IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Moawiah Ibrahim
10/16/2024

Outline

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

Executive Summary

Space travel missions using the Falcon 9 spacecraft are the most successful in the field of commercial space travel, in large part thanks to the ability to reuse stage 1 of the spacecraft.

We have used the REST API of SpaceX and webscraping for historical data available on Wikipedia to assess which are the most important factors contributing to the reuse of stage 1.

After an initial Exploratory Data Analyis through visualization and database querying using SQL, we have isolated and formatted the most relevant data in preparation for predictive model building.

We have built 4 classification models to yield optimized predictions of stage 1 reuse : a Logistic Regression model, an SVM model, a Decision Tree model and a KNN model. These 4 models have high performance in predicting success at above 83% of exact predictions. The Decision Tree model had slightly better results, while the other 3 models had the same accuracy.

Introduction

- Development of commercial space travel. Already several players in the field.
- Cost is substantially reduced, hence competitive advantage gained, when the first stage of the spacecraft can be reused.
- SpaceX's Falcon 9 rocket is the only successful reusable rocket to date
- We are SpaceY, a new company in the field. Our commercial success is largely determined by our ability to predict reuse of the first stage.

- We want to know the key factors that determine our ability to reuse the first stage of the spacecraft using Falcon 9's historical records
- In order to determine the price of each launch, we need to train a Machine Learning Model that will predict if SpaceY will reuse the first stage.

Section 1

Methodology

Methodology

Data collection methodology:

Describe how data was collected

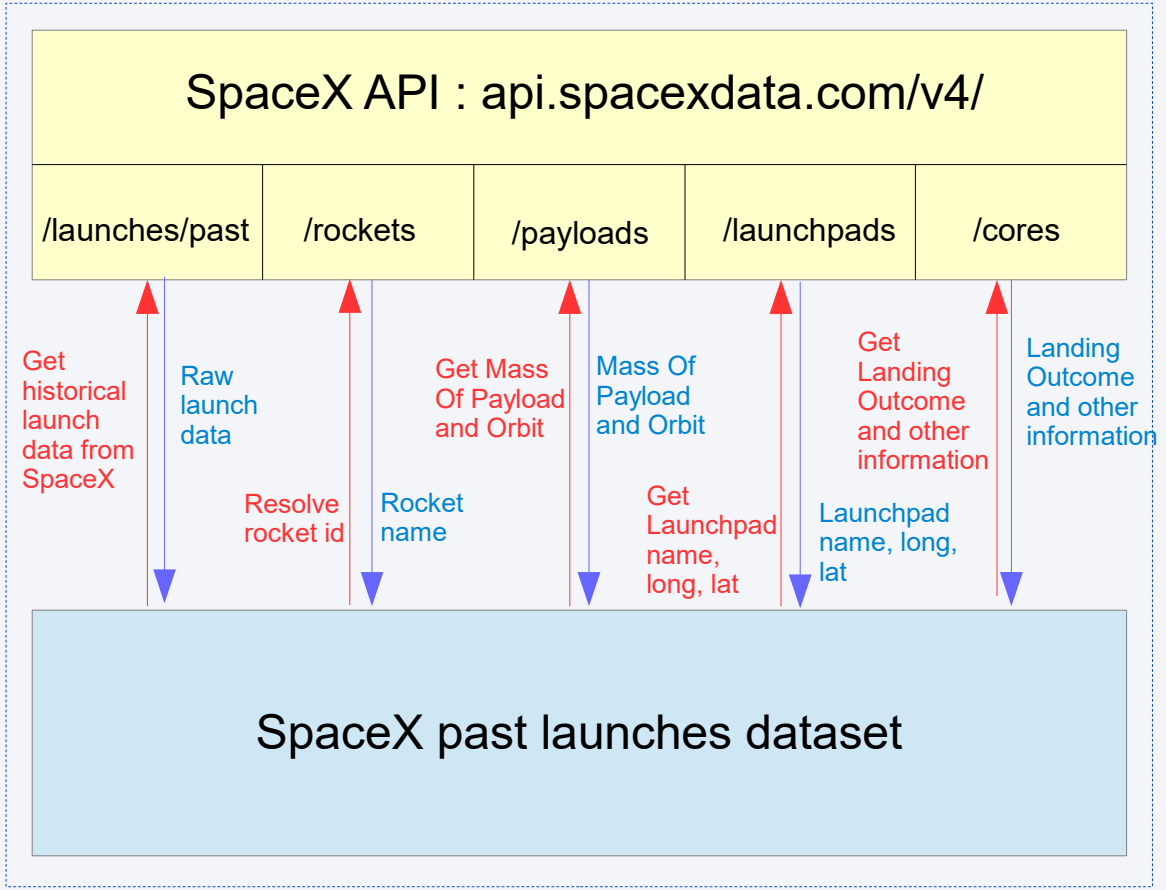
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- To build predictive models for SpaceY's future launches, historical data describing SpaceX's rocket launches from 2010 to 2020 was collected
- The historical data is available on SpaceX's REST API and wikipedia
- We used python's webscraping library Beautiful soup to collect the data from the wikipedia page
- We used python's Pandas library to collect, clean and manipulate the dataset

Data Collection – SpaceX API

- We use SpaceX launch data available on a REST API at api.spacexdata.com/v4/
- First, we get past launch data at endpoint :
  api.spacexdata.com/v4/launches/past
- We use other endpoints to resolve references gathered in the first phase:
  api.spacexdata.com/v4/rockets
  api.spacexdata.com/v4/launchpads
  api.spacexdata.com/v4/payloads
  api.spacexdata.com/v4/cores

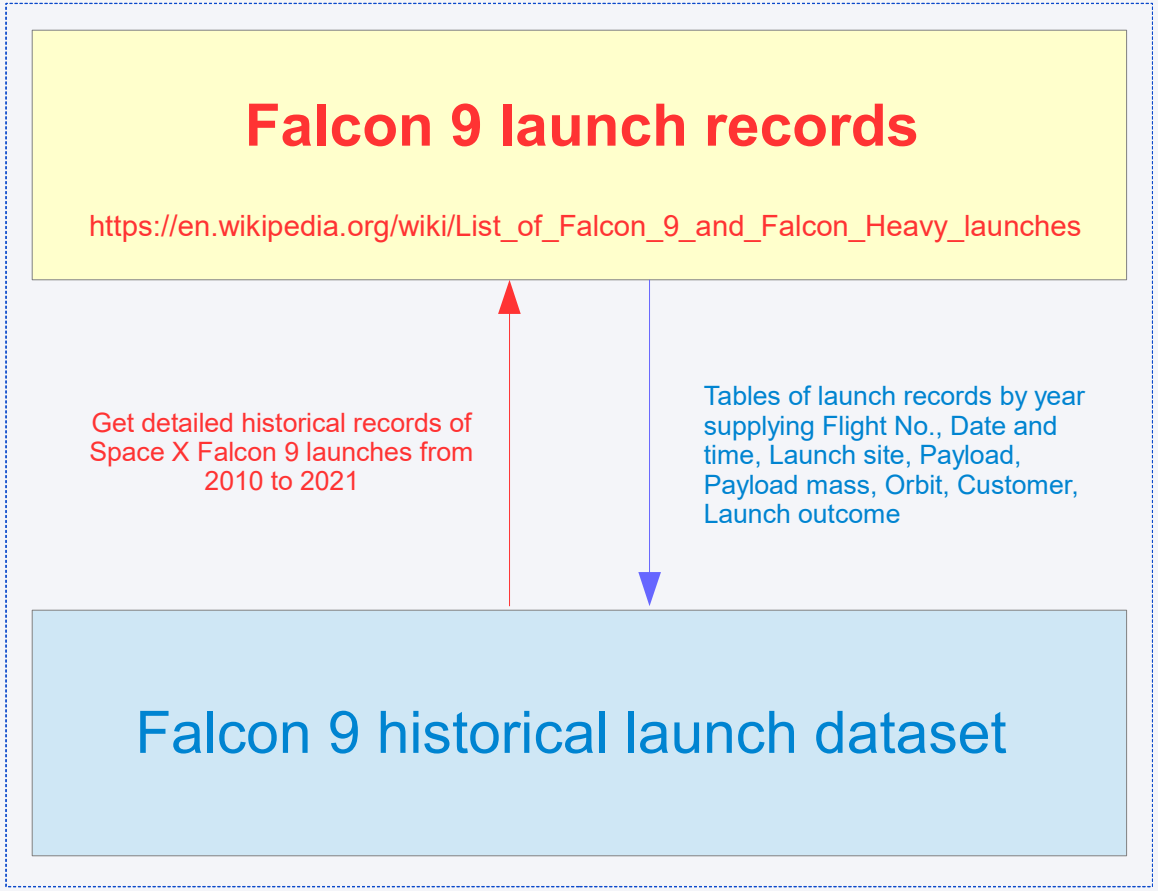


GitHub URL of the SpaceX API calls notebook

- https://github.com/MoawiahDev/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Falcon 9's historical launch records are available on
  https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- We use the Python BeautifulSoup package to web scrape HTML tables describing past launches
- From these tables we extract key data that we will use to build our model



GitHub URL of the web scraping notebook

https://github.com/MoawiahDev/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

Data was collected from SpaceX's REST API and wikipedia. The following steps were taken in order to obtain the best input for predicitive model building:

- Replacing missing Payload Mass values by the mean of the available payload mass data
- Adding a numerical column named “class” to the dataframe representing success or failure
- After EDA, selecting the features that appear to have the greatest impact on the outcome
- Applying one hot encoding : creating parallel numerical features for the categorical ones and include them into the dataframe
- Create a dataframe with only the numerical features and convert all values to float

GitHub URL of data wrangling related books

https://github.com/MoawiahDev/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data wrangling.ipynb

https://github.com/MoawiahDev/Applied-Data-Science-Capstone/blob/main/edadataviz.ipynb

https://github.com/MoawiahDev/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

EDA with Data Visualization

Exporatory Data Analyis using Data Visualization was done using the following charts :

- A categorical plot of flight number vs. payload mass where data points are colored according to the success of the recovery of stage 1. This showed that starting from flight no 52, success ratio was higher and that failures were less frequent above 6k payload mass. It also showed that very high payload mass were much more frequent in the later flights.
- A categorical plot of flight number vs. launch site where data points are colored according to the success of the recovery of stage 1. This showed that CCAFS SLC-40 was the most used launch site and had the worst success ratio
- A categorical plot of payload mass vs. launch site where data points are colored according to the success of the recovery of stage 1. This showed that no high payload mass flights (above 10k) were launched from VAFB SLC 4E
- A bar plot of success rate per orbit type. This showed that 4 orbit types had 100% success rate (ES-L1,SSO,GEO,HEO) among whom only SSO was targeted several times, the rest only once. GTO was the most targeted orbit and had the worse success rate, followed by ISS
- A categorical plot of flight number vs. orbit type where data points are colored according to the success of the recovery of stage 1. This showed that VLEO orbits where targeted starting from flight 65 and had a better success rate than other frequently targeted orbit types.
- A categorical plot of payload mass vs. orbit type where data points are colored according to the success of the recovery of stage 1. This showed that payloads lower than 6k and with orbit types GTO and ISS had a high failure rate
- A line plot of success rate per year that showed that success rate grew from around 35% in 2015 to around 80% in 2020 with only a setback in 2018

GitHub URL of the EDA with Data Visualization

https://github.com/MoawiahDev/Applied-Data-Science-Capstone/blob/main/edadataviz.ipynb

EDA with SQL

Some examples of Exploratory Data Analysis using SQL are :

- Getting the names of the launch sites :

```
SELECT DISTINCT(Launch_Site) FROM SPACEXTABLE
```

- List the date when the first succesful landing outcome in ground pad was achieved:

```
SELECT Min(Date) FROM SPACEXTABLE where Landing_Outcome like '%Success (ground pad)%'
```

- List the total number of successful and failure mission outcomes:

```
SELECT count(CASE WHEN mission_outcome like '%Success%' THEN 1 END) as 'Successes', count(CASE WHEN mission_outcome like '%Failure%' THEN 1 END) as 'Failures'from SPACEXTABLE
```

- List the names of the booster_versions which have carried the maximum payload mass

```
select Booster_Version FROM SPACEXTABLE where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

GitHub URL of EDA with SQL

https://github.com/MoawiahDev/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- A Folium map was augmented with circles, markers and lines
- Circles were added to localize the launch sites on the map
- Markers were added to palce the names of the launch sites on the map
- Lines where added to display the distances between launch sites and significant close location like the sea or the closest city

GitHub URL of interactive map with Folium

https://github.com/MoawiahDev/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- A Plotly Dash application was created to perform interactive visual analytics
- Two types of charts were used:
  - A Pie Chart representing launch success
  - A Scatter Chart representing Payload Mass vs Launch Success
- The user can select from a dropdown list either to display the ratio that the launch sites had in overall successes or the ratio of successful vs failed missions for each specific launch site
- When the selection is made in the dropdown list, a scatter plot shows the successful missions vs the failed once according to booster type and payload mass. The range of payload masses displayed is determined through an interactive range slider.

GitHub URL of Plotly Dash lab

https://github.com/MoawiahDev/Applied-Data-Science-Capstone/blob/main/lab_plotly_dash.ipynb

Predictive Analysis (Classification)

- Using the historical data of SpaceX's 90 launches using Falcon 9, we want to find the best predictive model for SpaceY's missions.
- The data was first standardized the StandardScaler class of the preprocessing python package
- In order to build the best model based on the available data, we split the data into a train set of 72 records and a test set 18 records (20% of the Falcon 9 records) for testing the predicitive model
- The process is run on 4 classification models : Logistic regression, SVM, Decision Tree and KNN
- For each of the models, the tuned parameters are found as well as the corresponding accuracy score
- We run all the models on the test data, which allows us to evaluate the accuracy of the model and build a confusion matrix to visualize the results

GitHub URL of Predictive Analysis Lab

https://github.com/MoawiahDev/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine Learning Prediction_Part_5.ipynb

Results of Exploratory Data Analysis

Through Exploratory Data Analysis, either through visualization or the use of SQL, we can see that:

- Success rate improves noticeably after flight number 60, regardless of the launch site
- Launch site CCAFS is used noticeably more than the other sites and VAFB SLC noticeably less
- Among Orbit Types which are often targeted, GTO has the worst success rate, VLEO the best

Results of Interactive Analytics

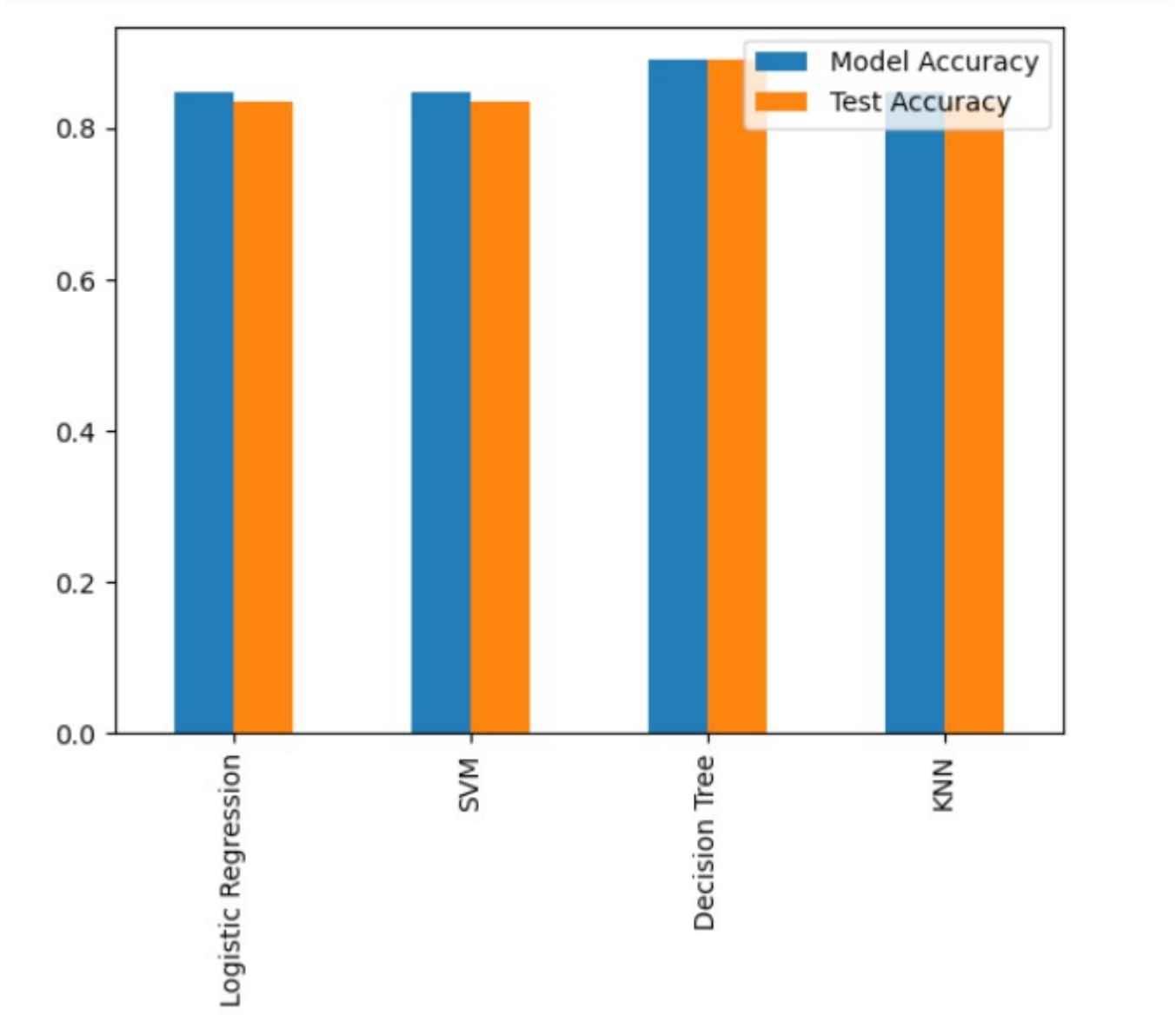
- Interactive Data Analytics using Dash reveals that:
  - Success is clearly the most advantageous launch site

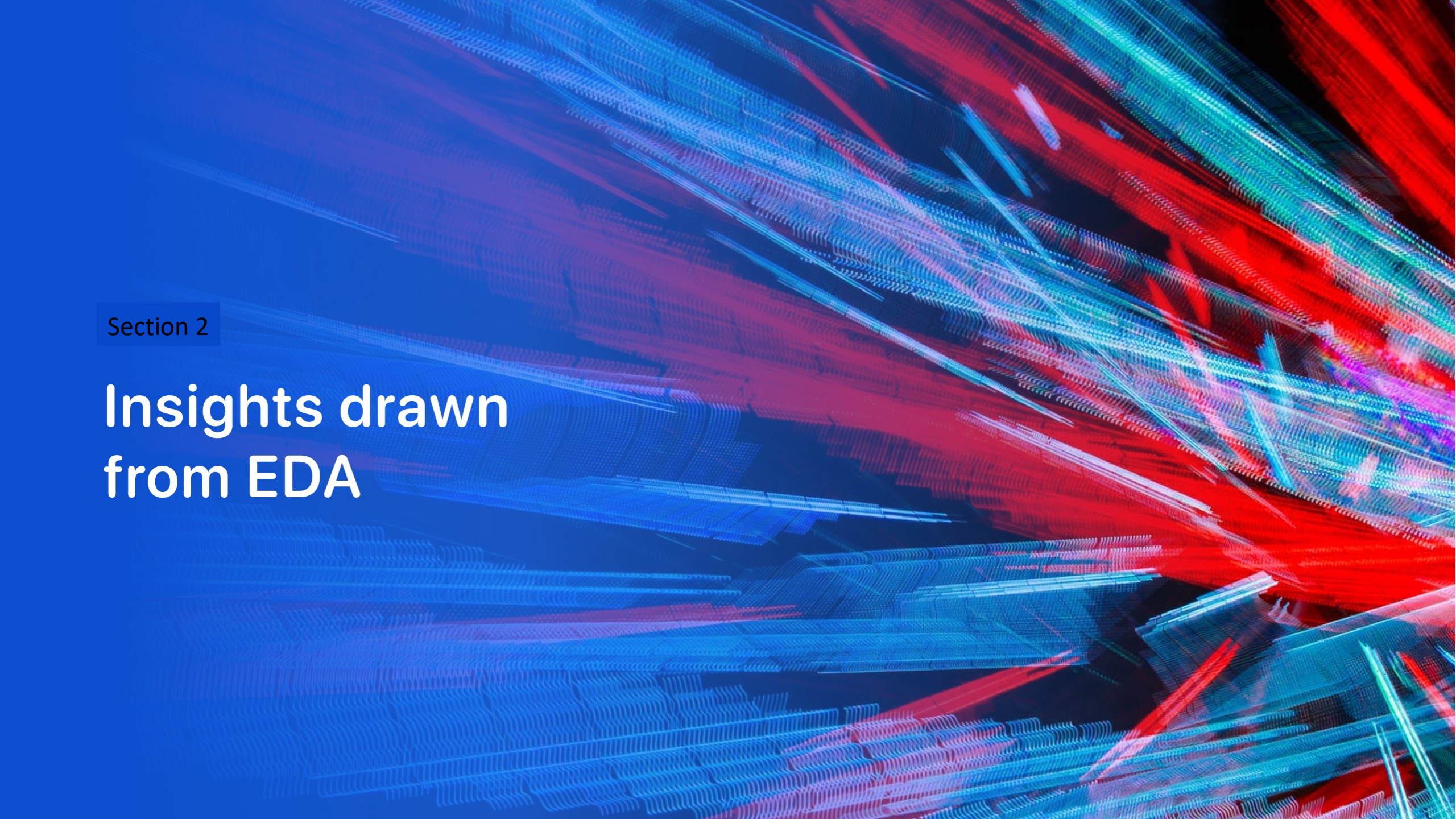
Results of Predictive Analytics

Predictive Analysis displays high predictive capacity for all 4 models when applied to the test set.

Once the parameters are tuned, all models have an accuracy score of 84% at least with a slight edge for decision tree model at almost 89%.

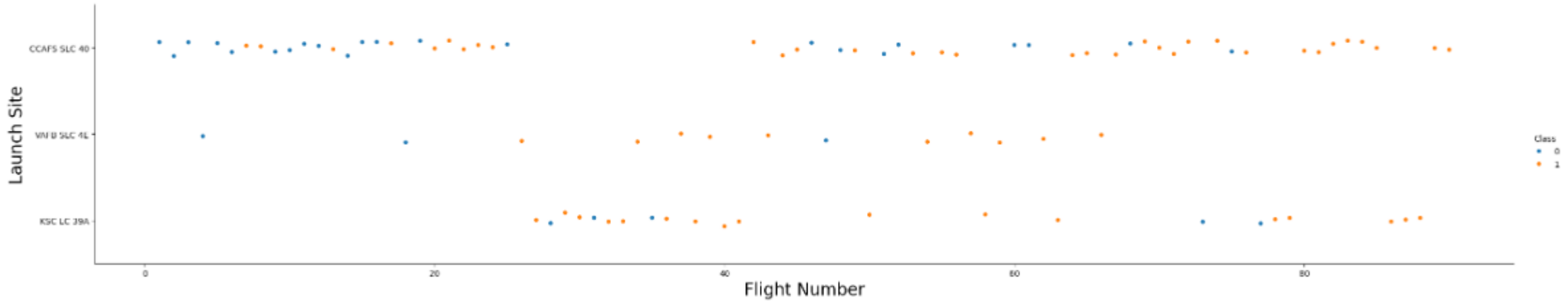
On the test set, the results are 83.3% accuracy for Logistic Regression, SVM and KNN. For Decision Tree, it is almost 89%




Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

```
[5]: # Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

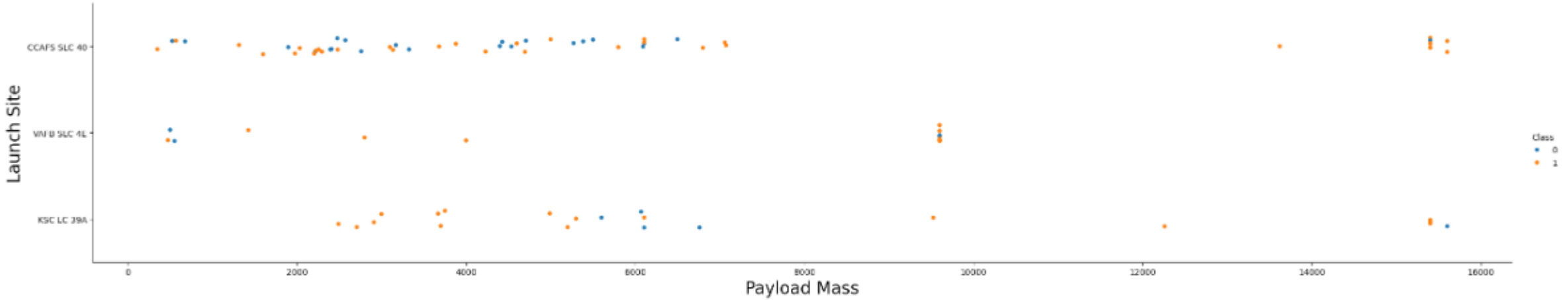


Thanks to the scatter plot of Flight Number vs. Launch Site, two important points are very clear:

- For all Launch Sites, the success rate improves with time i.e. as the flight number increases
- Cape Canaveral Space Launch Complex 40 is the preferred launch site for Falcon 9

Payload vs. Launch Site

```
[6]: # Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

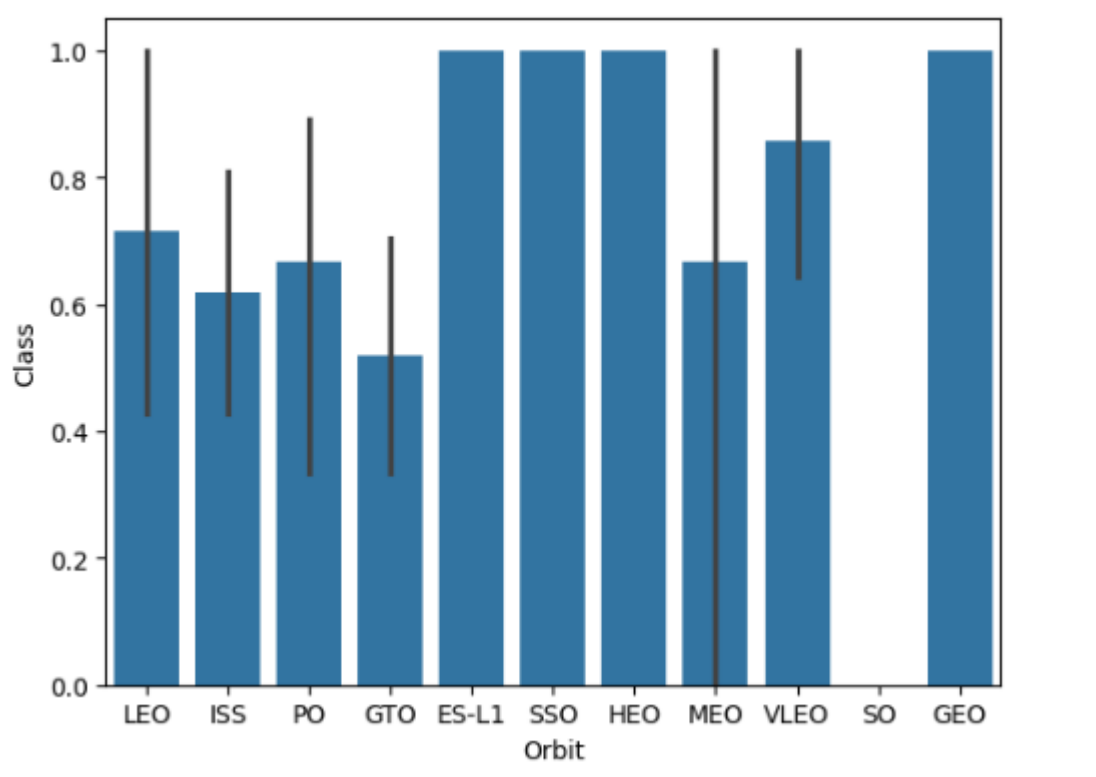


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Thanks to the scatter plot of Payload vs. Launch Site, we see that :

- Success rate does not seem to depend on Payload Mass
- For high payload masses, only Kennedy Space Center Launch Complex 39A and Cape Canaveral Space Launch Complex 40 are used

Success Rate vs. Orbit Type



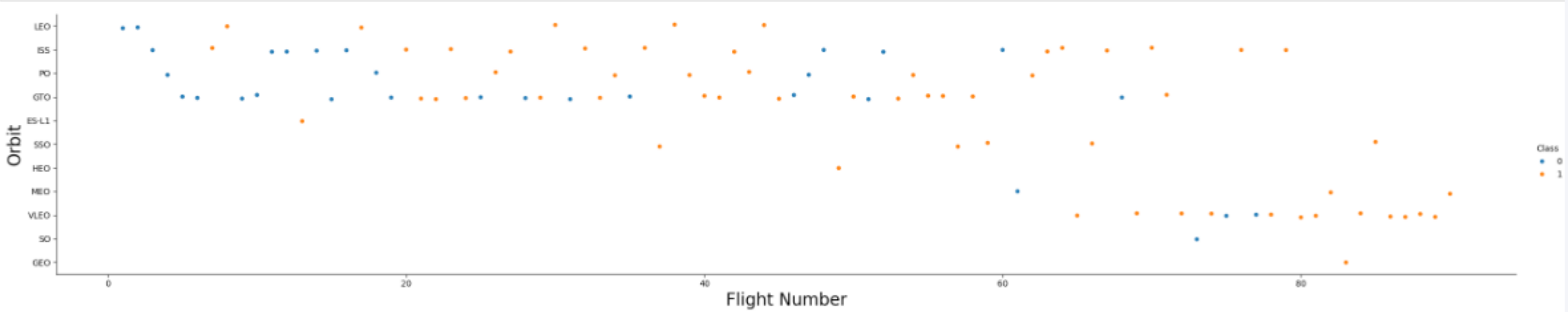
Thanks to the bar plot of Succes Rate vs. Orbit Type, we can see that success rate varies with Orbit Type:

- Success rate is perfect for orbit types ES-L1, SSO, HEO and GEO
- Success rate is lowest for orbit types GTO and ISS

Flight Number vs. Orbit Type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

[8]:
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



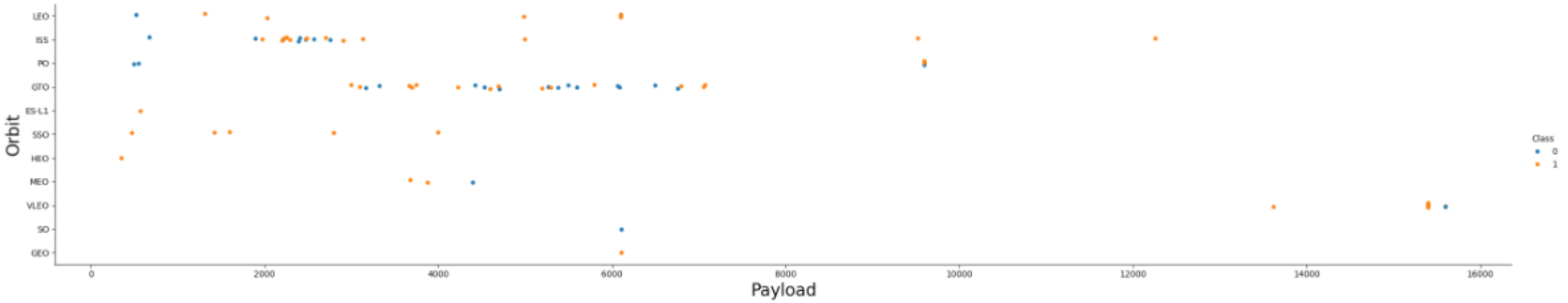
Thanks to the scatter plot of Flight Number vs. Orbit Type, we see that :

- Only 4 orbit types were almost exclusively targeted til flight number 60: LEO, ISS, PO and GTO

Payload vs. Orbit Type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

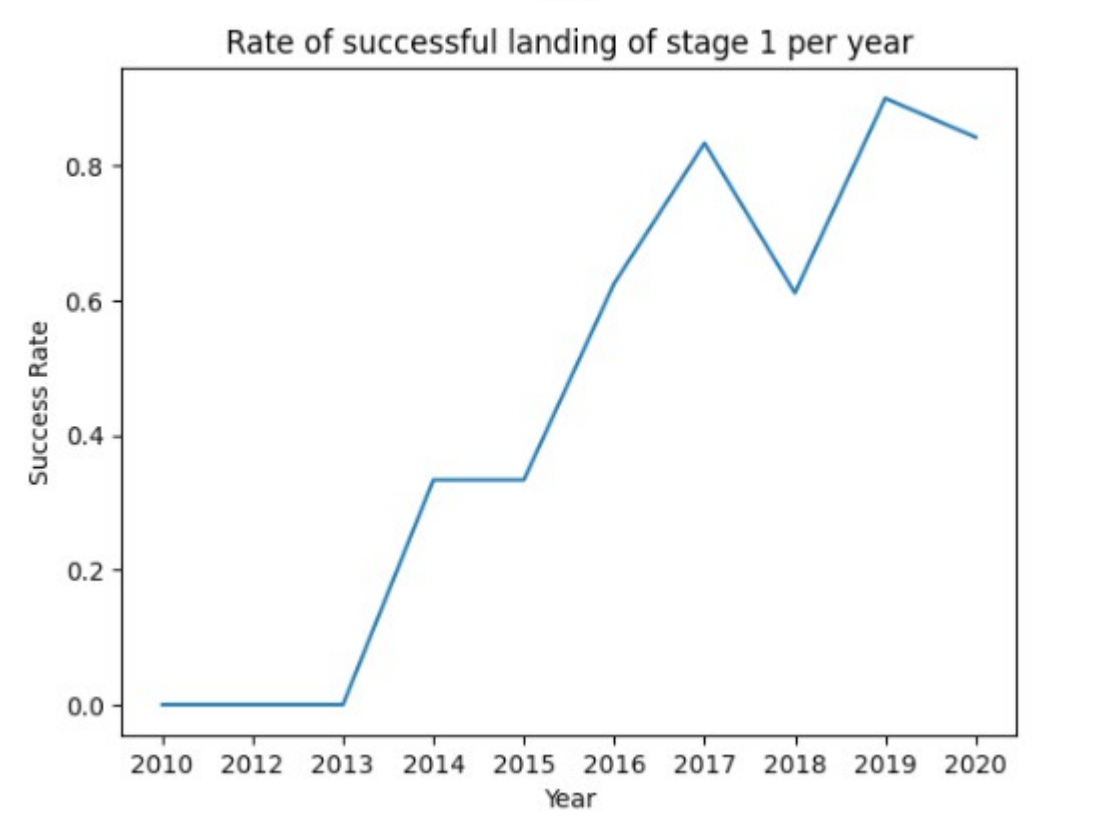
[10]:
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Thanks to the scatter plot of Payload vs. Orbit Type, we see that :

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- GTO has a higher failure rate than other Orbit Types

Launch Success Yearly Trend



From the line plot to Success Rate by Year, we can clearly see an increase of success rate from 0 in 2013 to around 80% in 2020 with a temporary setback in 2018.

All Launch Site Names

- To find the names of the launch sites, we use the following query:

```
SELECT DISTINCT(Launch_Site) FROM SPACEXTABLE
```

- Results :

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

- Where
  - CCFAS LC-40 and CCAFS SLC-40 are Cape Canaveral Space Launch Complex in Florida
  - VAFB SLC-AE is Vanderberg Space Launch Complex in California
  - KSC LC-39E is Kennedy Space Center Launch Complex in Florida

Launch Site Names Begin with 'CCA'

- To find 5 records where launch sites begin with `CCA`, we use query:

```
select * from SPACEXTABLE 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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Total Payload Mass

- To calculate the total payload carried by boosters from NASA, we use :

```
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE where Payload like '%CRS%'
```

- Result :

SUM(PAYLOAD_MASS__KG_)
111268

Average Payload Mass by F9 v1.1

- To calculate the average payload mass carried by booster version F9 v1.1:

```
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE where Booster_Version like '%F9 v1.1%'
```

- Result :

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- To find the date of the first successful landing outcome on ground pad :

```
SELECT Min(Date) FROM SPACEXTABLE where Landing_Outcome like '%Success (ground pad)%'
```

- Result :

Min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
SELECT distinct(Booster_Version) FROM SPACEXTABLE where Landing_Outcome like '%Success (drone ship)%' and
PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

- Result :

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

Total Number of Successful and Failure Mission Outcomes

- To calculate the total number of successful and failure mission outcomes :

```
SELECT count(CASE WHEN mission_outcome like '%Success%' THEN 1 END) as 'Successes', count(CASE WHEN mission_outcome like '%Failure%' THEN 1 END) as 'Failures'from SPACEXTABLE
```

- Result :

Successes	Failures
100	1

- Mission Outcome is the answer to : was the payload placed on orbit ?

Boosters Carried Maximum Payload

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

```
select Booster_Version FROM SPACEXTABLE where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

- Result :

Booster_Version
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

- To list the failed landingoutcomes in drone ship, their booster versions, and launch site names for in year 2015 :

```
select substr(Date,6,2) as Month, landing_outcome, booster_version, launch_site from SPACEXTABLE where substr(Date,0,5)='2015' and landing_outcome like '%Failure (drone ship)%'
```

- Result :

Month	Landing_Outcome	Booster_Version	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

- To 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 :

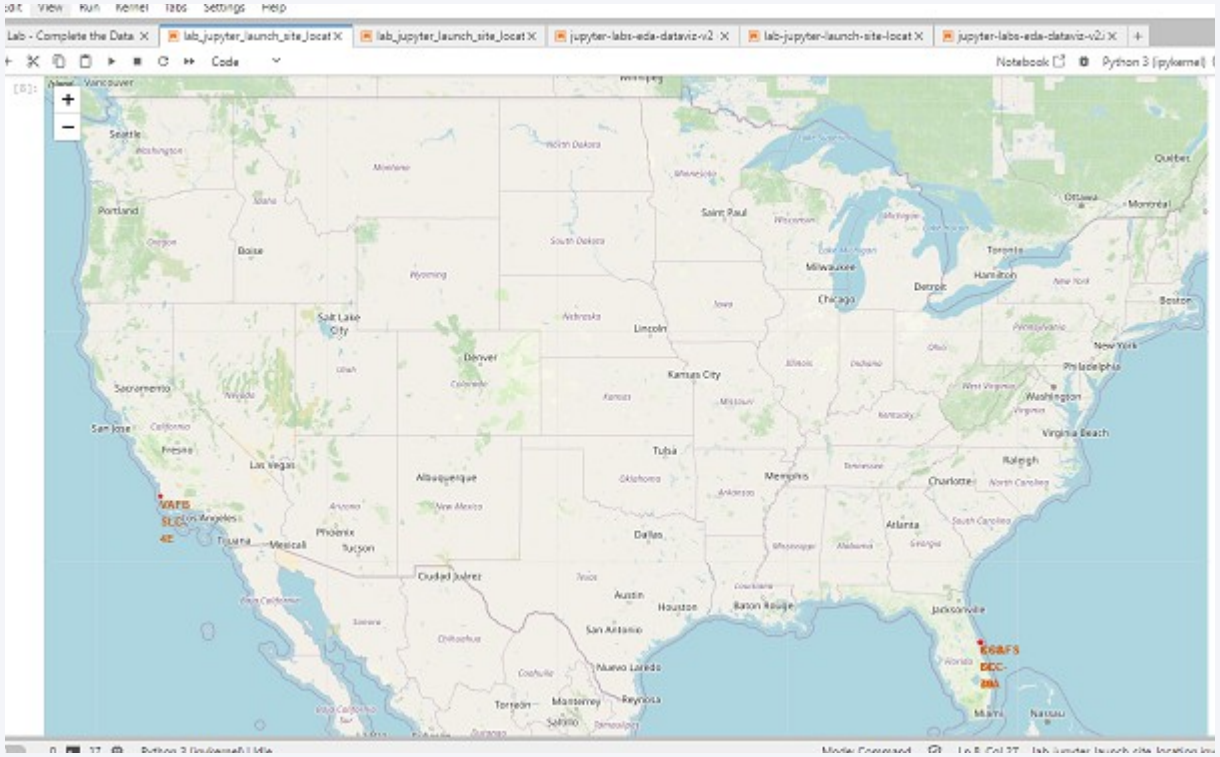
```
select landing_outcome,count(landing_outcome) from spacextable where date > '2010-06-04' and date < '2017-03-20' group by landing_outcome order by count(landing_outcome) desc
```

- Result :

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

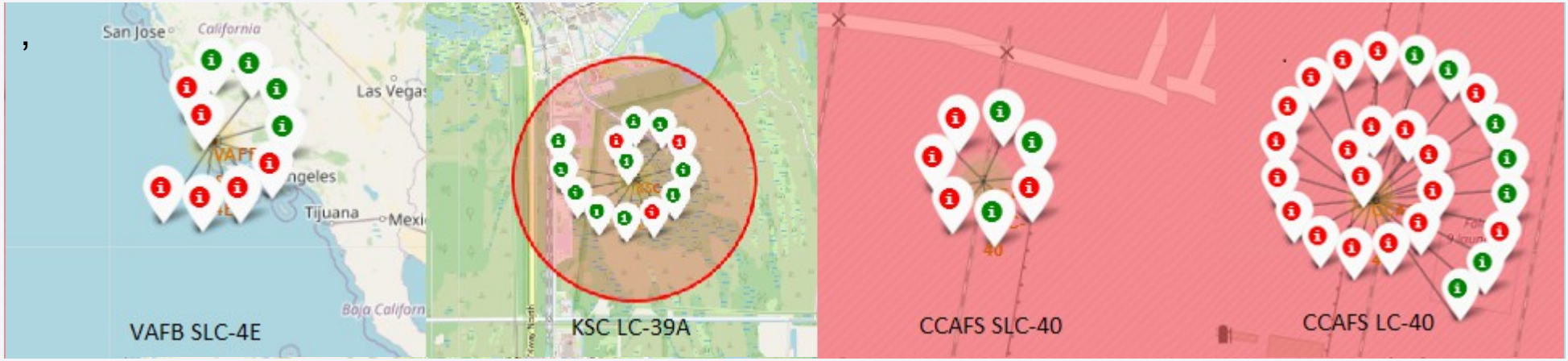

Section 3
Launch Sites
Proximities Analysis

Map with Launch Sites



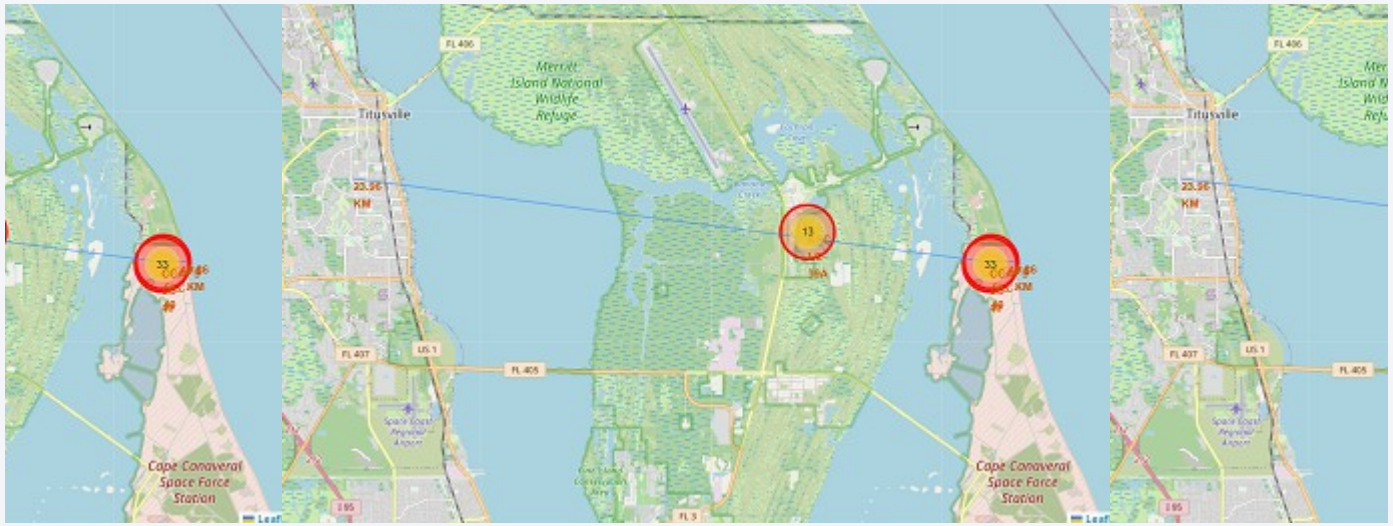
- One launch site is in California, while the other three are very close to each other in Florida

Successful vs. Failed Launches Map

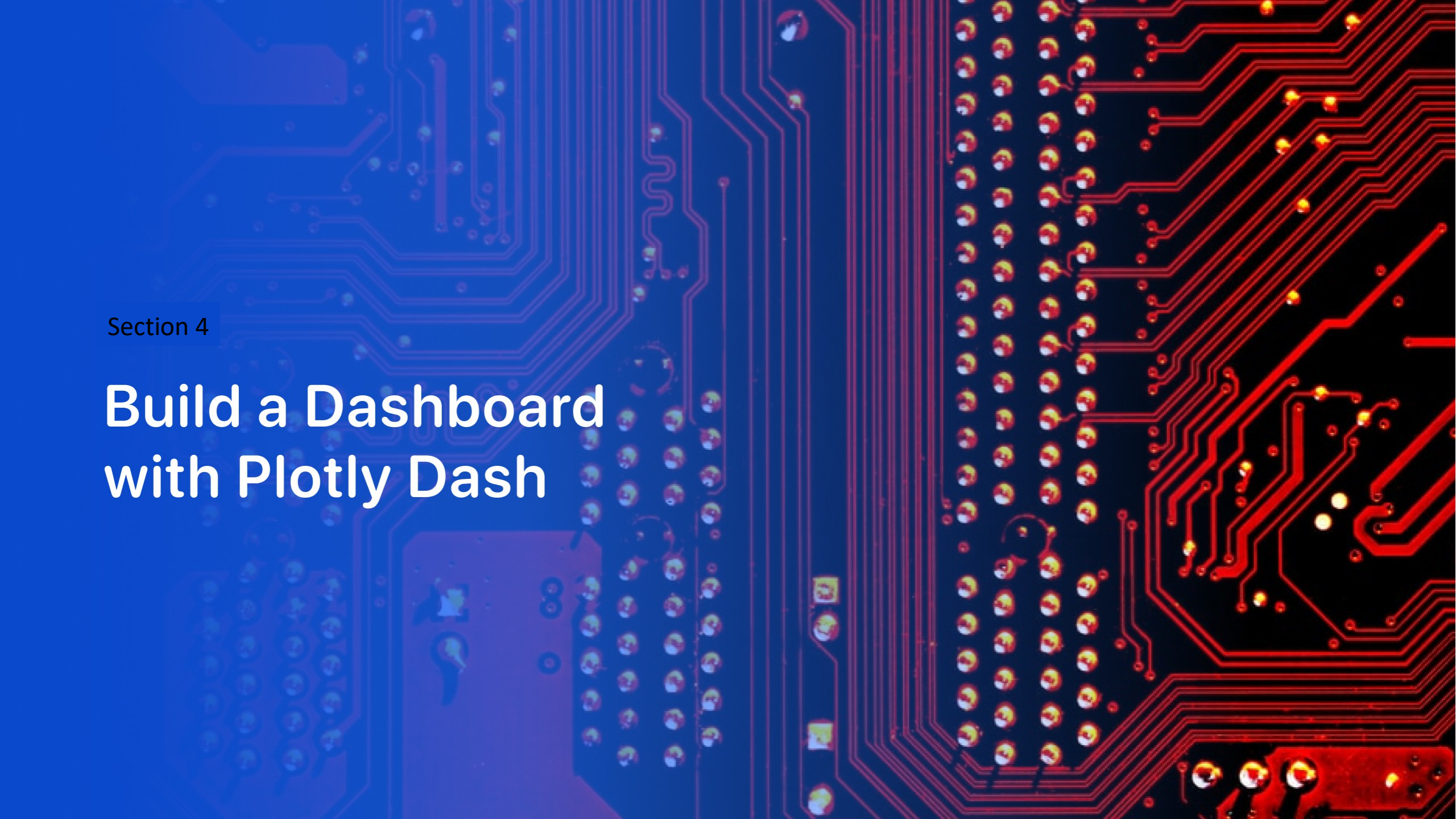


- The map displays the successful launches in green vs the failed ones in red.
- The data only covers launches from launch number 1 until launch number 56 on june,4 2018. This explains the very high proportion of fails
- During this period, launch site KSC LC-39A had the best results (3 fails for 13 launches) and site CCAFS LC-40 had the worst results (19 fails for 26 launches)

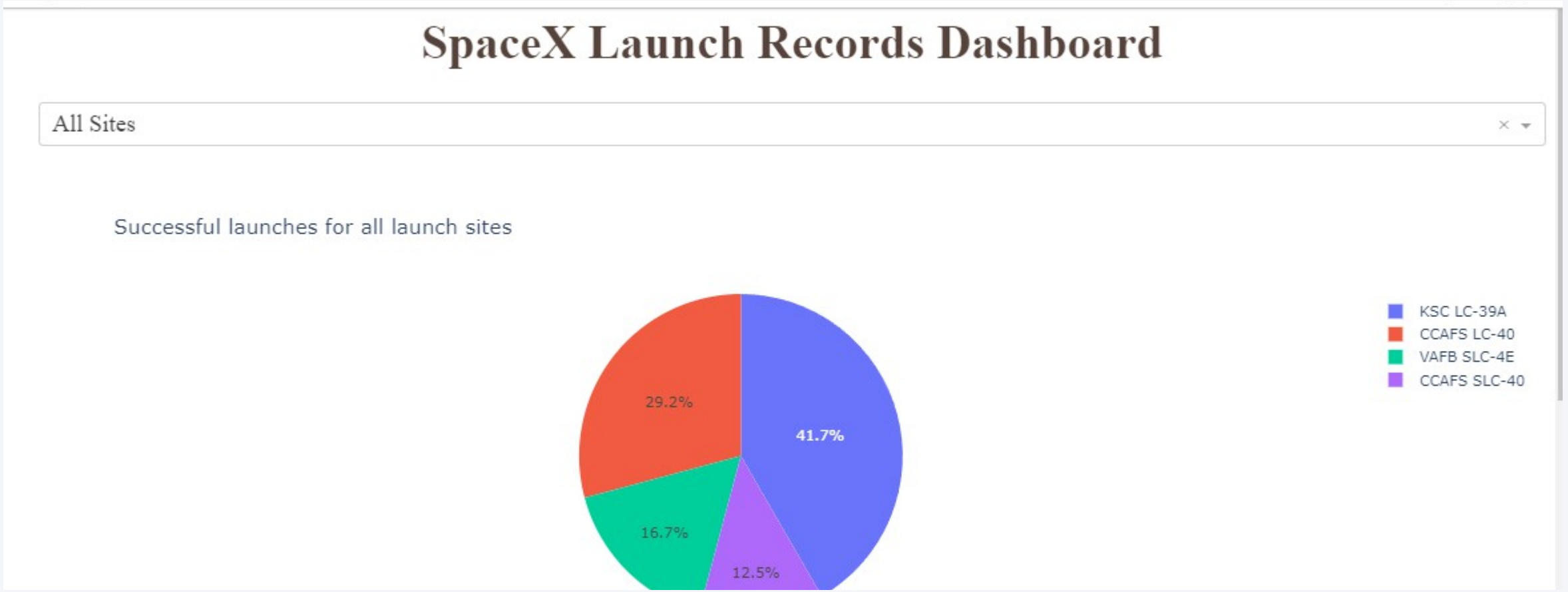
Distances between launch site and proximities



- Using Folium, we are able to determine that launch site CCAFS SLC-40 is only 860m from the sea
- Using Folium, we are able to determine that the distance between launch site CCAFS SLC-40 and the closest town (Titusville) is only 23.96 km.

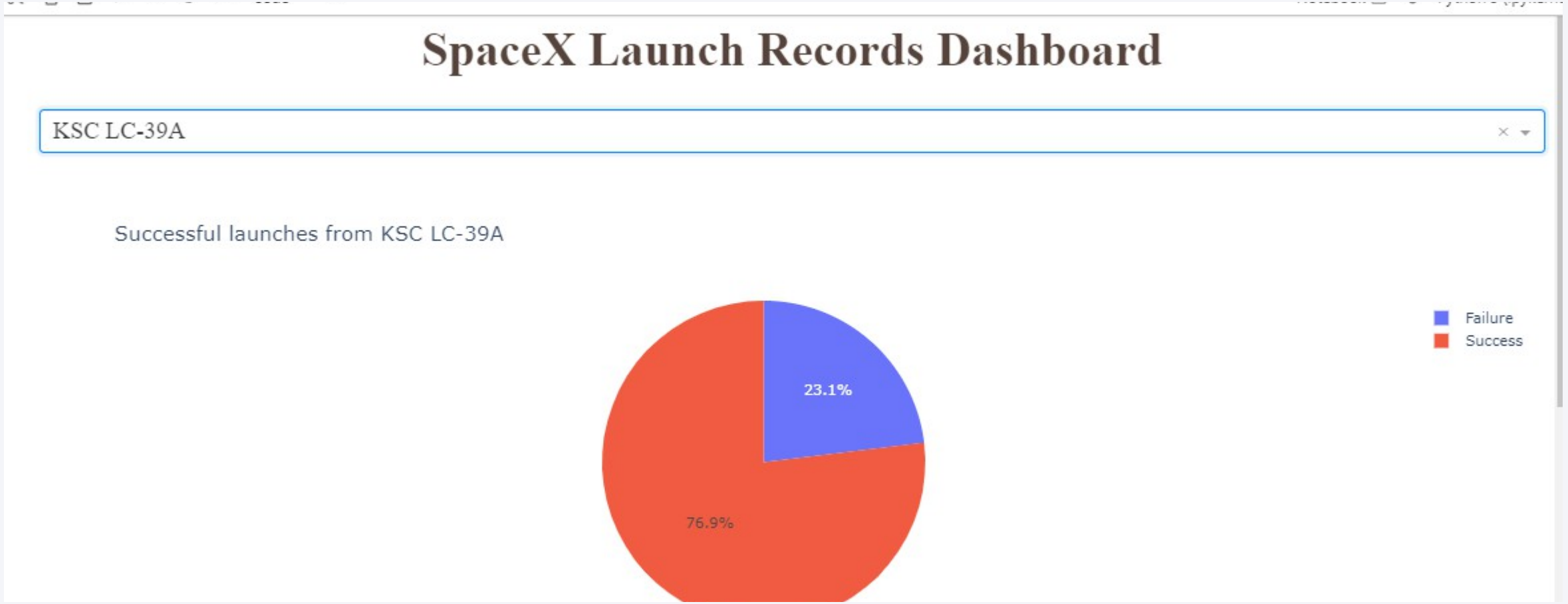

Section 4
Build a Dashboard with Plotly Dash

Successful launches for all sites



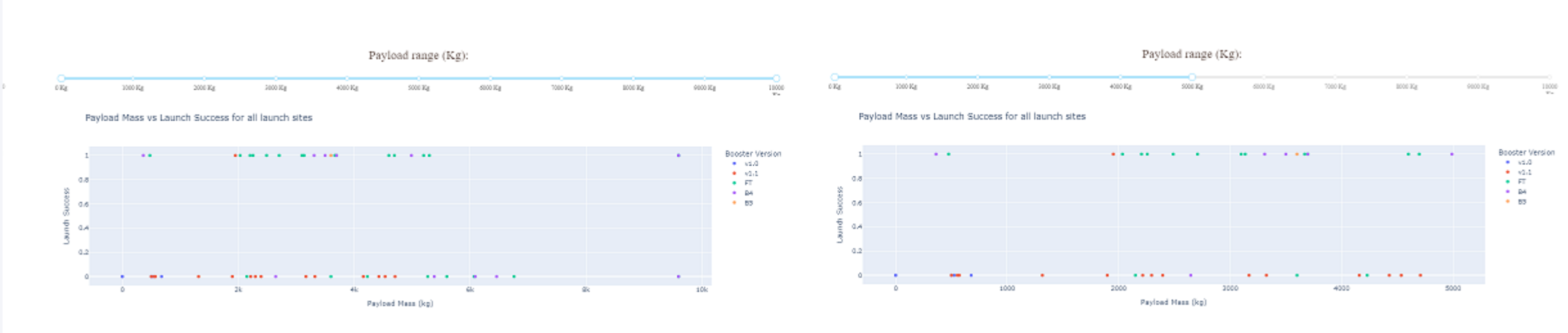
- From the pie chart, we see that most successful launches were done from KSC LC-39A, then from CCAFS LC-40
- Only a small proportion of successful launches were done from VAFB SLC-4E and CCAFS SLC-40

Launch site with highest success ratio



- Kennedy Space Center KSC LC-39A is the launch site with the highest success ratio with more than three out of four launches being successful at reusing stage 1

Payload Mass vs. Launch Outcome



- From the scatter plot, it appear that the success ratio is higher when the payload mass is in the range of 2000kg to 6000kg
- Booster v1.1 has a very high rate of failure, regardless of Payload Mass
- Booster FT seems to have the best success ratio if we except booster B5 which has only been used once
- The best scenario seems to be a launch with payload mass between 2k and 6k, using booster FT

Section 5

Predictive Analysis (Classification)

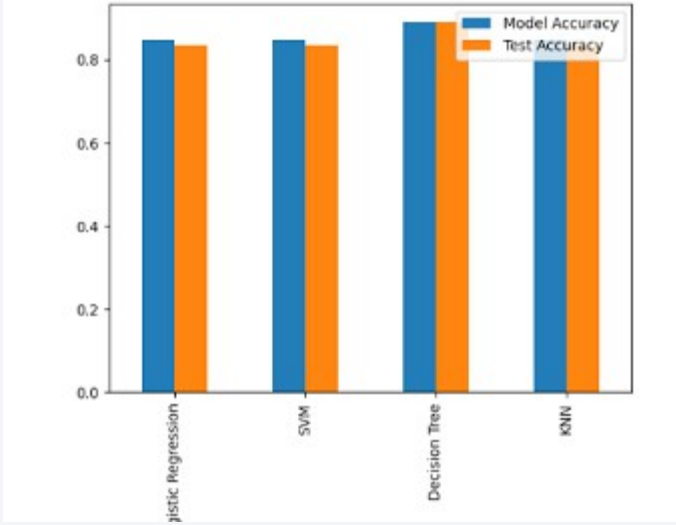
Classification Accuracy

[41]:

	Model Accuracy	Test Accuracy
Logistic Regression	0.8464	0.833333
SVM	0.8482	0.833333
Decision Tree	0.8893	0.888889
KNN	0.8482	0.833333

[43]:
```
model_accuracy_df.plot.bar()
```

[43]: <Axes: >



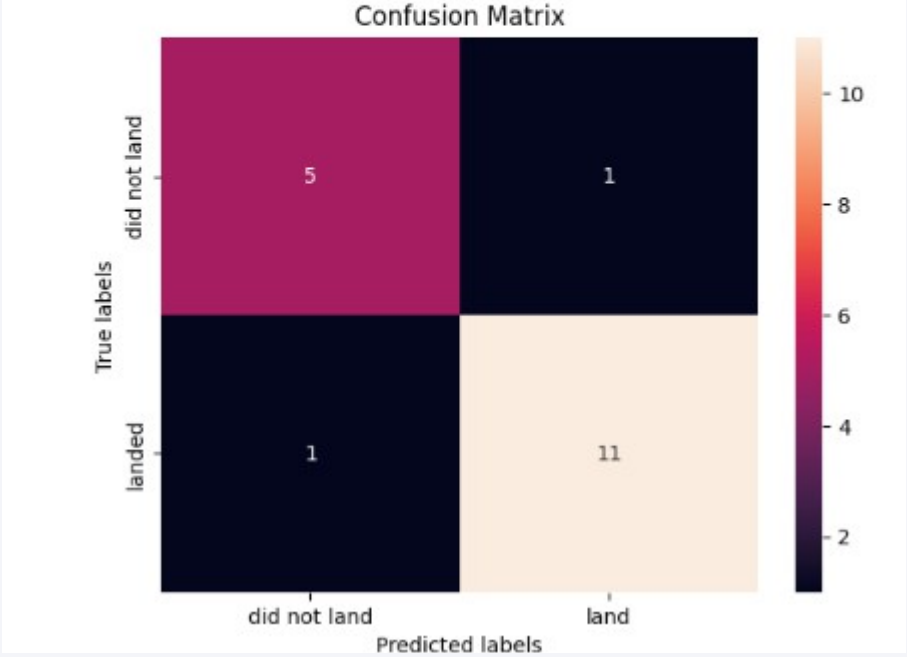
- Model accuracy after training on 80% of the dataset is best for the decision tree model
- Classification accuracy on the remaining 20% of the dataset is also best for the decision tree model

Confusion Matrix

```
[31]: tree_cv.score(X_test,Y_test)
,
[31]: 0.8888888888888888

      We can plot the confusion matrix

[33]: yhat = tree_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



- The confusion matrix for the decision tree model shows very hich accuracy on the test data : out of 18 test cases, 16 outcomes are correctly predicted

Conclusions

- This study has allowed us to isolate factors that contribute to the reuse of stage 1 of the spacecraft during commercial launches and to develop an efficient predictive model that will allow better assessment of cost
- Key factors were the type orbit, the type of booster, the launch site. But above all, accumulation of experiences allowed a steady improvement in success rates
- We built 4 optimized classification models to predict stage 1 reuse using Logisitic Regression, SVM, Prediction Tree and KNN
- All 4 predictive models performed highly during the testing phase, yield above 83% correct predictions, with a slight edge for the Prediction Tree Model

Appendix

- Python code to compare accuracy scores for the 4 predictive models:

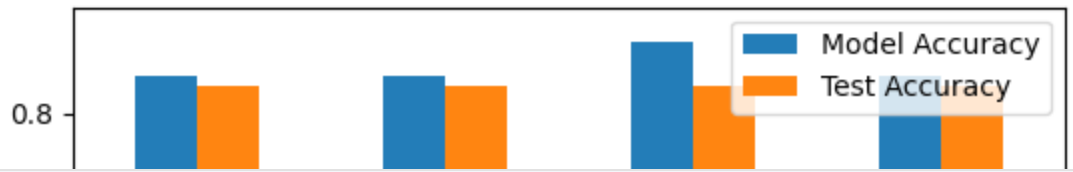
```
[43]: model_accuracy_dict = {}
model_accuracy_dict["Logistic Regression"] = [logreg_cv.best_score_.round(4),logreg_cv.score(X_test,Y_test)]
model_accuracy_dict["SVM"] = [svm_cv.best_score_.round(4),svm_cv.score(X_test,Y_test)]
model_accuracy_dict["Decision Tree"] = [tree_cv.best_score_.round(4),tree_cv.score(X_test,Y_test)]
model_accuracy_dict["KNN"] = [knn_cv.best_score_.round(4),knn_cv.score(X_test,Y_test)]
model_accuracy_df = pd.DataFrame.from_dict(model_accuracy_dict,orient="index")
model_accuracy_df.columns = ["Model Accuracy","Test Accuracy"]
model_accuracy_df
```

[43]:

	Model Accuracy	Test Accuracy
Logistic Regression	0.8464	0.833333
SVM	0.8482	0.833333
Decision Tree	0.8893	0.833333
KNN	0.8482	0.833333

```
[44]: model_accuracy_df.plot.bar()
```

```
[44]: <Axes: >
```



Thank you!