
# Winning Space Race with Data Science

Marek Jindrich
19/11/2023

# Outline

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

# Executive Summary

- ## Summary of methodologies

- Data collection, Data preparation, Data Wrangling, EDA, Machine Learning predictions. Conclusions.

- ## Summary of all results

- SpaceX has a great product and betting agains them will be difficult
- They have state of the art boosters mainly FT booster seems to be the best for any weight load and V1.1 beign extremely unstable
- They have a high success launch rate especially from sites KSC LC-39A and VAFB SLC 4E at 77% !!
- ML has been done on a small dataset the the results might not be satisfactory
- **BUSINESS** oportunity might be at weight loads over 6000 kg, since SpaceX has been having many rapid unscheduled disesembly events.

# Introduction

- Project background and context
- To present a competition to spaceX
- Problems you want to find answers
- We want to determine weather first stage of Falcon 9 rocket will land to be able to determine the cost of the launch, this can be benefitial for other company if the will bit against the space X

Section 1

# Methodology

# Methodology

Executive Summary

- 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

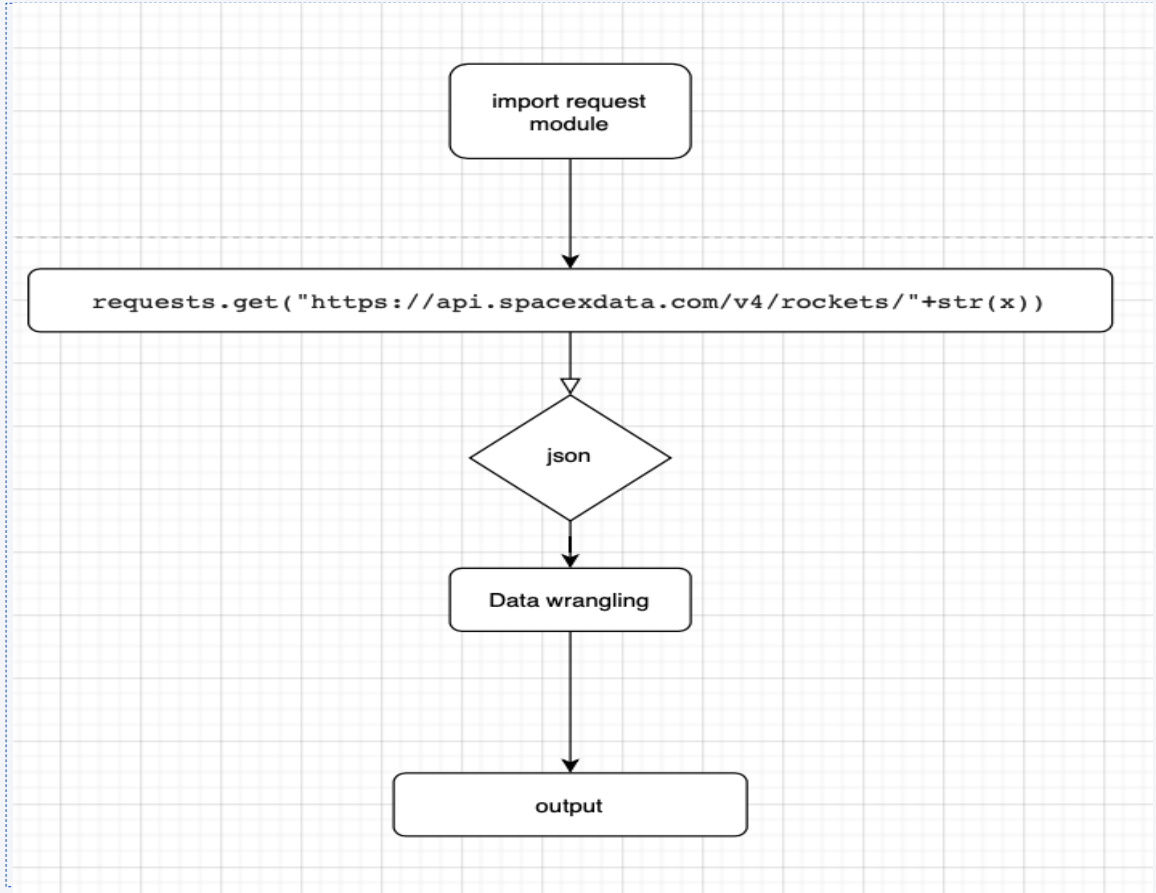
- Data were collected via Spacex api ( api.spacexdata.com) and scraping the wikipedia

# Data Collection – SpaceX API

- For data collection we can you spaceX API, with request module from python

GitHub URL

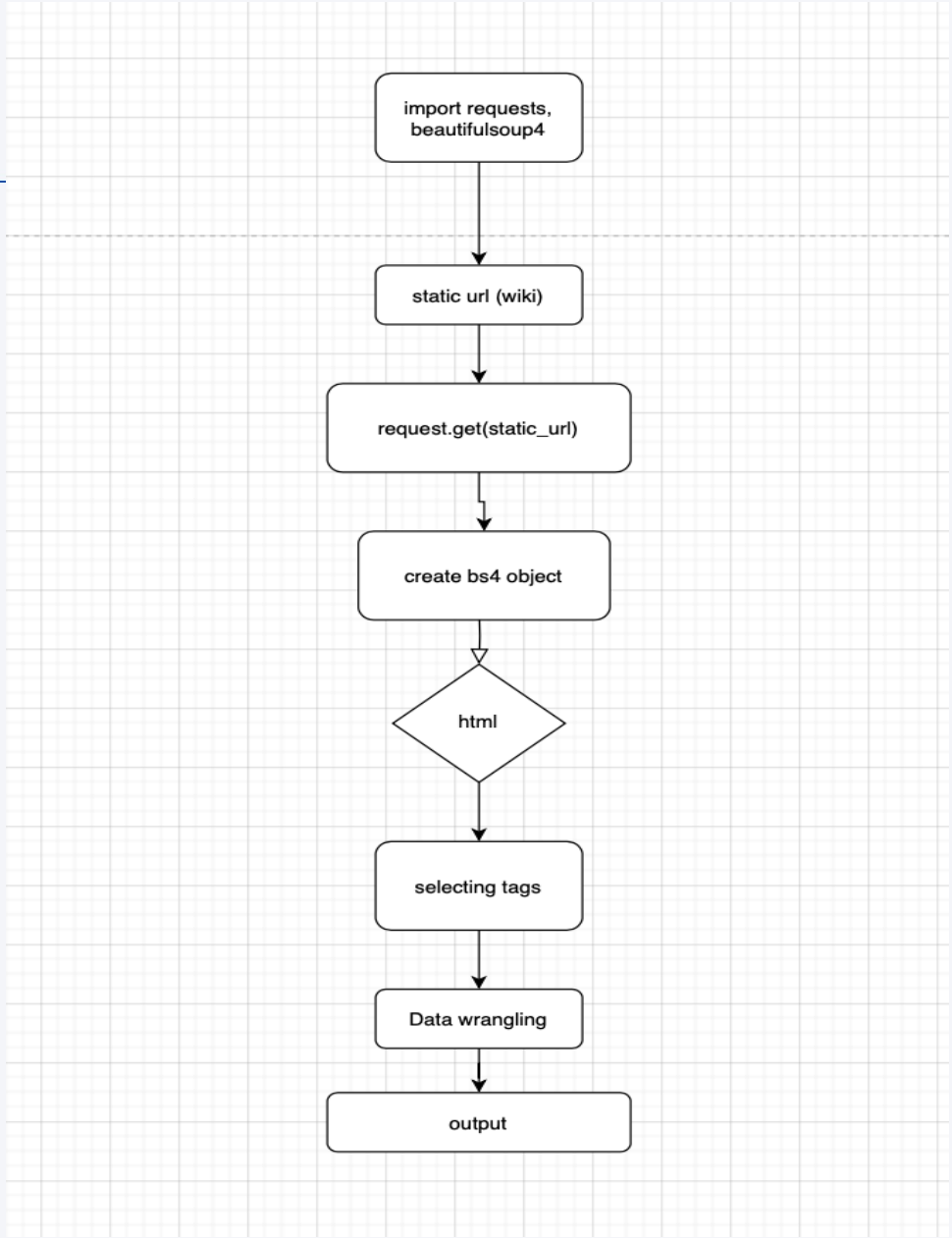
https://github.com/fxmooger/testrepo/blob/main/module_1_data-collection-api.ipynb



# Data Collection - Scraping

- For additional data we used scraping method to obtain data from wikipedia

GitHub URL
https://github.com/fxmooger/testrepo/blob/main/module_1_webscraping.ipynb



# Data Wrangling

- Data were processed using Pandas library, using various techiniques
- Replacing missing values, basic statistics, exploratory analysis

GitHub URL
https://github.com/fxmooger/testrepo/blob/main/module_1_data_wrangling.ipynb

# EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts

I explored FlightNumber and PayloadMass relationship, indicating that higher flight numbers and lower payload masses correlate with successful Falcon 9 first stage landings.

I analyzed LaunchSite impact on success, revealing varying success rates among sites, with CCAFS LC-40 at 60% and KSC LC-39A/VAFB SLC 4E at 77%.

I examined Orbit types, showing high success rates for orbits like ES-L1, GEO, HEO, and SSO, and revealed varying success factors in LEO and GTO orbits.

I visualized yearly success rate trend, illustrating a consistent increase from 2013 to 2020, providing insights for future success predictions.

GitHub URL
https://github.com/fxmooger/testrepo/blob/main/module_2_EDA_Visuals.ipynb

# EDA with SQL

- Using bullet point format, summarize the SQL queries you performed

**Task 1:** Explored unique launch sites.

**Task 2:** Examined 5 records from launch sites starting with 'KSC'.

**Task 3:** Calculated the total payload mass for NASA (CRS) missions.

**Task 4:** Found the average payload mass for booster version F9 v1.1.

**Task 5:** Identified the earliest successful landing on a drone ship.

**Task 6:** Listed boosters with successful ground pad landings and payload mass between 4000 and 6000.

**Task 7:** Counted the total number of successful and failed missions.

**Task 8:** Identified booster versions that carried the maximum payload mass.

**Task 9:** Explored records for successful ground pad landings in 2017.

**Task 10:** Ranked landing outcomes between June 4, 2010, and March 20, 2017, in descending order.



GitHub URL https://github.com/fxmooger/testrepo/blob/main/module_2_EDA_SQL.ipynb

# Build an Interactive Map with Folium

**Map Elements:**

*Markers:* Launch site locations.

*Circles:* Highlighted areas.

*Marker Clusters:* Grouped same-location markers.

*PolyLines:* Distances to proximities.

**Purpose:**

Visualize launch sites and areas efficiently.

Improve readability with marker clusters.

Show distances to key locations.

**Findings:**

Launch sites aren't uniformly distributed along the Equator line.

All launch sites strategically located close to the coast.

Proximity to railways indicates a vital transportation link for materials and rockets.

One launch site (KSC LC-39A) is strategically close to a highway, potentially in a military area.

Launch sites maintain a distance from cities, likely to mitigate noise impact.

**GitHub** URL https://github.com/fxmooger/testrepo/blob/main/module_3_folium.ipynb

# Build a Dashboard with Plotly Dash

- **Summarize what plots/graphs and interactions you have added to a dashboard**

I added a Launch Site Dropdown for users to select specific launch sites or view data for all sites. The Success Pie Chart visualizes launch success outcomes, showing either site-specific or overall success percentages. A Payload Range Slider allows users to explore the correlation between payload mass and launch success. The Success-Payload Scatter Chart provides a detailed view based on launch site and payload range selections.

- **Explain why you added those plots and interactions**

I added these elements to offer users an interactive exploration of SpaceX launch data. The dropdown and pie chart give quick insights, while the slider and scatter chart provide more detailed analyses, allowing users to understand the impact of launch site and payload mass on mission success.

.

**GitHub** URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

https://github.com/fxmooger/testrepo/blob/main/module_3_DASH_spacex_dash_app.py

# Predictive Analysis (Classification)

- **Summarize how you built, evaluated, improved, and found the best performing classification model**

I performed Exploratory Data Analysis (EDA) and created a classification model for SpaceX launch success prediction. Here is a summary of the key steps and results:

- **You need present your model development process using key phrases and flowchart**

In this project, the goal was to predict SpaceX launch success through exploratory data analysis (EDA) and classification modeling. The data underwent preprocessing, including standardization and splitting into training and testing sets. Four classification models—Logistic Regression, SVM, Decision Tree, and KNN—were employed, and hyperparameter tuning was performed using GridSearchCV. Evaluation on test data revealed accuracy scores: Logistic Regression (88.89%), SVM (88.89%), Decision Tree (77.78%), and KNN (83.33%). The Decision Tree model demonstrated the highest accuracy, making it the recommended choice for predicting SpaceX launch success based on the provided features.

- **GitHub URL**

https://github.com/fxmooger/testrepo/blob/main/module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

# Results

- ## Exploratory data analysis results

Exploratory data analysis reveals a majority of successful SpaceX launches, varied payload masses, four launch sites, and the need for further investigation into the correlation between different variables.

- ## Interactive analytics demo in screenshots

- Later in the presentation

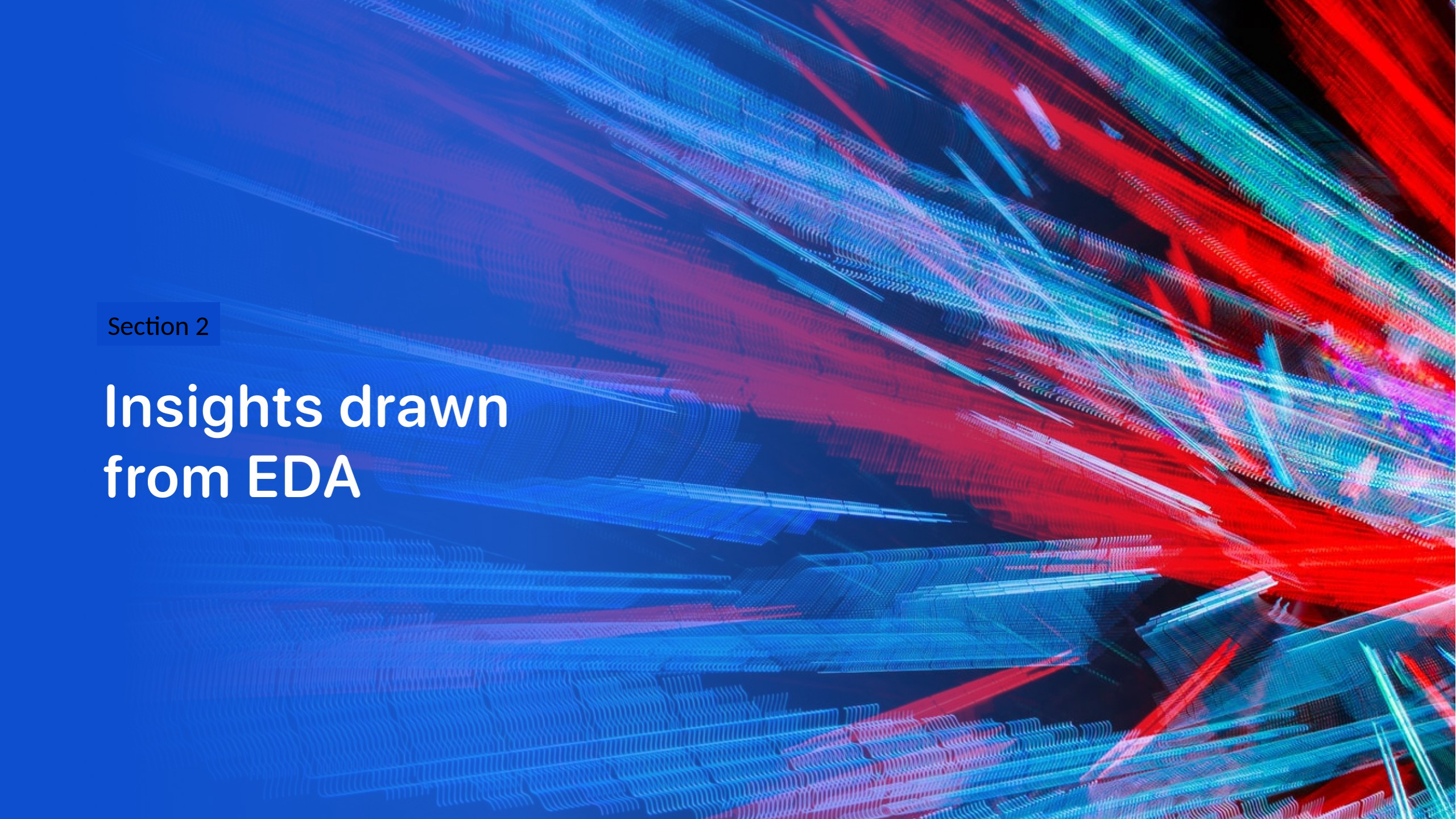
- ## Predictive analysis results

•

The predictive analysis results indicate the accuracy of different classification models in predicting SpaceX launch success based on the provided features. Here are the accuracy scores for each model on the test data:

Logistic Regression: 83.33%
SVM (Support Vector Machine): 83.33%
Decision Tree: 88.89%
KNN (K Nearest Neighbors): 83.33%

Among these models, the Decision Tree model achieved the highest accuracy of 88.89%. Therefore, the Decision Tree model is recommended for predicting SpaceX launch success using the given dataset and features.


Section 2
Insights drawn from EDA

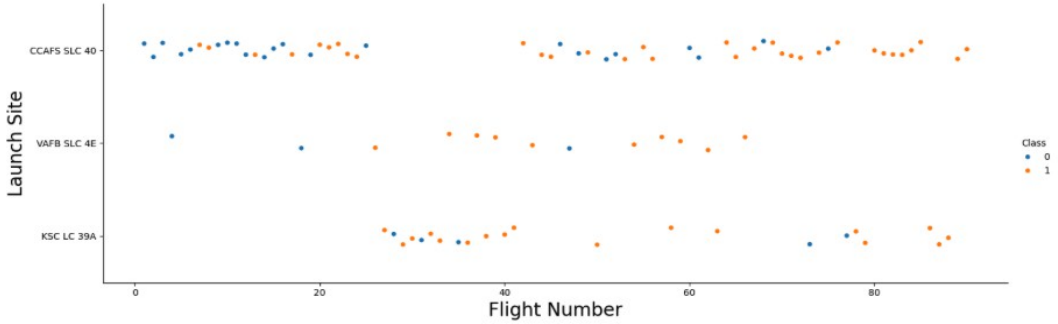
# Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site

Expl: The scatter plot shows the relationship between Flight Number and Launch Site.

Different launch sites have different success rates for landing.

```
In [8]: # 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
sns.catplot(x="FlightNumber", y="LaunchSite", hue="Class", data=df, aspect=3)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
plt.show()
```



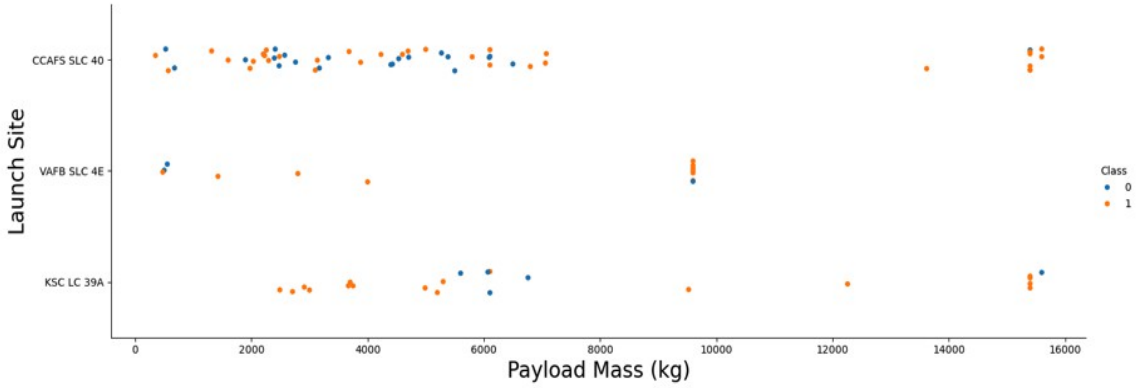
# Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

Expl: The scatter plot shows the relationship between Payload Mass and Launch Site and success of starts.

VAFB-SLC launch site has no rockets launched for heavy payload mass (greater than 10000).

In [9]:
```
# 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
sns.catplot(x="PayloadMass", y="LaunchSite", hue="Class", data=df, aspect=3)
plt.xlabel("Payload Mass (kg)", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
plt.show()
```



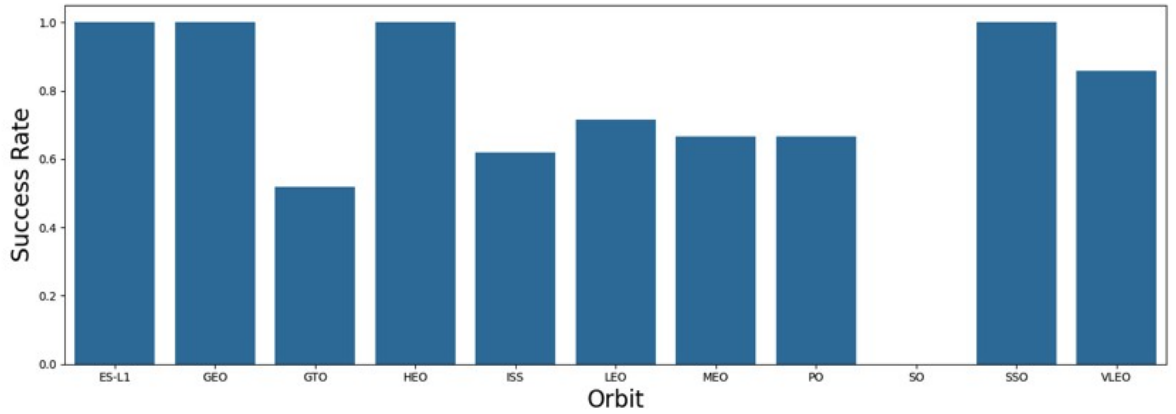
# Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

Expl: Orbits like ES-L1, GEO, and HEO and SSO have high success rates.

In [11]:

```
# HINT use groupby method on Orbit column and get the mean of Class column
orbit_success_rate = df.groupby('Orbit')['Class'].mean().reset_index()
sns.barplot(x='Orbit', y='Class', data=orbit_success_rate)
plt.xlabel("Orbit", fontsize=20)
plt.ylabel("Success Rate", fontsize=20)
plt.show()
```

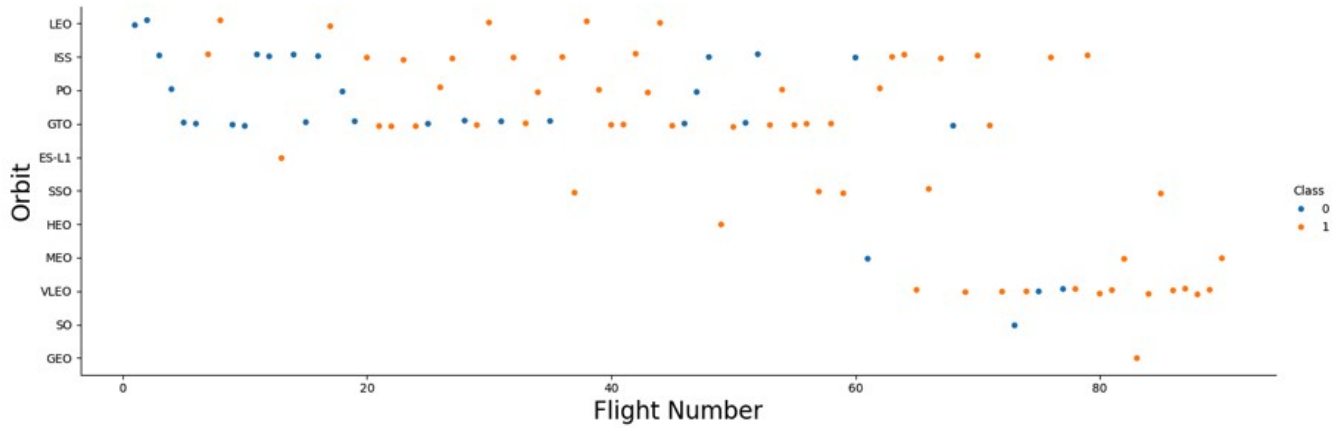


# Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

In the LEO orbit, the success appears related to the number of flights; no clear relationship in GTO orbit.

In [12]:
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class v
sns.catplot(x="FlightNumber", y="Orbit", hue="Class", data=df, aspect=3)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```



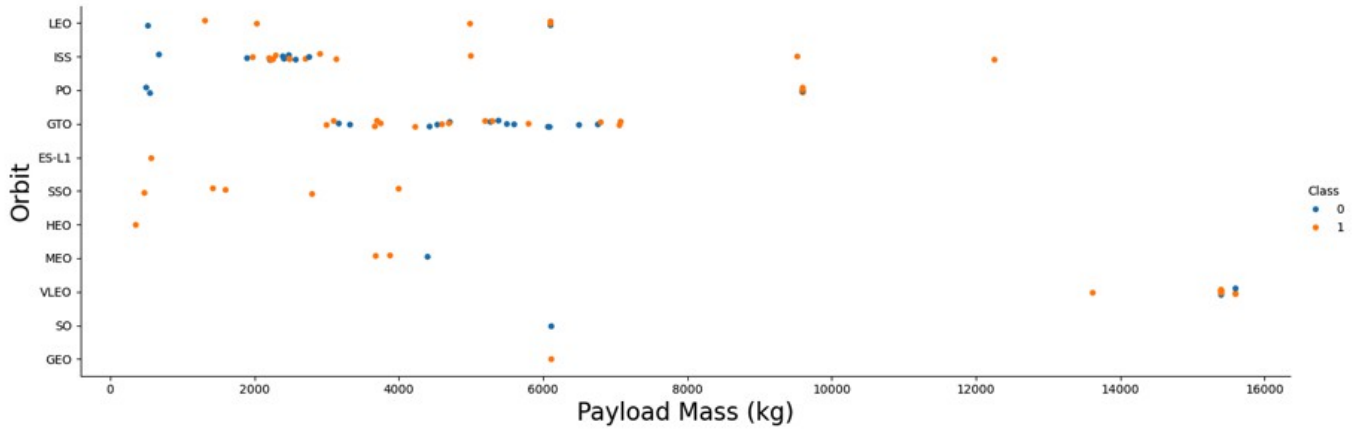
# Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

Expl: With heavy payloads, the successful landing or positive landing rate is more for Polar, LEO, and ISS.

For GTO, it's challenging to distinguish between positive and negative landing.

In [13]:
```
# 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(x="PayloadMass", y="Orbit", hue="Class", data=df, aspect=3)
plt.xlabel("Payload Mass (kg)", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```



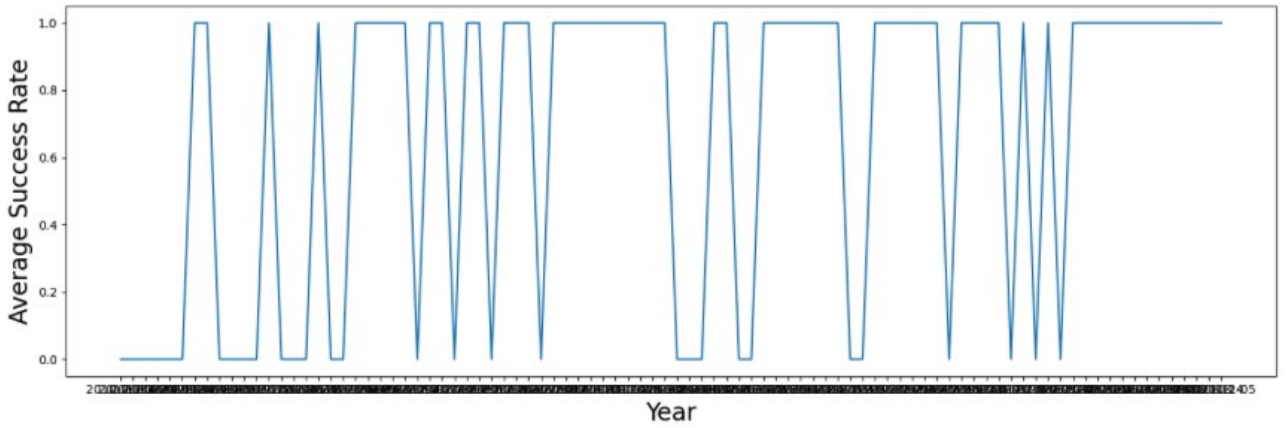
# Launch Success Yearly Trend

- Show a line chart of yearly average success rate

The success rate has been increasing since 2013 until 2020.

In [14]:

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
yearly_success_rate = df.groupby('Date')['Class'].mean().reset_index()
sns.lineplot(x='Date', y='Class', data=yearly_success_rate)
plt.xlabel("Year", fontsize=20)
plt.ylabel("Average Success Rate", fontsize=20)
plt.show()
```



# All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here
-

- CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, CCAFS SLC-40
- This is the query

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

# Launch Site Names Begin with 'KSC'

- Find 5 records where launch sites' names start with `KSC`
- Present your query result with a short explanation here

```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'KSC%' LIMIT 5;
```

Out[10]:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Ou |
|---|---|---|---|---|---|---|---|---|---|
| 2017-02-19 | 14:39:00 | F9 FT B1031.1 | KSC LC-39A | SpaceX CRS-10 | 2490 | LEO (ISS) | NASA (CRS) | Success | Success ( |
| 2017-03-16 | 6:00:00 | F9 FT B1030 | KSC LC-39A | EchoStar 23 | 5600 | GTO | EchoStar | Success | No a |
| 2017-03-30 | 22:27:00 | F9 FT B1021.2 | KSC LC-39A | SES-10 | 5300 | GTO | SES | Success | Success |
| 2017-05-01 | 11:15:00 | F9 FT B1032.1 | KSC LC-39A | NROL-76 | 5300 | LEO | NRO | Success | Success ( |
| 2017-05-15 | 23:21:00 | F9 FT B1034 | KSC LC-39A | Inmarsat-5 F4 | 6070 | GTO | Inmarsat | Success | No a |

# Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here
-

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TotalPayloadMass FROM SPACEXTABLE WHERE Customer = 'NASA
(CRS)';
```

```
TOTAL PAYLOADMASS IS 45596
```

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here
-

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AveragePayloadMass FROM SPACEXTABLE WHERE Booster_Version
= 'F9 v1.1';
```

```
AVERAGE PAYLOAD MASS for boosters version F9 v1.1 is 2928.4
```

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on drone ship.Present your query result with a short explanation here

```
%sql SELECT MIN(Date) AS SuccessfulLandingDate FROM SPACEXTABLE WHERE Landing_Outcome = 'Success
(drone ship)';
```

FOUND JUST ONE DATE : 2016-04-08

# 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

- Present your query result with a short explanation here

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)' AND
PAYLOAD_MASS__KG_ BETWEEN 4000 AND 600
```

```
BOOSTER VERSIONS:

F9 FT B1032.1

F9 B4 B1040.1

F9 B4 B1043.1
```

# Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here
-

```
%sql SELECT COUNT(*) AS TotalSuccessfulMissions FROM SPACEXTABLE WHERE Mission_Outcome LIKE
'Success%' UNION ALL SELECT COUNT(*) AS TotalFailureMissions FROM SPACEXTABLE WHERE
Mission_Outcome LIKE 'Failure%';
```

```
100 Successful only 1 Failure.
```

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here
-

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

Out[19]:

| 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

- List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

- Present your query result with a short explanation here

%sql SELECT SUBSTR(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE SUBSTR(Date, 0, 5) = '2017' AND Landing_Outcome LIKE 'Success (ground pad)'

Out[20]:

| Month | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|
| 02 | Success (ground pad) | F9 FT B1031.1 | KSC LC-39A |
| 05 | Success (ground pad) | F9 FT B1032.1 | KSC LC-39A |
| 06 | Success (ground pad) | F9 FT B1035.1 | KSC LC-39A |
| 08 | Success (ground pad) | F9 B4 B1039.1 | KSC LC-39A |
| 09 | Success (ground pad) | F9 B4 B1040.1 | KSC LC-39A |
| 12 | Success (ground pad) | F9 FT B1035.2 | CCAFS SLC-40 |

# 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

- Present your query result with a short explanation here

%sql SELECT Landing_Outcome, COUNT(*) AS OutcomeCount FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY OutcomeCount DESC;

Out[21]:

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


Section 3
Launch Sites
Proximities Analysis

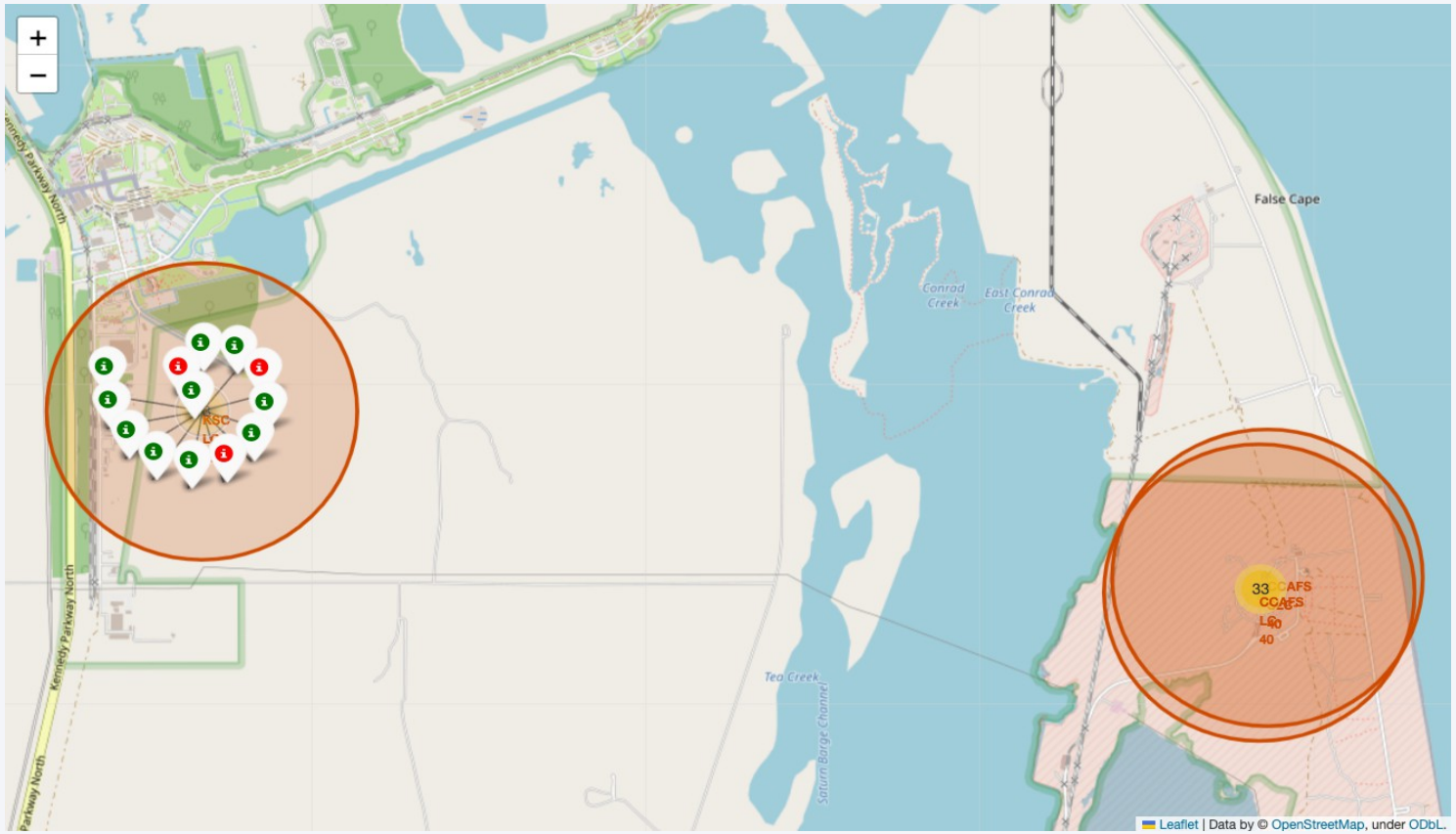
# All launch sites

Explain the important elements and findings on the screenshot

- There are launching sites in California and Florida, they all are close to the ocean,

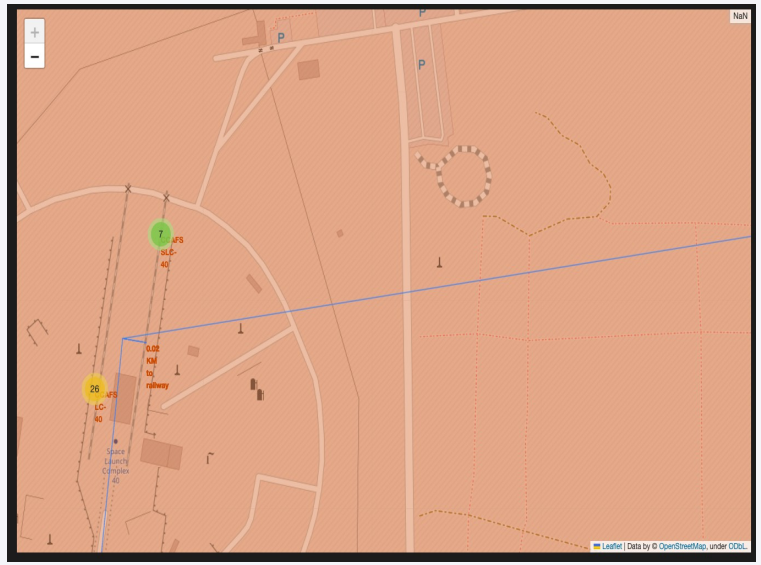
# Color labeles launch outcomes

- Explain the important elements and findings on the screenshot
- It represents the successes and failures in the launches, for wach launching site there are multiple outcomes

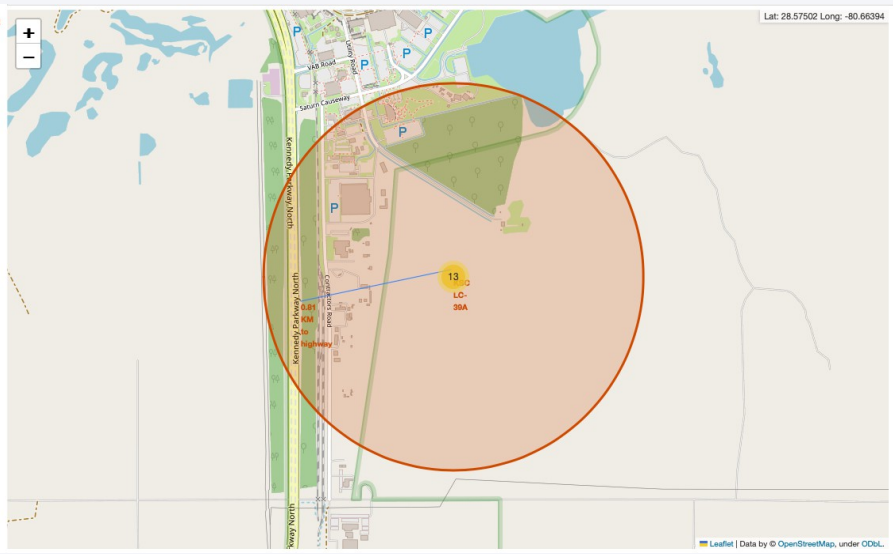


# Calculated distances

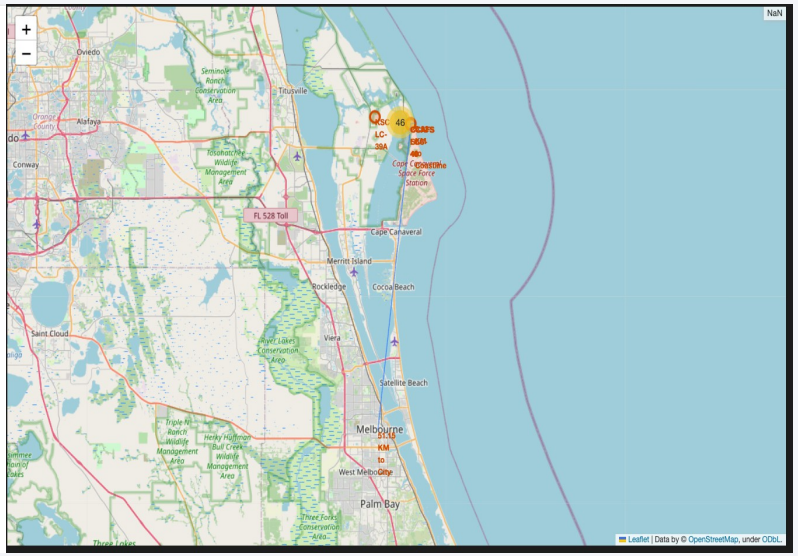
- From train 0.02km

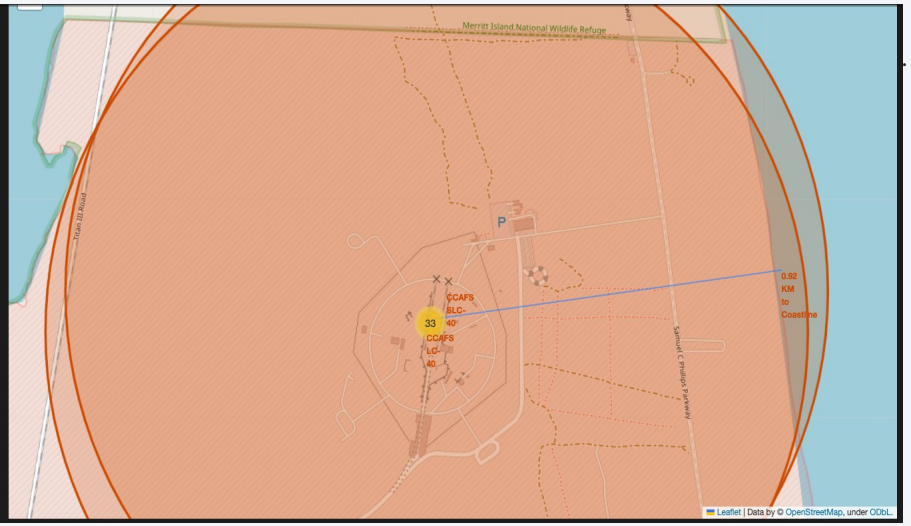



- From highway 0.81km

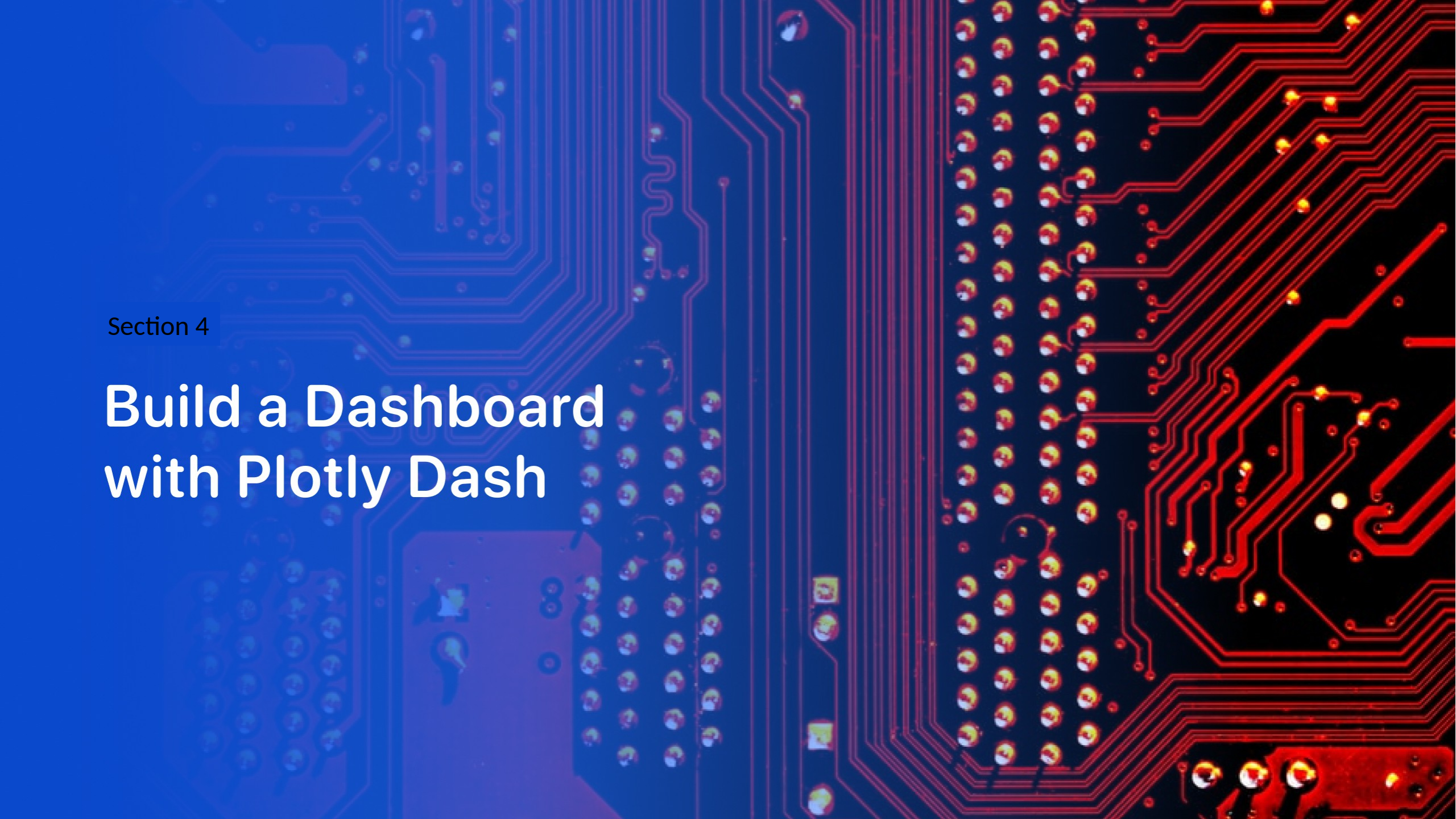



- From city 51km



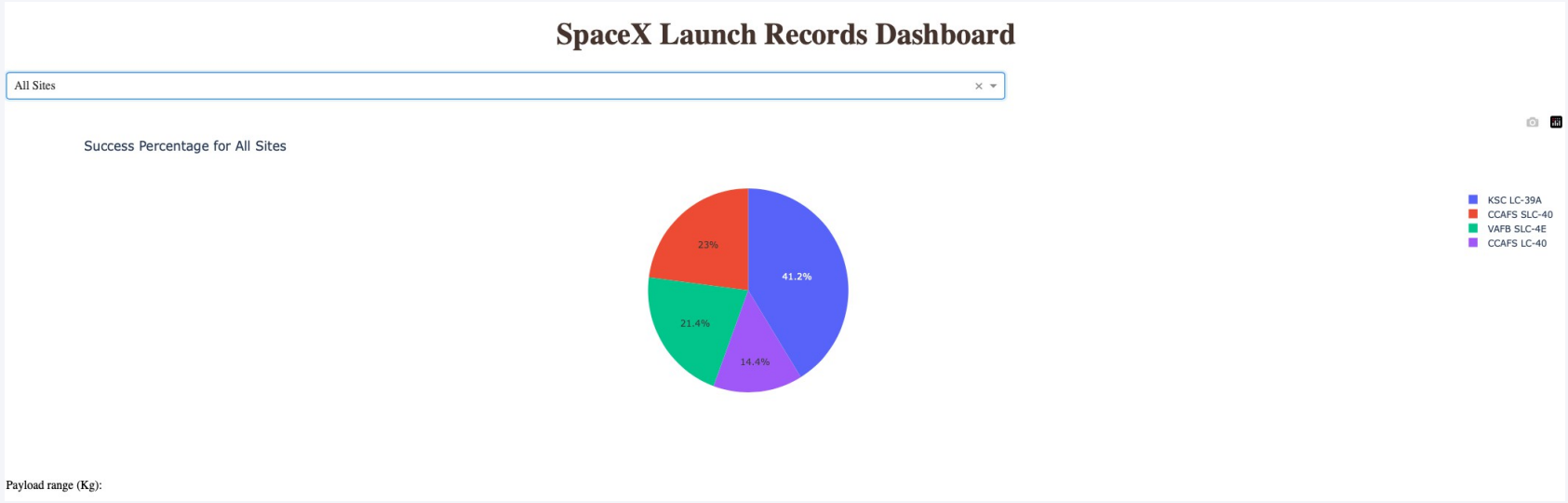

- From ocean 0.92km


Section 4
Build a Dashboard with Plotly Dash

# Launch success count for all sites

Explain the important elements and findings on the screenshot

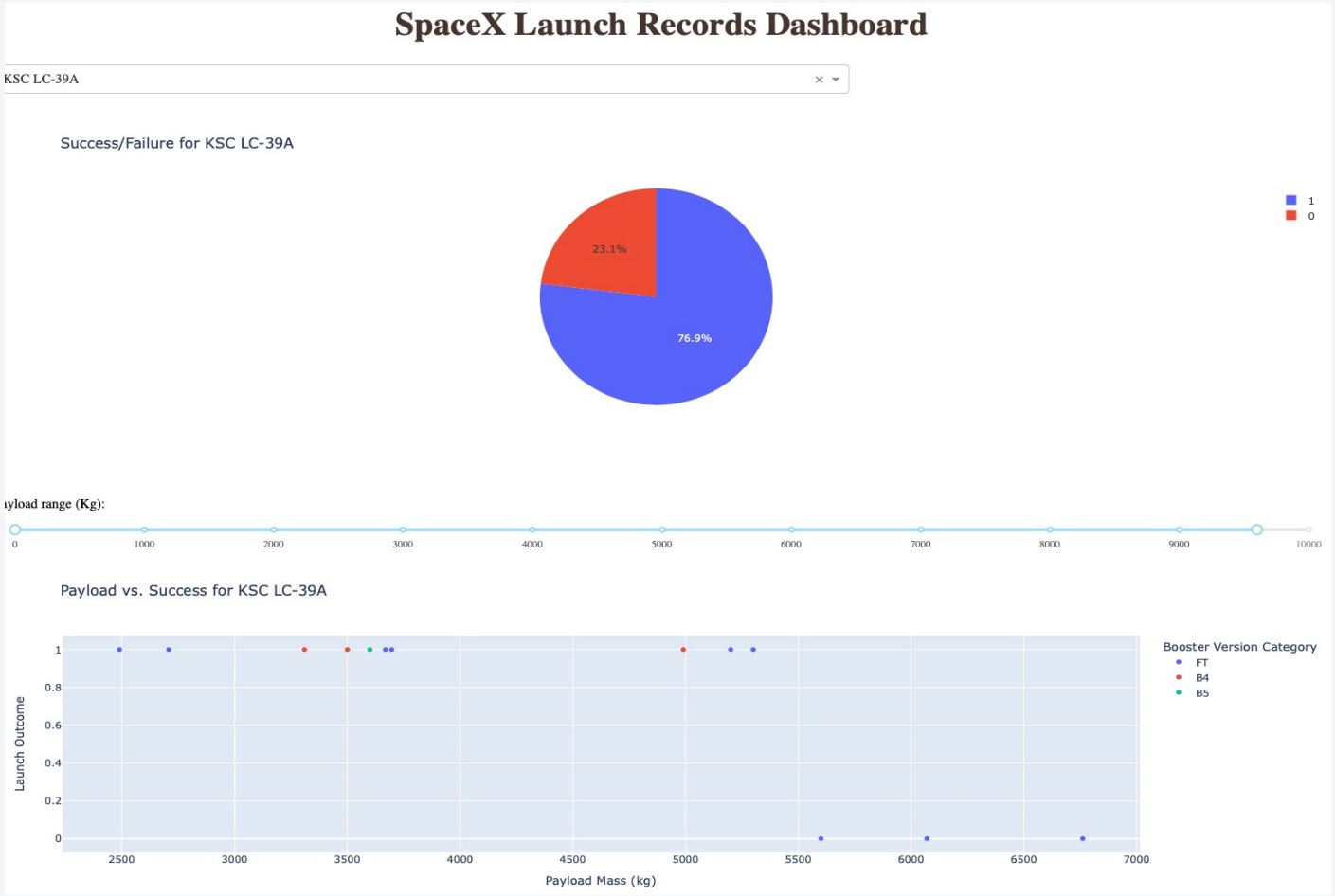
- KSC LC-39A is the most successful launching site with over 40% success rate.



# Highest launch success ratio

## Explain the important elements and findings on the screenshot

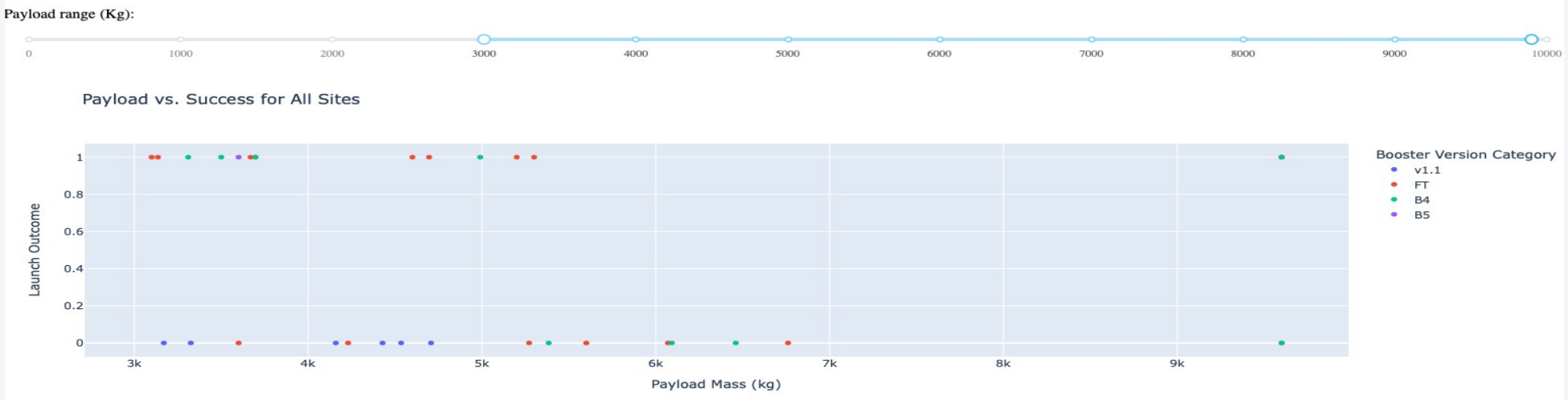
- The most successful site is KSC LC-39A with up to 5500 kg payload, all attempts above that weight failed.



# Payload vs. Success for All sites

Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

- V1.1 is extremely unstable booster, the best seems to be FT for ANY weight load, TOP load is about 9000kg with 50% chance of success

Section 5

# Predictive Analysis (Classification)

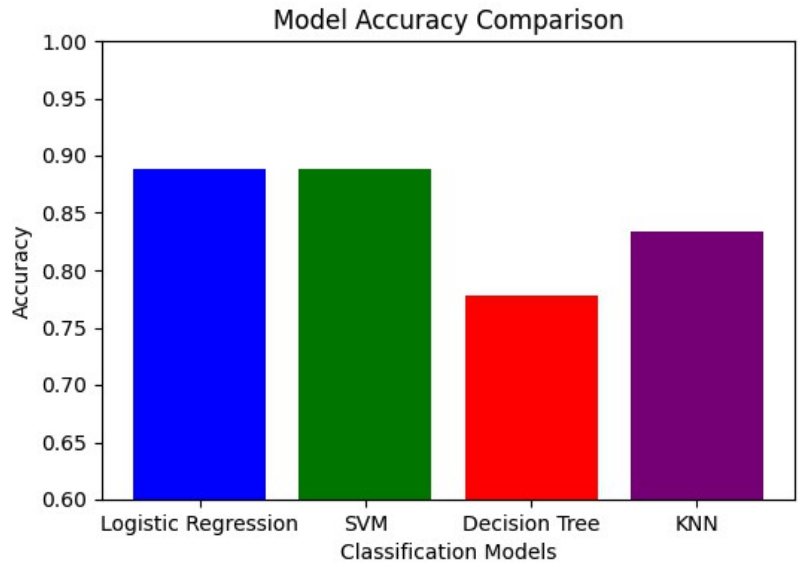
# Classification Accuracy

2 models had the same score with cross validation, even the confusion matrix showed same results. Main reason might be not enough data.

```
[78]: accuracy_scores = [accuracy_logreg, accuracy_svm, accuracy_tree, accuracy_knn]


      model_names = ['Logistic Regression', 'SVM', 'Decision Tree', 'KNN']


      plt.bar(model_names, accuracy_scores, color=['blue', 'green', 'red', 'purple'])
      plt.ylim(0.6, 1.0)
      plt.title('Model Accuracy Comparison')
      plt.xlabel('Classification Models')
      plt.ylabel('Accuracy')
      plt.show()
```



# Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation
- Model is good at prediciting successful launch, and less good predicting failed lauch

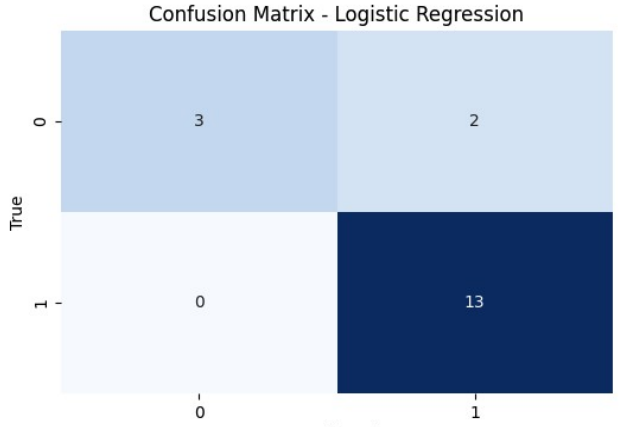
[179]:
```
def plot_confusion_matrix(y_true, y_pred, method_name):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
    plt.title(f'Confusion Matrix - {method_name}')
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()

# Logistic Regression
yhat_logreg = logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat_logreg, 'Logistic Regression')

# SVM
yhat_svm = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat_svm, 'SVM')

# Decision Tree
yhat_tree = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat_tree, 'Decision Tree')

# KNN
yhat_knn = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat_knn, 'KNN')
```



# Conclusions

1. In examining the classification models, it's intriguing that both Support Vector Machine (SVM) and Logistic Regression yielded identical accuracy scores. However, a closer inspection using precision, recall, and F1-score metrics exposed nuances in their performances. The confusion matrices further illuminated their strengths and weaknesses, offering valuable insights into true positives, true negatives, false positives, and false negatives.

2. An interesting finding emerged when tweaking hyperparameters through GridSearchCV, suggesting that optimizing these parameters could potentially enhance the models. This indicates the significance of fine-tuning for achieving better results.

3. The nature of the dataset seems to play a pivotal role, with similar accuracy scores possibly hinting at a balanced influence that neither strongly favors SVM nor LogReg. This underscores the importance of understanding dataset characteristics.

4. As we navigate through this iterative process of evaluation metrics, confusion matrices, and hyperparameter tuning, a deeper understanding of each model's behavior unfolds. This iterative analysis serves as a roadmap for refining model performance, ensuring adaptability to the specific intricacies of the given problem.

5. More data would have been benefitial.

# Appendix

- I used Python, Jupyter notbook, Plotly, FlowChart editor…
- I added all the notebooks, functions all the pictures and flowcharts into GitHub.
- All the python code added is included in the notebooks.

# Thank you!