IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kexin Jiao
04/17/2024

Outline

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

Executive Summary

- **Summary of methodologies**

➢ Data collection through API

➢ Data collection with web scraping

➢ Data wrangling

➢ Exploratory data analysis with SQL

➢ Exploratory data analysis with data visualization

➢ Interactive visual analytics with folium

➢ Machine learning prediction

- **Summary of all results**

➢ Exploratory data analysis result

➢ Interactive analytics in screenshots

➢ Predictive analytics result

Introduction

- Project background and context

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. Spaces X's Falcon 9 launch like regular rockets. we will determine if the first stage will land, and determine the cost of a launch. We will train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

- Problems you want to find answers

➢ The factors that determine if the rocket will land successfully

➢ How do those factors interact to affect the success rate of landing

➢ Build a mode for predicting the success rate of landing based on the major factors

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 collection methods.

➢ Use get request from the SpaceX API

➢ Data was also collected using web scraping from website

➢ Decode the requested content as a Json file followed by turning it into a pandas dataframe

➢ The data collected from website was extracted as HTML tables followed by converting the tables to pandas dataframes

➢ Clean data, fill the missing values with the average value

Data Collection – SpaceX API

- We used get request from the SpaceX API. The collected data was cleaned and reformatted .

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

Now let's start requesting rocket launch data from SpaceX API with the following URL:

In [9]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [10]:
```
response = requests.get(spacex_url)
```

Check the content of the response

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

In [12]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

We should see that the request was successfull with the 200 status response code

In [13]:
```
response.status_code
```

Out[13]: 200

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [14]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data=pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

Data Collection - Scraping

- Web scraping using BeautifulSoup
- The table was parsed and converted into pandas dataframe
- https://github.com/Kexin-Jiao/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb

In [6]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [7]:
```
# use requests.get() method with the provided static_url
# assign the response to a object=requests.get(static_url)
response=requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

In [8]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup=BeautifulSoup(response.content)
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [9]:
```
# Use soup.title attribute
soup.title
```

Out[9]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

- The training labels were determined by performing exploratory data analysis
- The number of each orbits and launches at each site were calculated
- https://github.com/Kexin-Jiao/IBM-Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

The data contains several Space X launch facilities: Cape Canaveral Space Launch Complex 40 **VAFB SLC 4E** , Vandenberg Air Force Base Space Launch Complex 4E **(SLC-4E)**, Kennedy Space Center Launch Complex 39A **KSC LC 39A** .The location of each Launch Is placed in the column `LaunchSite`

Next, let's see the number of launches for each site.

Use the method `value_counts()` on the column `LaunchSite` to determine the number of launches on each site:

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

In [8]:
```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

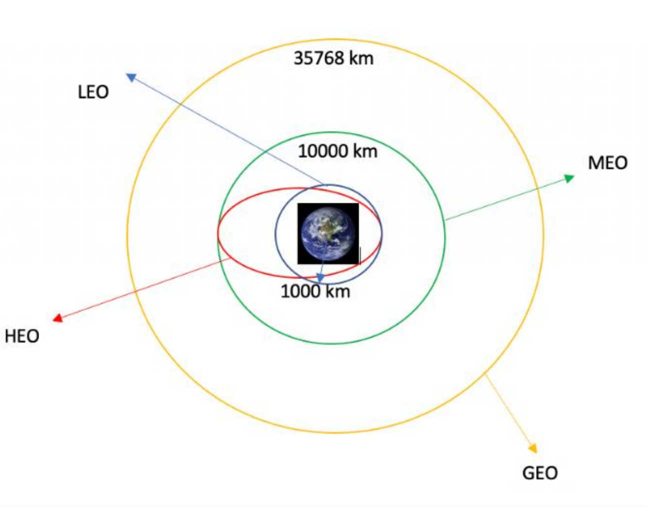
Out[8]:
```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

TASK 3: Calculate the number and occurence of mission outcome of t

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing` variable landing_outcomes.

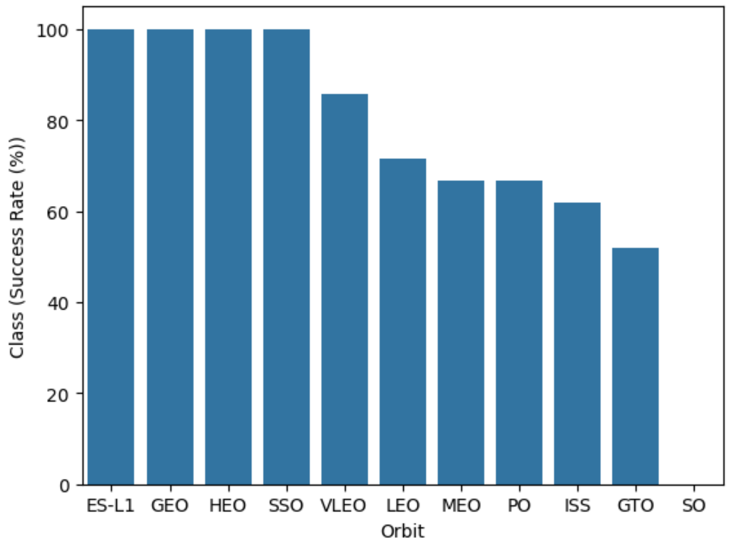
In [28]:
```
# landing_outcomes = values on Outcome column
landing_outcomes=df['Outcome'].value_counts()
landing_outcomes
```

Out[28]:
```
True ASDS     41
None None     19
```

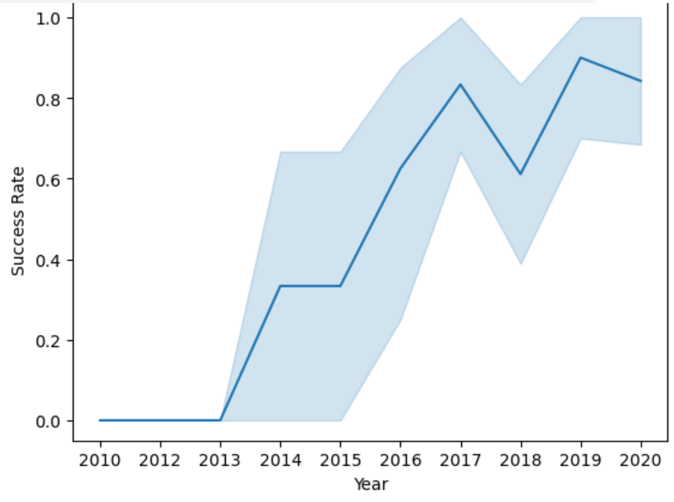


EDA with Data Visualization

- The data was explored using the visualization of the relationship between flight number and launch site, the payload and launch site, the flight number and orbit type, and the launch success trend.
- https://github.com/Kexin-Jiao/IBM-Applied-Data-Science-Capstone/blob/main/edadataviz.ipynb



Analyze the ploted bar chart try to find which orbits have high sucess rate.



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

EDA with SQL

- The SpaceX dataset was loaded into a SQL database in the jupyter notebook. The data was read and explored by applying EDA with SQl.
- Several queries were performed and shown in the figure
- https://github.com/Kexin-Jiao/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

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

Display the total payload mass carried by boosters launched by NASA (CRS)

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

List the date when the first succesful landing outcome in ground pad was acheived.

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Build an Interactive Map with Folium

- All the launch sites were marked with additional map objects to label the success or failure of launches for each site.
- We used the color-labled marker clusters to identify the launch sites having high success rate.
- We calculated the distance between a launch site to the proximities
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

https://github.com/Kexin-Jiao/IBM-Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

The generated map with marked launch sites should look similar to the following:

Now, you can explore the map by zoom-in/out the marked areas , and try to answer the following questions:

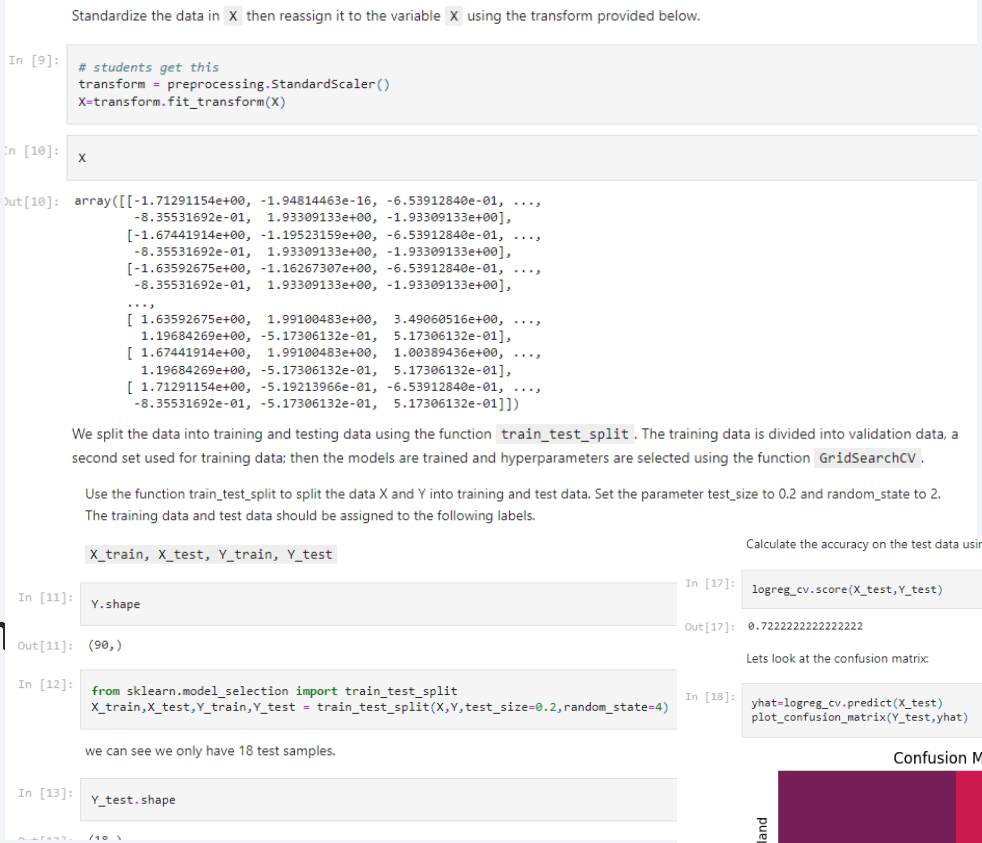
- Are all launch sites in proximity to the Equator line?
- Are all launch sites in very close proximity to the coast?

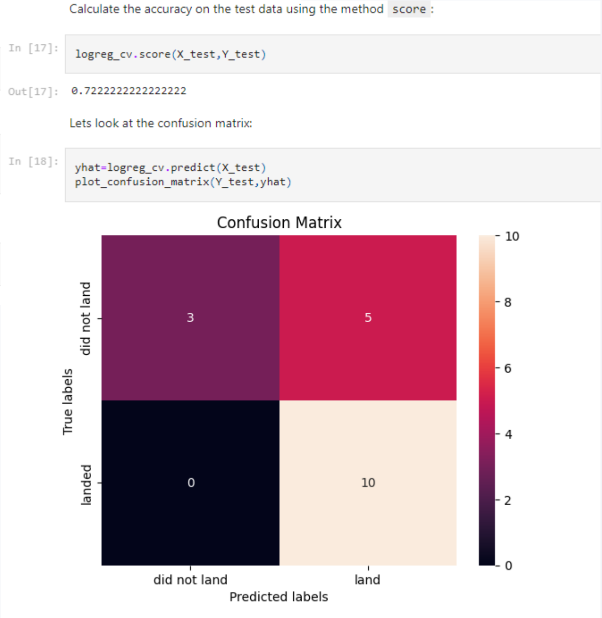
Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- We used pandas and numpy to transform and assign training and testing data groups
- Multiple machine learning models were built and tuned with different hyper parameters using GridSearchCV
- The machine learning models were evaluated using accuracy followed by being optimized using feature engineering and algorithm tunin
- https://github.com/Kexin-Jiao/IBM-Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb





Results

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

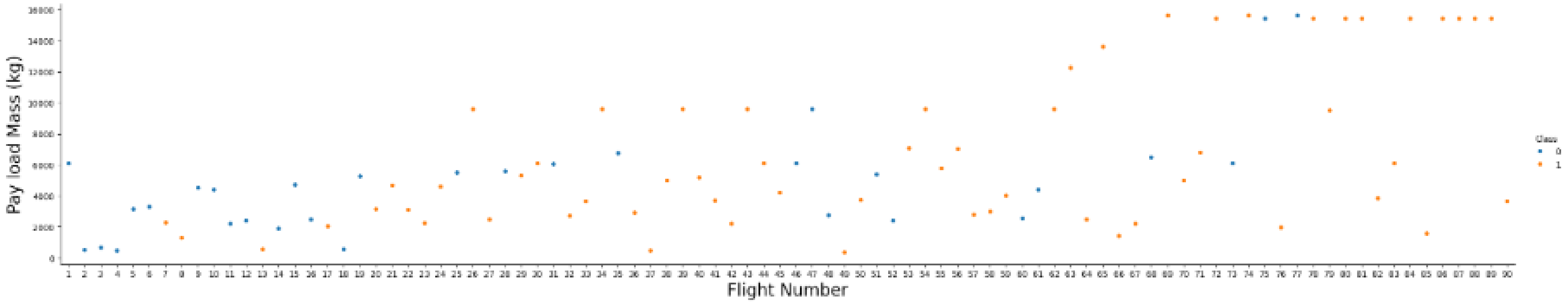
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Conclusion: the larger the flight amount at a launch site, the greater the success rate at a launch site.

In [9]:
```
sns.catplot(y="PayloadMass", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```

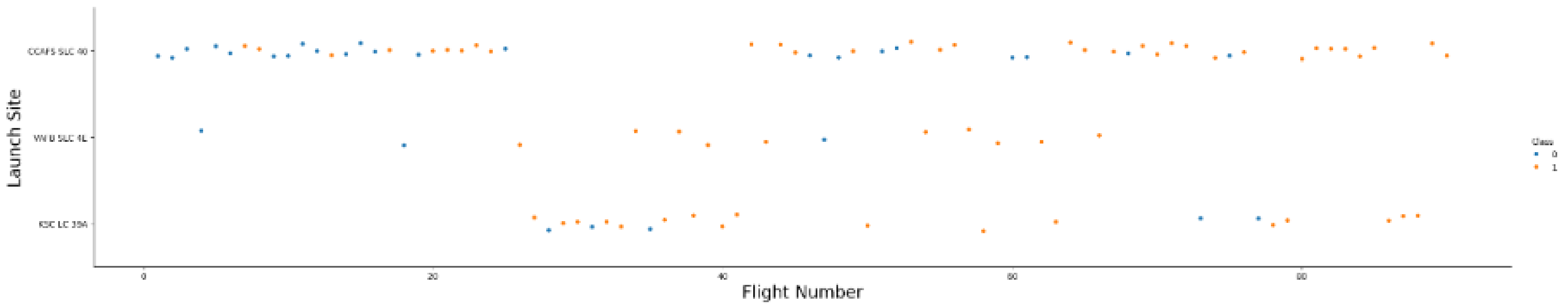


Payload vs. Launch Site

- We found that the greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket

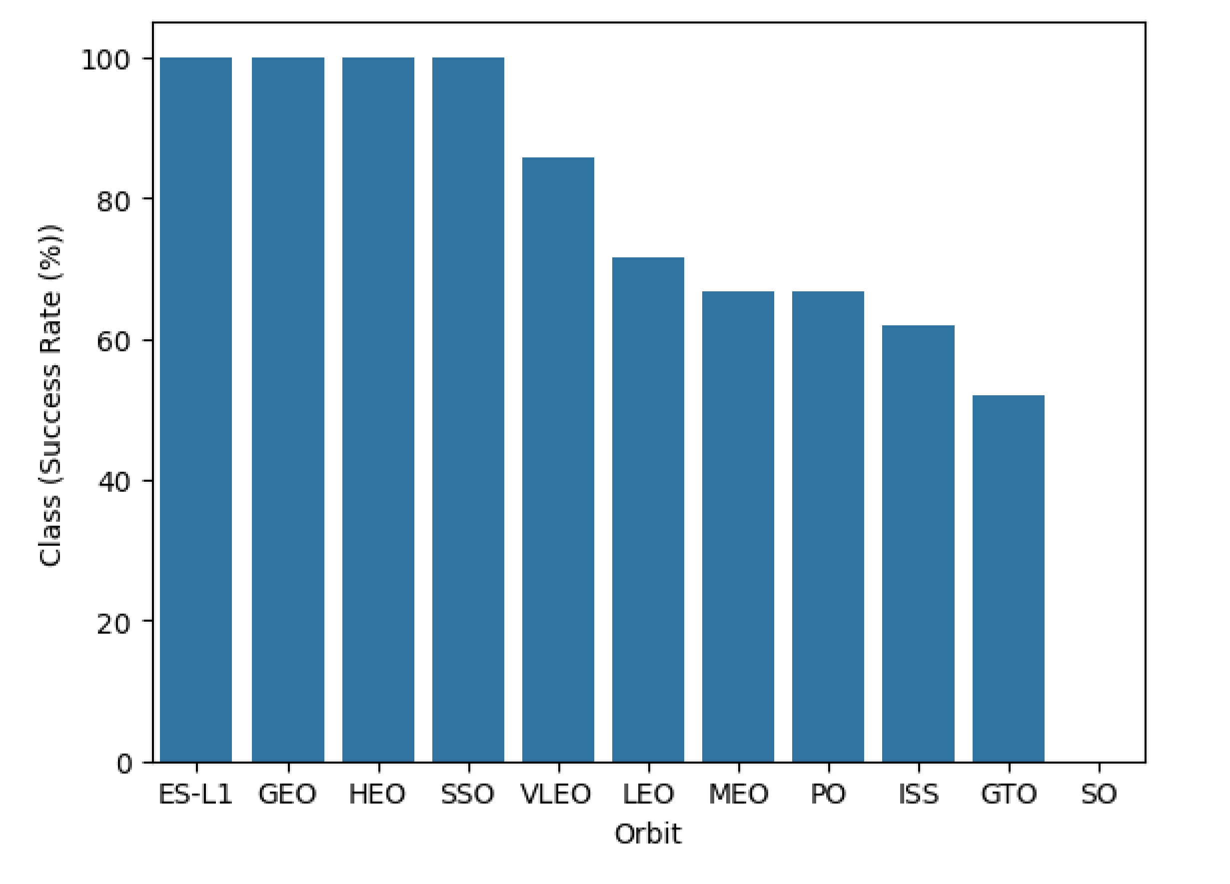
6]:

```
### TASK 1: Visualize the relationship between Flight Number and Launch Site
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()
```



Success Rate vs. Orbit Type

- The figure shows that ES-L1, GEO, HEO, SSO, and VLEO had the most success rate.



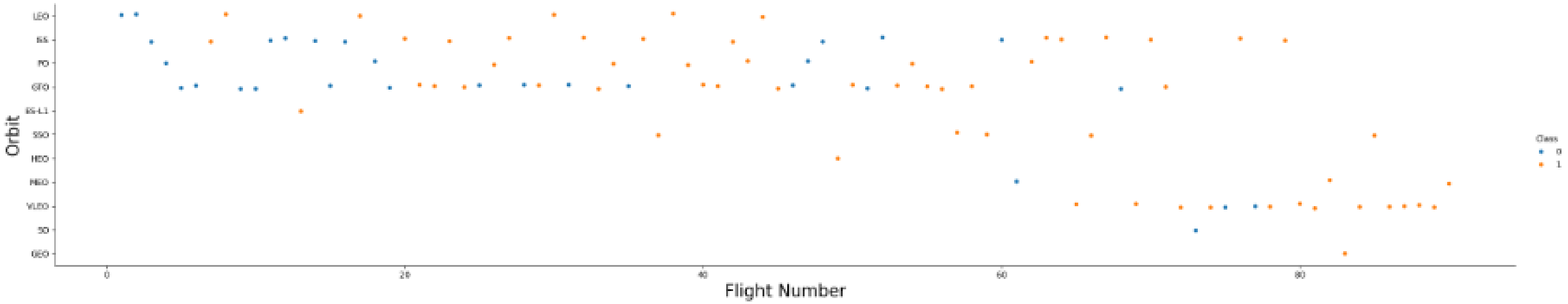
Flight Number vs. Orbit Type

- This plot shows the change at Orbit type as a function of Flight Number. It shows that the LEO orbit, success is related to the number of flights. However, the GTO orbit has no significant relationship with the flight number.

In [14]:

```
# 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()
```



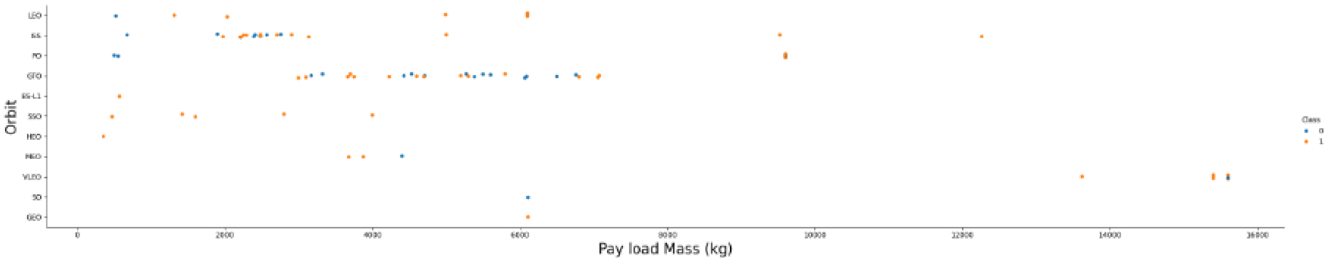
Payload vs. Orbit Type

- It is observed that the PO, LEO, and ISS orbits have higher successful landing rate with heavy payloads.

In [16]:

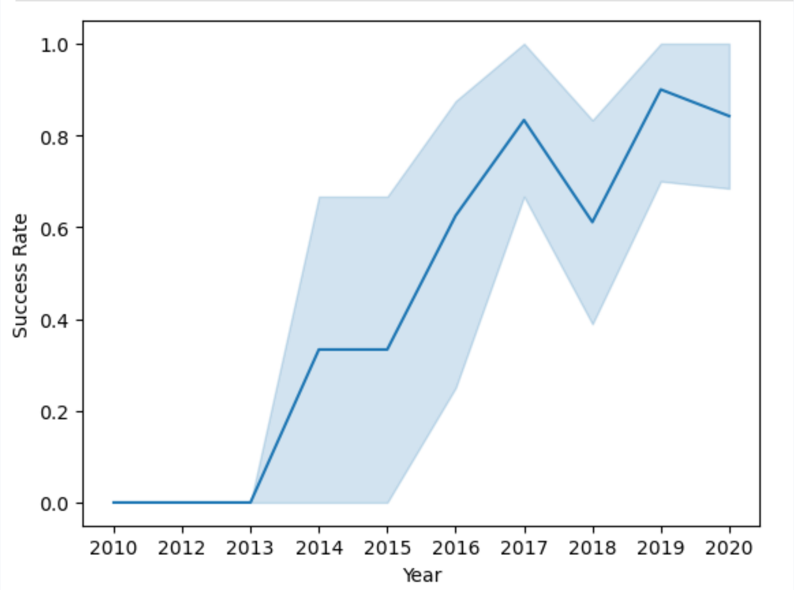
```
### TASK  5: Visualize the relationship between Payload and Orbit type
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)

plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch Success Yearly Trend

- It is observed from the plot that the success rate increased since 2013 till 2020



All Launch Site Names

- The unique launch sites from the SpaceX data were shown using the key word DISTINCT

In [31]:
```
%%sql

select DISTINCT Launch_Site from SPACEXTABLE
```

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

Out[31]:

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

Launch Site Names Begin with 'CCA'

- Five record where launch sites begin with "CCA" were displayed

Display 5 records where launch sites begin with the string 'CCA'

In [32]:

```
%%sql
select* from SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' limit 5
```

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

Out[32]:

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

- The total payload carried y boosters from NASA as 48213 was calculated using the query below

Display the total payload mass carried by boosters launched by NASA (CRS)

In [33]:
```
%%sql
select sum(payload_mass__kg_) from SPACEXTABLE  where customer LIKE '%CRS%'
```

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

Out[33]:

sum(payload_mass__kg_)
48213

Average Payload Mass by F9 v1.1

- The average payload mass carried by the booster version F9 v1.1 is 2928.4 kg

```
In [34]: %%sql
         select avg(payload_mass__kg_) from SPACEXTABLE where booster_version='F9 v1.1'

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

Out[34]:

avg(payload_mass__kg_)
2928.4

First Successful Ground Landing Date

- The first successful landing outcome on ground pas happened on December 22nd 2015.

```
In [48]: %%sql

         select Date  from  SPACEXTABLE where Landing_Outcome ='Success (ground pad)' limit 1;

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

Out[48]:

Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- We list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 using WHERE clause with AND condition

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [50]:
```
%%sql
select Booster_Version from SPACEXTABLE where landing_outcome like '%drone ship%' and payload_mass__kg_>=4000 and payload_ma
```

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

Out[50]:

Booster_Version
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- We used WHERE and “%” to filter the desired values

List the total number of successful and failure mission outcomes

In [51]:
```
%%sql
SELECT Count(mission_outcome) from SPACEXTABLE where mission_outcome like '%Success%'
```

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

Out[51]:

Count(mission_outcome)
100

In [52]:
```
%%sql
SELECT Count(mission_outcome) from SPACEXTABLE where mission_outcome not like '%Success%'
```

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

Out[52]:

Count(mission_outcome)
1

Boosters Carried Maximum Payload

- The booster that carried the maximum payload was determined using the subquery clause and the MAX() function

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [53]:
```
%%sql
SELECT  booster_version FROM SPACEXTABLE where  payload_mass__kg_ = (Select Max(payload_mass__kg_) from SPACEXTABLE
```

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

Out[53]:

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

- We used WHERE clause with LIKE, AND, and BETWEEN conditions to filter out the failed landing outcomes in drone ship, the booster versions, and the launch site names in 2015

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [64]:

```
%%sql
select booster_version,launch_site,landing_outcome,substr(Date,6,2) as Month from SPACEXTABLE where landing_outcome LIKE '%
```

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

Out[64]:

Booster_Version	Launch_Site	Landing_Outcome	Month
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)	01
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)	04
F9 v1.1 B1018	CCAFS LC-40	Precluded (drone ship)	06

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

- We used COUNT, BETWEEN condition in WHERE clause to filter out the landing outcomes between 2010/06/04 to 2010/03/20
- WE applied GROUP BY and ORDER BY clauses to group then order the landing outcome in a descending order.

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.

In [67]:

```
%%sql
select landing_outcome as OUTCOME,count(landing_outcome) as TOTAL from SPACEXTABLE where Date>'2010-06-04' AND Date<='2
```

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

Out[67]:

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

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- The SpaceX launch sites are in Florida and California in United States.

The generated map with marked launch sites should look similar to the following:

Now, you can explore the map by zoom-in/out the marked areas , and try to answer the following questions:

- Are all launch sites in proximity to the Equator line?
- Are all launch sites in very close proximity to the coast?

<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

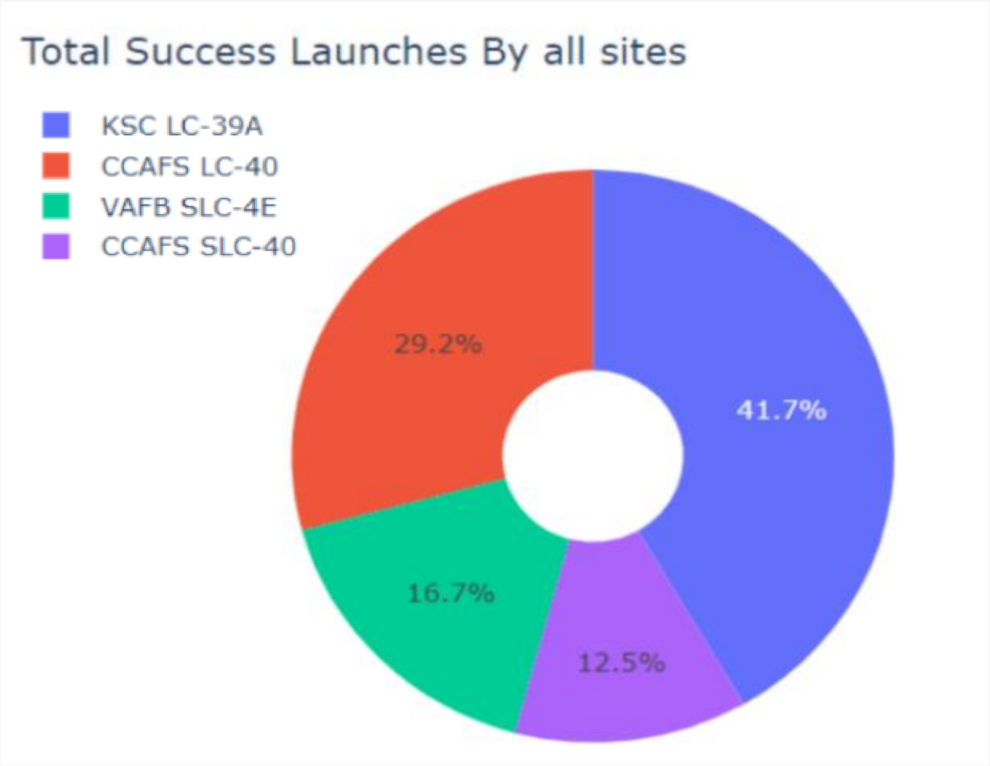
- Explain the important elements and findings on the screenshot

Section 4

Build a Dashboard with Plotly Dash

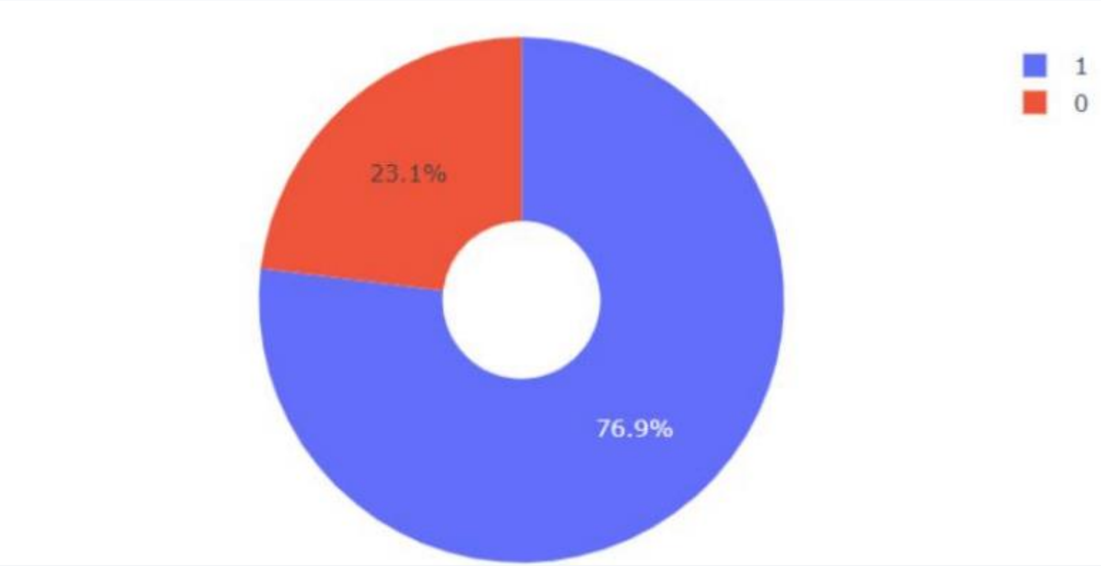
<Dashboard Screenshot 1>

- KSC LC-39A had the most successful launches from all the sites



<Dashboard Screenshot 2>

- KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate



<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier model has the highest classification accuracy

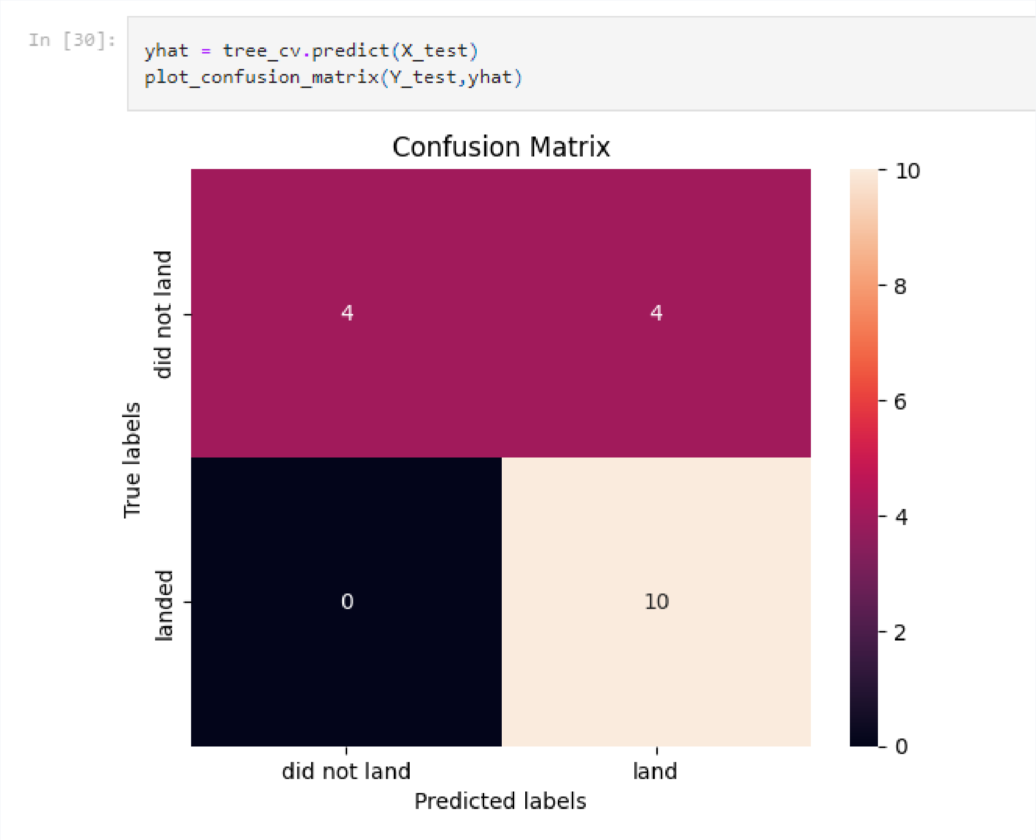
Find the method performs best:

```
In [41]:
algorithms = {'KNN':KNN_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',KNN_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.9053571428571429
Best Params is : {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2,
'splitter': 'random'}
```

Confusion Matrix

- The confusion matrix shows that the model has good accuracy for those successful landings but low accuracy for not successful landings. That means the major issue is the false positives.



Conclusions

- The success rate at a launch site increases as the flight amount at the launch site increases
- The launch success rate increased since 2013 to 2020
- KSC LC-39A has the highest number of successful launches for any launch sites
- The decision tree classifier is the best machine learning model for this task

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

Thank you!