IBM Developer
SKILLS NETWORK

WINNING THE SPACE RACE

WITH DATA SCIENCE

Zachary Dunnell
September 9, 2022
ztdpro@outlook.com

OUTLINE

1. Executive Summary
2. Introduction
3. Methodology
4. Results from EDA
5. Launch Site Proximity Analysis
6. Building a dashboard
7. Predictive Analysis
8. Conclusion
9. Appendix

EXECUTIVE SUMMARY

- SpaceX Data Collection API
  - Normalized
  - Cleaned
  - Stored in Db2
- Queried with SQL
- Plotted and Visualized
  - Scatter Plots, Line Graphs, Pie Charts, etc.
  - Location map w/ Folium
  - Interactive dashboard
- Predictive Analysis
  - K-Nearest Neighbor
  - Decision Tree
  - Support Vector Machine
  - Logistic Regression

Ability Gained:

Having done all this, it will be possible to predict the outcome of a landing based on any given set of variables to a reasonably accurate degree.

Application:

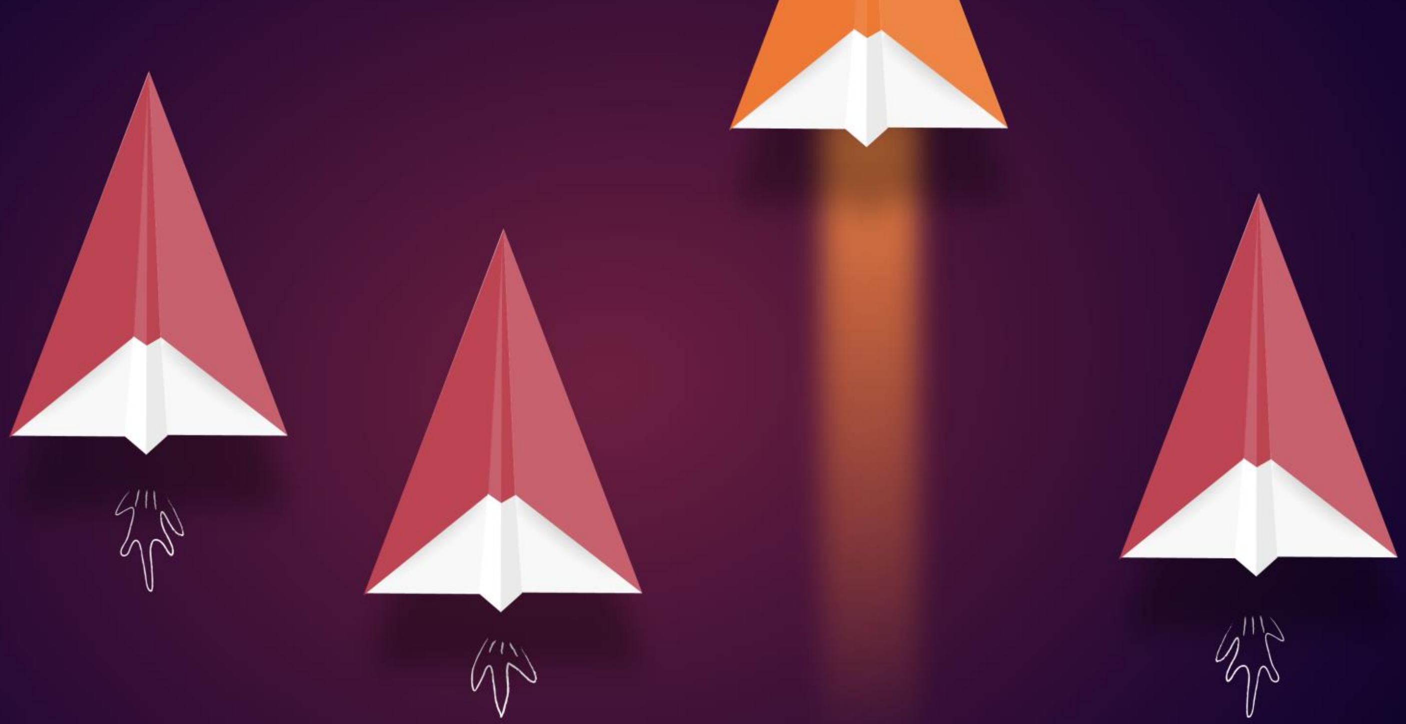
With this knowledge, it should be possible to choose only missions with the highest chance of success, thereby only bidding on profitable missions.

Introduction

The race is on!

Once again, humanity has turned its sights to the stars. In recent years, the new space race has been dominated by private companies: SpaceX, Blue Origin, Virgin Galactic – the list goes on.

Now, a newcomer has arrived!



Billionaire Allon Mask has started SpaceY with the intention of outbidding his counterpart at SpaceX on future rocket launches!

INTRODUCTION

BUT WHAT DOES HE NEED TO KNOW?

- Which launch site has the largest successful launches?
- Which site has the highest launch success rate?
- Which payload range has the highest launch success rate?
- Which payload range has the lowest launch success rate?
- Which Falcon 9 booster version has the highest launch success rate?

METHODOLOGY

Section 1

Methodology:

- Data Collection:
  - Using SpaceX REST API to collect, filter, and store data in Python
  - Finding and replacing missing values using mean() + replace() methods

- Data Wrangling:
  - Exploring data types and variable distribution; what relationships exist between variables
  - Finding the target variable and creating a class; converting the data to fit the class

- Exploratory Data Analysis:
  - Querying with SQL
  - Visualization with scatter plots and other charts

- Building an Interactive Map with Folium:
  - Marking locations and their proximities

- Building a Plotly Dashboard:
  - Which charts were used and why

- Predictive Analysis:
  - Preparing and splitting the data
  - Building a predictive model

Data Collection

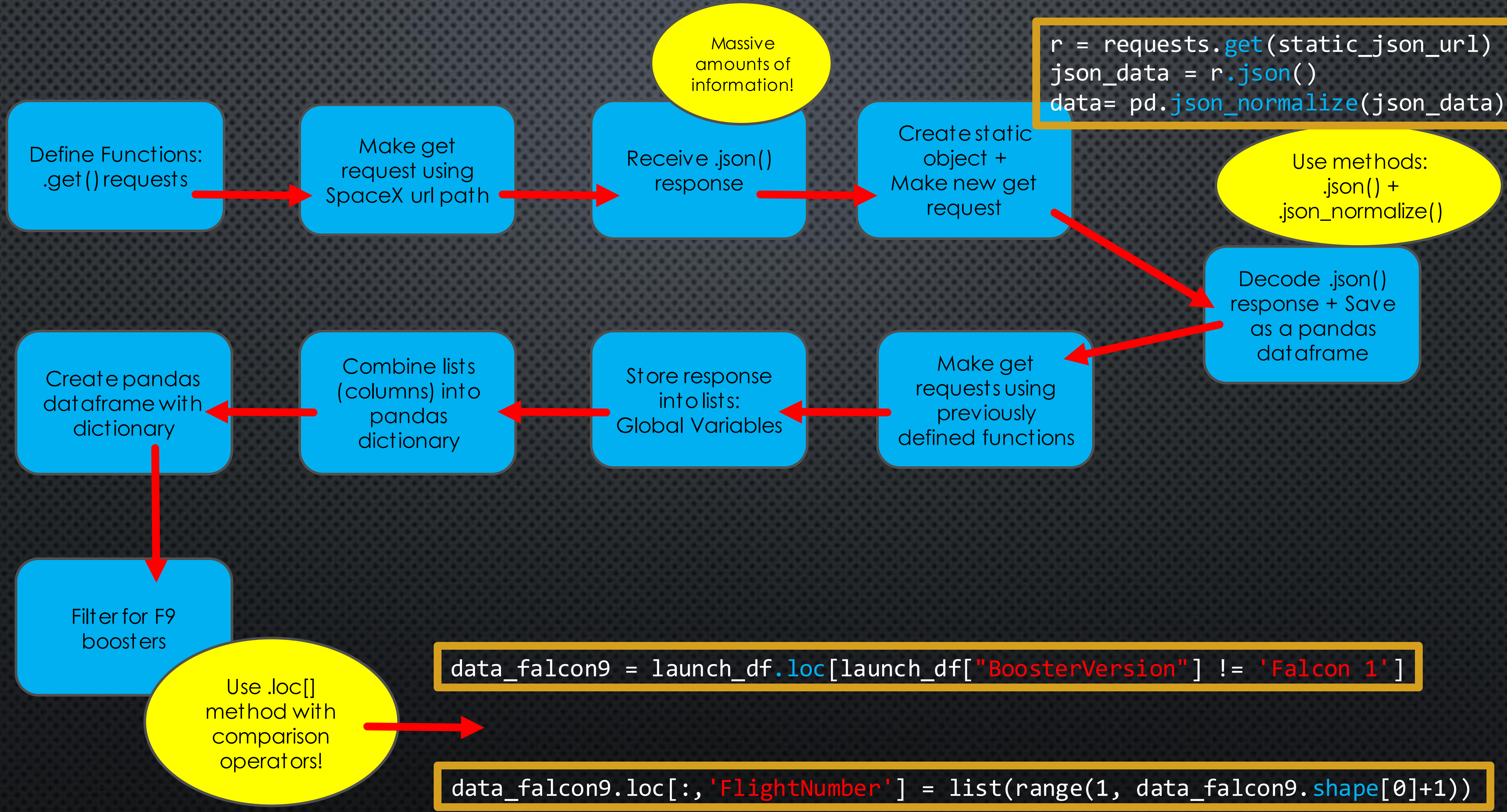
Link to GitHub repository:

https://github.com/ZTD273/AppliedDataScience

Link to view notebook directly using Jupyter Notebook Viewer:

https://nbviewer.org/github/ZTD273/AppliedDataScience/blob/ff47e1e2f111fa1387f733914a545fa0e2b2106c/Data%20Collection%20API.ipynb

Data Collection:




```
r = requests.get(static_json_url)
json_data = r.json()
data= pd.json_normalize(json_data)
```

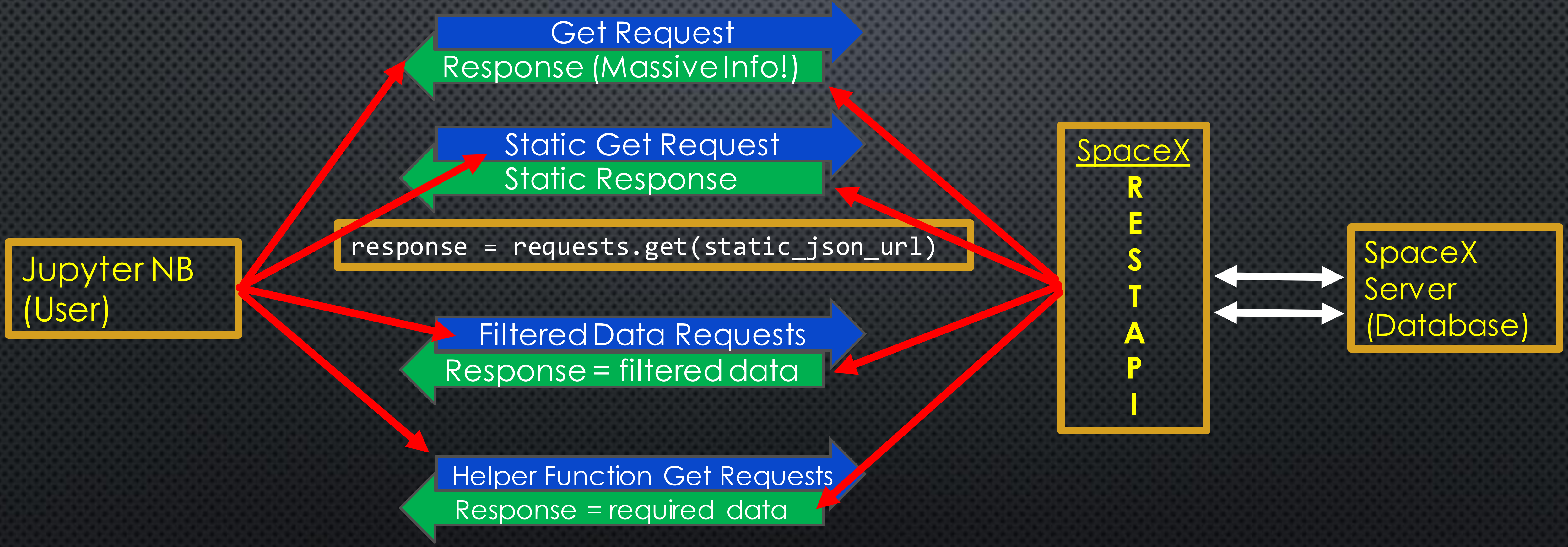
```
data_falcon9 = launch_df.loc[launch_df["BoosterVersion"] != 'Falcon 1']
```

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
```

Data Collection: SpaceX API

Use the REST API to make .get() requests for relevant data

- Helper functions = previously defined functions



Data Collection: Handle Missing Values

Check for missing values

```
data_falcon9.isnull().sum()
```

```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        5
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int6
```

Calculate average value of the column

```
avg_PayloadMass = data_falcon9["PayloadMass"].astype("float").mean(axis=0)
```

Replace missing values with calculated mean

```
data_falcon9["PayloadMass"].replace(np.nan, avg_PayloadMass, inplace=True)
```

Double-check for success!

```
data_falcon9.isnull().sum()
```

```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        0
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
```

(. . .)

Data Wrangling

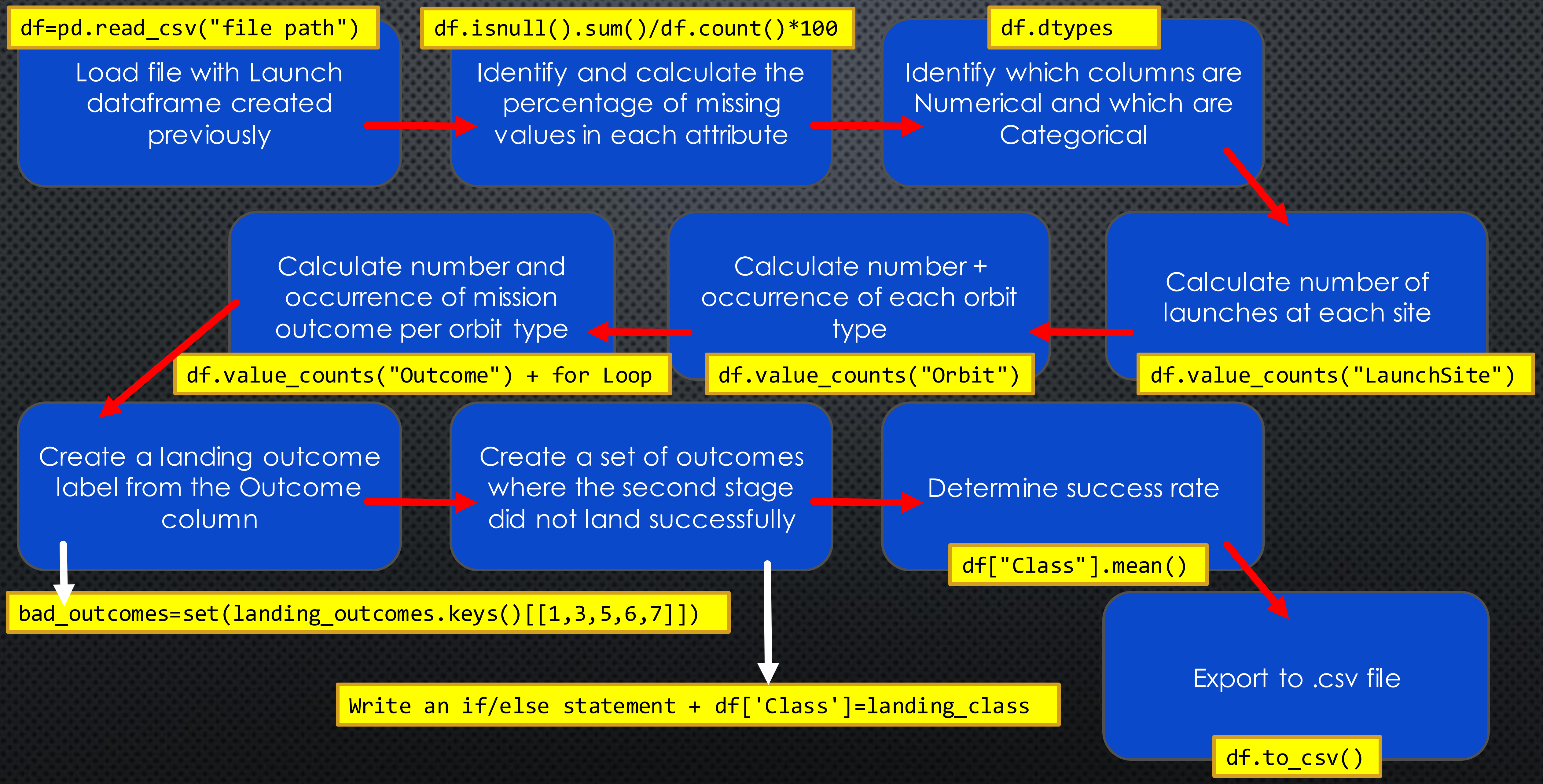
Link to GitHub repository:

https://github.com/ZTD273/AppliedDataScience

Link to view notebook directly using Jupyter Notebook Viewer:

https://nbviewer.org/github/ZTD273/AppliedDataScience/blob/ff47e1e2f111fa1387f733914a545fa0e2b2106c/Data%20Wrangling.ipynb

Data Wrangling:



Exploratory Data Analysis

Link to GitHub repository:

https://github.com/ZTD273/AppliedDataScience

Link to view notebook directly using Jupyter Notebook Viewer: with SQL:

https://nbviewer.org/github/ZTD273/AppliedDataScience/blob/ff47e1e2f111fa1387f733914a545fa0e2b2106c/EDA%20with%20SQL.ipynb

Link to view notebook directly using Jupyter Notebook Viewer: with Visualization:

https://nbviewer.org/github/ZTD273/AppliedDataScience/blob/ff47e1e2f111fa1387f733914a545fa0e2b2106c/EDA%20with%20Visualization.ipynb

EDA with SQL:

Task	SQL used
Find unique launch sites	DISTINCT clause
Display 5 records starting with string 'CCA'	LIKE predicate
Find total payload mass carried by boosters launched by NASA (CRS)	sum() + WHERE w/ LIKE
Find average payload mass carried by Falcon9 booster v1.1	avg() + WHERE w/ LIKE
When was the first successful landing achieved?	min() + WHERE =
Find which boosters have success in drone ship while carrying a payload mass between 4,000kg and 6,000kg	DISTINCT + WHERE, AND, BETWEEN
What is the total number of successful missions? Total of failed missions?	SELECT as, count()
Which booster versions have carried the maximum payload?	WHERE = (subquery)
Find which records showing failure in drone ship during year 2015. Include: month, booster version, and launch site	WHERE = AND
Rank the count of successful landing outcomes between June 4, 2010 and March 20, 2017 in descending order	GROUP BY, ORDER BY, desc

EDA with Visualization:

Chart Type	Purpose of use
1. Scatter Plots:	• What is the relationship between Payload Mass and Flight Number? • What is the relationship between Flight Number and Launch Site? • What is the relationship between Payload Mass and Launch Site? • What is the relationship between Flight Number and Orbit type? • What is the relationship between Payload Mass and Orbit type?
2. Bar Graph:	• What is the success rate for each Orbit type?
3. Line graph:	• How has the success rate changed over the years?

Interactive Maps with Folium

Link to GitHub repository:

https://github.com/ZTD273/AppliedDataScience

Link to view notebook directly using Jupyter Notebook Viewer:

https://nbviewer.org/github/ZTD273/AppliedDataScience/blob/5b5058beda39663548f6086e1264f14d4877e0fe/Launch%20Site%20Location%20Maps%20%28Folium%29.ipynb

Building an Interactive Map using Folium:

Locations Mapped: *Marked with circles*

CCAFS SLC-40 (Cape Canaveral Space Launch Complex)

CCAFS LC-40 (Cape Canaveral Launch Complex)

KSC LC-39A (Kennedy Space Center Launch Complex)

VAFB SLC-4E (Vandenberg Air Force Base Space Launch Complex)

Other Objects Mapped:

Failed and Successful Launches at each location

Mouse position on the map

Distance lines from each location to the nearest:

- City
- Railroad
- Highway
- Coast

Methods Used:

Folium.circle()

Folium.marker()

Marker_cluster()

Mouse_position()

Folium.Polyline()

Site_map.add_child()

Why?

Marking these objects gives at least a preliminary idea as to what is necessary for deciding where to build a launch site, as well as giving consideration to how easily parts and labor can reach the site.

Plotly Dashboard

Link to GitHub repository:

https://github.com/ZTD273/AppliedDataScience

Direct Link to dashboard code file:

https://github.com/ZTD273/AppliedDataScience/blob/ab8ba2abc0548fe903fd1f848bec35c27f89444d/SpaceX_Dash_App.py

Plotly Dashboard: Includes:

- Pie Chart: Successful outcomes for all launch sites
- Drop-down menu to choose a specific launch site
- Pie Chart: Total successes and failures for a selected launch site
- Payload range slider: 0 kg -> 10,000 kg
- Scatter plot: All successful and failed missions compared to payload mass
  - Default is all launches, all sites
  - Drop-down menu will narrow data to selected launch site

- Compare total successes between sites
- Added to specify returned data
- Specific site data to find success rate
- Narrows data to find most successful payload mass
- Gives a detailed look at how the size of a payload may affect a mission's success

Predictive Analysis

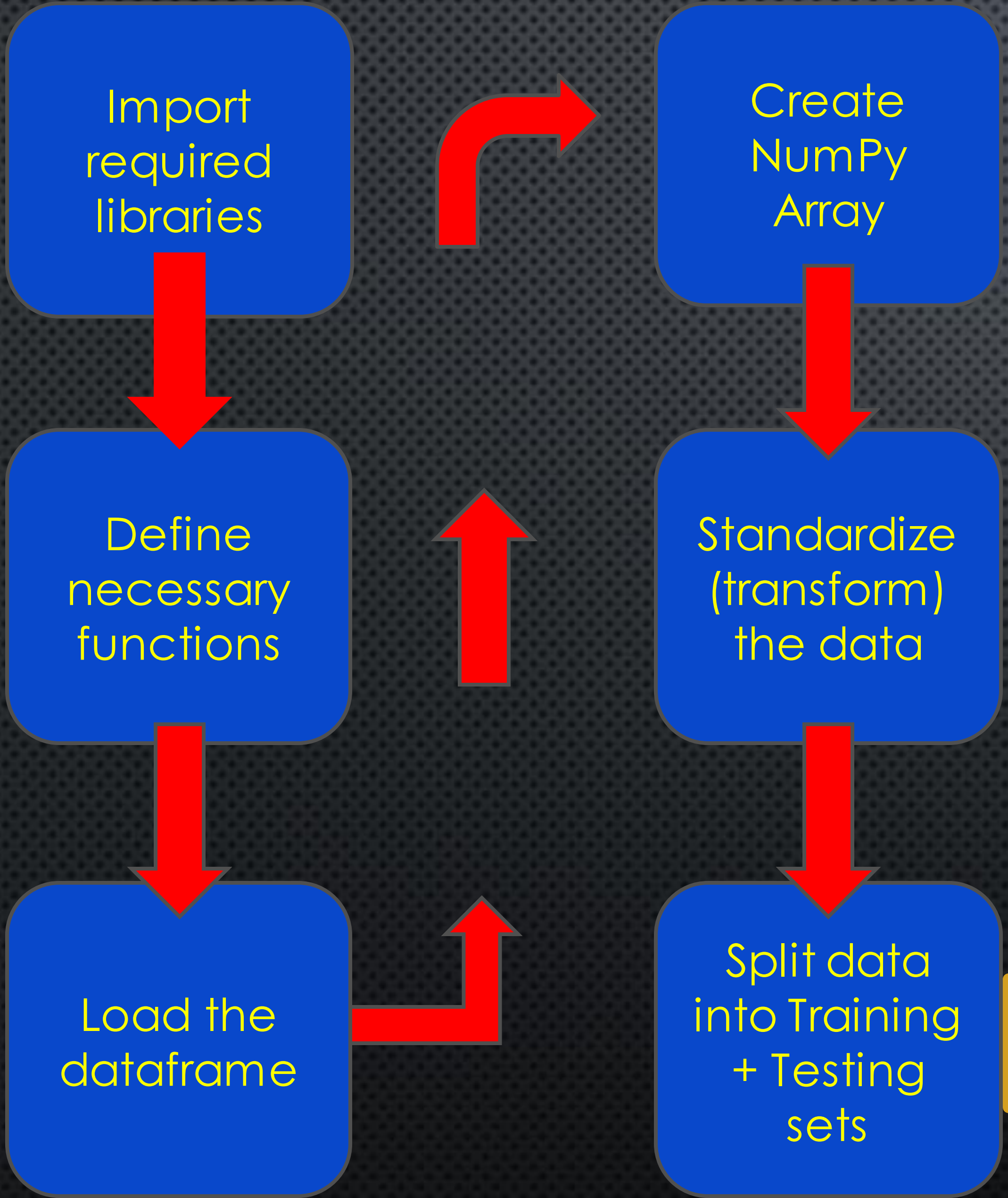
Link to GitHub repository:

https://github.com/ZTD273/AppliedDataScience

Direct Link to dashboard code file:

https://nbviewer.org/github/ZTD273/AppliedDataScience/blob/ff47e1e2f111fa1387f733914a545fa0e2b2106c/Machine%20Learning%20Predictions.ipynb

Prepare the Data:

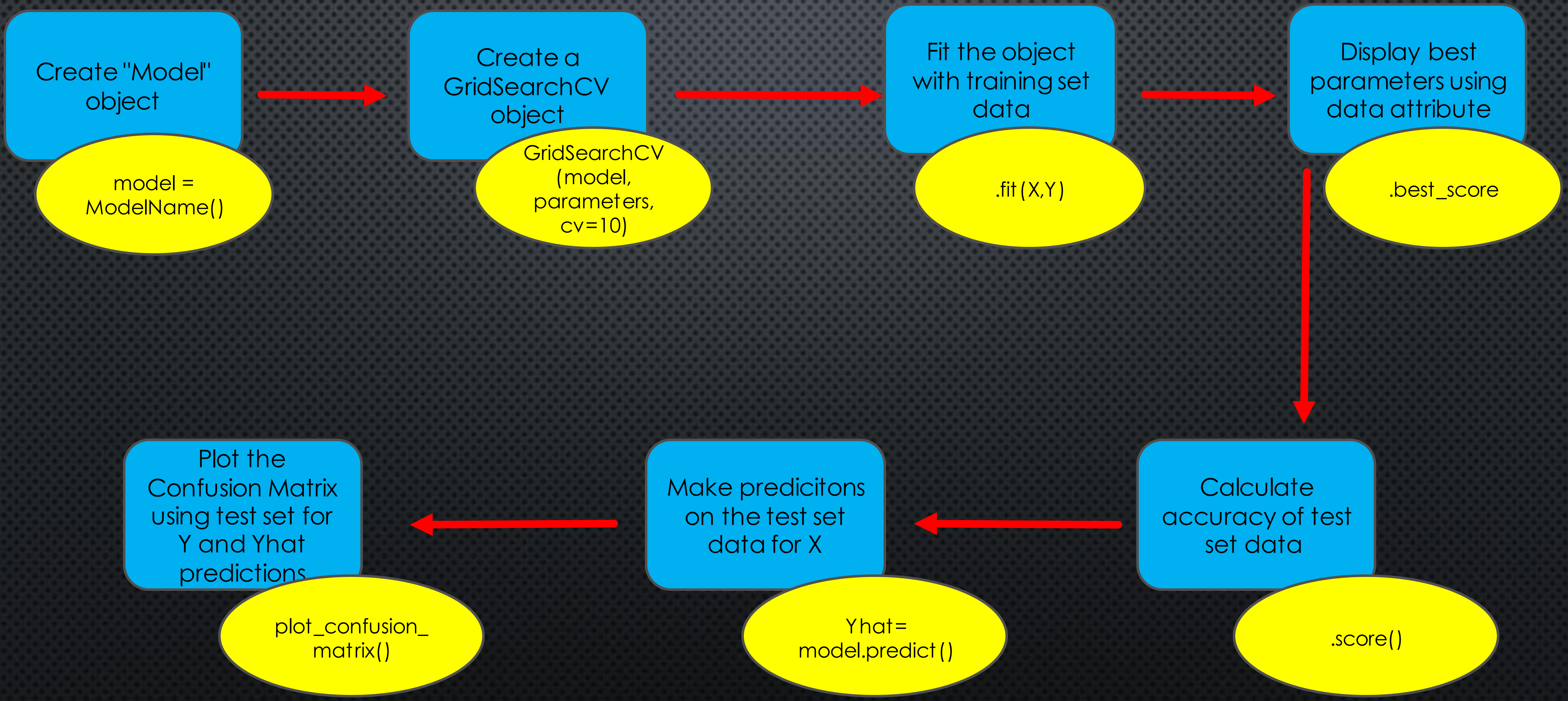



```
Y = data['Class'].to_numpy()
```

```
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
```

```
x_train, x_test, y_train, y_test =
train_test_split(X, y, test_size=0.2, random_state=2)
```

Building a Predictive Model:



RESULTS FROM EDA

Section 2

ALL LAUNCH SITE NAMES

```
select distinct(LAUNCH_SITE)
from SPACEXTBL
```

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

LAUNCH SITE NAMES BEGIN WITH 'CCA'

```
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
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

TOTAL PAYLOAD MASS

```
select sum(PAYLOAD_MASS__KG_)
as "Total Payload Mass in KG launched by Nasa (CRS)"
from SPACEXTBL where "Customer" like "NASA%CRS%"
```

Total Payload Mass in KG launched by Nasa (CRS)
48213

AVERAGE PAYLOAD MASS BY F9 V1.1

```sql
select avg(PAYLOAD_MASS__KG_)
as "Average Payload Mass carried by F9v1.1 Booster"
from SPACEXTBL where "Booster_Version" like "F9%v1.1%"
```

Average Payload Mass carried by F9v1.1 Booster
2534.6666666666665

FIRST SUCCESSFUL GROUND LANDING DATE

```
select min("Date")
as "First Successful Landing on Ground Pad"
from SPACEXTBL
where "LANDING _OUTCOME" = "Success (ground pad)"
```

First Successful Landing on Ground Pad
01-05-2017

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

```
select distinct("Booster_Version")
as "Successful Boosters of the Given Parameters"
from SPACEXTBL
where "LANDING _OUTCOME" = "Success (drone ship)"
and PAYLOAD_MASS__KG_ between 4000 and 6000
```

Successful Boosters of the Given Parameters
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

```
select MISSION_OUTCOME as "Outcome", count(*)
as "Total Missions" from SPACEXTBL
group by MISSION_OUTCOME
```

Outcome	Total Missions
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

BOOSTERS CARRIED MAXIMUM PAYLOAD

```
select "Booster_Version"
as "Boosters That Carried Max Payload" from SPACEXTBL
where PAYLOAD_MASS__KG_ =
(select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

Sub-Query

Boosters That Carried Max Payload
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) as "Month",
"LANDING _OUTCOME" as "Landing Outcome",
"Booster_Version" as "Booster",
LAUNCH_SITE as "Launch Site"
from SPACEXTBL
where "LANDING _OUTCOME" = "Failure (drone ship)"
and substr(Date, 7, 4) = '2015'
```

Note: SQLLite does not support monthnames

Month	Landing Outcome	Booster	Launch Site
01	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

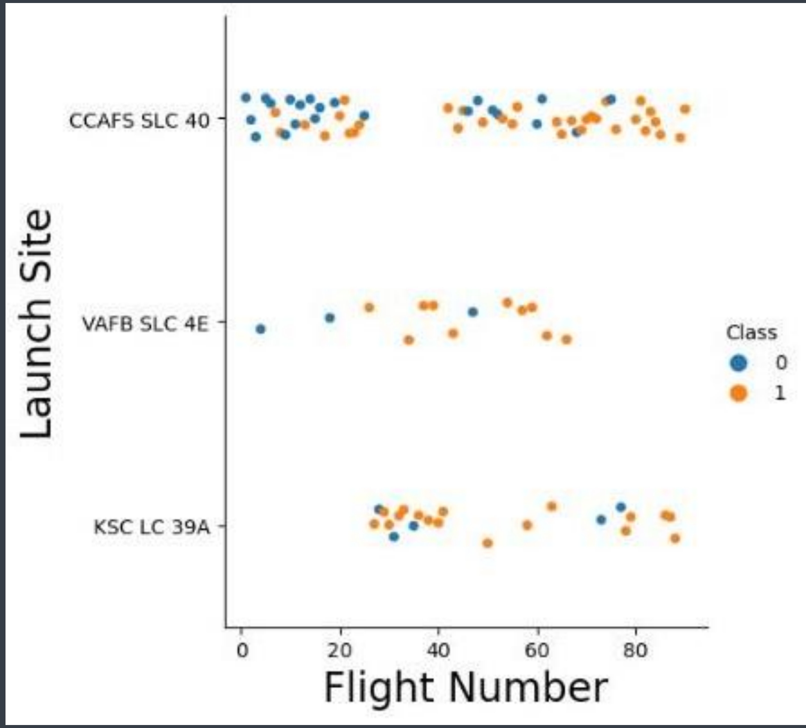
```
select "Date", "LANDING _OUTCOME" as "Outcome",
count("LANDING _OUTCOME") as "Number of Successes"
from SPACEXTBL
where  substr(Date,7,4)||substr(Date,4,2)||substr(Date,1,2)
between '20100604' and '20170320'
group by "LANDING _OUTCOME"
order by count("LANDING _OUTCOME") desc
```

Note: SQLLite does not support monthnames

Date	Outcome	Number of Successes
07-08-2018	Success	20
08-10-2012	No attempt	10
08-04-2016	Success (drone ship)	8
18-07-2016	Success (ground pad)	6
10-01-2015	Failure (drone ship)	4
05-12-2018	Failure	3
18-04-2014	Controlled (ocean)	3
04-06-2010	Failure (parachute)	2
06-08-2019	No attempt	1

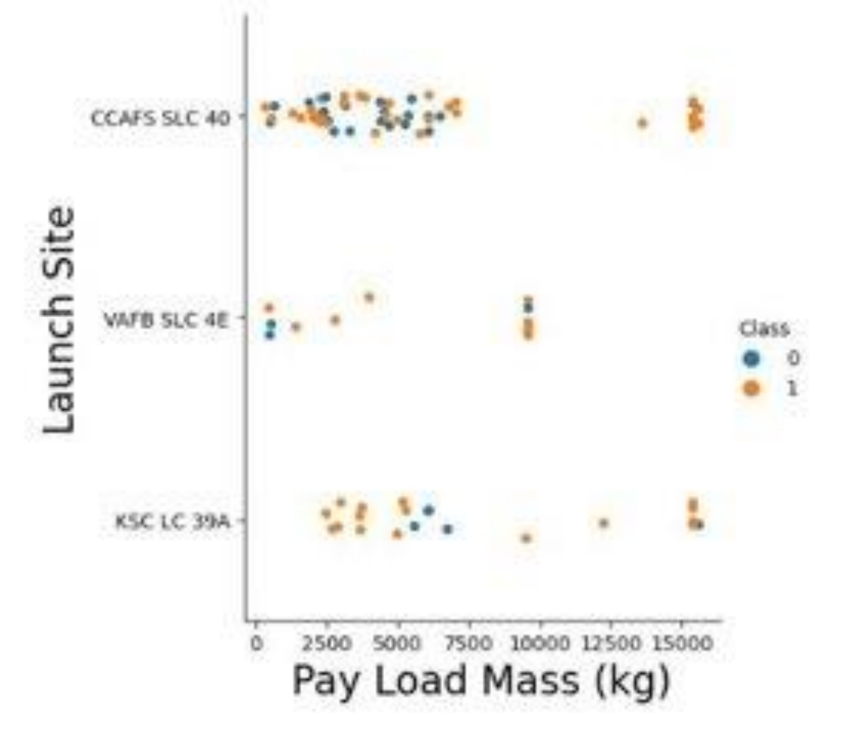
FLIGHT NUMBER VS. LAUNCH SITE

Scatterplot shows the distribution and frequency of launches between sites, as well as how many launches were successes or failures.



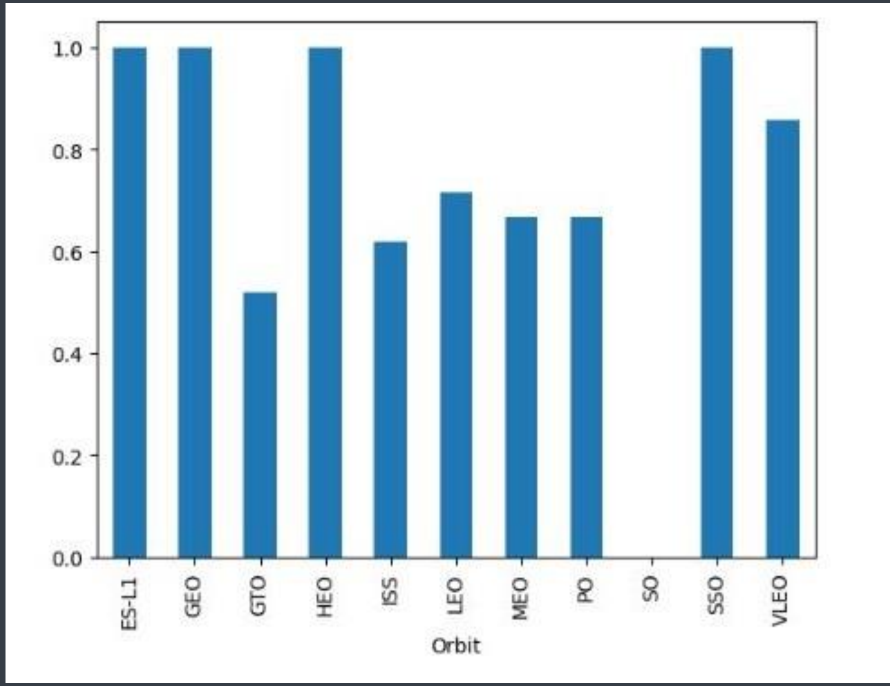
PAYLOAD MASS VS. LAUNCH SITE

Scatterplot shows which sites launch the heaviest payloads, which payload sizes are launched most frequently, and whether they are successful or not.



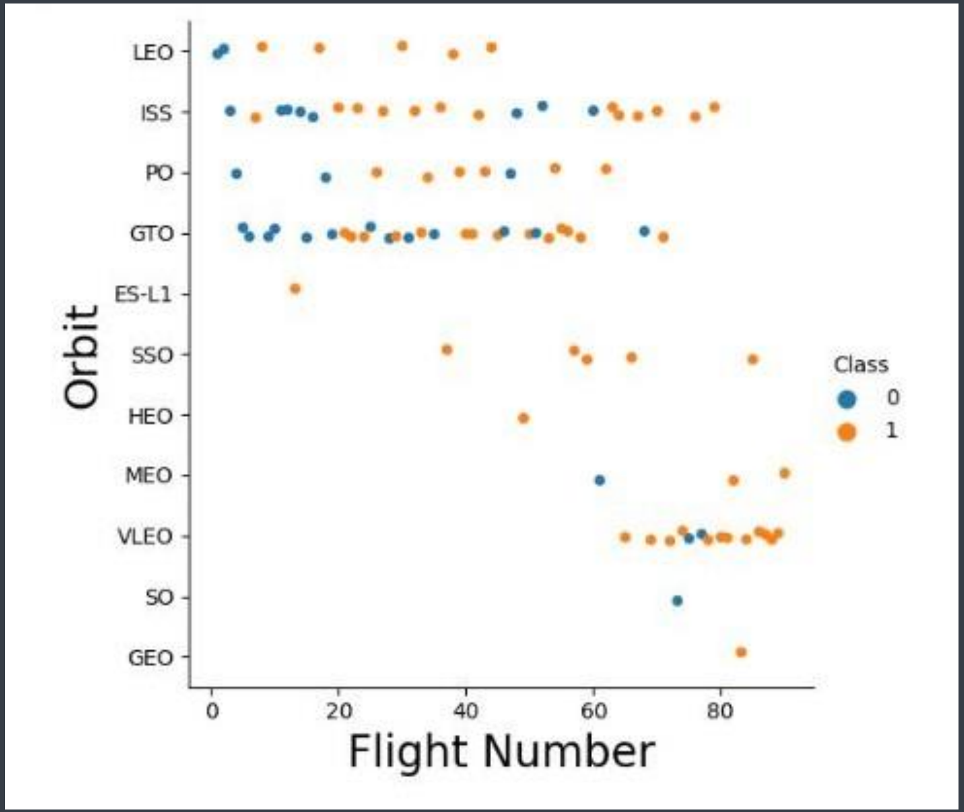
SUCCESS RATE VS. ORBIT TYPE

Bar chart shows which orbits are the most successful.



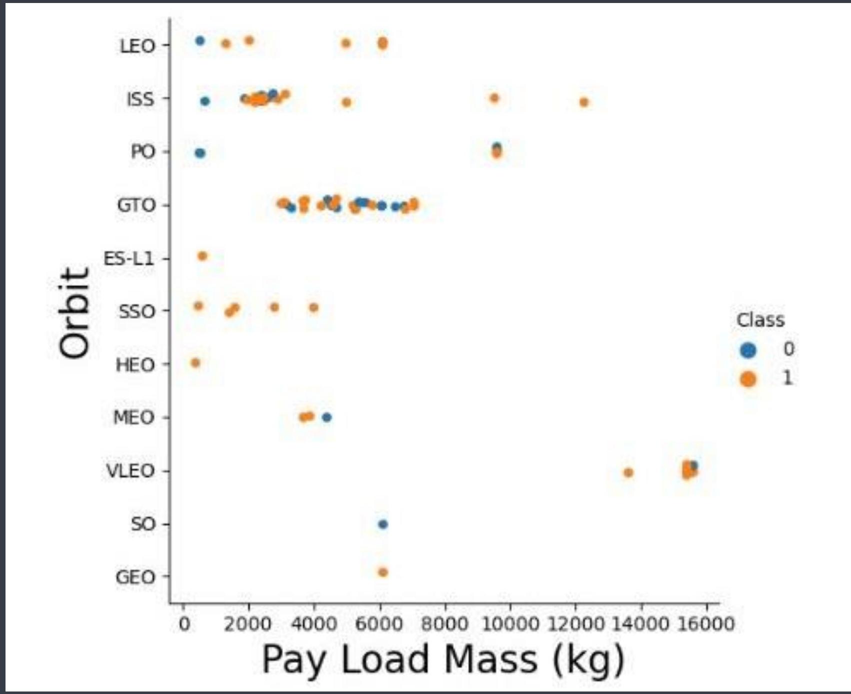
FLIGHT NUMBER VS. ORBIT TYPE

Scatterplot shows how often different orbit types are launched and if they are successful or not.



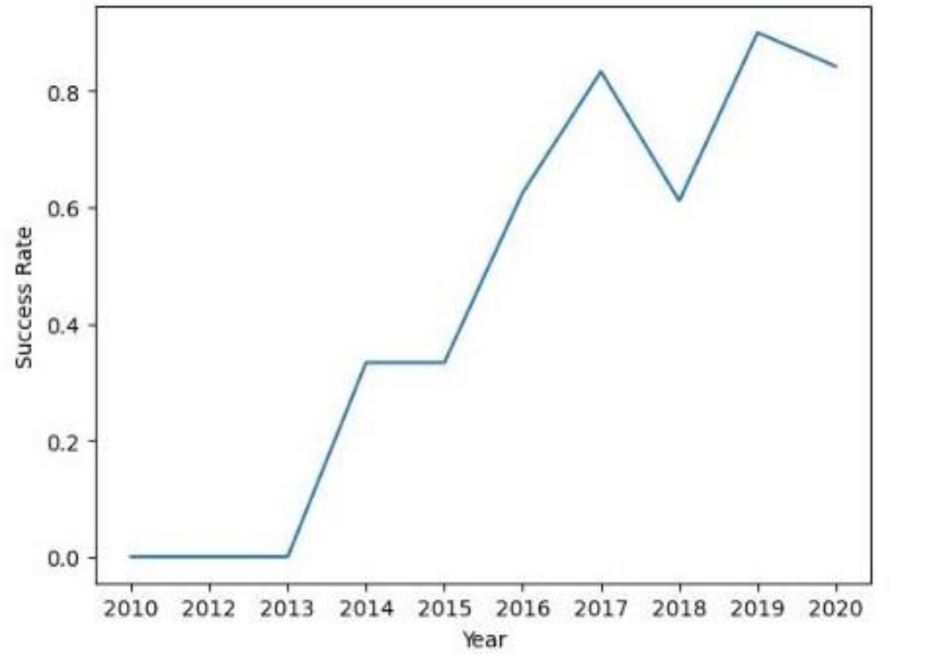
PAYLOAD MASS VS. ORBIT TYPE

Scatterplot shows how often different payload sizes are launched into what orbit and if they are successful or not.



YEARLY TREND OF LAUNCH SUCCESSES

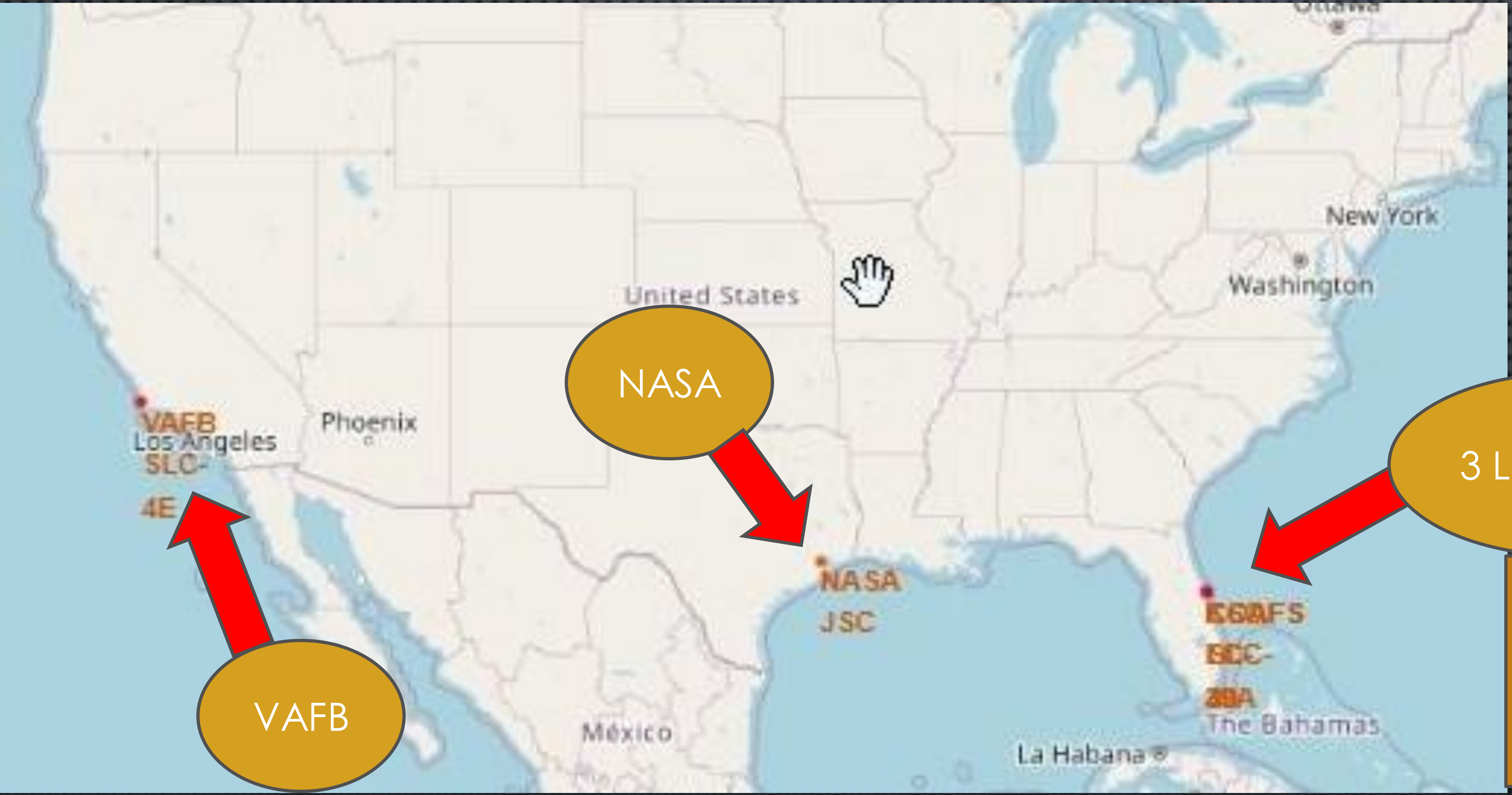
Line graph shows the success rate over time.



LAUNCH SITE PROXIMITY ANALYSIS

Section 3

Launch Site Locations



NASA

VAFB

3 Locations!

KSC LC-39A
CCAFS SLC-40
CCAFS LC-40

Majority of launches take place in Florida near Cape Canaveral

Launch outcomes at each launch site

VAFB SLC-4E

4 Successes
10 Attempts
40% Success

CCAFS LC-40

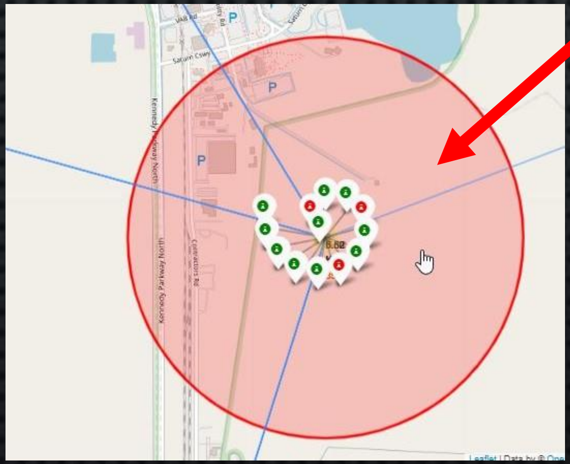
Most launches!

7 Successes
26 Attempts
26.9%
Success

Green = Successful
Red = Failure

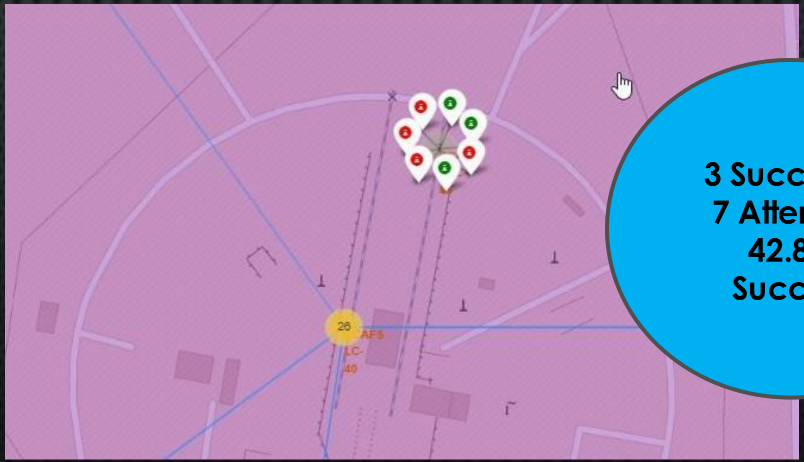
KSC LC-39A

Highest rate of success!



10 Successes
13 Attempts
76.9%
Success

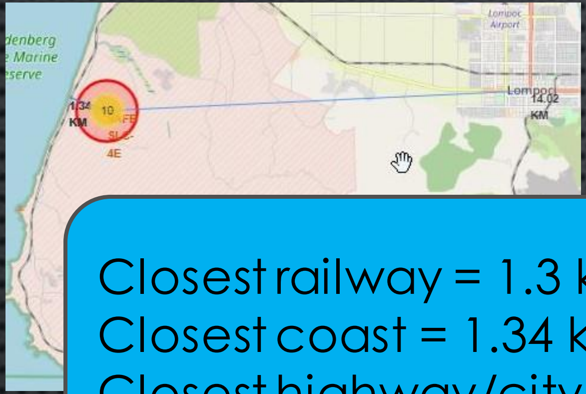
CCAFS SLC-40



3 Successes
7 Attempts
42.8%
Success

Launch site proximities

VAFB SLC-4E



Closest railway = 1.3 km
Closest coast = 1.34 km
Closest highway/city = 14.02 km

Closest railway = 9.4 km
Closest coast = 6.78 km
Closest highway = 5.89 km
Closest city = 26.29 km

Closest railway = 1.33 km
Closest coast = 0.93 km
Closest highway = 6.89 km
Closest city = 18.05 km

KSC LC-39A

CCAFS LC-40

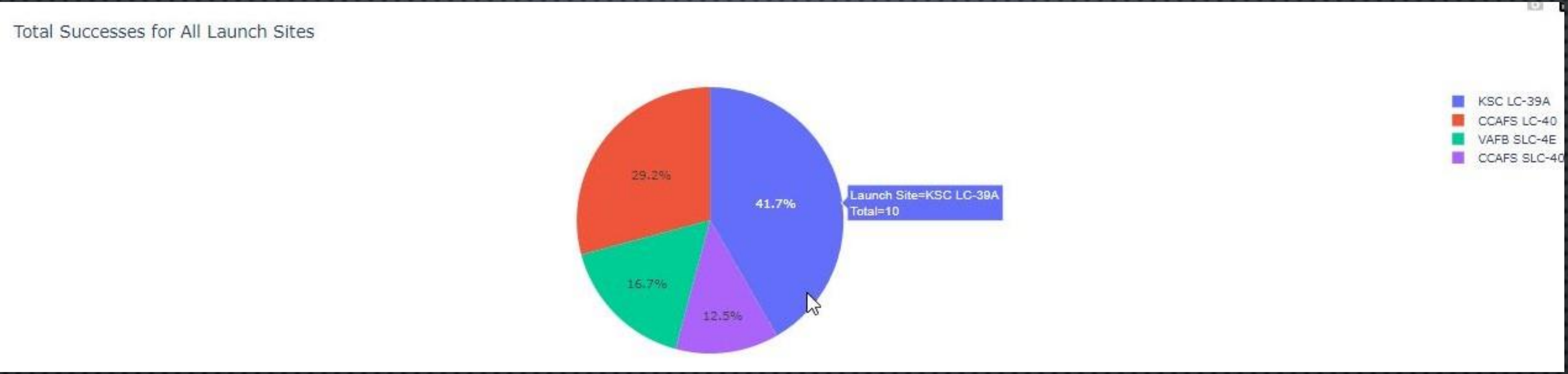
CCAFS SLC-40

Launch sites need to be close to a coast, a railway, and a highway while being far away from population centers.

BUILD A DASHBOARD WITH PLOTLY DASH

Section 4

Number of successful launches: All Launch Sites



Success totals in descending order:

KSC LC-39A = 10 successes

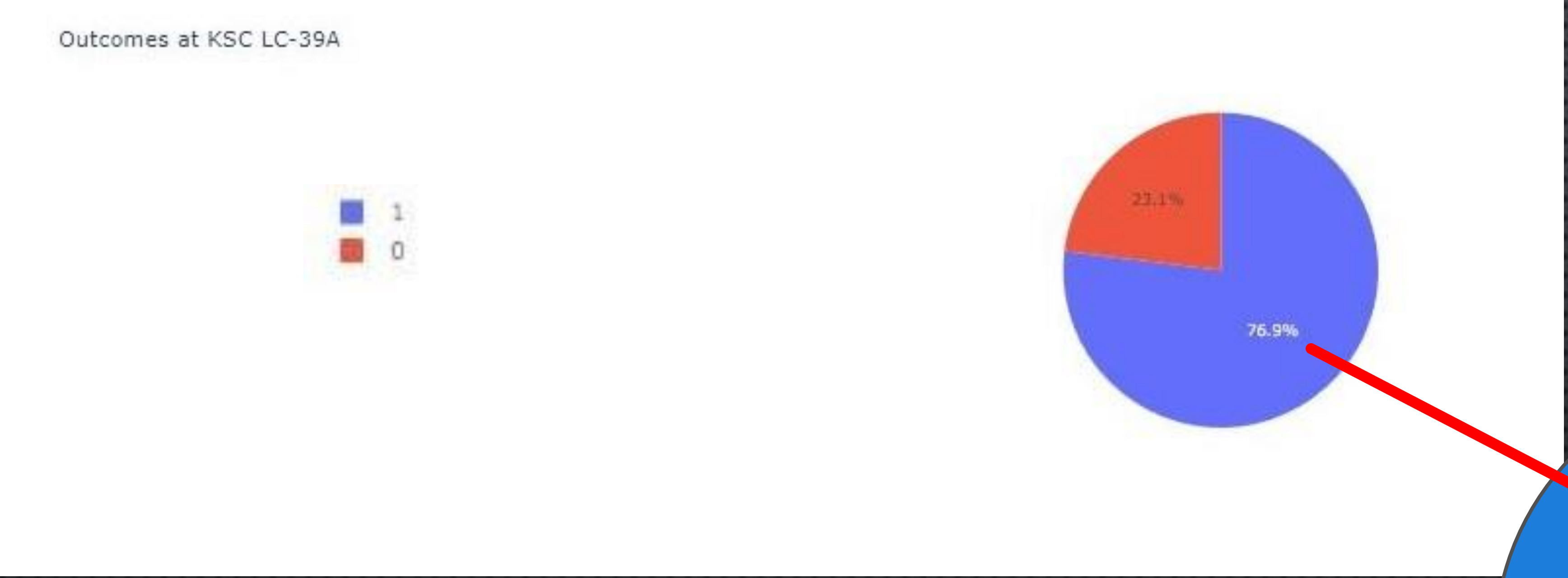
CCAFS LC-40 = 7 successes

VAFB SLC-4E = 4 successes

CCAFS SLC-40 = 3 successes

Which Launch Site has the highest success ratio?

KSC LC-39A : Kennedy Space Center Launch Complex 39-A



Success Ratio = (Successes) / (Attempts)

Success Ratio = 10 / 13

Success Ratio = 0.7692307692

Success =
76.9%

Payload Range vs. Launch Outcome: All Launch Sites

0 kg

2,000 kg

3/11 Succeed

Success= 27.27 %

Payload Range vs. Launch Outcome: All Launch Sites

2,000 kg → 4,000 kg

Best performing Booster = FT

12/20 Succeed

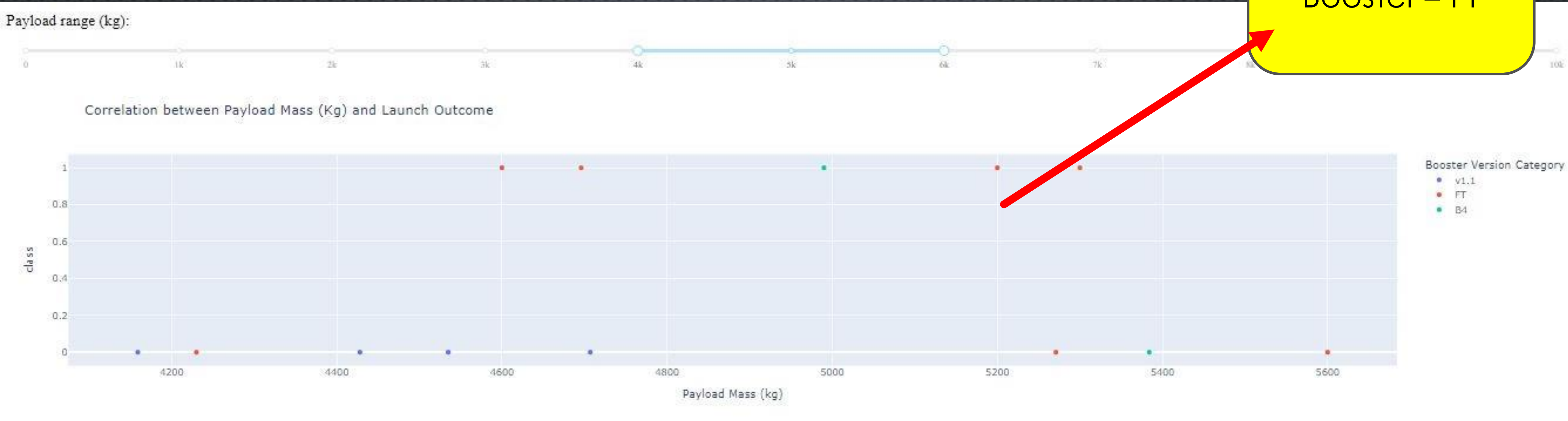
Success= 60 %

Payload Range vs. Launch Outcome: All Launch Sites

4,000 kg

6,000 kg

Best performing Booster = FT



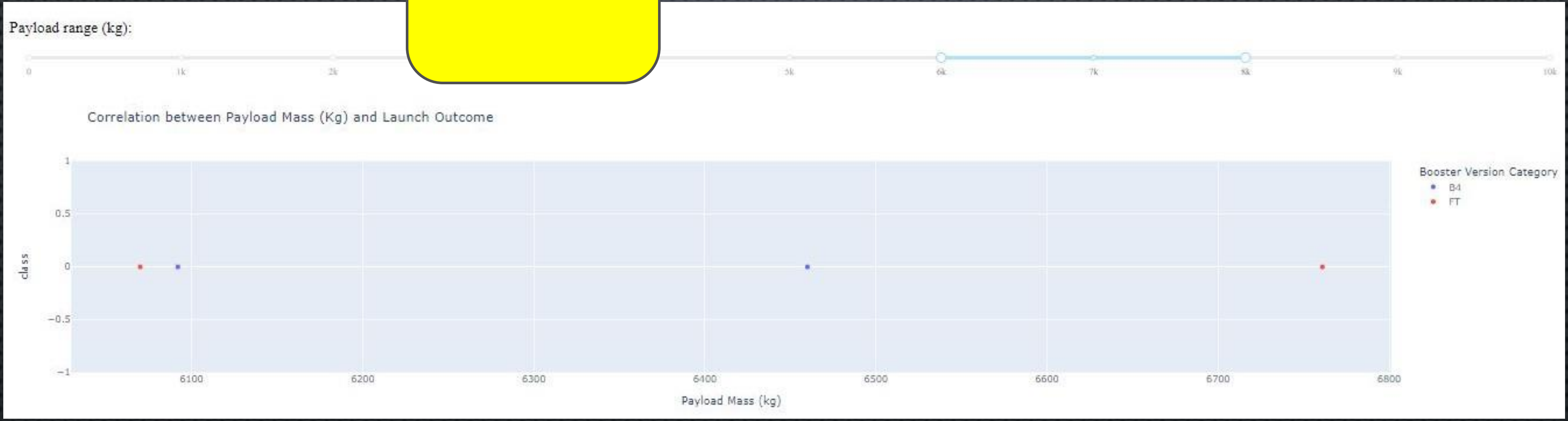
5/13 Succeed

Success= 38.46 %

Payload Range vs. Launch Outcome: All Launch Sites

6,000 kg → 8,000 kg

No Successes



0/4 Succeed

Success= 0.0 %

Payload Range vs. Launch Outcome: All Launch Sites

8,000 kg

10,000 kg

Only Success: Booster B4

1/2 Succeed

Success= 50 %

PREDICTIVE ANALYSIS: CLASSIFICATION

Section 5

LOGISTIC REGRESSION

```
LR = LogisticRegression()
```

```
logreg_cv = GridSearchCV(LR, parameters, cv=10)
```

```
logreg_cv.fit(X_train,Y_train)
```

```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

```
print("tuned hyperparameters:(best parameters)", logreg_cv.best_params_)
print("accuracy:", logreg_cv.best_score_)
```

```
tuned hyperparameters: (best parameters)
{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8472222222222222
```

```
logreg_cv.score(x_test, y_test)
```

```
0.8333333333333334
```

```
log_hat=logreg_cv.predict(x_test)
plot_confusion_matrix(y_test,log_hat)
```

SUPPORT VECTOR MACHINES

```
svm = SVC()
```

```
svm_cv = GridSearchCV(svm, parameters, cv=10)
```

```
svm_cv.fit(X_train,Y_train)
```

```
parameters = {'kernel':
              ('linear','rbf','poly','rbf','sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}
```

```
print("tuned hyperparameters:(best parameters)", svm_cv.best_params_)
```

```
print("accuracy:", svm_cv.best_score_)
```

```
tuned hyperparameters:(best parameters)
{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8472222222222222
```

```
svm_cv.score(x_test, y_test)
```

```
0.8333333333333334
```

```
svm_hat=svm_cv.predict(x_test)
plot_confusion_matrix(y_test,svm_hat)
```

DECISION TREE CLASSIFIER

```
tree = DecisionTreeClassifier()
```

```
tree_cv = GridSearchCV(tree, parameters, cv=10)
```

```
tree_cv.fit(X_train,Y_train)
```

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}
```

```
print("tuned hyperparameters:(best parameters)", tree_cv.best_params_)
```

```
print("accuracy:", tree_cv.best_score_)
```

```
tuned hyperparameters:(best parameters)
{'criterion':'entropy','max_depth':4, 'max_features': 'sqrt',
'min_samples_leaf':2, 'min_samples_split':2, 'splitter':'best'}
accuracy : 0.875
```

```
tree_cv.score(x_test, y_test)
```

```
0.8333333333333334
```

```
tree_hat=tree_cv.predict(x_test)
plot_confusion_matrix(y_test,tree_hat)
```

K-NEAREST NEIGHBOR

```
KNN= KNearestNeighbor()
```

```
knn_cv = GridSearchCV(KNN, parameters, cv=10)
```

```
parameters = {
  'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
  'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
  'p': [1,2]}
```

```
knn_cv.fit(X_train,Y_train)
```

```
print("tuned hyperparameters:(best parameters)", knn_cv.best_params_)
print("accuracy:", knn_cv.best_score_)
```

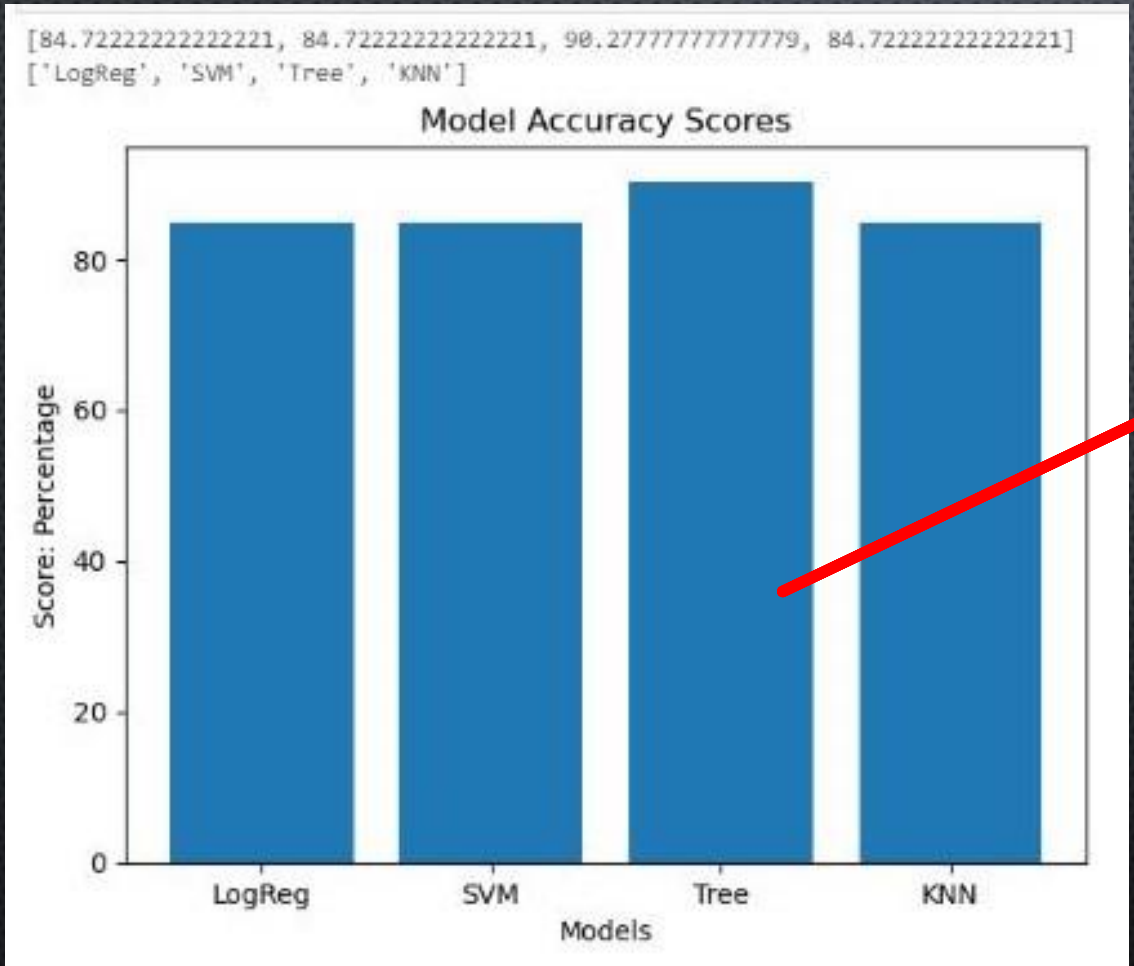
```
tuned hyperparameters:(best parameters)
{'algorithm': 'auto', 'n_neighbors': 9, 'p': 1}
accuracy : 0.8472222222222222
```

```
knn_cv.score(x_test, y_test)
```

```
0.8333333333333334
```

```
knn_hat=knn_cv.predict(x_test)
plot_confusion_matrix(y_test,knn_hat)
```

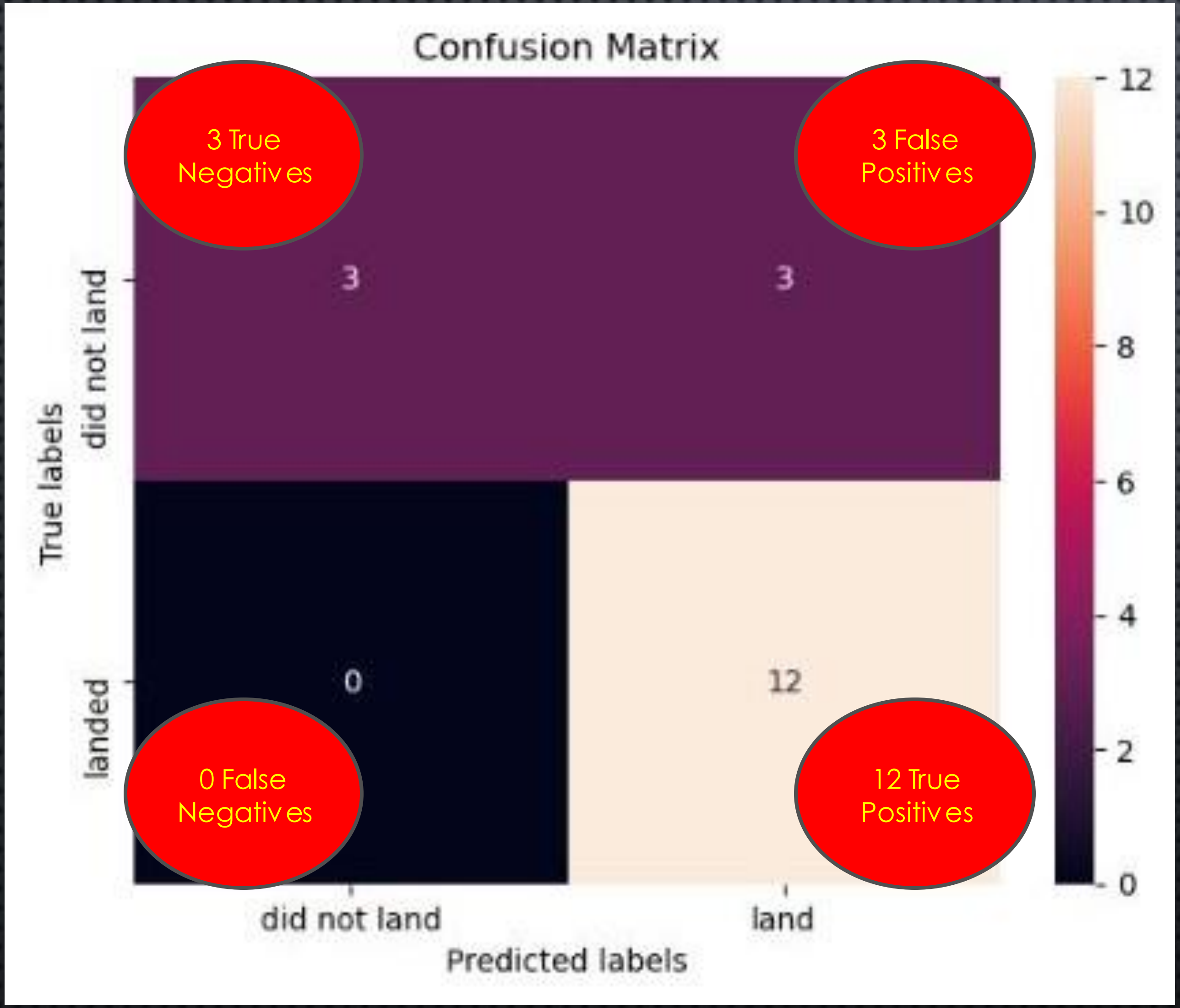
CLASSIFICATION ACCURACY



Most accurate model = Decision Tree Classifier

90.27 % Accuracy!

CONFUSION MATRIX



This is the confusion matrix for the best performing model: <u>Decision Tree</u>

This is also the same exact Confusion Matrix for all 3 of our other models.

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

83.3 % Accuracy

CONCLUSION

Section 6

Launch Site with the largest successful launch?

VAFB SLC-4E: Vandenberg Air Force Base
9,600 Kg

9,600 Kg

Launch Site with the highest success rate?

KSC SLC-39A: Kennedy Space Center

76.9 %
Success

Payload Range with the highest success rates? The lowest?

Highest Success Rate = 2,000 – 4,000 Kg
Lowest Success Rate = 6,000 – 8,000 Kg

60 %
Success

0 %
Success

Falcon 9 Booster Version with the highest success rate?

"FT" Booster Version:
14 Successes of 20 Attempts

70 %
Success

So, what does that all mean?

YES!
The outcome of a successful landing after launch CAN be reasonably predicted.

Therefore, it **IS** possible for SpaceY to correctly make these predictions to successfully outbid SpaceX.

Most Profitable Parameters?

Launch Site:
KSC LC-39A

Payload Mass:
2,000–4,000 Kg

Booster:
Version FT

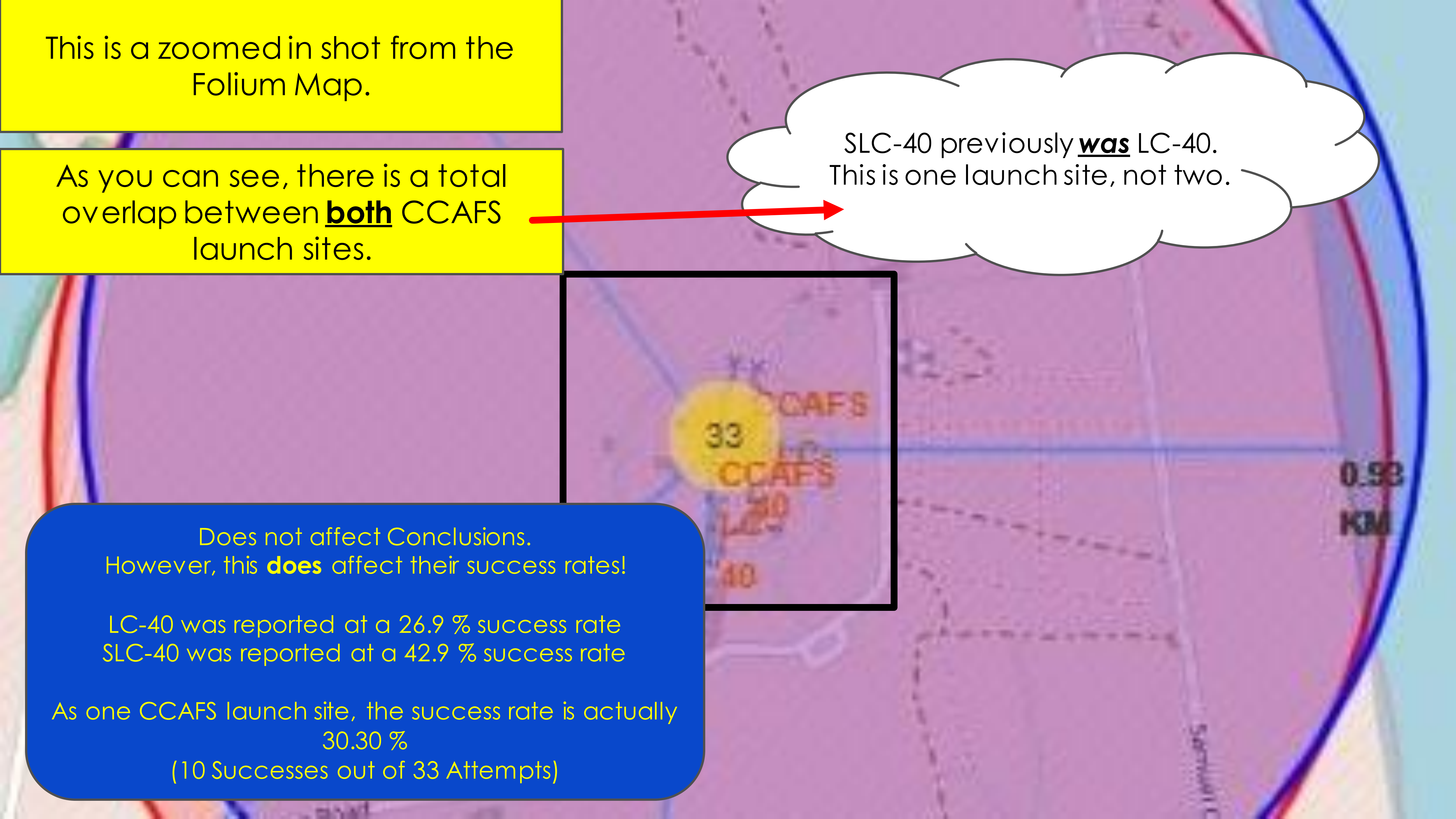
APPENDIX

Section 7

This is a zoomed in shot from the Folium Map.

As you can see, there is a total overlap between **both** CCAFS launch sites.

SLC-40 previously ***was*** LC-40.
This is one launch site, not two.



Does not affect Conclusions.
However, this **does** affect their success rates!

LC-40 was reported at a 26.9 % success rate
SLC-40 was reported at a 42.9 % success rate

As one CCAFS launch site, the success rate is actually 30.30 %
(10 Successes out of 33 Attempts)

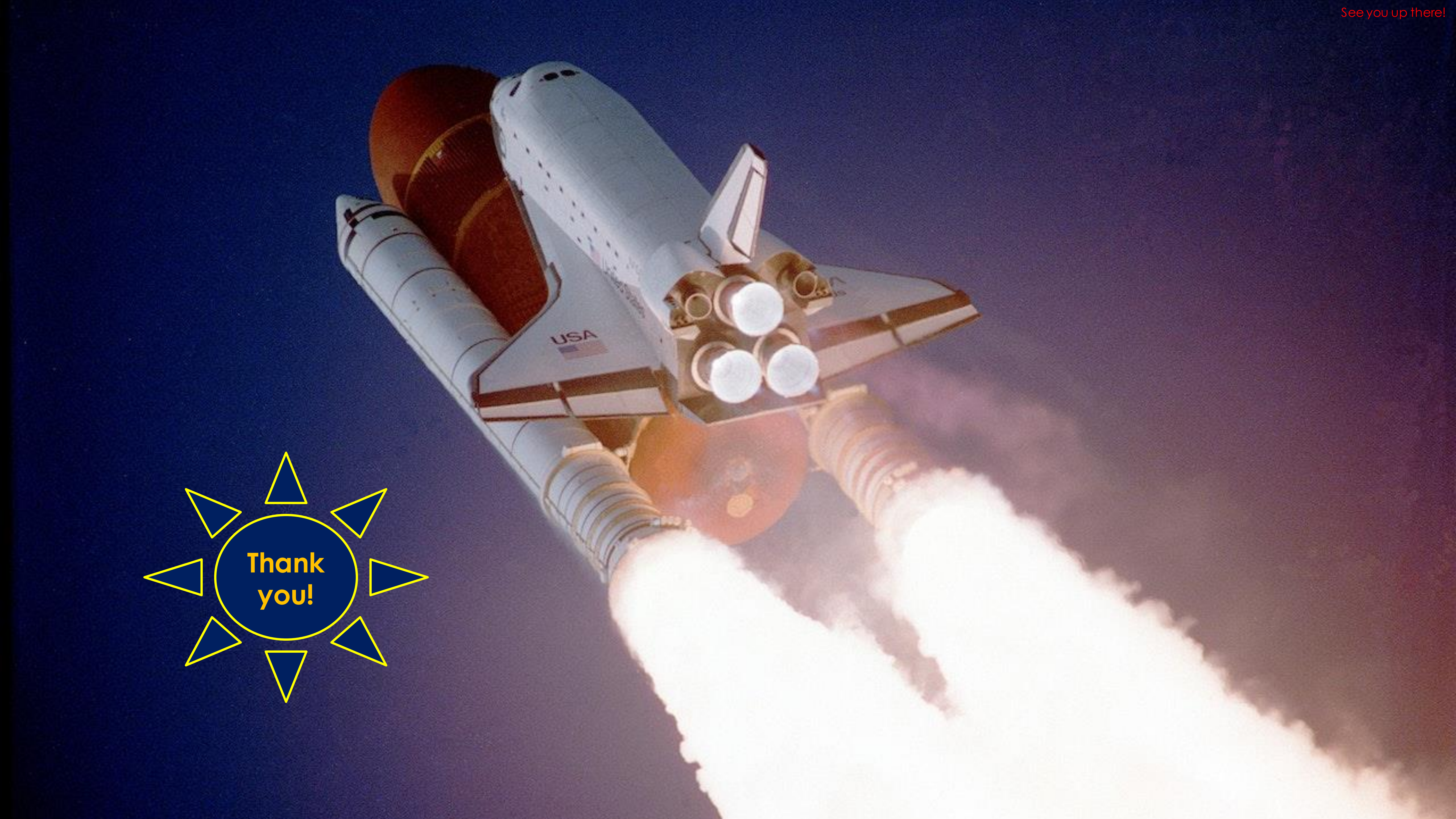
Conclusions were drawn from the data in the Plotly Dashboard

SQL Query gives a different conclusion for largest successful launch

Largest Successful Launches had a payload mass of 15,600 Kg.
These launches took place at both KSC LC-39A and CCAFS SLC-40

See Scatter Plot:
Payload Mass vs. Launch Site

See you up there!



Thank you!