IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

SpaceY, inc.
John Smith
Data Scientist
07062023
https://example.spacey.edu
johnsmith@example.com

Outline

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

Executive Summary

Project Goals:

- Develop a means to predict Successful recovery of Falcon 9 Stage 1 Boosters

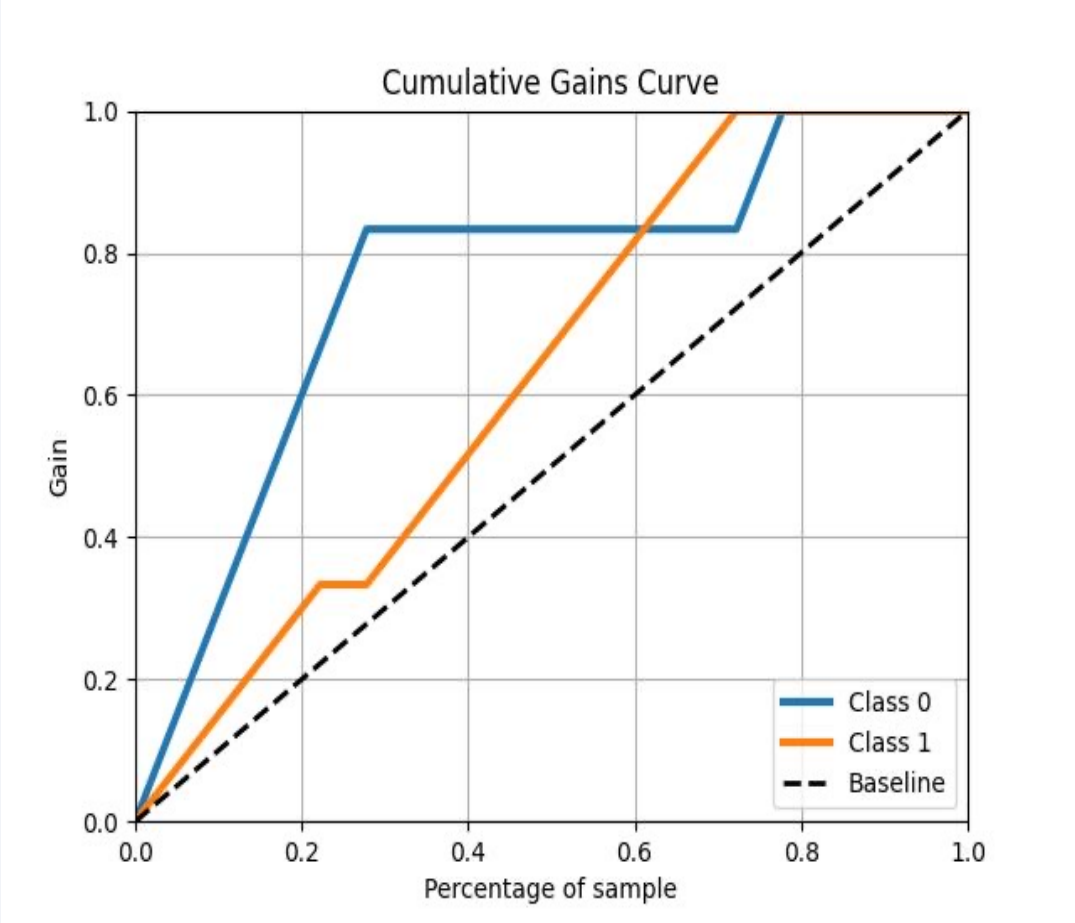
Methodologies

- Collect Historical Data on failed / successful Launches
- Analyze Data for Predictive Patterns
- Develop a classification machine model to predict successful recoveries given certain parameters

Results

- Identified Predictive Factors to Successful recovery of Falcon 9 Stage 1 Boosters
- Developed a classification model to accurately predict successful recoveries with 80% confidence and 80% accuracy

Model Gains due to Prediction



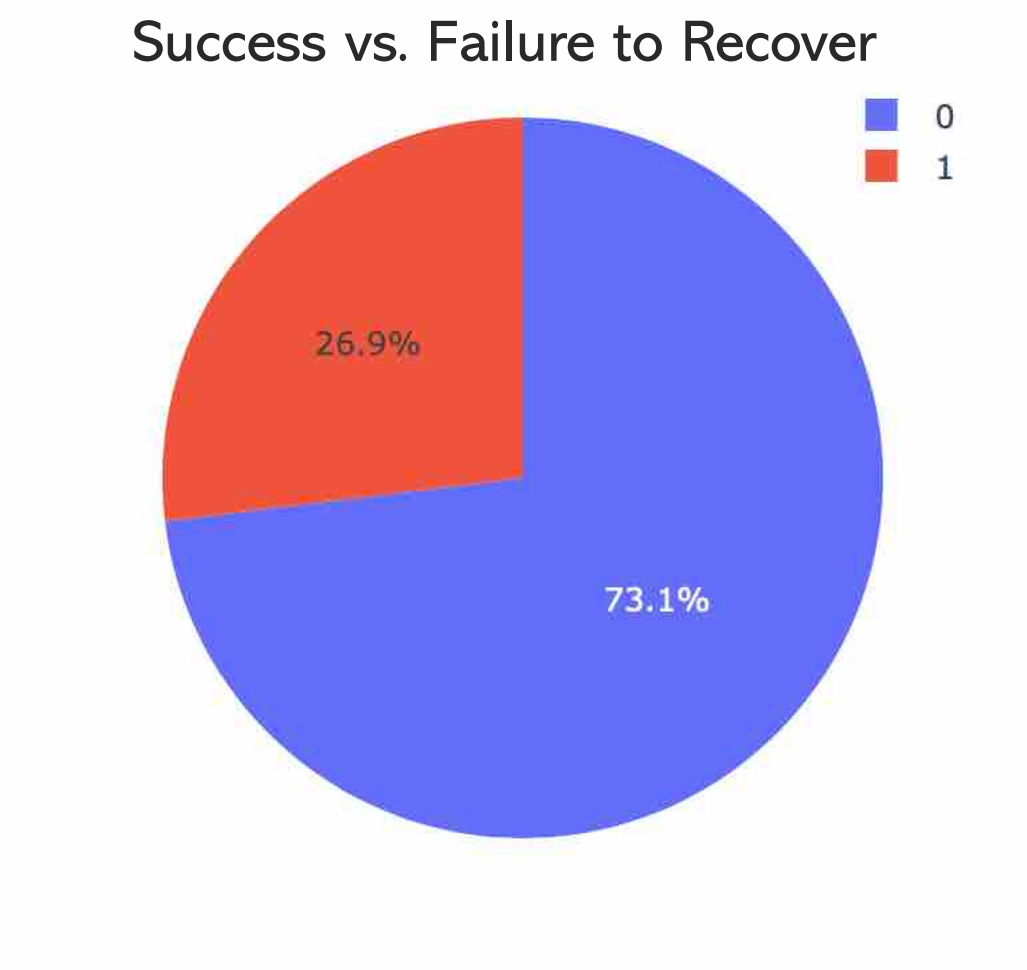
Introduction

Project background and context

- SpaceX advertises Falcon9 Launches as almost half their competition. This is due largely to the ability to recover and re-use their 1st stage boosters.

Problem to Solve:

1. Identify what features are predictors of successful recovery
2. If we could accurately predict the success rate of the recovery, we could use this method to determine the actual cost per launch for future proposals.

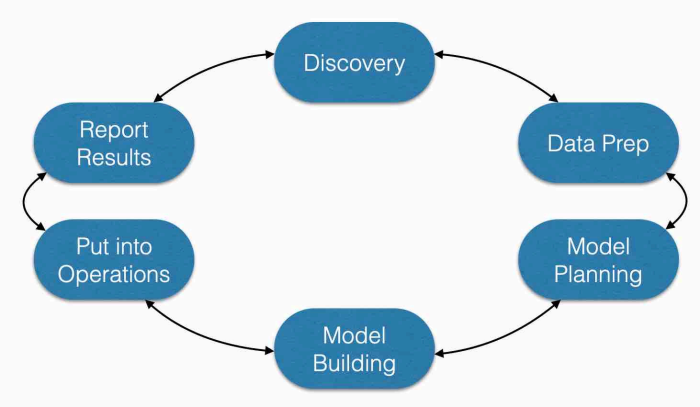


Section 1

Methodology

Methodology

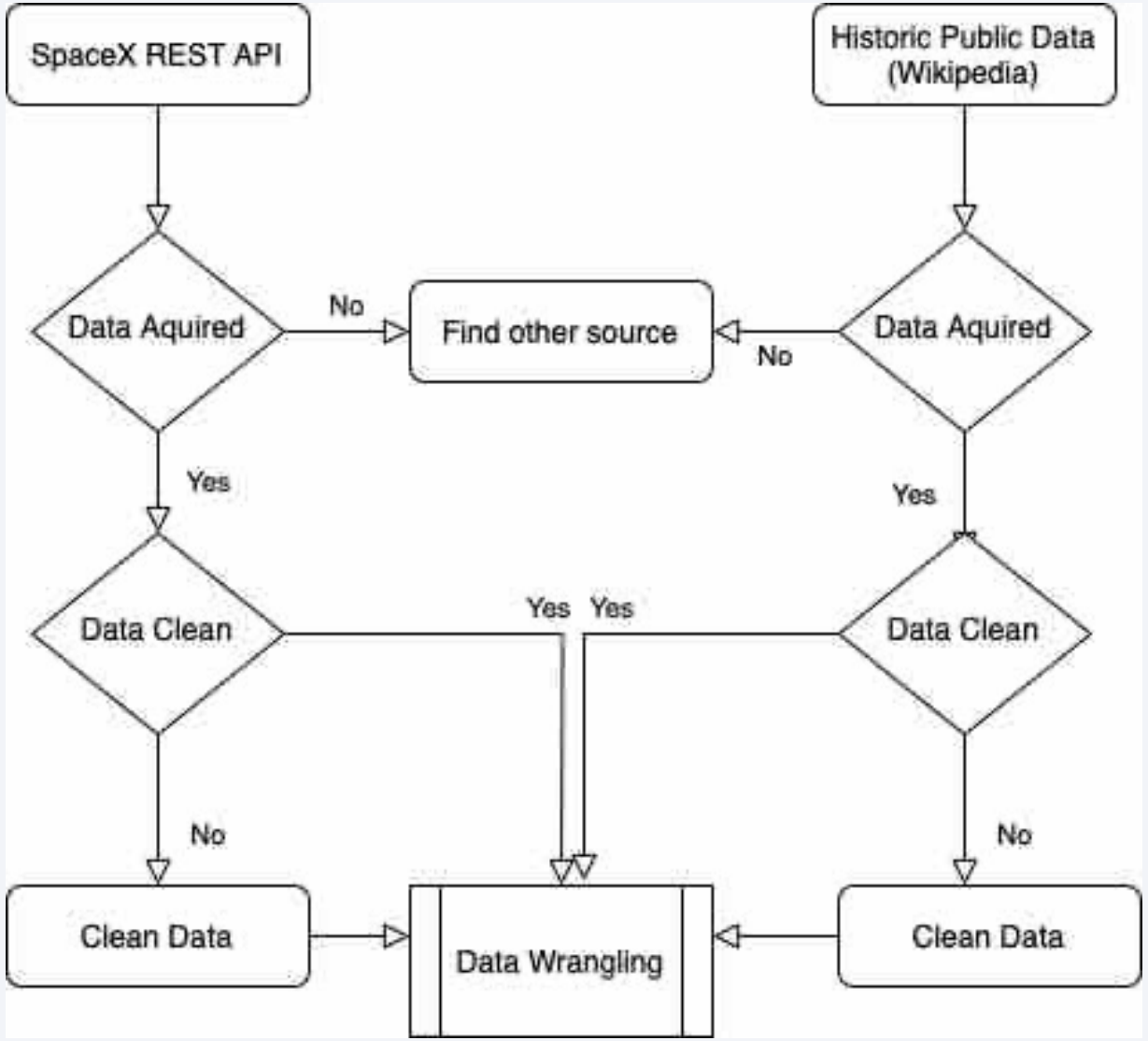
Executive Summary



- Data collection methodology:
  - Web Scraping from public web sites (Wikipedia)
  - Historical Information provided by SpaceX REST API
- Perform data wrangling
  - Data brought together in an SQL database for initial quantative analysis
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Feature selection guided by EDA
  - Logistic Regression, SVM, KNN and Decision Tree models tested and evaluated by metric

Data Collection

- Historical Launch Data Sources:
  - SpaceX public REST API
  - Wikipedia.com page for Falcon 9 launches
  - SpaceX public Website (Pricing)
- Process Methods:
  - Web Scraping tabular data
  - Normalizing JSON data from API
  - Missing / empty values removed
  - Data filtered for Falcon 9 Boosters
  - Missing Payload Data filled with Mean

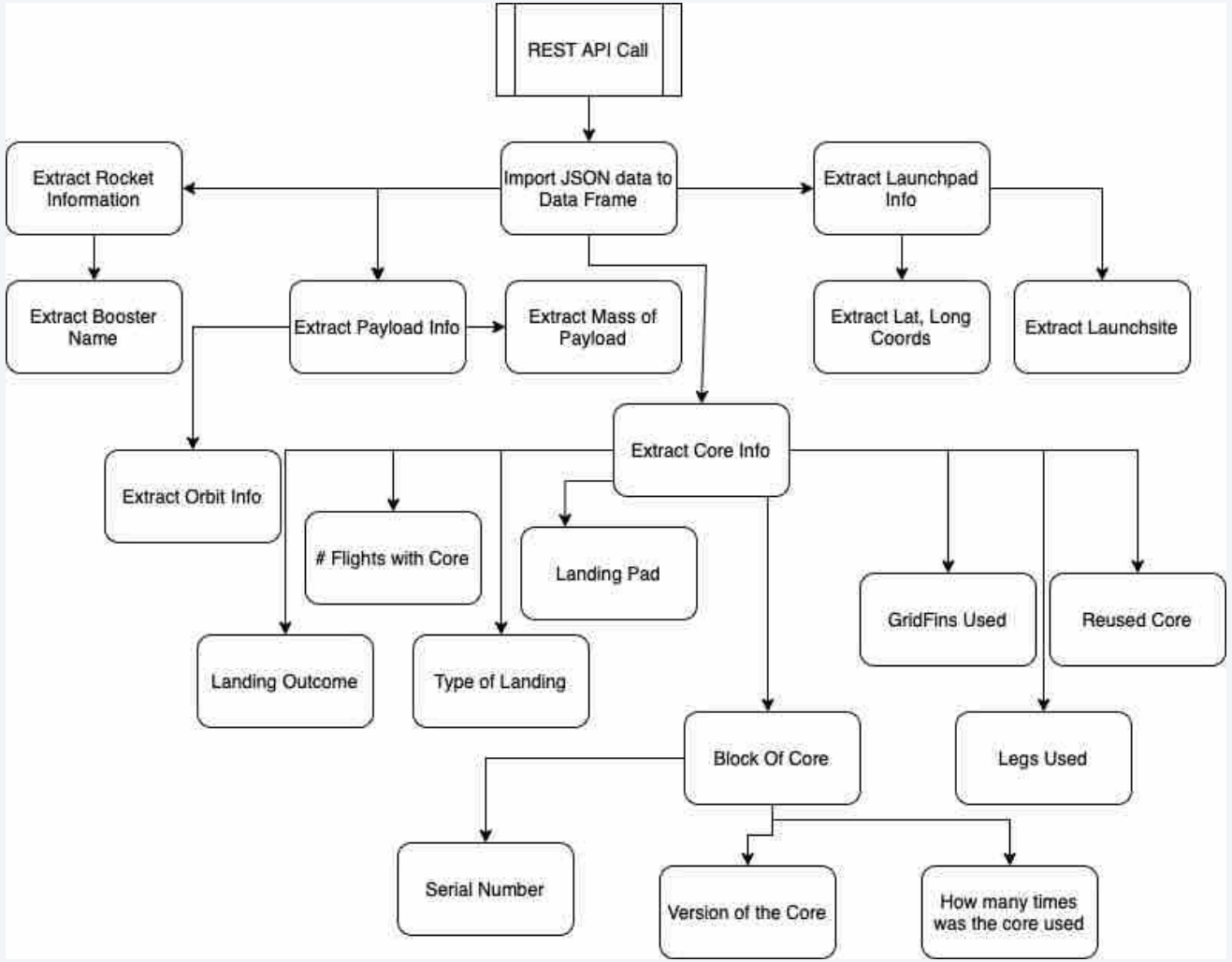


Data Collection – SpaceX API

- SpaceX REST API Data:
  - Booster Name
  - Launch site Name
    - Coordinates
  - Mass of Payload
  - Orbit information
  - Landing Outcome
  - Number of Flights
  - Type of Landing
  - Landing Pad
  - Grid Fin Usage
  - Leg Usage
  - Reused Core
  - Booster Serial Number
  - Core Version
  - Number of Flights with particular core
  - Missing Payload Information filled with mean value

- Data Archive:

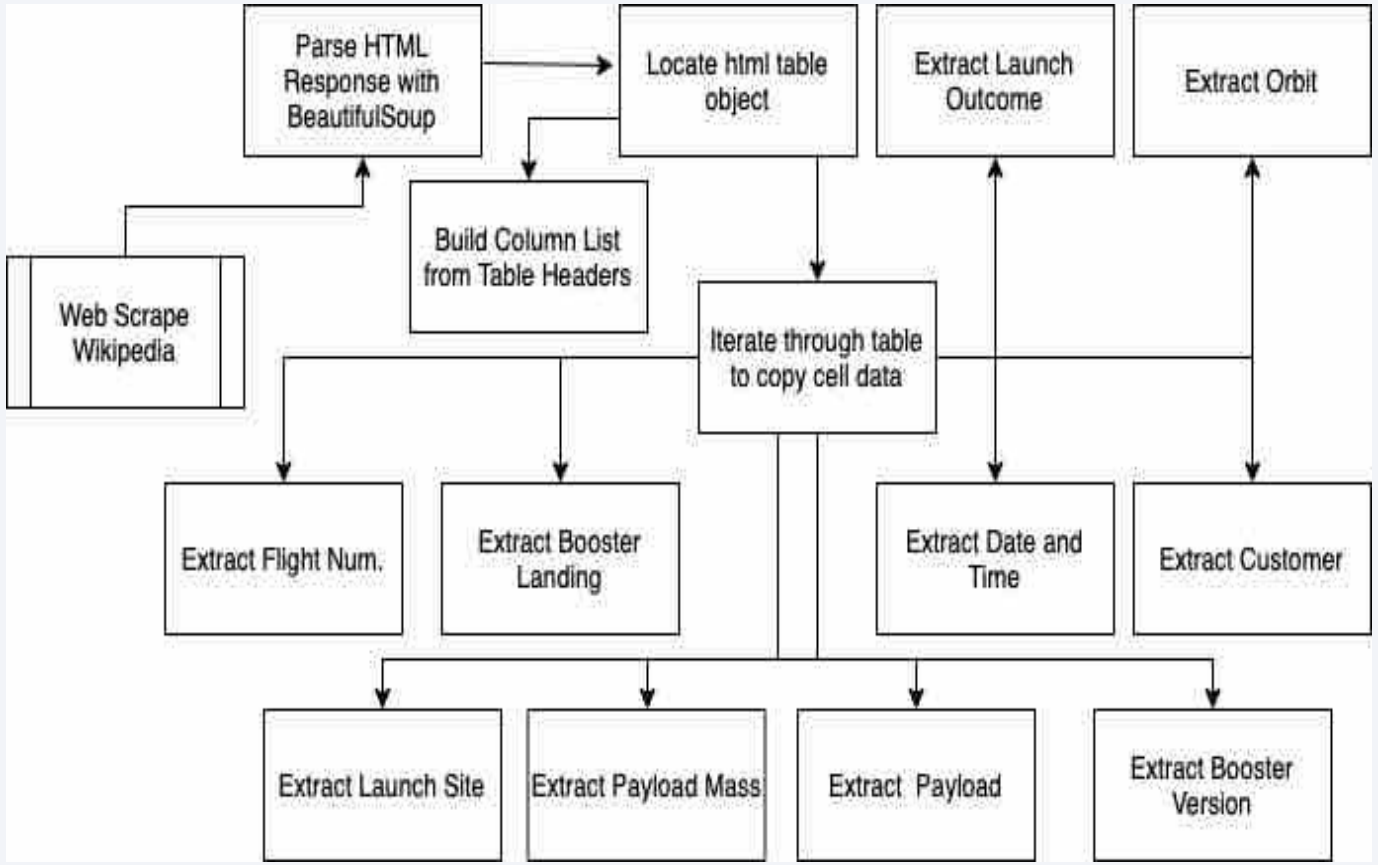
https://github.com/Cstudent76543/Data_Collection.git



Data Collection - Scraping

- Public Wikipedia information about Falcon 9:
  - Launch Outcome
  - Orbit
  - Customer
  - Date / Time
  - Booster Version
  - Payload info
  - Payload Mass
  - Launch Site
  - Booster Landing
  - Flight Num.
- Data Archive:

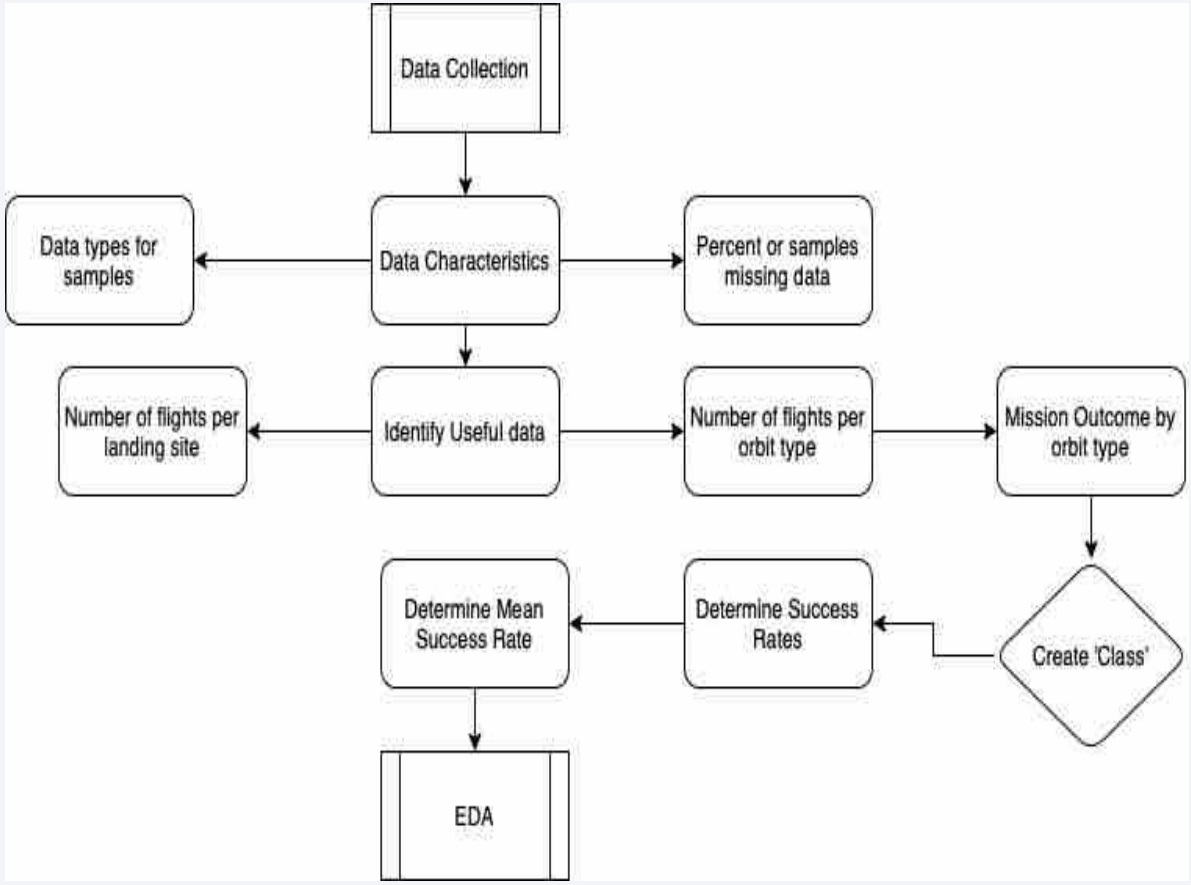
https://github.com/Cstudent76543/WebScrape.git



Data Wrangling

- Data Loaded from CSV
- Characterize the data:
  - Quantify missing values
  - Validate data types
  - Collect number of flights per landing site
  - Collect number of flights by orbit
  - Collect mission outcome by orbit
  - Feature Engineering
    - Create a Categorical Feature 'Class' to identify successful and failed landings
  - Collect Success Rates
  - Calculate average success rate
- Data Archive:

https://github.com/Cstudent76543/Data-Wrangle.git



EDA with Data Visualization

- Charts Selected for EDA
  - Payload Mass vs. Flight Number
    - Scatter Plot & Time Series
    - Success Increased as Flight numbers increased
    - Massive payloads were less likely to return
  - Launch Site vs. Flight Number
    - Scatter Plot & Time Series
    - Frequency of launches per site
    - As success became more common, other sites used
  - Payload Mass vs Launch Site
    - Scatter Plot & Time Series
    - Some sites did not deal with Massive Payloads
  - Success Rate vs. Orbit Type
    - Some orbits had much higher success rates than others
  - Flight Number vs. Orbit Type
    - Very Low Earth Orbit Flights occur later
    - Some orbits have little correlation with flight number
  - Payload Mass vs. Orbit Type
    - Some orbits were more successful with massive payloads
  - Launch Success by Year
    - Launch Success increased by year
    - ~5% in 2015 -> ~60% in 2020

- Data Archive:

https://github.com/Cstudent76543/Data_Visualization.git

EDA with SQL

- SQL Queries run on data:
  - Unique Launch Site Names
  - Selected 5 launches from sites beginning with 'CCA'
  - Calculated total Payload Mass (kg) for all missions on behalf of NASA
  - Calculated mean Payload mass for the Falcon 9 booster v1.1
  - Listed date of first successful landing to a ground pad
  - Selected names from Boosters with mass between 4000 & 6000 successfully landing on drone ships
  - Selected Total success and failure outcomes from all the missions
  - Selected Booster Versions having carried the maximum payload
  - For 2015, Display months with failure outcomes on droneships, launch site and booster versions.
  - From 20100604 to 20170320: Count successful landing outcomes
- Data Archive:

https://github.com/Cstudent76543/SQL_EDA.git

Build an Interactive Map with Folium

- Folium Objects generated:
  - Map Object:
    - The Background map where all the other objects are placed
  - Circle Objects:
    - Created location shading around each launch site region
  - Marker Objects:
    - Marked Prominent nearby features for each launch site
    - Marked each launch site
    - Color coded mission success and failure by launch site and mission number
  - MarkerCluster Object:
    - Collected each launch outcome at each launch site
  - MousePosition Object:
    - Displayed lat long for feature identification and distance calcs
  - PolyLine Object:
    - Graphically represent distance from the launch site to various prominent features
- Data Archive:

https://github.com/Cstudent76543/Folium_notebook.git

Build a Dashboard with Plotly Dash

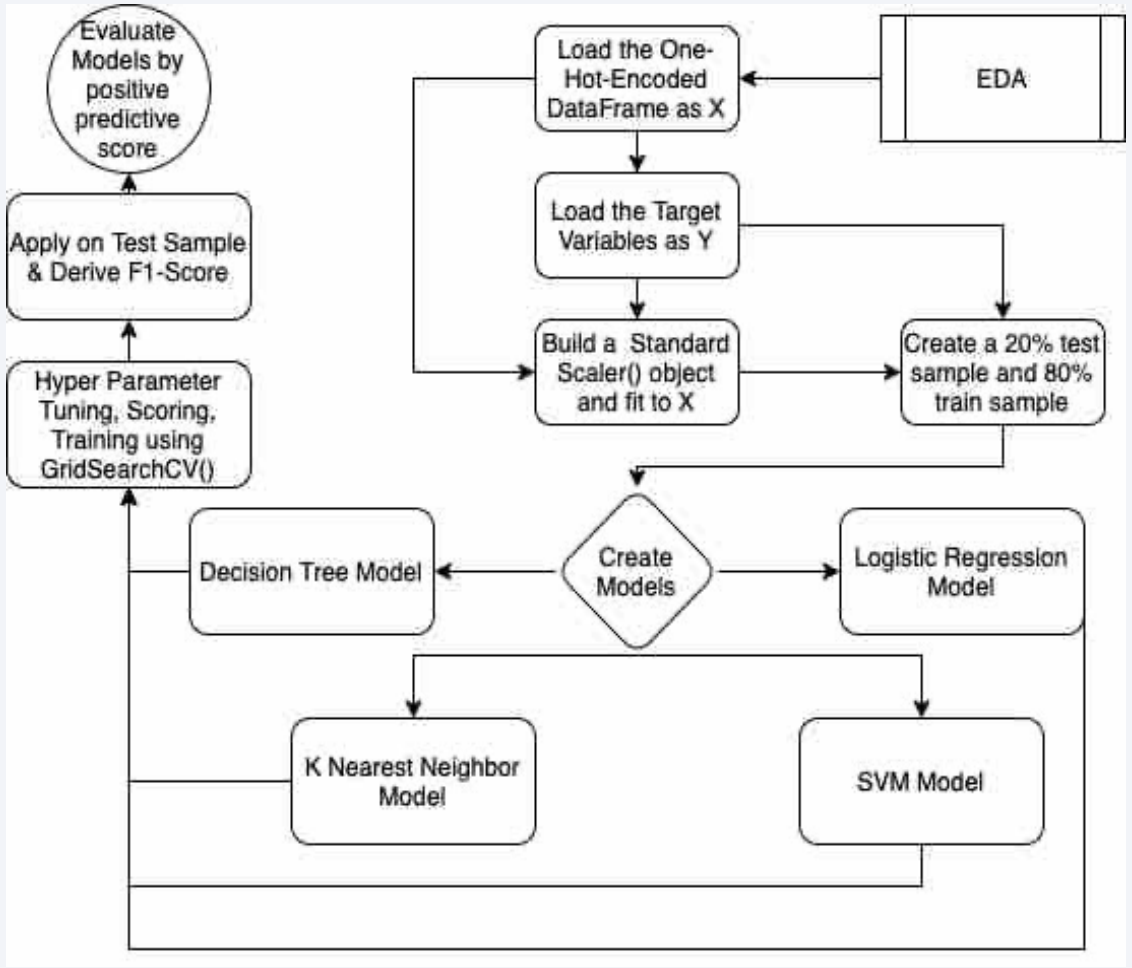
- Dashboard Components
  - Site Selection Drop Down:
    - Default to 'ALL' – Gives overview of success rate for all sites, easy to identify higher rates
    - Per Launch Site: Allowed a deeper dive into the success rates and frequency at each particular site
  - Success Rate Pie Charts:
    - Frequency of Success and total missions from each site
  - Payload Range Slider:
    - Provided an easy way to visualize how payload impacted success rate by launch site
  - Payload Mass vs. Mission Outcome Scatter Chart:
    - Classified by mission outcome, easy to identify correlation between sites and payload on mission outcome
- Data Archive:

https://github.com/Cstudent76543/Plotly_Dash.git

Predictive Analysis (Classification)

- Model Development & Assessment
  - One-Hot-Encoded Independent Variables
  - Scaled Independent Variables to common range
  - Set aside 20% of samples for testing
  - Built 4 different classification models
  - Performed hyperparameter tuning for f1 score and k-fold validation using GridSearchCV()
  - Tested each model on Test Data
  - Further assessment for Positive Predictive score from confusion matrix
  - Calculated Relative impact of predictive features on successful recovery of Falcon 9 boosters
- Data Archive:

https://github.com/Cstudent76543/Model_Development.git



Results

- Exploratory data analysis results
  - Predictive Features Identified
    - Grid Fins & Legs, Serial number of the booster, orbit, payload, launch site, re-use of the booster all play important roles in the successful recovery of the Falcon 9 boosters. Launches have been increasing success frequency over time, from 5% -> 60% in under a decade.
  - Strong trends in payload, orbit, launch site and flight number to successful outcomes.
- Interactive analytics: See Below Sections for Screen Shots
  - Some Launch sites have had higher frequency of success in recovery of the Falcon 9 Boosters, but this is likely due to the payload and destination orbit of later launches.
- Predictive analysis results
  - While three of the 4 chosen models achieved 88% in harmonic accuracy (f1-Score) the logistic regression model lends itself better to our use case. The ability to generate consistent direct probabilities without additional pre-processing is exactly what we want to address the original problem.

  i.e. To predict which launches will result in a successful landing, we want to assign a probability to the outcome and have confidence in that prediction when it occurs.

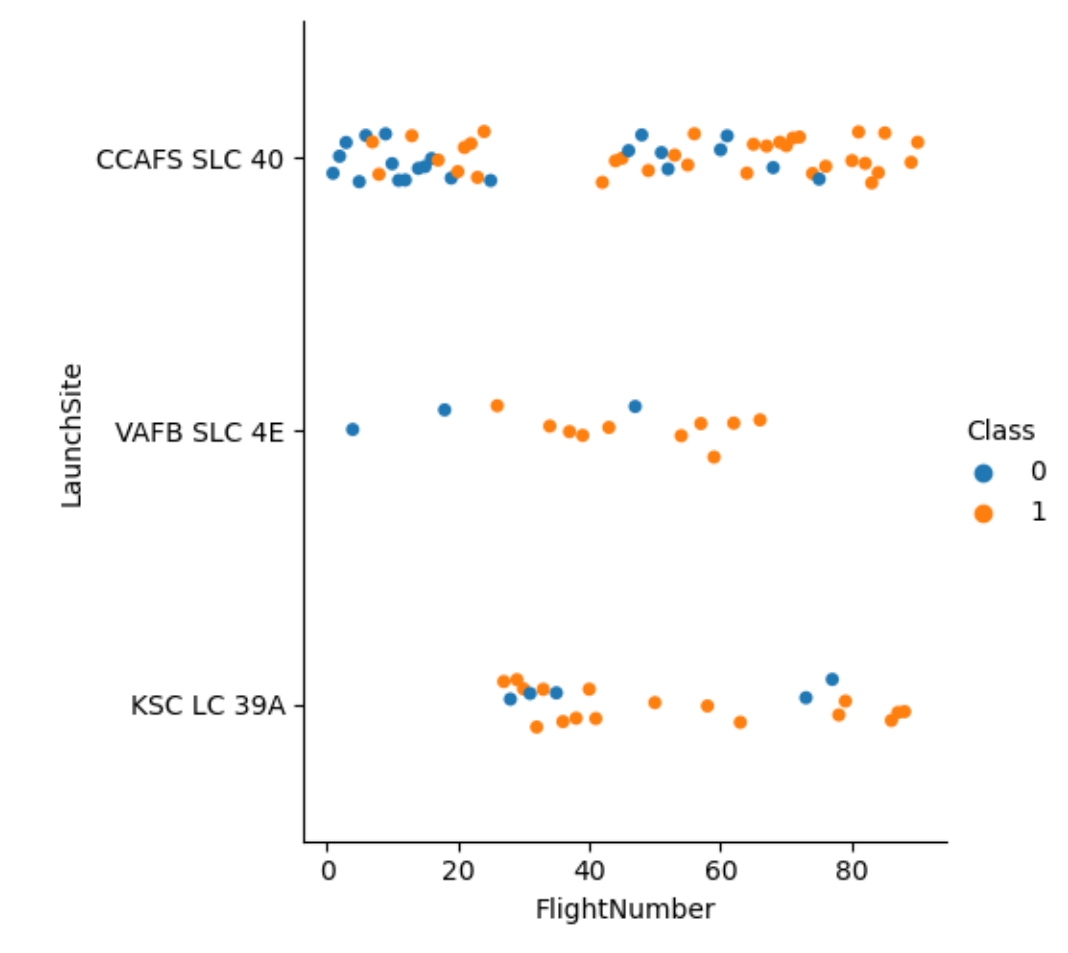
  The Logistic Regression Model satisifies this requirement with 80%+ F1 score and 80% Positive Predictive score.

Section 2

Insights drawn from EDA

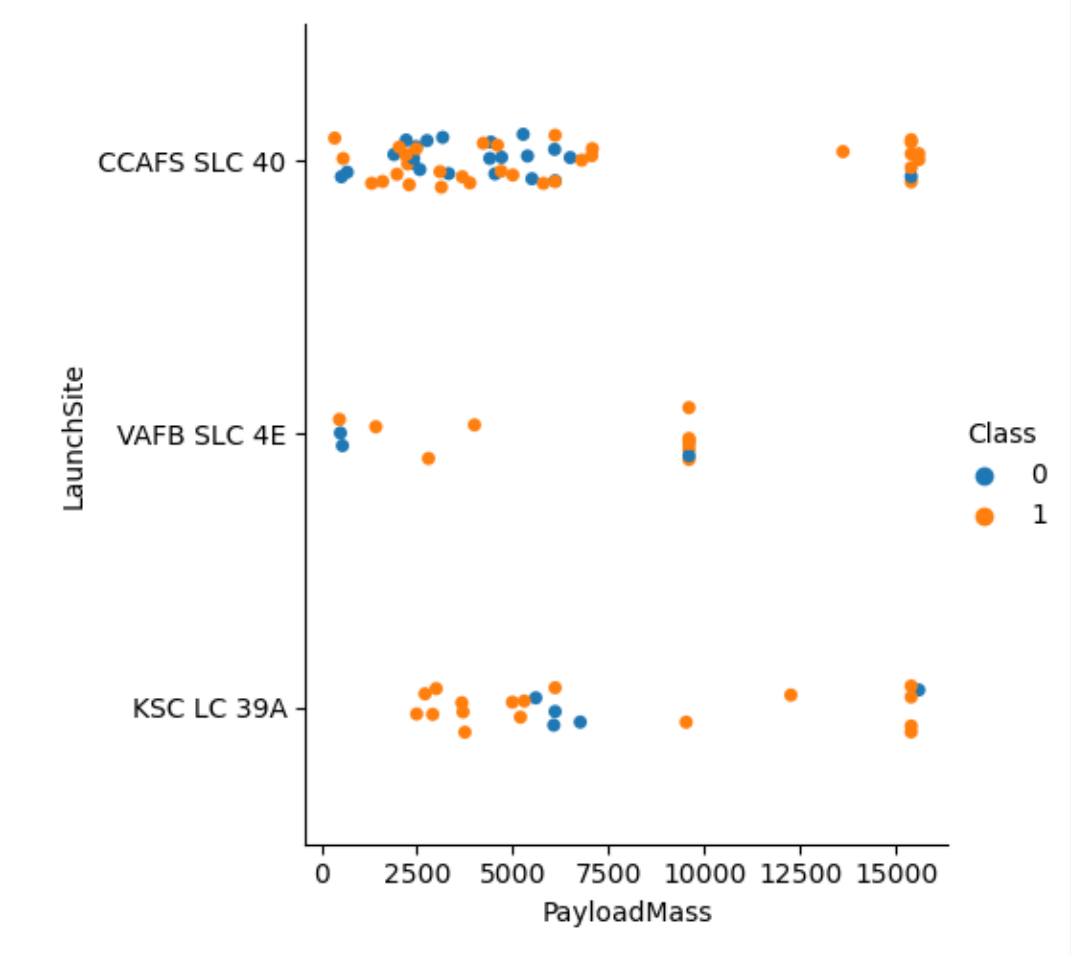
Flight Number vs. Launch Site

- CCAFS SLC-40 was the primary launch site for early, failure prone launches. It remains the highest frequency launch site.
- After ~20 launches the success rate began to climb at all sites and KSC-LC 39A started to be used.
- KSC-LC 39A has a high frequency of success.



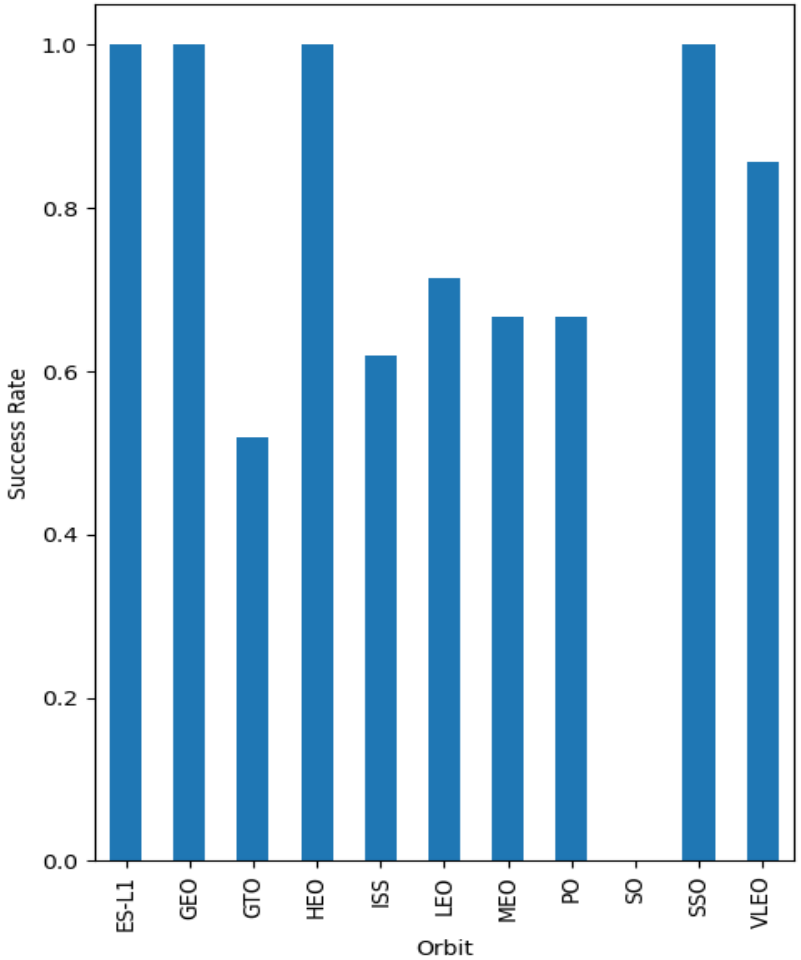
Payload vs. Launch Site

- VAFB SLC-4E has not launched a payload over 10000 kg in the sample data.
- CCAFS SLC-40 has a wide range of payload mass in it's launch history. Most of the heaviest payloads successfully launched with a recovered Falcon 9 booster come from this site.



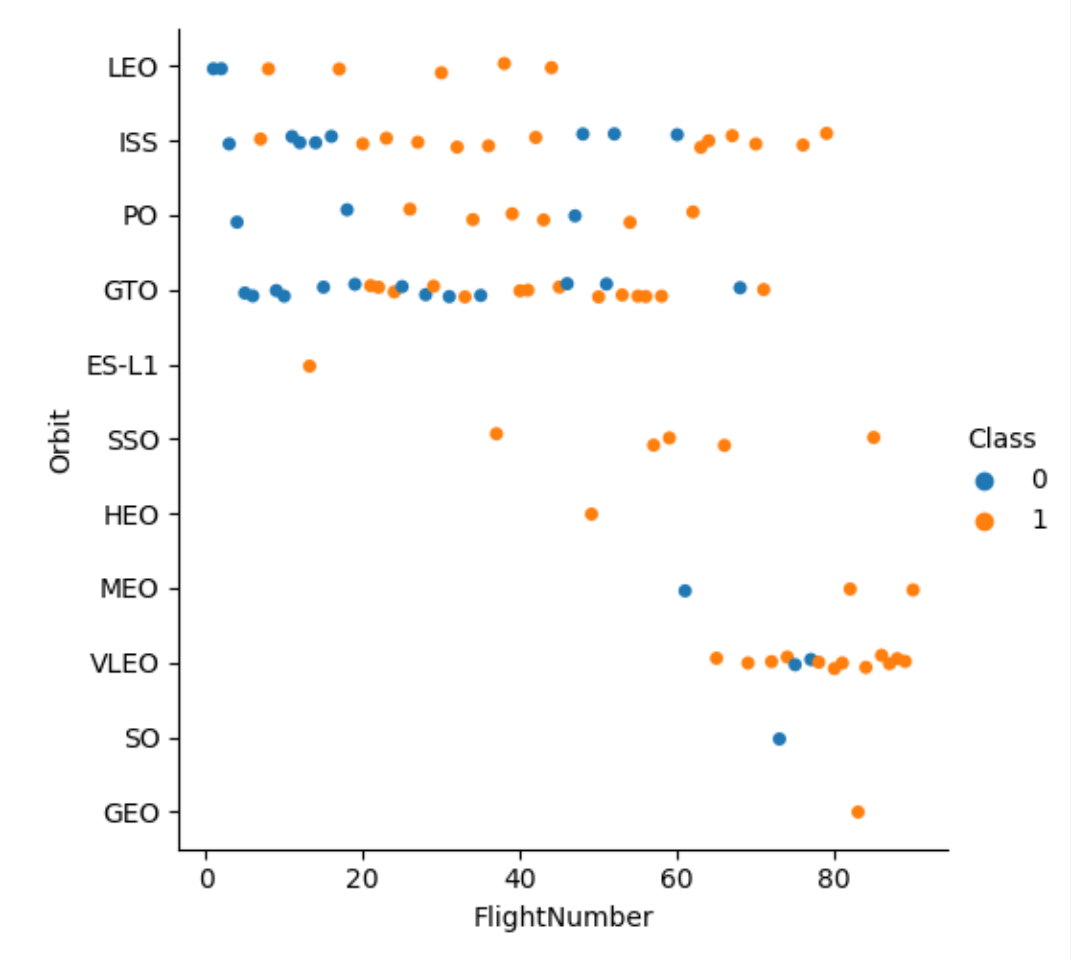
Success Rate vs. Orbit Type

- SSO, HEO, GEO and ES-L1 Orbits have close to 100% success rates in recovery of the booster.
- VLEO orbits have success rates over 80%.
- GTO Orbits are the worst orbits, only about 50% success rate.



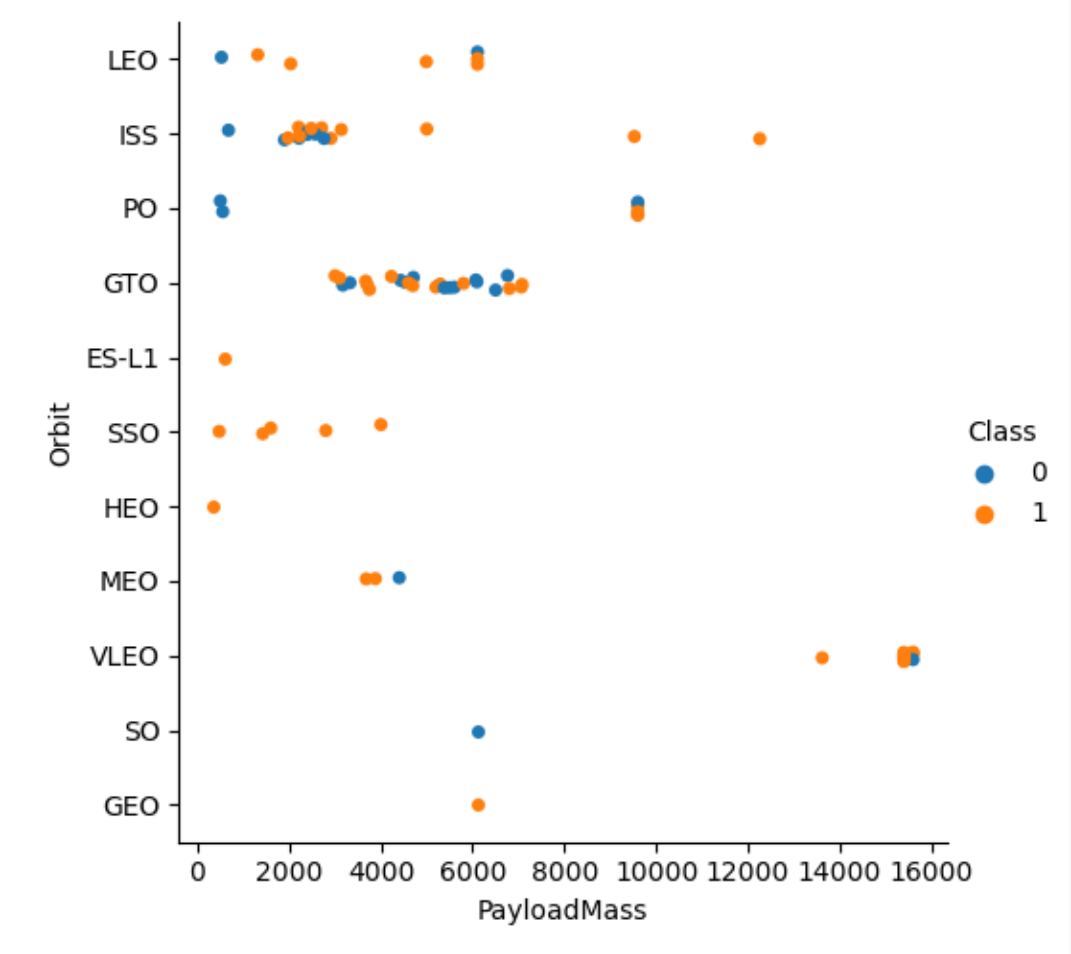
Flight Number vs. Orbit Type

- VLEO Launches have only recently been done, which could explain their higher success rate in recovery.
- GTO and ISS orbits have continued to be prone to failure despite higher flight numbers.



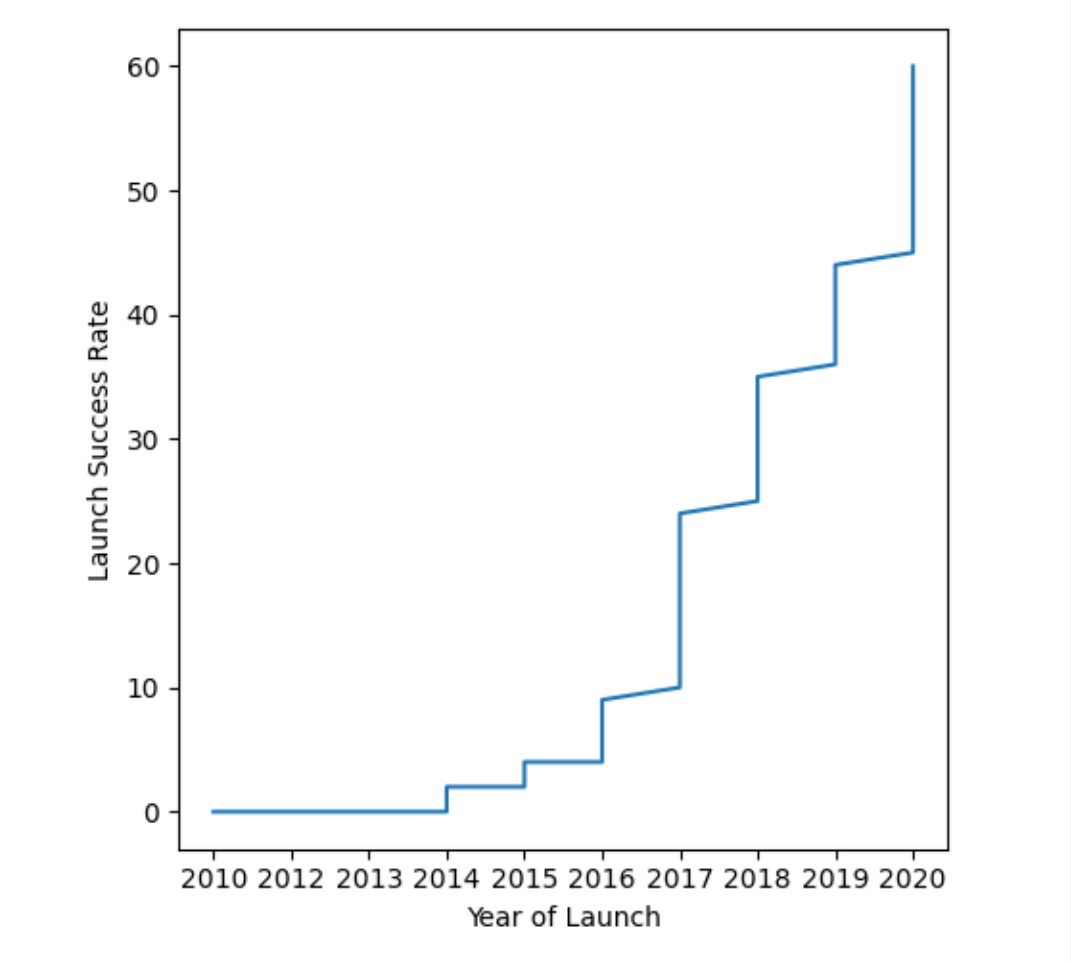
Payload vs. Orbit Type

- GTO Orbits seem to be unaffected by Payload Mass.
- Almost all VLEO Orbit missions were performed with close to Maximum payload mass.
- Most missions carry between 2000 and 6000 kg in payload mass.



Launch Success Yearly Trend

- Successful recovery of the Falcon 9 booster has become more frequent with time.
- From ~5% in 2015 to nearly 60% in 2020.
- As flights continue, they are likely to result in more successful booster recovery efforts.



All Launch Site Names

- Find the names of the unique launch sites
- The Unique names were pulled from the SQL database table with a SELECT statement combined with DISTINCT

Task 1

Display the names of the unique launch sites in the space mission

```
%%sql
SELECT DISTINCT "Launch_Site"
FROM "SPACEXTBL"
ORDER BY "Launch_Site"
```

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

Launch_Site
None
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Applying WHERE, LIKE and LIMIT clauses to the SELECT statement generated this list.

```
%%sql
SELECT *
FROM "SPACEXTBL"
WHERE "Launch_Site" LIKE "CCA%"
LIMIT 5
```

Pyth

* sqlite:///my_data1.db
Done.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcom
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Succes
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	Succes
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Succes
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Succes
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Succes

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Used the SUM() function combined with WHERE and LIKE with the SELECT Statement

Task 3

Display the total payload mass carried by boosters launched

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS Total_kg_NASA_Payload
FROM SPACEXTBL
WHERE "Customer" like "%NASA%"
```

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

Total_kg_NASA_Payload
107010.0

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Applied the AVG() function combined with a WHERE and LIKE clause for the SELECT statement.

Display average payload mass carried by booster version F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS Mean_payload_kg_F9v11
FROM SPACEXTBL
WHERE "Booster_Version" like "F9 v1.1%"
```

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

Mean_payload_kg_F9v11
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Applied the MIN() function in combination with WHERE and LIKE clauses to the SELECT statement.

Hint:Use min function

```
%%sql
SELECT MIN("Date") AS Earliest_groundpad_Success
FROM SPACEXTBL
WHERE "Landing_Outcome" LIKE "%Success (ground pad)%"
```

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

Earliest_groundpad_Success
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- Used Logical Operators in the WHERE clause to combine the Landing Outcome with Success.

Task 6

List the names of the boosters which have success in drone ship and h
but less than 6000

```
%%sql
SELECT DISTINCT "Booster_Version" AS Booster_droneship_success
FROM SPACEXTBL
WHERE "Landing_Outcome" LIKE "%Success (drone ship)%"
    AND (PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000)
ORDER BY "Booster_Version"
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Used the COUNT() function in combination with logical OR in the WHERE clause for the SELECT statement

List the total number of successful and failure mission outcomes

```
%%sql
SELECT COUNT(*) AS total_success_failures
FROM SPACEXTBL
WHERE "Landing_Outcome" LIKE "%Success%"
    OR "Landing_Outcome" LIKE "%Failure%"
```

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

total_success_failures
71

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Used a sub-query in the WHERE clause to calculate the maximum payload and compare it to a given records payload value.
- Of those returned records, select the DISTINCT version values to display.

```
%%sql
SELECT DISTINCT "Booster_Version" AS have_carried_max_payload
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL
    )
ORDER BY "Booster_Version"
```

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

have_carried_max_payload
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

2015 Launch Records

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

+ Code + Markdown

```
%%sql
SELECT substr("JanFebMarAprMayJunJulAugSepOctNovDec",1+3*substr("Date",4,2)-3, 3) as Month,
    "Landing_Outcome", "Booster_Version", "Launch_Site"
FROM SPACEXTBL
WHERE "Landing_Outcome" LIKE "%Failure (drone ship)%"
    AND substr(Date,7,4) = '2015'
```

15] Pytho

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

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

Used the substr() function to parse the date field and assign a month name and to provide a logical operator in the WHERE clause.

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

- Rank 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
- Heavy use of Substr to parse the date field and use the BETWEEN operator.
- GROUP BY and ORDER BY required.

```
%%sql
SELECT "Date", COUNT(*) AS num_success
FROM SPACEXTBL
WHERE (substr("Date",7,4)||substr("Date",4,2)||substr("Date",1,2)
      BETWEEN '20100604' AND '20170320')
      AND
      ("Landing_Outcome" LIKE "%Success%")
GROUP BY "Date"
ORDER BY substr("Date",7,4) DESC, substr("Date",4,2) DESC, substr("Date",1,2) DESC
```

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

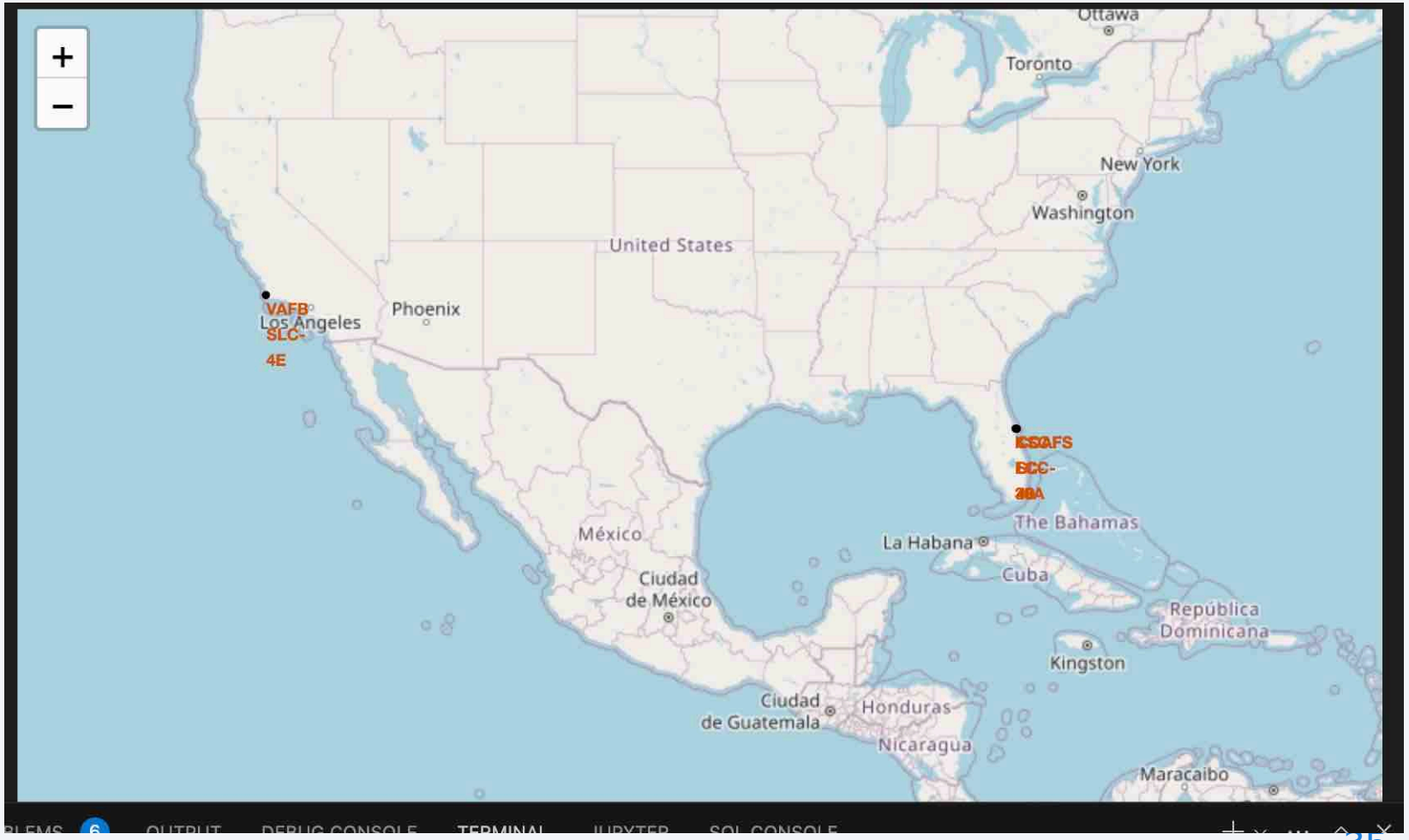
Date	num_success
06/03/2017	1
19/02/2017	1
14/01/2017	1
05/01/2017	1
14/08/2016	1
04/08/2016	1
18/07/2016	1
05/06/2016	1
27/05/2016	1
22/12/2015	1

Section 3

Launch Sites Proximities Analysis

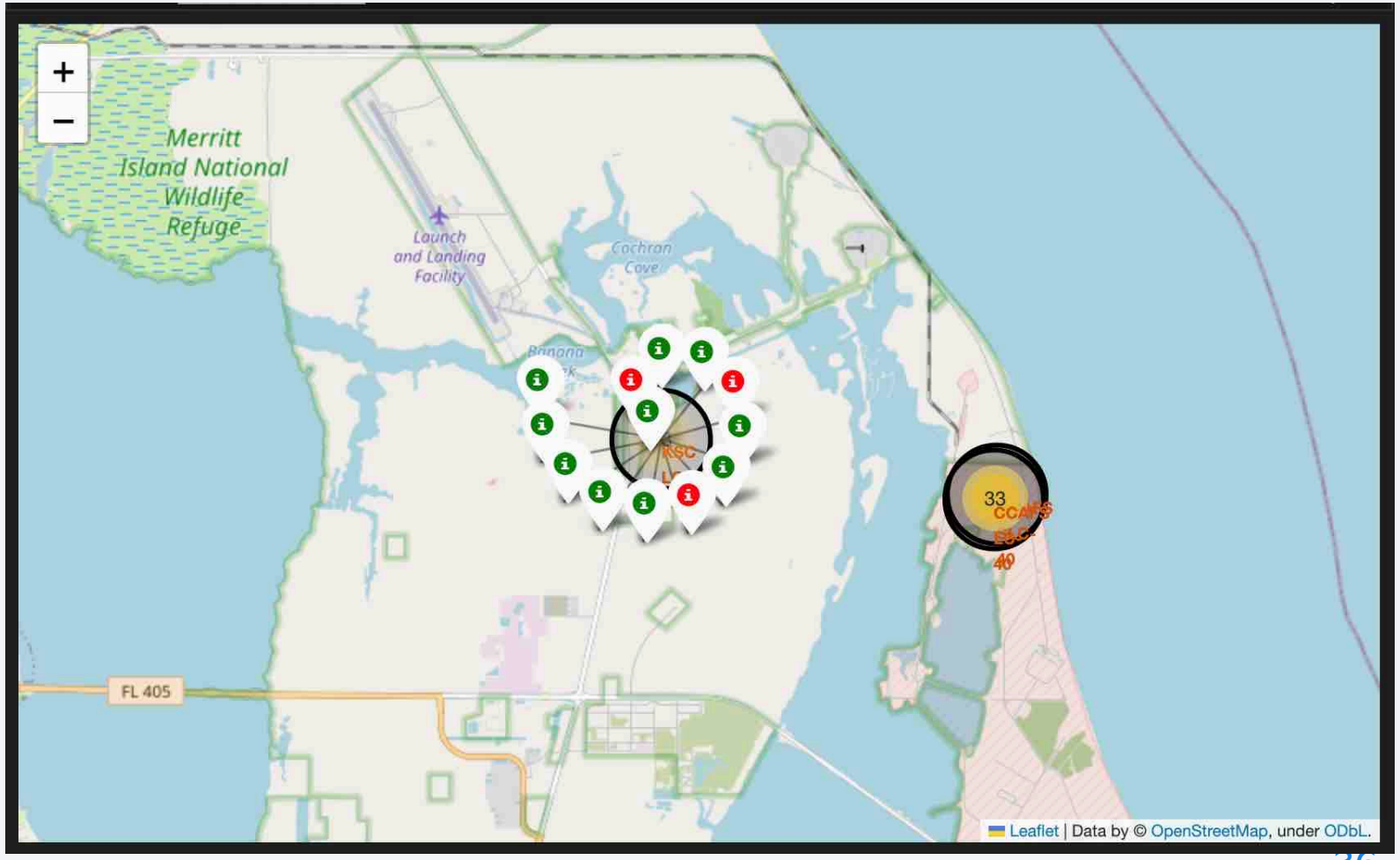
GeoSpatial Data for Launch Sites

- All Launch sites are in the southern region of the US located very close to the coast but distanced from major population centers.



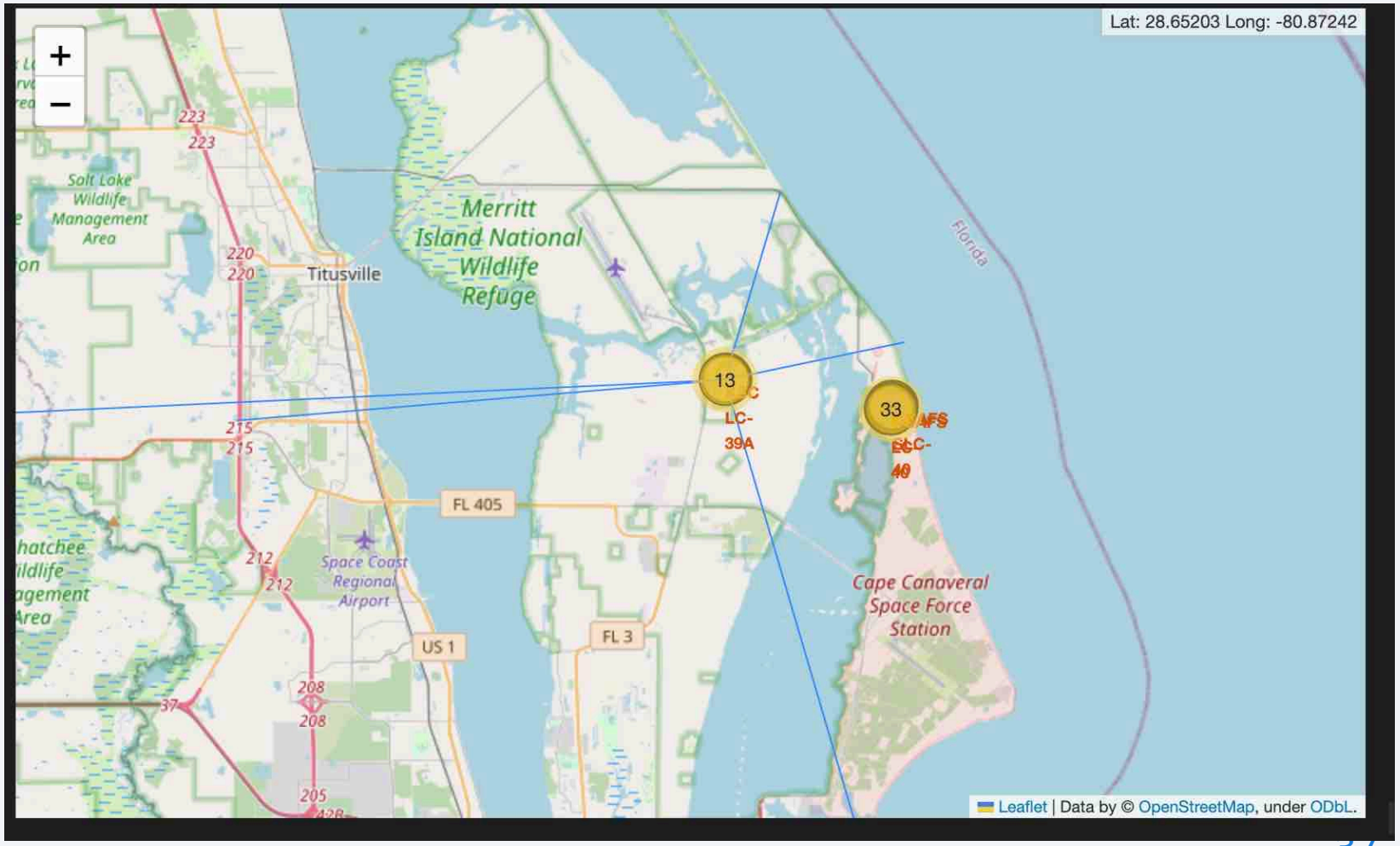
Recovery Success by Launch Site

- Some Launch Sites had a much higher frequency of success.



Proximity of Launch Sites to Landmarks

- The launch sites tend to be distant from major population centers but reasonably close to railways and highways.
- This could be due to the US N/S highways and railroads hugging the coast.
- If you are near the coast, you are near a major highway and railroad almost anywhere in the US

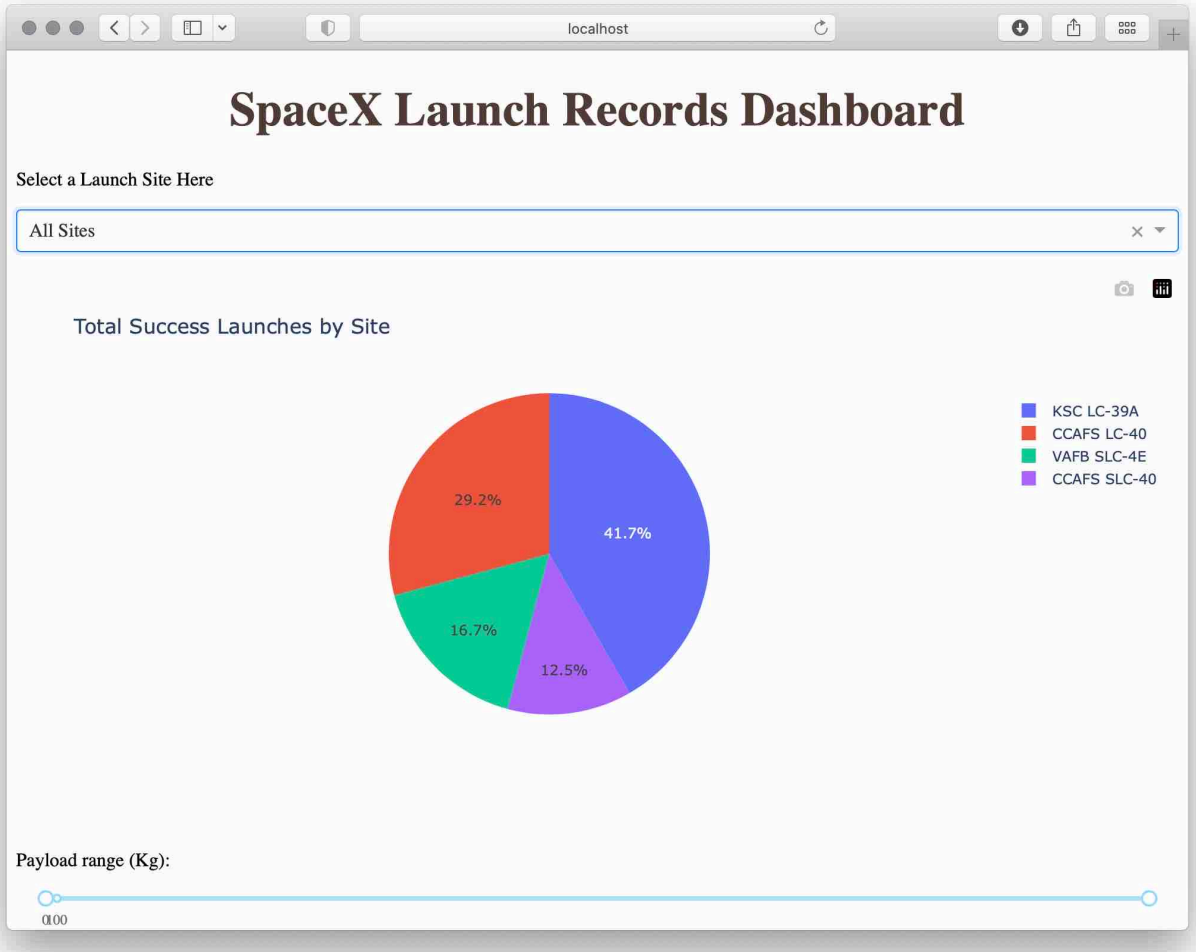


Section 4

Build a Dashboard with Plotly Dash

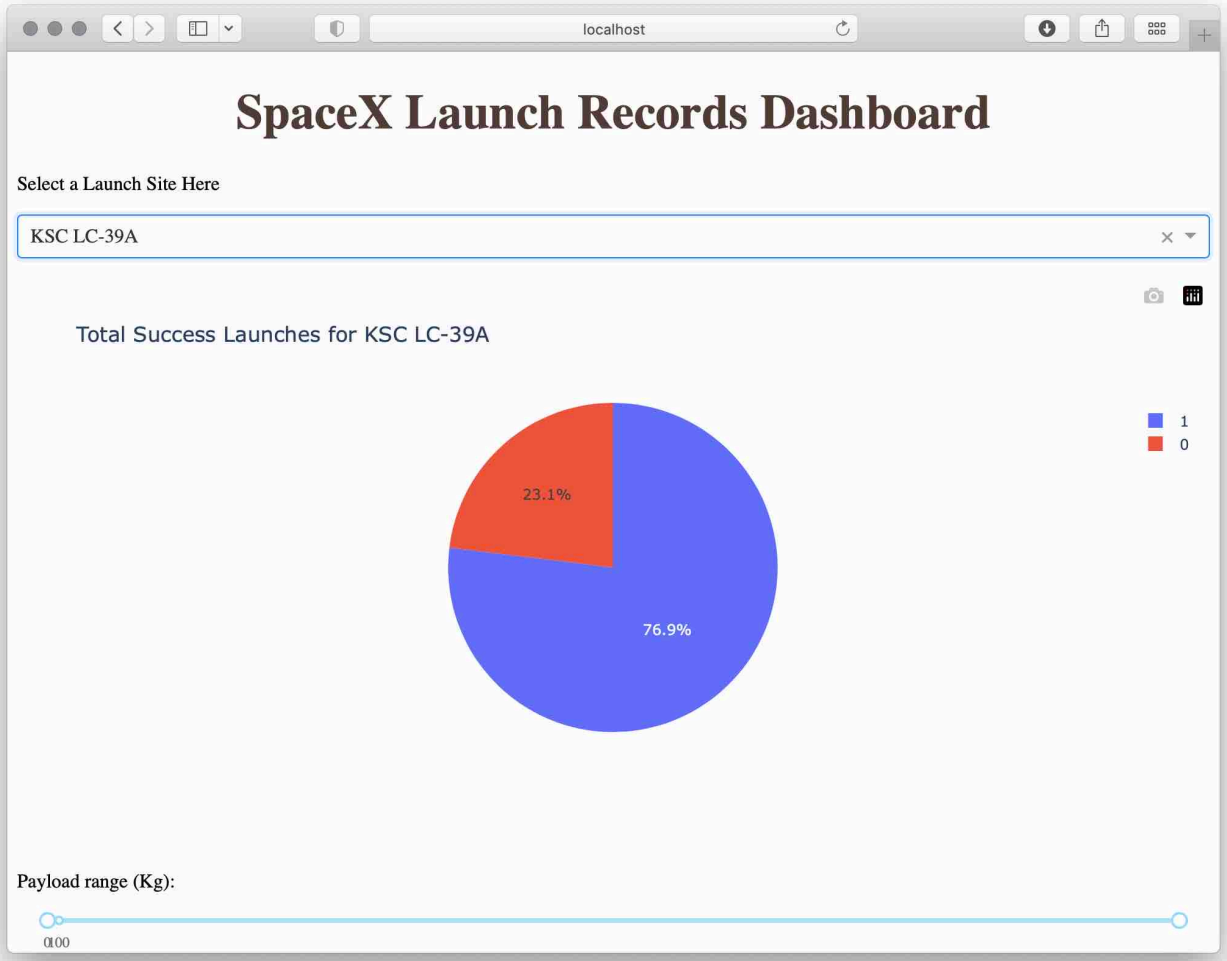
Success Rates by Site

- KSC-LC39A Has the highest frequency of successful launches
- CCAFS LC-40 has the second highest.
- No site has over 50% success rate.

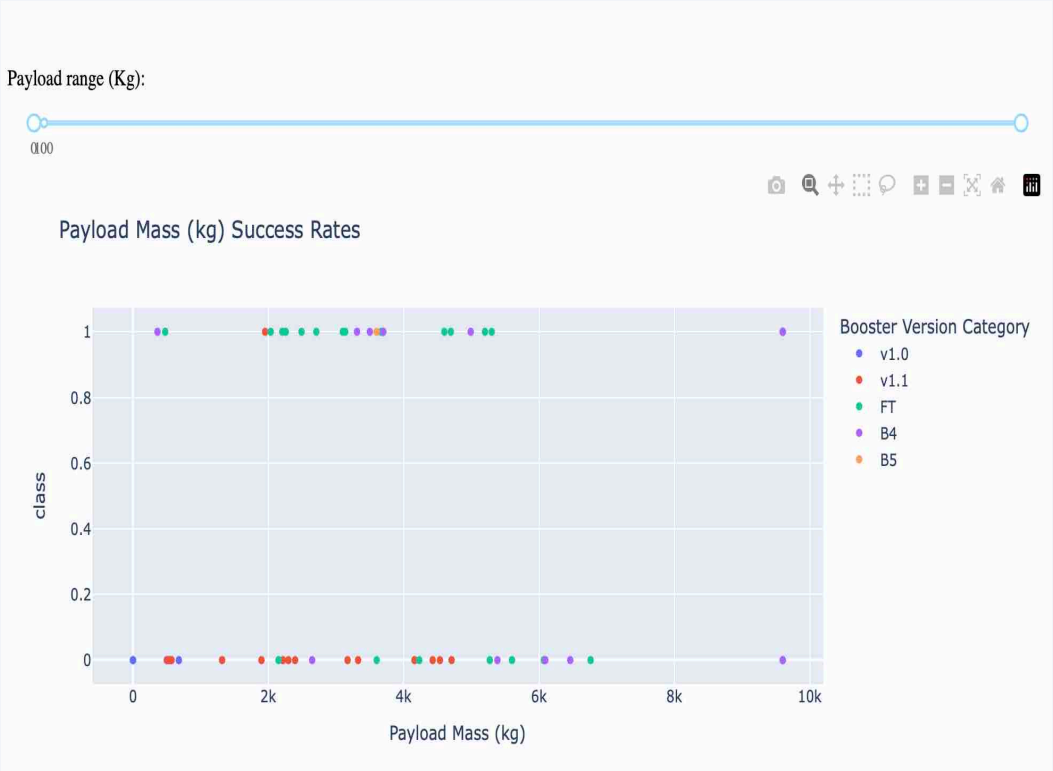


Success Rate for KSC LC

- Over 76% of all launches from KSC LC-39A have resulted in successful recovery of the Falcon 9 Booster.



Payload vs. Success Rate All Sites



- The FT Booster version seems more successful over the entire range of payloads

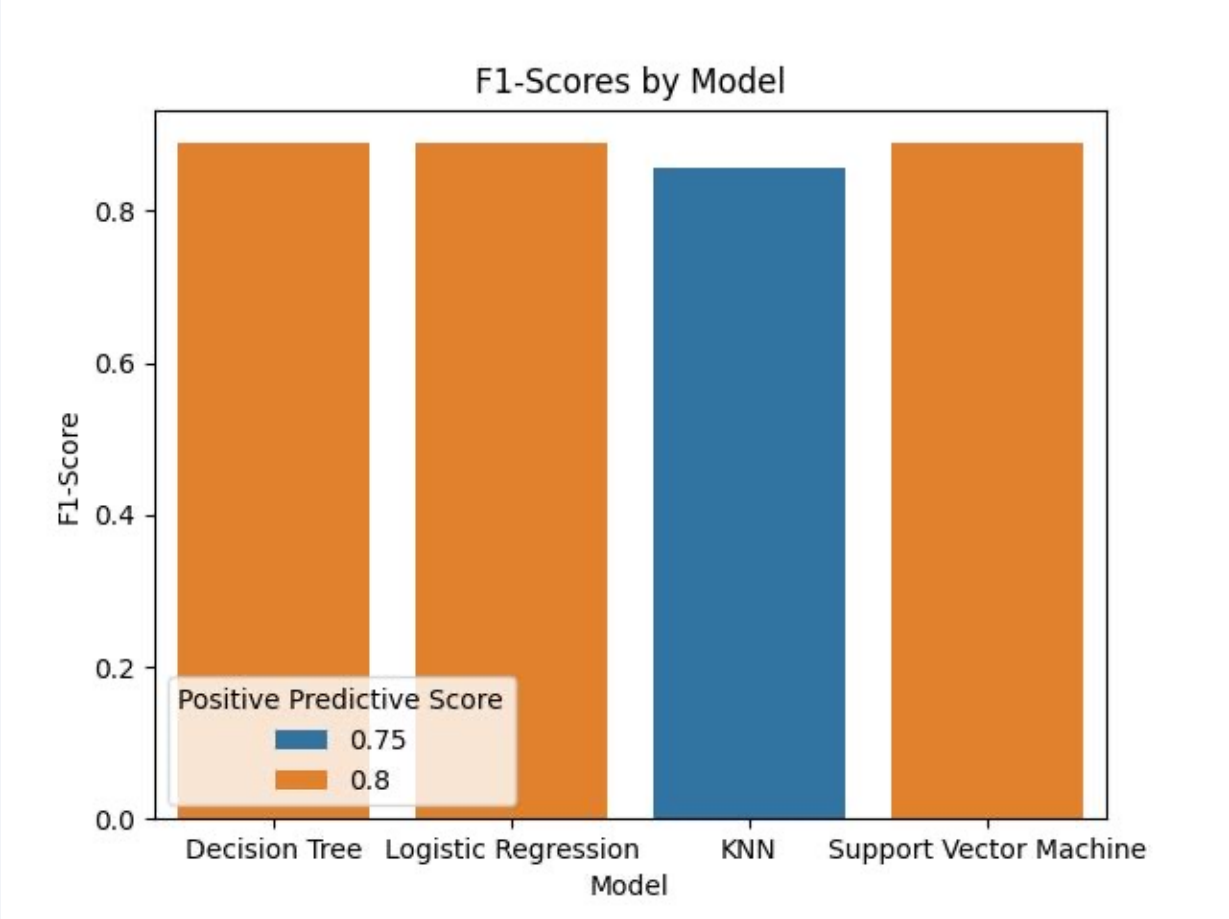
The v1.1 Booster does not have a good success rate, frequently resulting in a failed recovery

Section 5

Predictive Analysis (Classification)

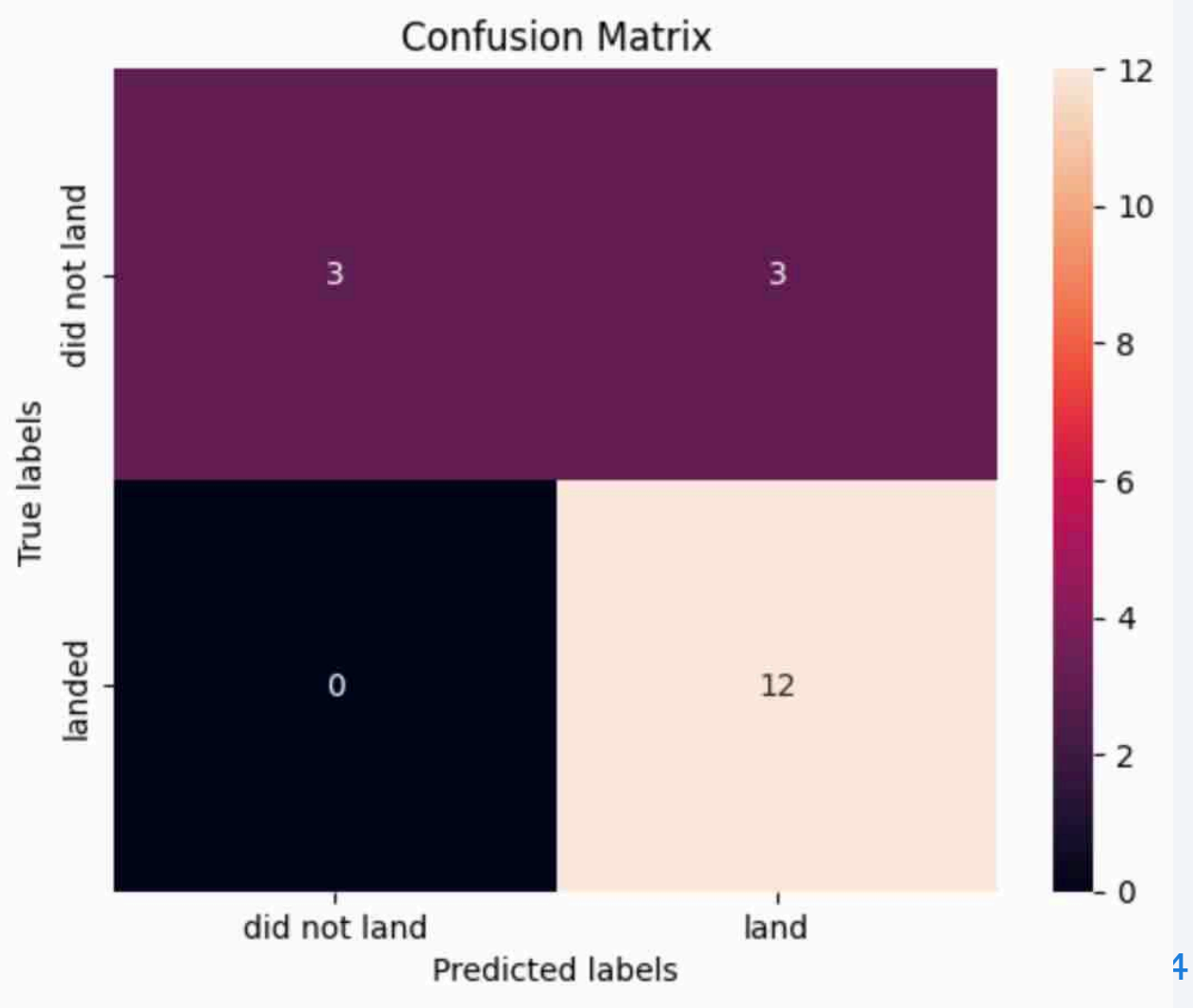
Classification Accuracy

- Decision Tree, Logistic Regression and SVM machines achieved over 80% F1 score.
- In addition all three achieved 80% on their Positive Predictive Score.
- Logistic Regression provides direct probabilities of success and is the best model for this problem.



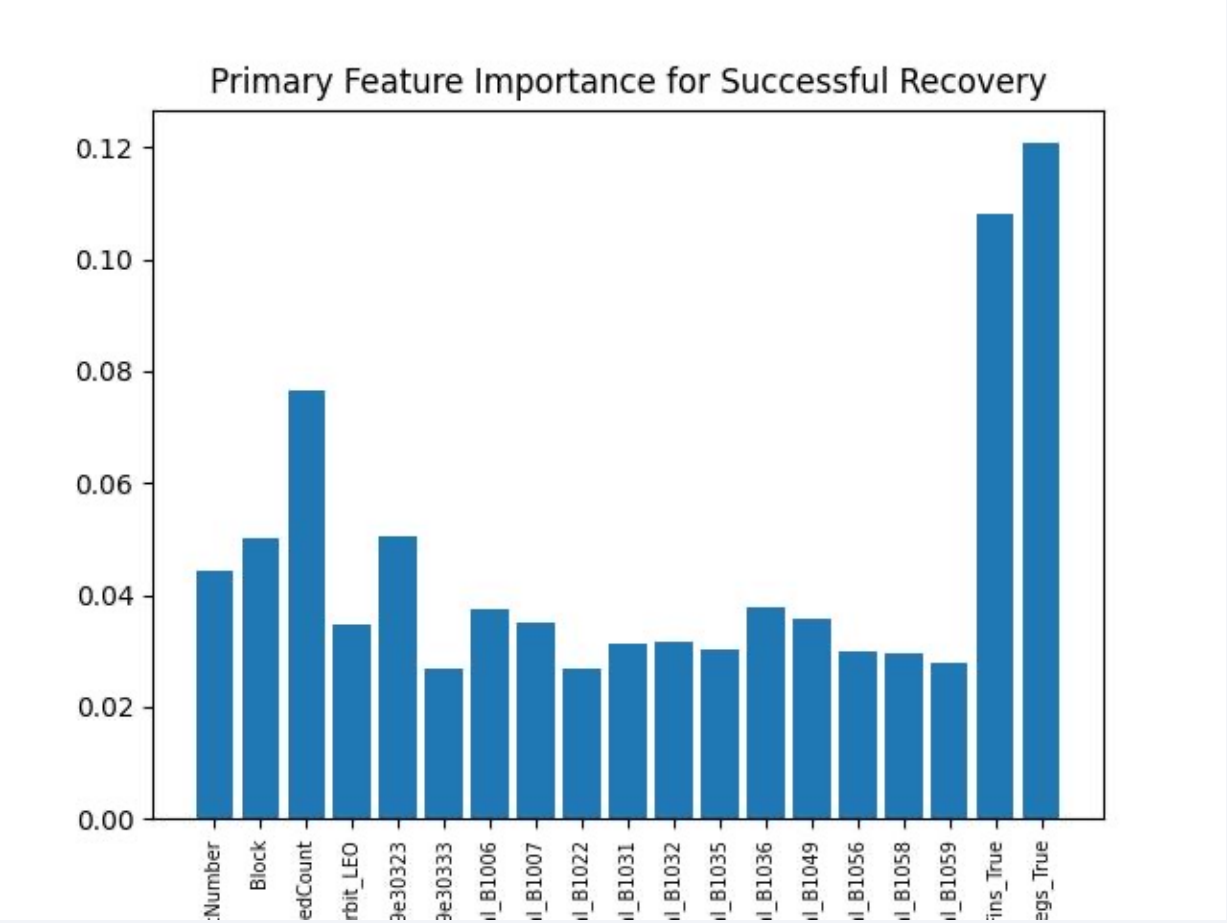
Confusion Matrix

- The confusion Matrix for the Logistic Regression model shows all 12 of 12 True Positives of the test sample were predicted correctly.
- 3 of the Failed recoveries were mis-classified as Fase Positives.
- This represents an 80% positive predictive confidence in the model to predict on onseen data



Conclusions

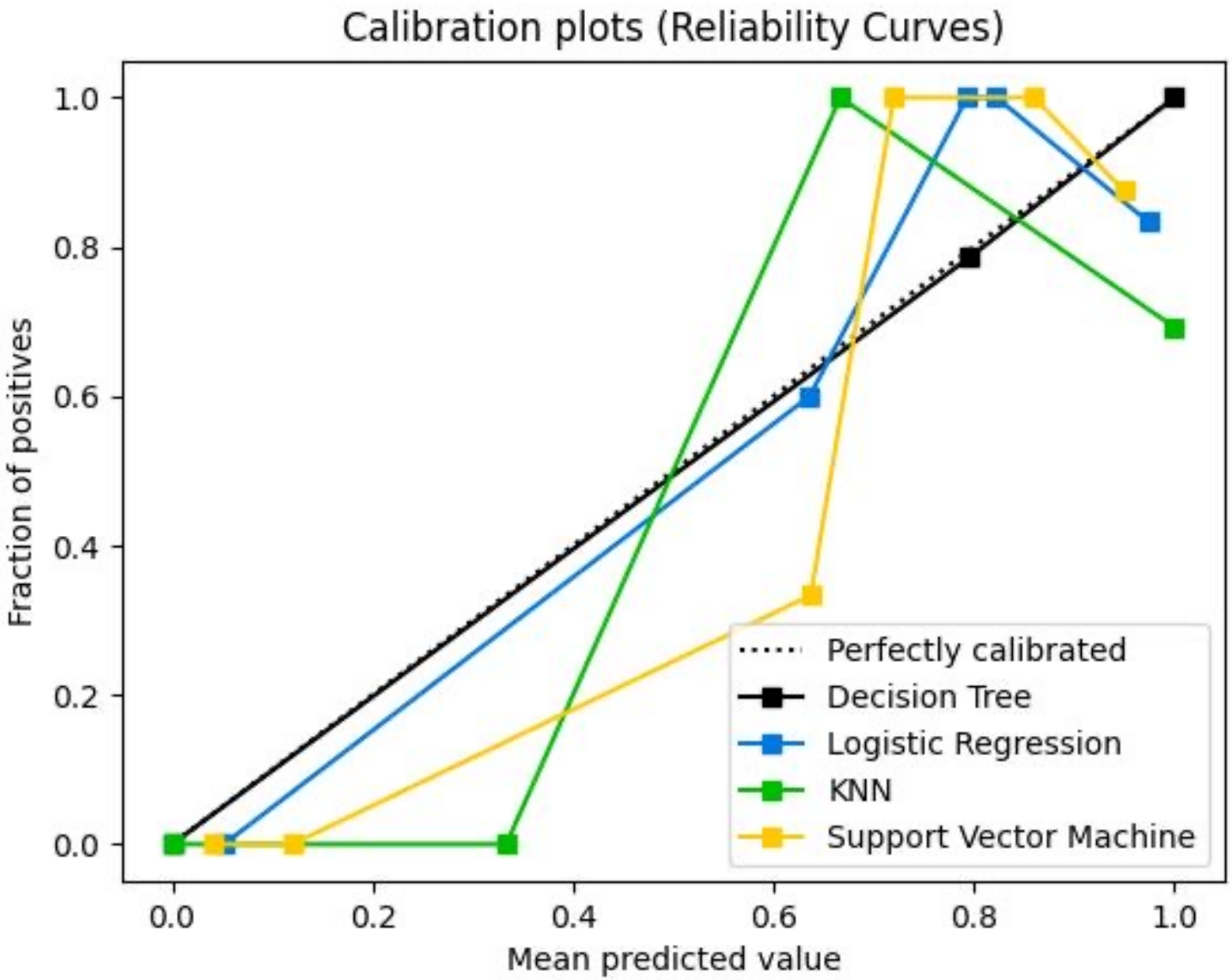
- Grid Fins and Legs are the most predictive factor for successful recovery of the Falcon 9 Booster
- This is followed by the re-use count of the booster. 'It worked before..'
- We can predict successful recovery of the Falcon 9 boosters with ~80% accuracy and confidence
- Recovery rates will likely continue to increase over time.



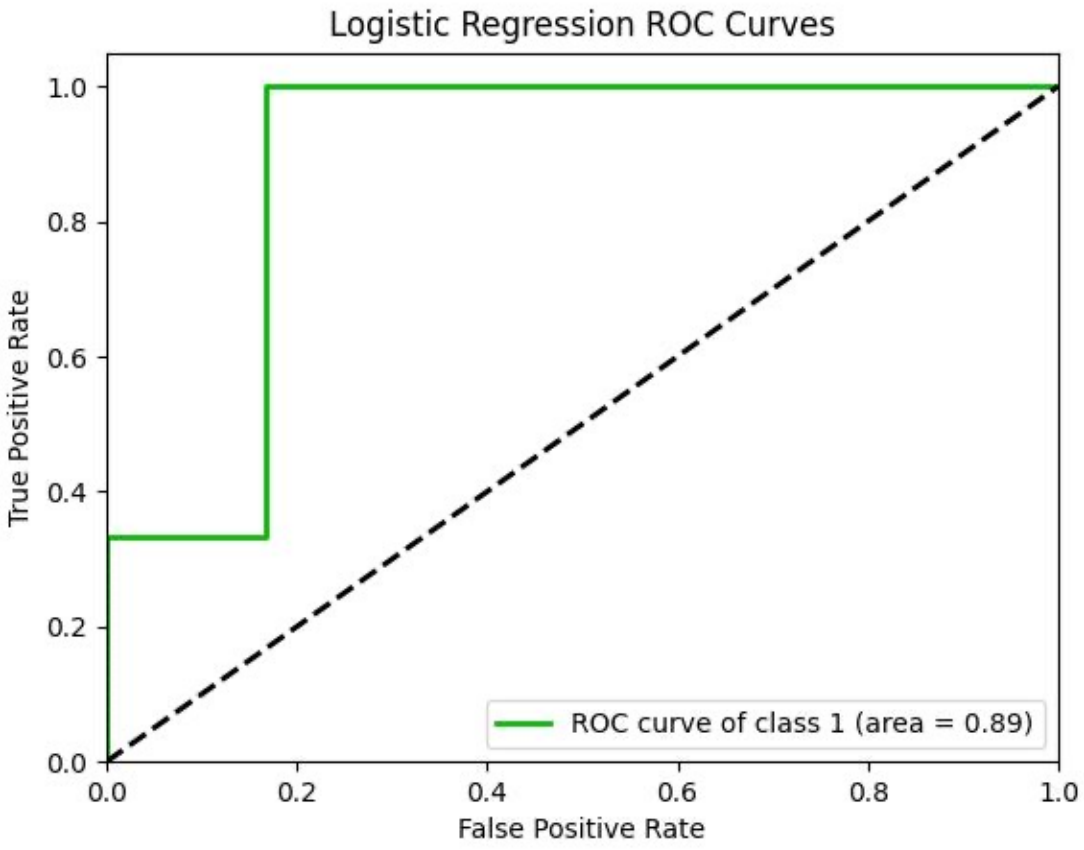
Appendix

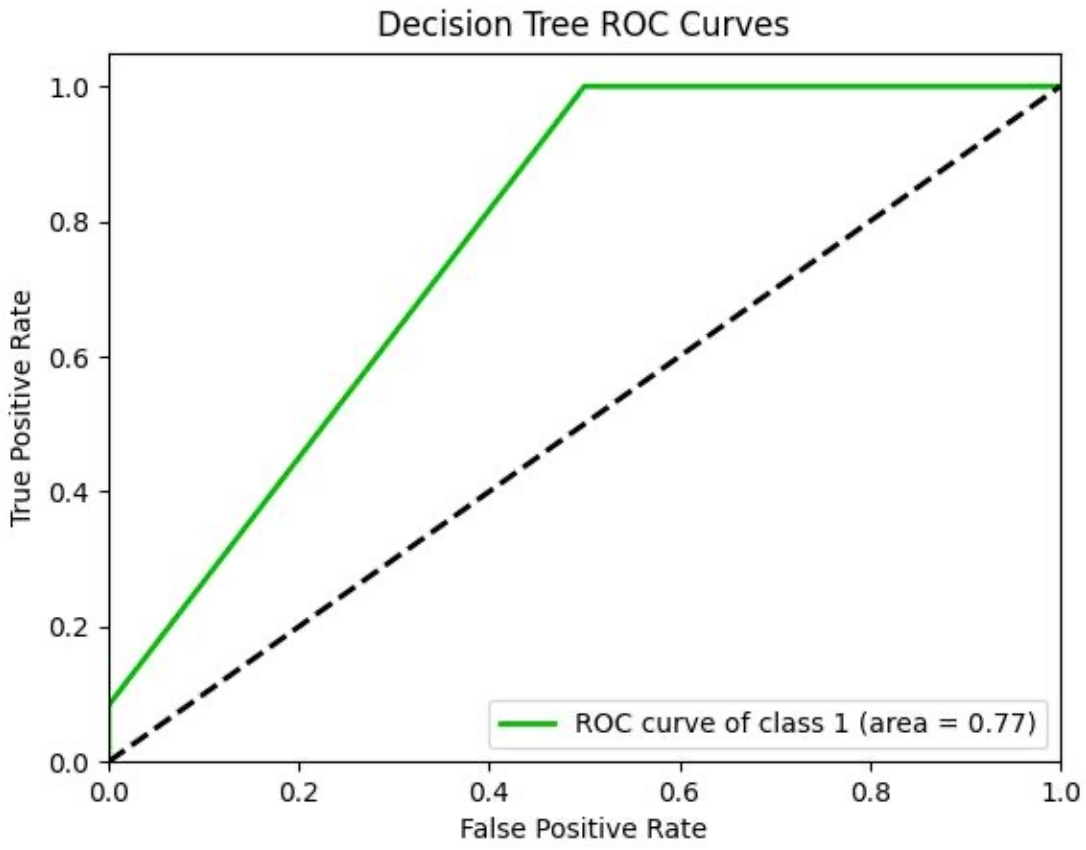
- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Model Calibration Comparison

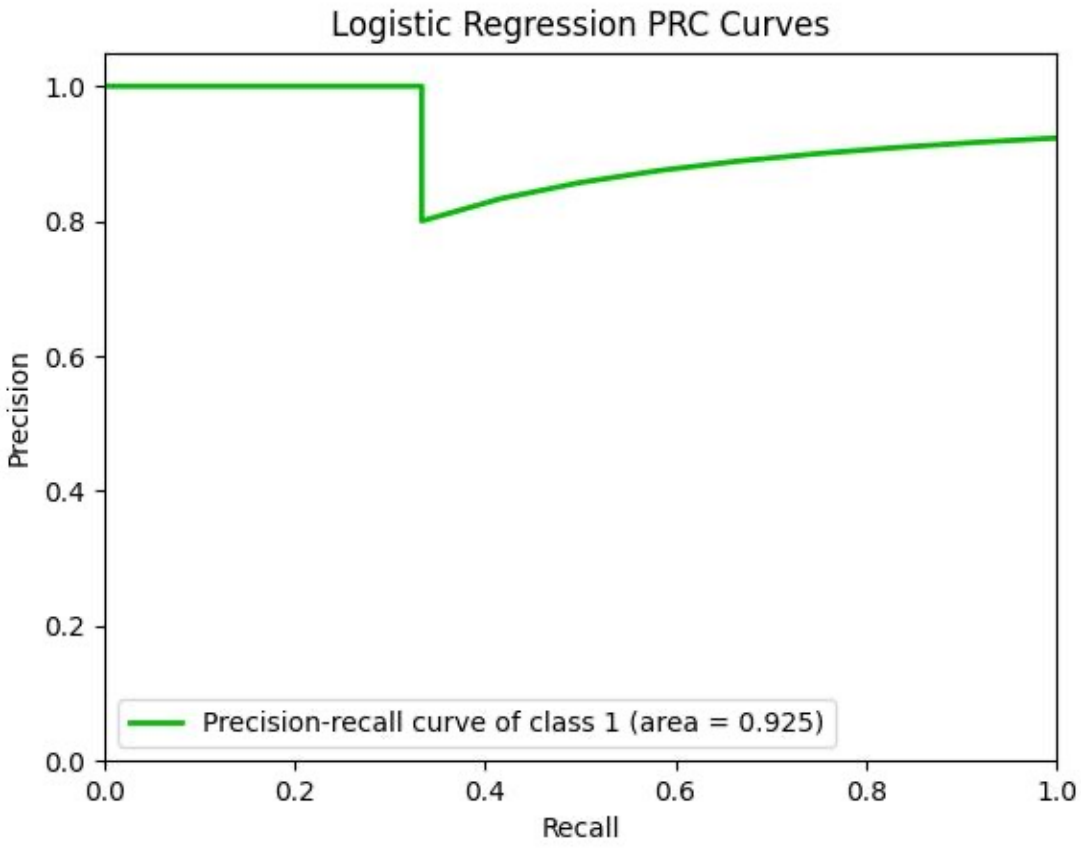


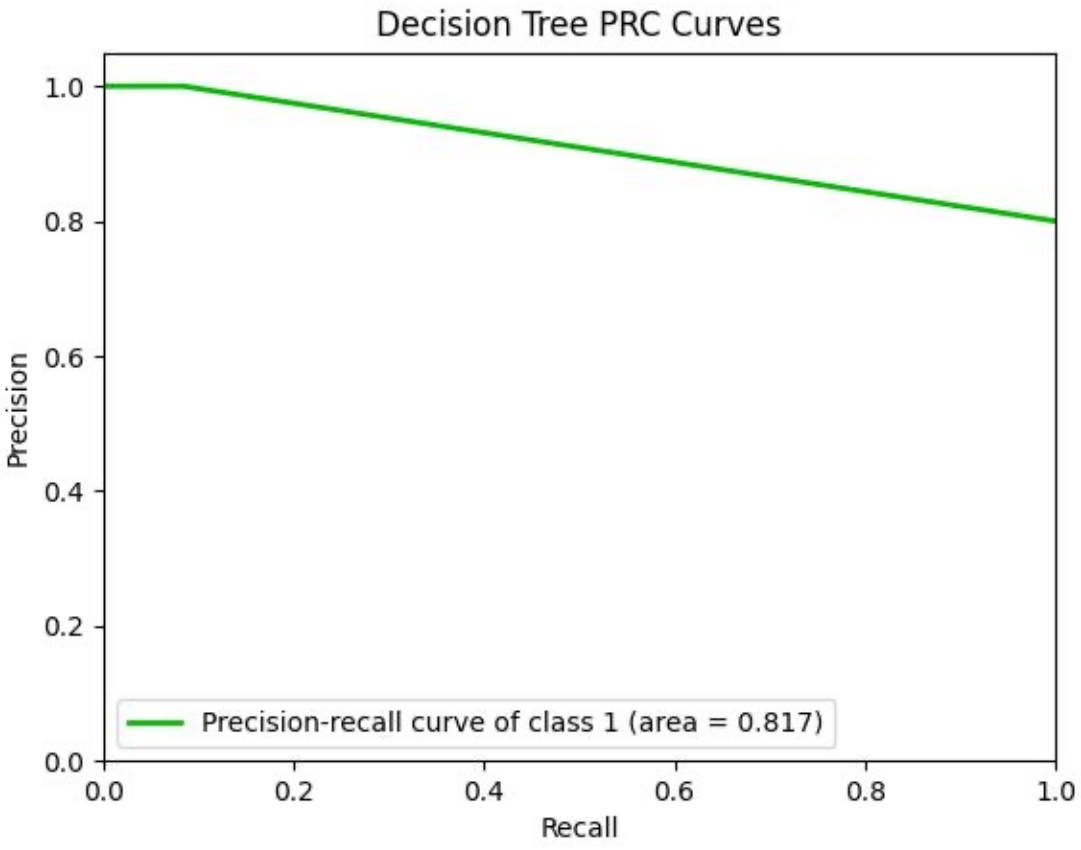
Model Calibration Comparison





Model Calibration Comparison





Thank you!