booser Version, Booster Landing, Date, Time
  - Saved to spacex_web_scraped.csv

Appendix

- Data Wrangling
  - Begin from dataset_part_1.csv
  - Check missing values and data types
  - Count number of launches at each site
  - Count number or each orbit types
  - Count number of landing outcomes 'Outcome' col
  - Create list of bad outcomes (lead value in Outcome is False or None)
  - Create label column 'Class'
  - 2/3 successful return rate
  - Save results to file dataset_part_2.csv

Appendix

- EDA With Visualization (Plots)
  - Started with dataset_part_2.csv
  - Features selected: FlightNumber, PayloadMass, Orbit, LaunchSite, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial
  - Dummy-encoded features: Orbit, LaunchSite, LandingPad, Serial
  - All features cast as Float
  - Save results to dataset_part_3.csv

Appendix

- Exploratory Data Analysis (SQL)
    - Begin from Spacex.csv (similar to SpaceX API dataset, but with more entries and some typos)
    - Upload data file to IBM Db2 SQL server
    - Connect to Db2 server through JupyterLab using *ibm_db_sa*, *ipython-sql*, and *sqlalchemy*
    - Run SQL queries to identify:
        - Unique launch sites
        - 5 records where the launch site begins with 'CCA'
        - Total payload mass with customer 'NASA (CRS)
        - Average payload mass carried by booser version F9 v1.1
        - Date of first successful booster landing on a ground pad
        - Names of boosters with successful drone ship landing and payload mass between 4000 and 6000 kg
        - Total number of successful and failure mission outcomes
        - Booster versions that have carried the maximum payload
        - List failed drone ship landings with booster version, launch site, and landing outcome in 2015
        - Rank the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order

Appendix

- Interactive Visual Analytics (Folium)
  - Begin from spacex_launch_geo.csv (apparently a variant of the SpaceX API data)
  - Analyze a subset of the data – launch site, latitude, longitude, and class
  - Circle – For indicating a region of interest
  - Marker - For placing an item on the map
    - DivIcon – For displaying arbitrary text/HTML, displayed by the Marker
    - Icon – For displaying a simple symbol (map pointer with circle of customizable color inside)
  - MarkerCluster – Plugin element for containing other elements (Markers), grouping them together when zoomed out
  - MousePosition – Plugin element for displaying the mouse coordinates in latitude and longitude
  - PolyLine – Draws a line between coordinates
  - Calculate distance from latitude and longitude with an explicit spherical approximation
  - Map with circles and labels for each launch site, a MarkerCluster containing a Marker for each launch that is color coded to the landing success, lines and labels for the map from launch site to the nearest railway, ocean, and city with distances in kilometers
  - Launch sites are in close proximity to railways (<1.5 km) due to railways being one of the most reasonable ways to transport large and heavy rocket components.
  - Launch sites are in close proximity to highways (<1.5 km) due to the need to transport personnel and light equipment to the launch site.
  - Launch sites are in close proximity to coastline to increase the nearby fraction of disaster-amenable surface area e.g. areas that will not catch fire or be caused damage by debris from an explosion post-liftoff.
  - Launch sites seem to keep at least 10 km away from cities (the closest approach to a notable city is VAFB SLC-4E to the city of Lompoc), to minimize risk of casualties in case of an accident at the launch complex or in the early rocket flight.

Appendix

- Dashboard
    - Variables:
        - Date
        - Time (UTC)
        - Booster_Version
        - Launch_Site
        - Payload
        - PAYLOAD_MASS__KG_
        - Orbit
        - Customer
        - Mission_Outcome
        - Landing _Outcome

Appendix

- Presentation improvements beyond the template
  - Included a Business Case slide
  - Considered launch statistics support in Dashboard slides
  - Expanded Appendix
- Innovative insights
  - Business Case addition
    - Quantified competitive edge in pricing: worst-case 10% price reduction (below average cost for predicted recovery while not exceeding total cost for predicted non-recovery) or accurate prediction of upcoming launch costs
  - Dashboard
    - Launch probabilities and support displayed for all selected launches and for each booster version
  - Machine Learning section extensions
    - Repeated k-fold cross-validation improves model training results
    - Principal Component Analysis for features makes Decision Tree and K Nearest Neighbor models perform worse
    - ANOVA p-values can be used to downselect features for models while maintaining performance
    - Feature preparation should be revisited to correct errant one-hot encoding and create continuous serial number and orbit variables

Thank you!