IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sarah Wolff
20/09/2022

Outline

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

Executive Summary

- Predicting whether a SpaceX flight will have a successful first-stage landing
- Using data visualization, exploratory analysis, classification methods
- Finding that which characteristics impact success

Introduction

- The rocket flights offered by SpaceY include reusable parts
- If the first-stage rocket part lands successfully, it can be reused and costs are severely cut
- The company wants to predict whether this will happen for a particular planned flight because it allows for precise cost estimation
- Further, the company wants to uncover which flight characteristics increase the chance for a successful first stage

Section 1

Methodology

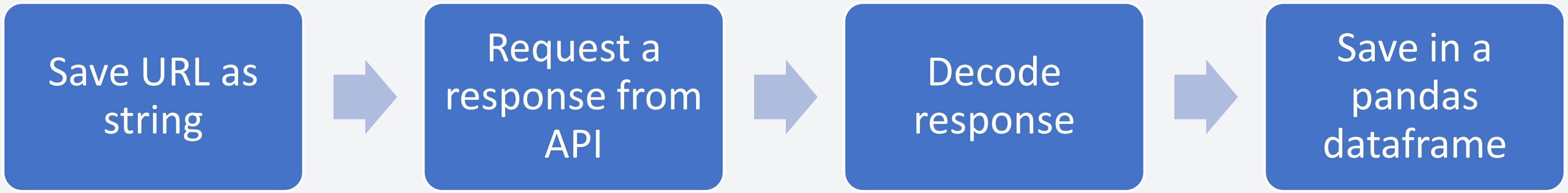
Methodology

Executive Summary

- Data collection methodology:
  - Request data from API
- Perform data wrangling
  - Clean and filter data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Find best hyperparameter
  - Evaluate using resulting accuracy score

Data Collection

Data Collection – SpaceX API

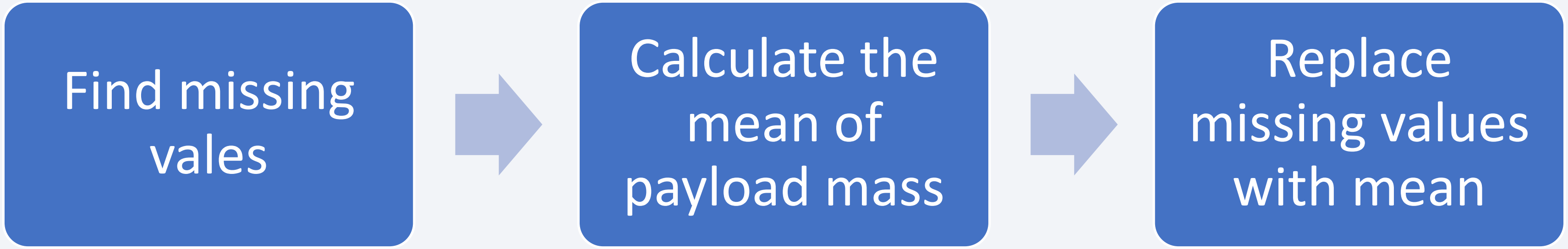


- https://github.com/pickel-bit/CapstoneAssignment/blob/dbfa0ecd3c44386e9d8efb8ef7d5c7230d6b7d77/jupyter-labs-spacex-data-collection-api%20(1).ipynb

Data Collection - Scraping

- https://github.com/pickel-bit/CapstoneAssignment/blob/dbfa0ecd3c44386e9d8efb8ef7d5c7230d6b7d77/jupyter-labs-webscraping.ipynb

Data Wrangling



- https://github.com/pickel-bit/CapstoneAssignment/blob/dbfa0ecd3c44386e9d8efb8ef7d5c7230d6b7d77/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Plot relationships between:
  - Flight number and launch sites -> success rates highly used launch sites
  - Payload and launch sites -> one of the sites does not launch high payload flights
  - Success rate and orbit type -> some orbits have a near 100% success rate
  - Flight number and orbit type -> in some orbits, flight number does not matter
  - Payload and orbits type -> some orbits cope well with higher payload
  - Launch success yearly trend -> from 2013 to 2020 the success rate increased
- https://github.com/pickel-bit/CapstoneAssignment/blob/dbfa0ecd3c44386e9d8efb8ef7d5c7230d6b7d77/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Get unique launch sites
- Calculate the total payload mass of NASA and F9 v1.1 boosters
- Get date of first successful landing
- Get number of successes and failures
- Get boosters that carry maximum payload
- Rank count of successful landing outcomes
- https://github.com/pickel-bit/CapstoneAssignment/blob/dbfa0ecd3c44386e9d8efb8ef7d5c7230d6b7d77/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- Show geographical location of launch sites
- Display success counts of all launch sites
- Display infrastructure in proximity
- https://github.com/pickel-bit/CapstoneAssignment/blob/dbfa0ecd3c44386e9d8efb8ef7d5c7230d6b7d77/lab_jupyter_launch_site_location.ipynb

Predictive Analysis (Classification)

- Find best hyperparameter
- Calculate accuracy score
- For the following methods:
  - Support vector machines
  - Decision trees
  - Logistics regression
  - K-nearest neighbour
- https://github.com/pickel-bit/CapstoneAssignment/blob/dbfa0ecd3c44386e9d8efb8ef7d5c7230d6b7d77/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

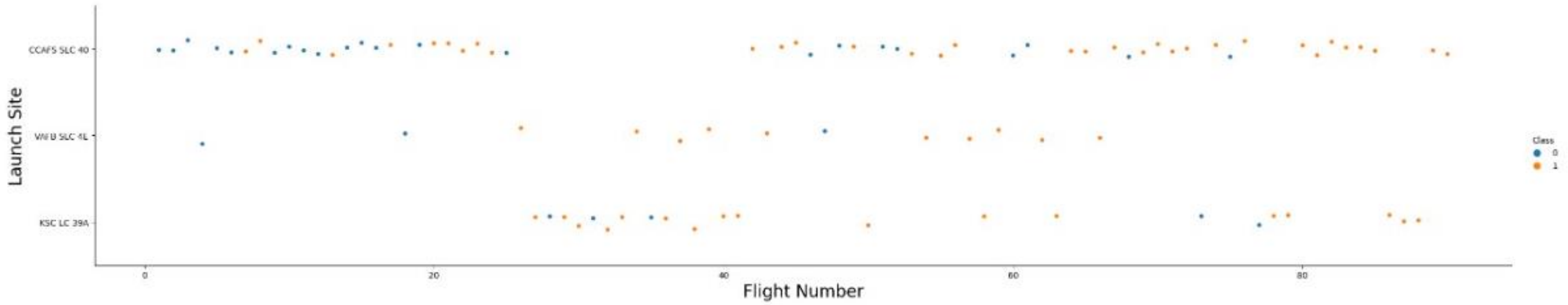
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

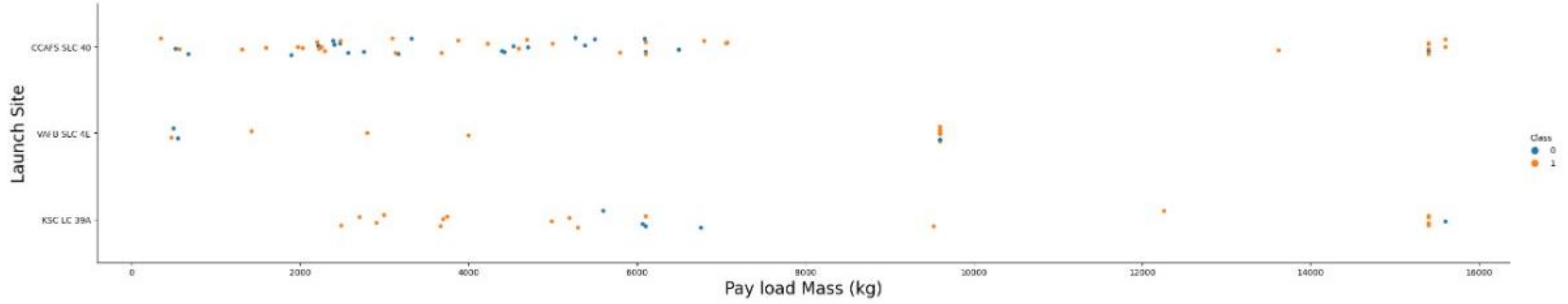
In [5]:

```
# 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()
```

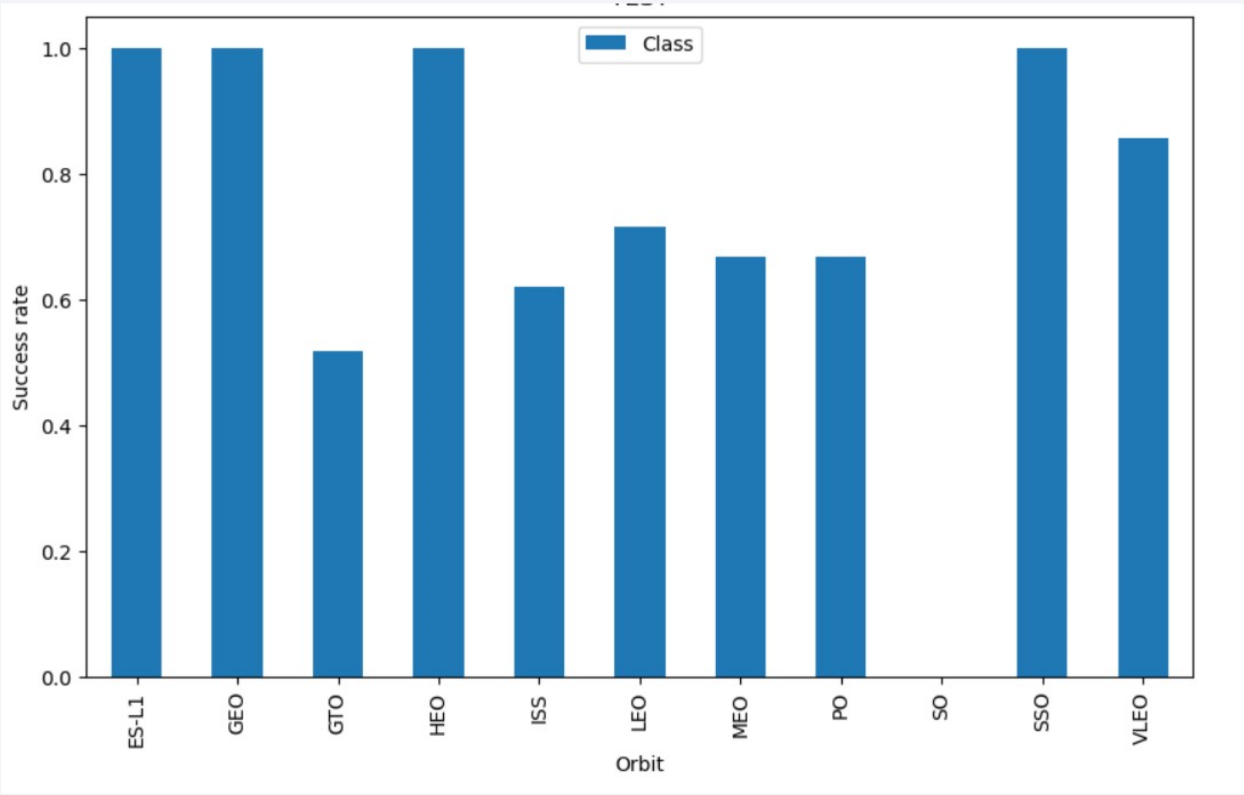


Payload vs. Launch Site

```
In [8]:  # 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 v
         sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
         plt.xlabel("Pay load Mass (kg)",fontsize=20)
         plt.ylabel("Launch Site",fontsize=20)
         plt.show()
```

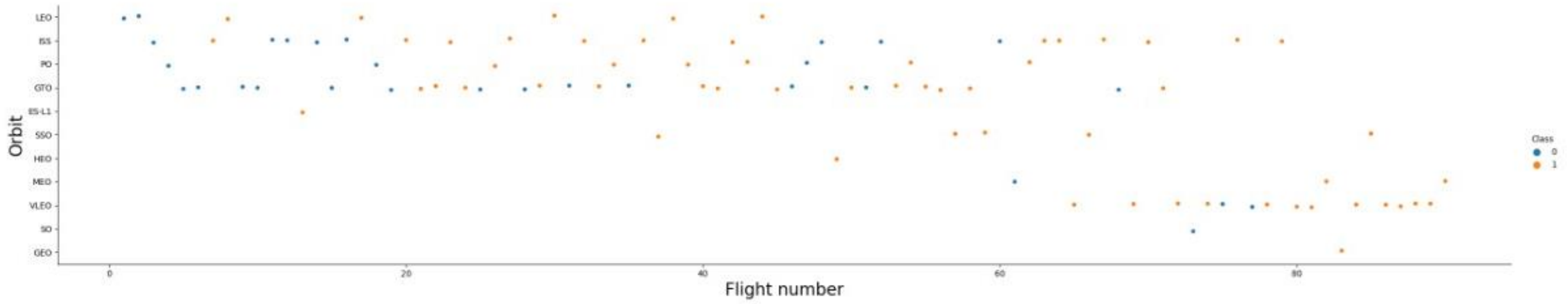


Success Rate vs. Orbit Type



Flight Number vs. Orbit Type

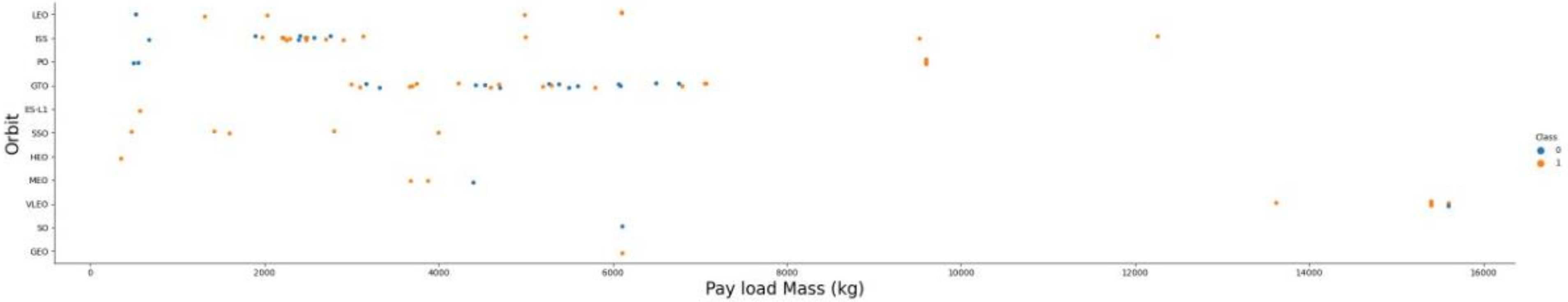
In [34]:
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Payload vs. Orbit Type

In [35]:

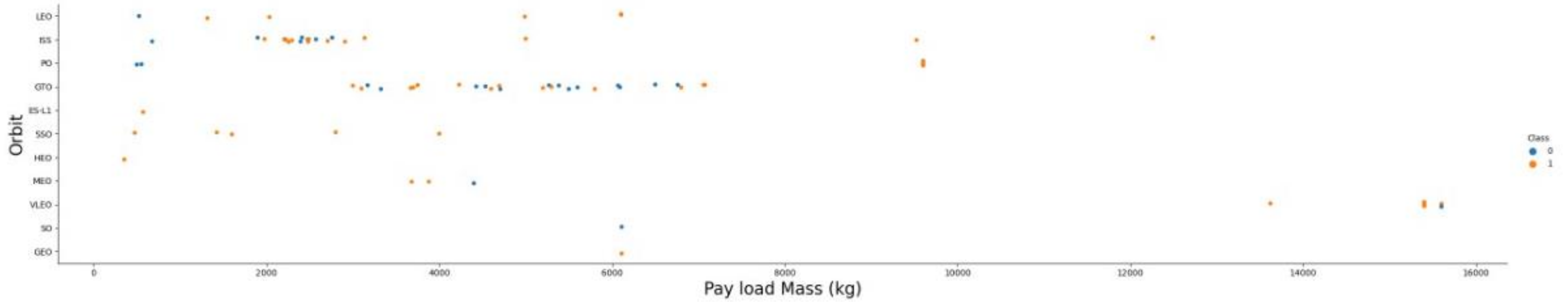
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch Success Yearly Trend

In [35]:

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



All Launch Site Names

- There are 4 unique launch sites:

```
In [22]: %sql select distinct "Launch_Site" from SPACEXTBL
 * sqlite:///my_data1.db
Done.
Out[22]:
```

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

Launch Site Names Begin with 'CCA'

```
In [24]: %sql select * from SPACEXTBL where "Launch_Site" = 'CCAFS LC-40' or "Launch_Site" = 'CCAFS SLC-40'
```

Out[24]:

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

- The total payload carried by boosters from NASA:

```
In [25]:  %sql select "Customer", sum("PAYLOAD_MASS__KG_") from SPACEXTBL group by "Customer"

 * sqlite:///my_data1.db
Done.

Out[25]:
```

Customer	sum("PAYLOAD_MASS__KG_")
ABS Eutelsat	7759
AsiaSat	8963
Bulsatcom	3669
CONAE	3000
CONAE, PlanetIQ, SpaceX	3130
Canadian Space Agency (CSA)	4200
EchoStar	5600
Es hailSat	5300
Hisdesat exactEarth SpaceX	2150
Hispasat NovaWurks	6092
Inmarsat	6070
Intelsat	6761

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1:

```
In [27]:  %sql select "Booster_Version", avg("PAYLOAD_MASS__KG_") from SPACEXTBL group by "Booster_Version"

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

Out[27]:

Booster_Version	avg("PAYLOAD_MASS__KG_")
F9 B4 B1039.2	2647.0
F9 B4 B1040.2	5384.0
F9 B4 B1041.2	9600.0
F9 B4 B1043.2	6460.0
F9 B4 B1039.1	3310.0
F9 B4 B1040.1	4990.0
F9 B4 B1041.1	9600.0
F9 B4 B1042.1	3500.0
F9 B4 B1043.1	5000.0
F9 B4 B1044	6092.0

First Successful Ground Landing Date

- First successful landing outcome on ground pad:

```
In [50]: %sql select min("Date") * from SPACEXTBL
```

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

Out[50]:

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)

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
In [54]: %sql select distinct "Booster_Version" from SPACEXTBL where "PAYLOAD_MASS__KG_" < 6000 and "PAYLOAD_MASS__KG" > 4000

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

Out[54]:

Booster_Version
F9 v1.0 B0003
F9 v1.0 B0004
F9 v1.0 B0005
F9 v1.0 B0006
F9 v1.0 B0007
F9 v1.1 B1003
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1010
F9 v1.1 B1012
F9 v1.1 B1013
F9 v1.1 B1014
F9 v1.1 B1015

Total Number of Successful and Failure Mission Outcomes

- Total number of successful mission outcomes

```
In [57]:  %sql select count(*) from SPACEXTBL where "Mission_Outcome" = 'Success'

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

Out[57]:

count(*)
98

Boosters Carried Maximum Payload

- Booster which have carried the maximum payload mass:

```
In [67]: %sql select "Booster_Version", "PAYLOAD_MASS__KG_" from SPACEXTBL where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_"

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

Out[67]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600

2015 Launch Records

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015:

In [67]:
```
%sql select "Booster_Version", "PAYLOAD_MASS__KG_" from SPACEXTBL where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_"
```

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

Out[67]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600

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:

```
In [67]: %sql select "Booster_Version", "PAYLOAD_MASS__KG_" from SPACEXTBL where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_"
```

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

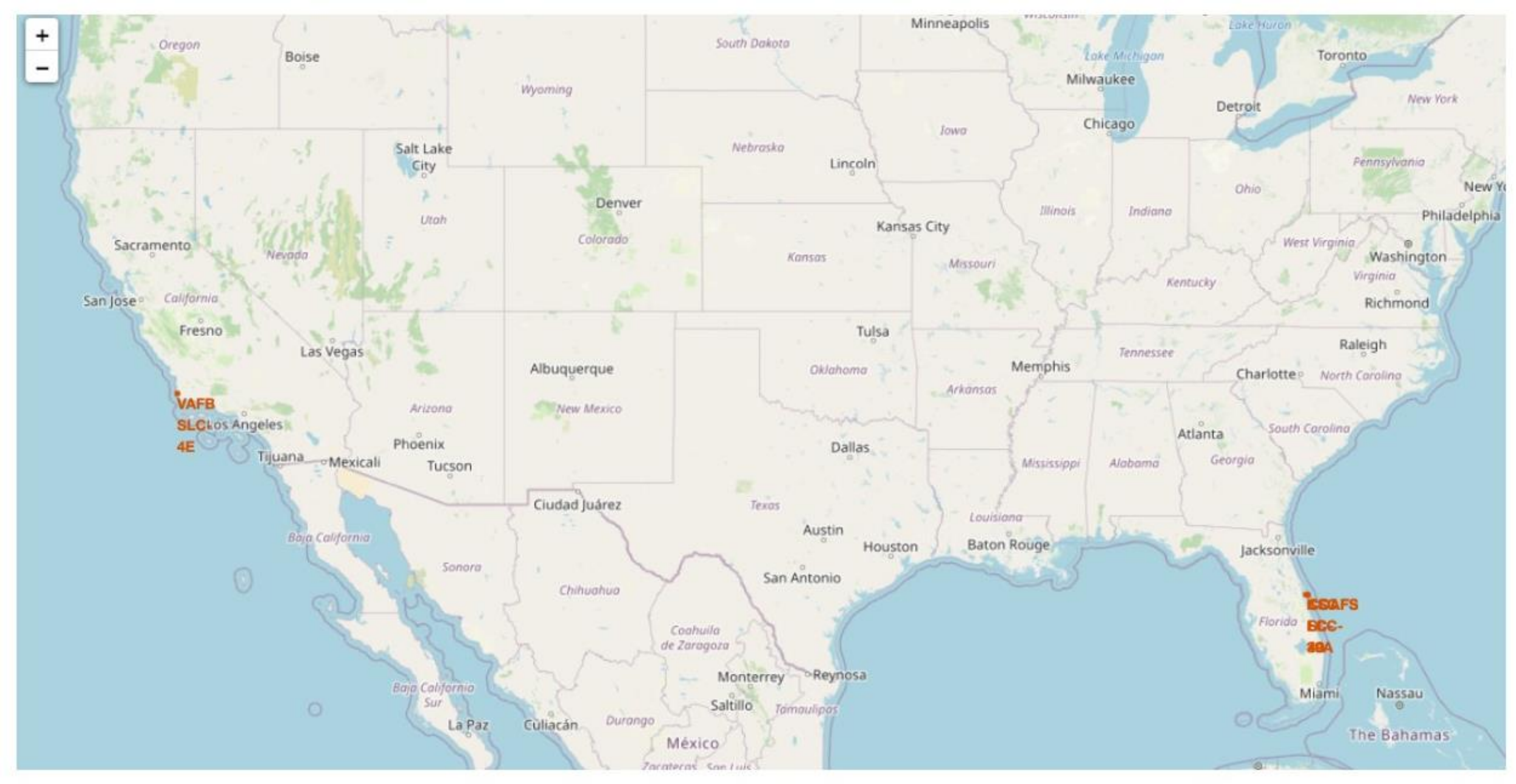
Out[67]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600

Section 3

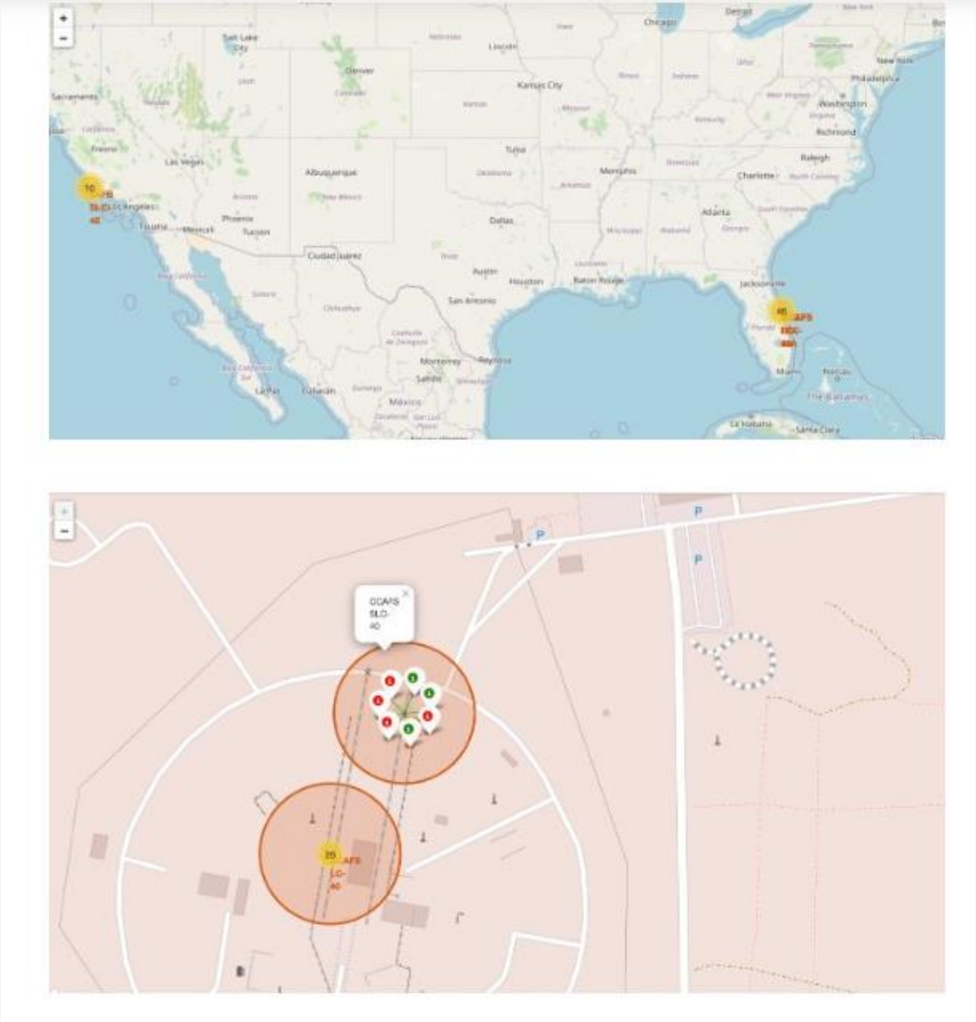
Launch Sites Proximities Analysis

Site locations



- There are launch sites in two geographical areas

Successes vs. failures



Infrastructure in proximity to launch sites

Section 5

Predictive Analysis (Classification)

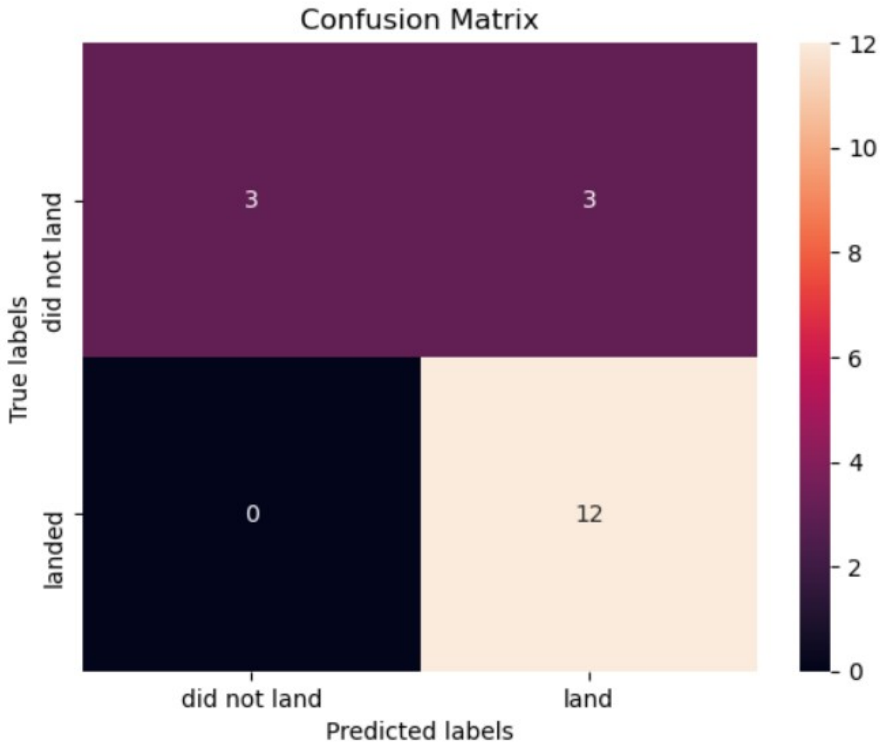
Classification Accuracy

- Tree model has the highest accuracy
- Tree: 0.888
- Logistic regression: 0.847
- Support vector machines: 0.847
- Neirest neighbour: 0.847

Confusion Matrix

- Tree model with an accuracy of 0.8333

In [40]:
```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Choose orbit and launch site carefully
- Use decision trees for prediction

Thank you!