IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Josie Barton
12/1/2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Methodology
  - Data was collected from the SpaceX REST API and using webscraping
  - Data was cleaned and analyzed
  - Data visualizations were built
  - Model creation to predict landing outcomes
- Results
  - All models performed similarly, with an accuracy score around 83%; the decision tree model slightly outperformed the others.

Introduction

- Problem Statement
  - The goal of this project is to predict if SpaceX's Falcon 9 rocket will successfully complete the first stage. If the rocket lands safely, they can reuse it. The benefit of reusing the first stage is major cost savings; a successful landing can mean the total cost if 62M, compared to 165M for competitors who do not reuse.

- Who is SpaceX?
  - SpaceX is a leader in the space exploration field. They use the newest technologies and are constantly innovating to drive down the cost of space travel.
  - It was founded by Allon Mask
  - The affordability of its space travel is heavily dependent on the stage 1 landings

Section 1

Methodology

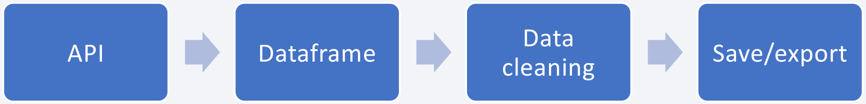
Methodology

Executive Summary

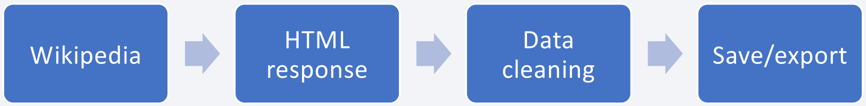
- Data collection methodology:
  - Data was collected using the SpaceX REST API and web scraping techniques
- Perform data wrangling
  - The data was processed using Pandas to clean and organize it properly
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- REST API
    - Request data from SpaceX REST API
    - Decode response and converted into a dataframe
    - Request information about the launches
    - Create a dictionary and dataframe
    - Filter dataframe and replace missing values

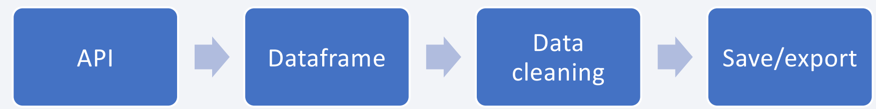


- Webscraping
    - Request data from Wikipedia
    - Create BeautifulSoup object
    - Extract data properties and format data
    - Create dictionary and dataframe



Data Collection – SpaceX API

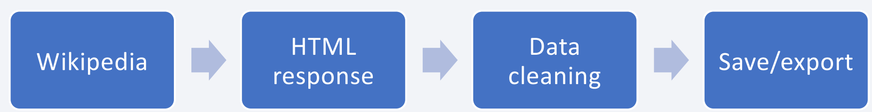
- Request data from SpaceX REST API
- Decode response and converted into a dataframe
- Request information about the launches
- Create a dictionary and dataframe
- Filter dataframe and replace missing values



- [GitHub URL]

Data Collection - Scraping

- Request data from Wikipedia
- Create BeautifulSoup object
- Extract data properties and format data
- Create dictionary and dataframe



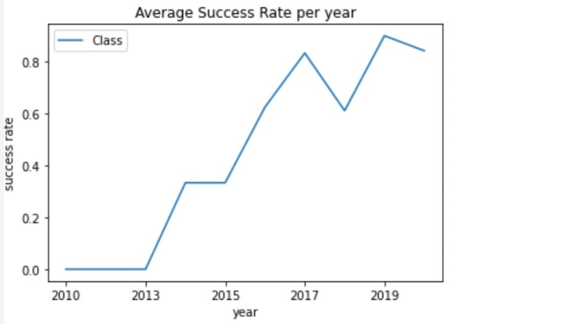
- [GitHub URL]

Data Wrangling

- Process
  - Perform EDA and determine data lables
  - Calculate necessary metrics
  - Create binary landing outcome
  - Export data to CSV
- Success rate of landings was 66.67%
- GitHub URL

EDA with Data Visualization

- MatplotLib and Seaborn were used
- Exploratory data analysis with Pandas dataframe
- Viz for relationship between flight number and launch site
- Viz for success rate of each orbit
- Viz for relationship between flight number and orbit type
- Viz for relationship between payload and orbit type
- Viz for launch success annually
- GitHub URL



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

EDA with SQL

- Download dataset
- Connect to database
- Display
  - launch sites names and beginning records
  - total payload mass
  - Average payload mass
  - Date for first successful landing outcome was achieved
  - Booster names
  - Booster versions
  - Landing outcomes
- GitHub URL

Build an Interactive Map with Folium

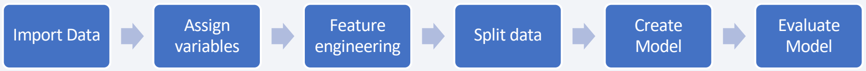
- I marked all launch sites on a map
- I marked the successes and failures for each launch and where they were on the map
- I calculated the distances between the launch site to each outcome
- GitHub URL

Build a Dashboard with Plotly Dash

- Launch site list
  - Allows selection of single or multiple launch sites
- Pie Chart
  - Displays successful and failed launches as percent of total
- Mass Range Slider
  - Playload mass range can be selected
- Scatter Chart
  - Mass vs, Success rate by Booster
- GitHub URL

Predictive Analysis (Classification)

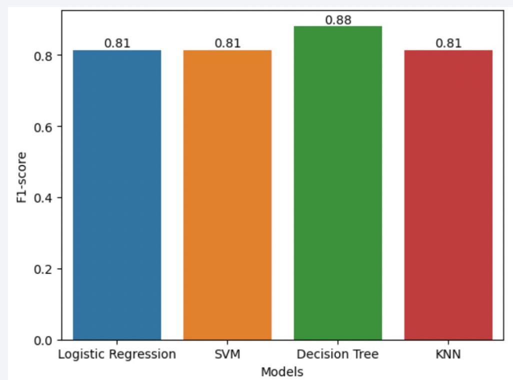
- Objectives
  - Create column for class
  - Standardize the data
  - Split into training and test data
- Process
  - Import and clean data
  - Assign feature and target variables
  - Scale features
  - Fit model
- GitHub URL



Results

- Exploratory data analysis results
  - NASA (CRS) had a total payload mass of ~46k.
  - Booster version had an avg payload mass of ~3k
  - Initial landing with positive outcome was in December 2015
  - 99 successful missions for the Falcon 9
  - Launch Successes increased over time
- Visual Results
  - Most launch sites are close to the equator and all are by the ocean
  - Launch sites are far enough away from important infrastructure to protect it and people
- Predictive analysis results
  - Decision tree was the best model for this particular dataset and our goals

	Models	F1-score
0	Logistic Regression	0.814815
1	SVM	0.814815
2	Decision Tree	0.882051
3	KNN	0.814815



Section 2

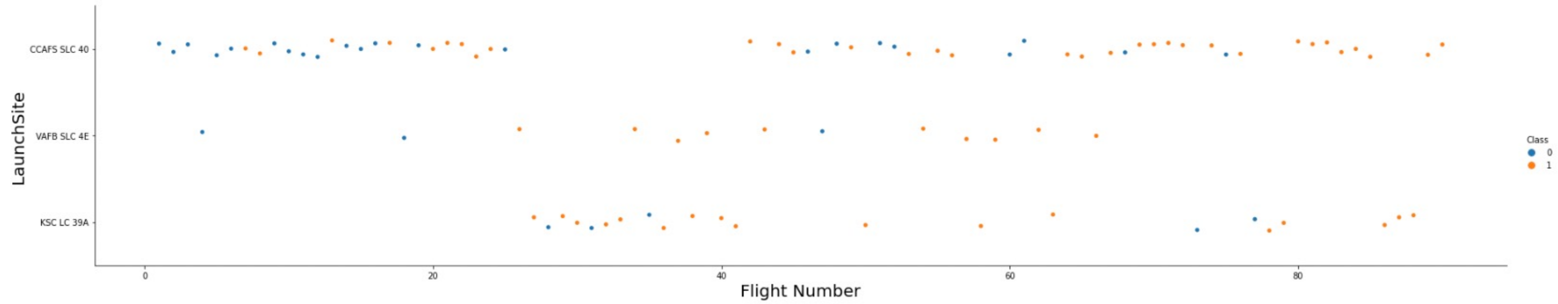
Insights drawn from EDA

Flight Number vs. Launch Site

TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

```
# 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(y='LaunchSite', x='FlightNumber', hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

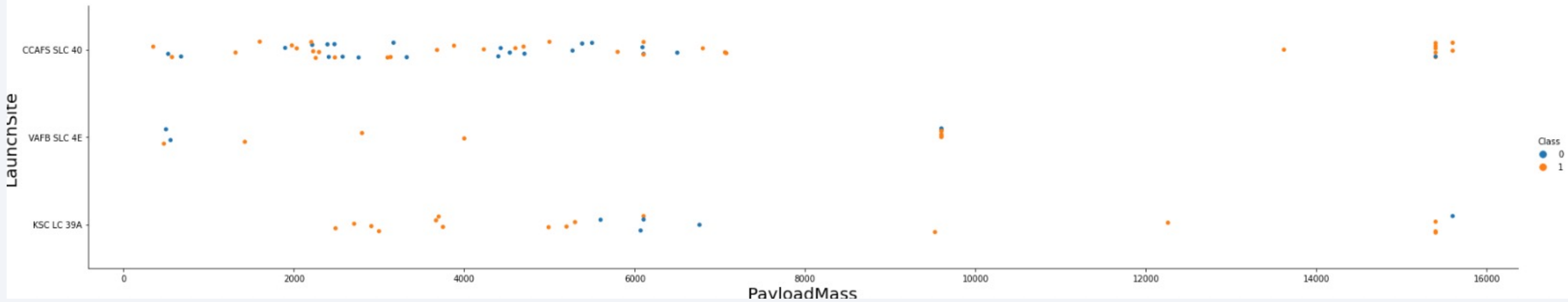


Payload vs. Launch Site

TASK 2: Visualize the relationship between Payload and Launch Site

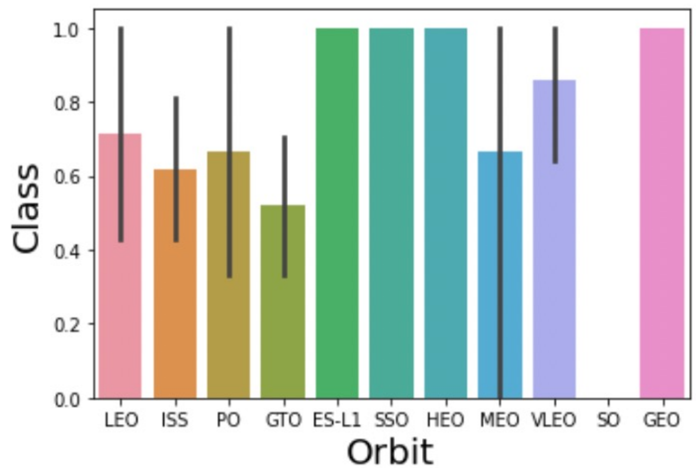
We also want to observe if there is any relationship between launch sites and their payload mass.

```
# 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(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Success Rate vs. Orbit Type

```
In [16]: sns.barplot(x='Orbit', y='Class', data=df)
         plt.xlabel('Orbit',fontsize=20)
         plt.ylabel("Class",fontsize=20)
         plt.show()
```



Flight Number vs. Orbit Type

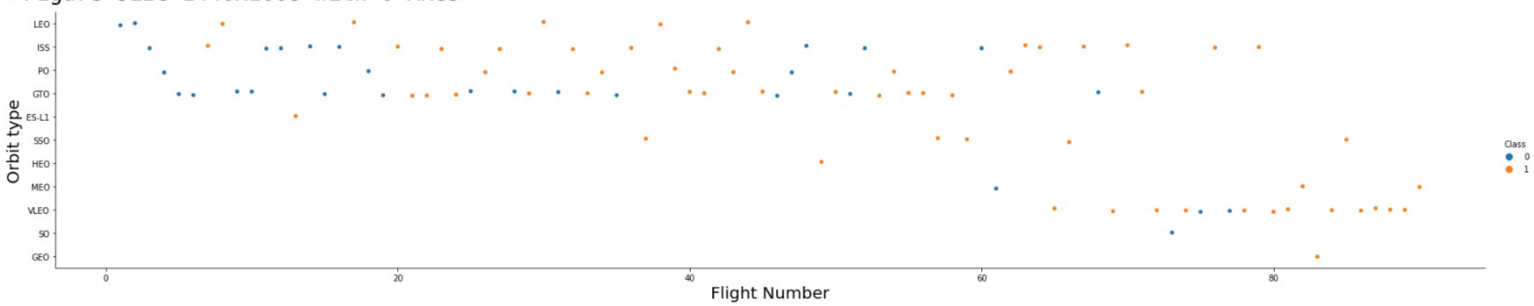
TASK 4: Visualize the relationship between FlightNumber and Orbit type

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

]:

```
# 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
plt.figure(figsize=(20,14))
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit type",fontsize=20)
plt.show()
```

<Figure size 1440x1008 with 0 Axes>

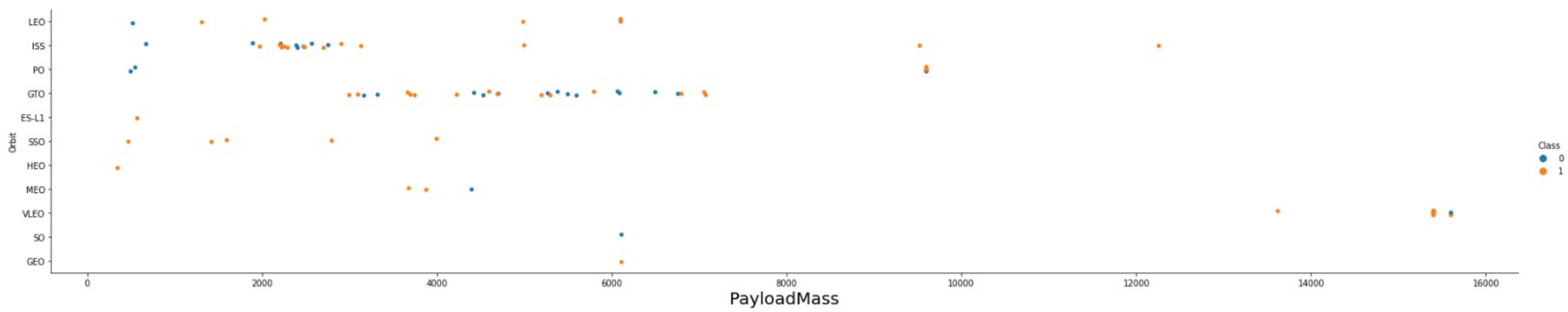


Payload vs. Orbit Type

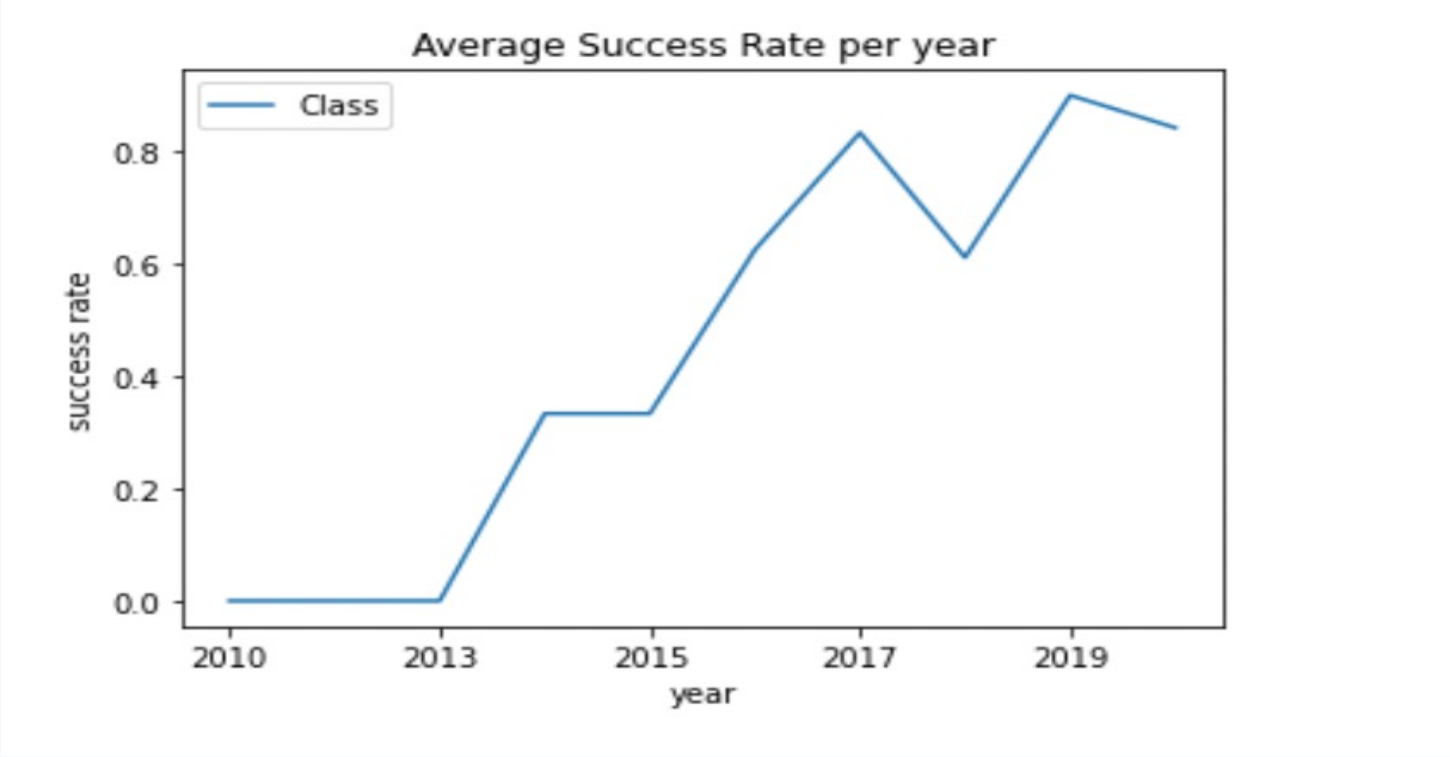
TASK 5: Visualize the relationship between Payload and Orbit type

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

```
# 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("Orbit type",fontsize=20)
plt.xlabel('PayloadMass',fontsize=20)
plt.show()
```



Launch Success Yearly Trend



All Launch Site Names

Launch Site Names

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

Landing Outcome Cont.

```
[30]: %sql ibm_db_sa://yyy33800:dwNKg8J3L0IBd6CP@1bbf73c5
      %sql SELECT Unique(LAUNCH_SITE) FROM SPACEXTBL;

 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4bb0-85b9
   sqlite:///my_data1.db
Done.
```

[30]:

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

Records with Launch Site Starting with CCA

- Displaying 5 records below

```
%sql SELECT * \
    FROM SPACEXTBL \
    WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;

 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/BLUDB
   sqlite:///my_data1.db
Done.
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Launch Site Names Begin with 'CCA'

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

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

```
%%sql
select *
from SPACEXTBL
where LAUNCH_SITE like "CCA%"
limit 5;
```

Total Payload Mass

Task 3

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

```
%%sql
select Customer, sum(PAYLOAD_MASS__KG_) as Total_NASA_CRS_mass
from SPACEXTBL
where Customer = "NASA (CRS)";
```

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

Customer	Total_NASA_CRS_mass
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

Task 4

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

```
]:  %%sql
    select Booster_Version, avg(PAYLOAD_MASS__KG_) as avg_Booster_versionF9_v1_1
    from SPACEXTBL
    where Booster_Version = "F9 v1.1";
```

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

]:

Booster_Version	avg_Booster_versionF9_v1_1
F9 v1.1	2928.4

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
%%sql
select  Mission_Outcome, min(Date) as Date_First_Succ_Land
from SPACEXTBL
where Landing_Outcome ='Success (ground pad)';
```

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

Mission_Outcome	Date_First_Succ_Land
Success	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

```
%%sql
select Booster_Version,Landing_Outcome, PAYLOAD_MASS__KG_
from SPACEXTBL
where (PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000)
      and Landing_Outcome = 'Success (drone ship)';
```

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

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%%sql
select Mission_Outcome, count(Mission_Outcome) as "Total (Success or failure)"
from SPACEXTBL
GROUP BY MISSION_OUTCOME;
```

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

Mission_Outcome	Total (Success or failure)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

```
%%sql
select Booster_Version,Landing_Outcome, PAYLOAD_MASS__KG_
from SPACEXTBL
where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_)
                            from SPACEXTBL);
```

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

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Success	15600
F9 B5 B1049.4	Success	15600
F9 B5 B1051.3	Success	15600
F9 B5 B1056.4	Failure	15600
F9 B5 B1048.5	Failure	15600
F9 B5 B1051.4	Success	15600
F9 B5 B1049.5	Success	15600
F9 B5 B1060.2	Success	15600
F9 B5 B1058.3	Success	15600
F9 B5 B1051.6	Success	15600
F9 B5 B1060.3	Success	15600
F9 B5 B1049.7	Success	15600

2015 Launch Records

Task 9

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, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%%sql
SELECT  Date, Booster_Version, Launch_Site, Landing_Outcome
FROM SPACEXTBL
where Landing_Outcome= 'Failure (drone ship)' and Date <= "2015-12-31";
```

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

Date	Booster_Version	Launch_Site	Landing_Outcome
2015-10-01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

Task 10

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.

```
%%sql
select Landing_Outcome, count(Landing_Outcome) as "Total Count"
from SPACEXTBL
where Landing_Outcome = "Failure (drone ship)" or Landing_Outcome = "Success (ground pad)" and
Date between "2010-06-04" and "2017-03-20"
GROUP BY Landing_Outcome
order by Landing_Outcome desc;
```

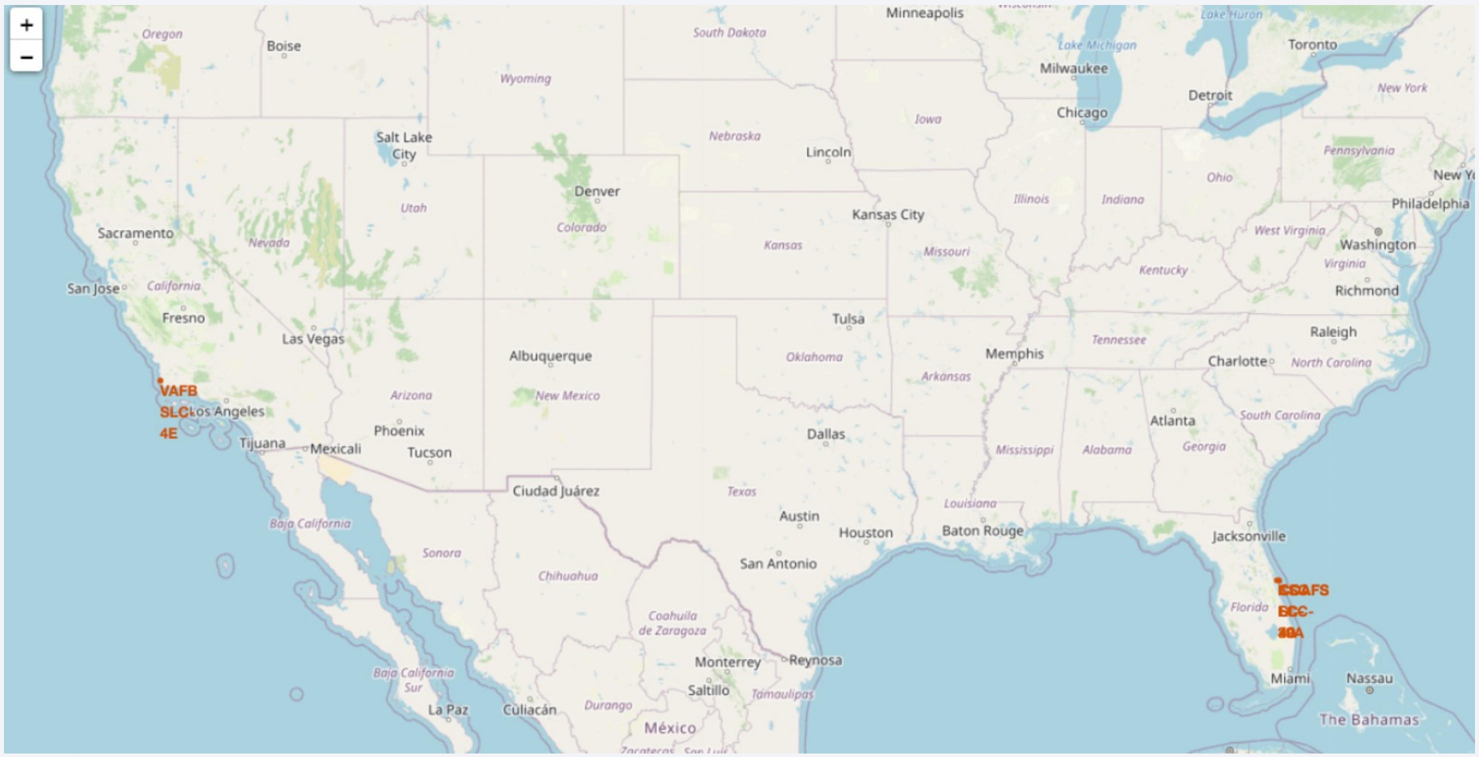
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Total Count
Success (ground pad)	5
Failure (drone ship)	5

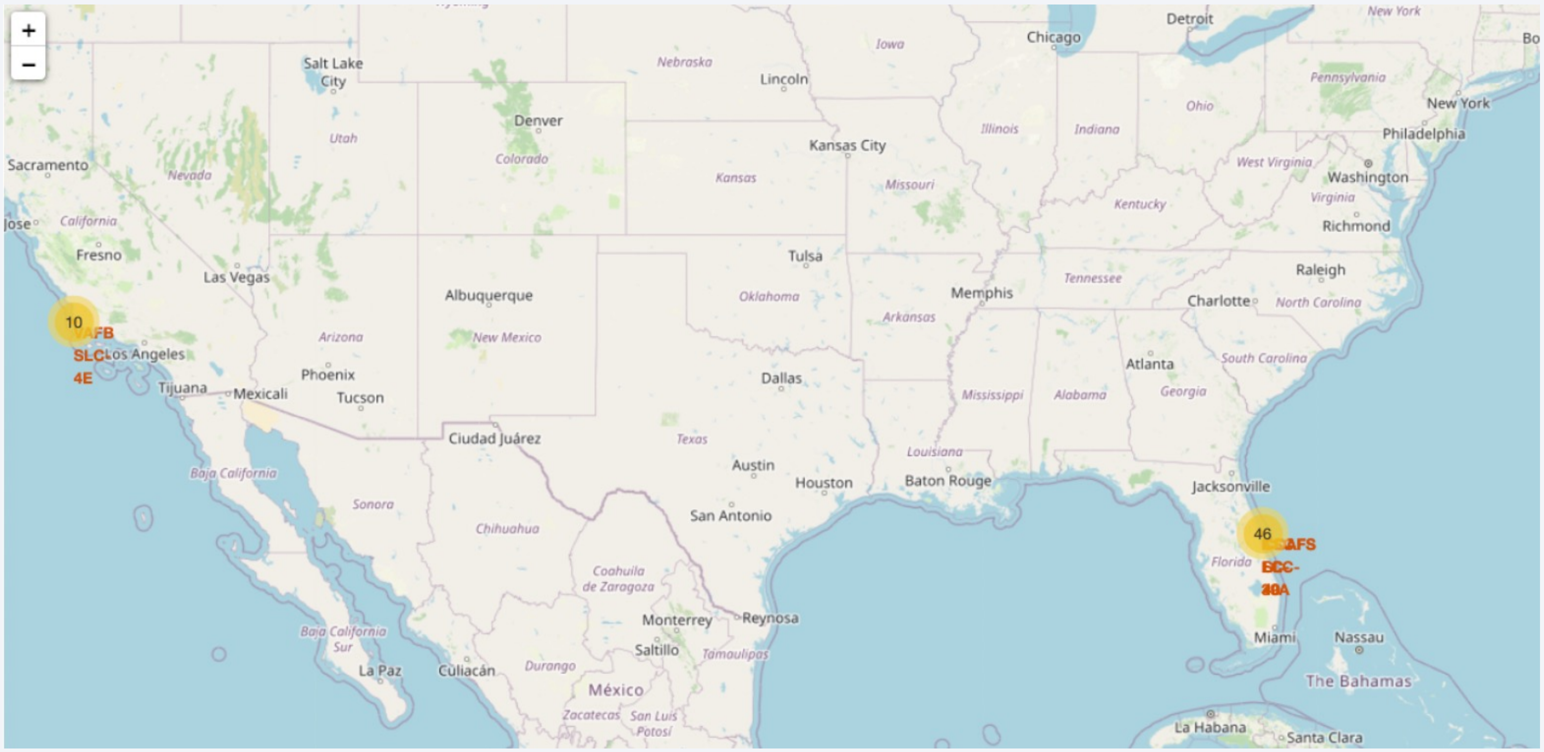
Section 3

Launch Sites Proximities Analysis

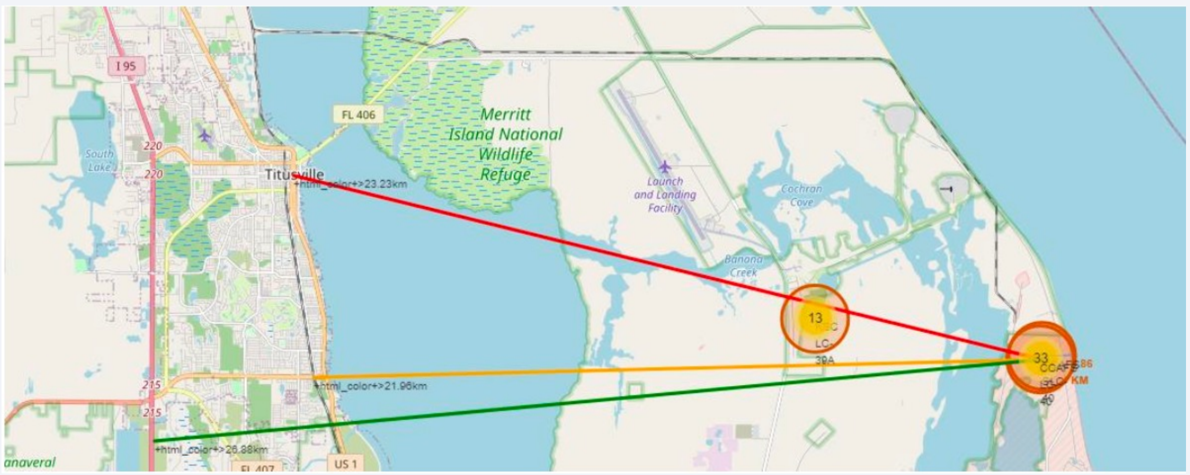
Identify Launch Sites



Mark Success/Failures on the Map



Calculate distances between launch site and proximities



Section 4

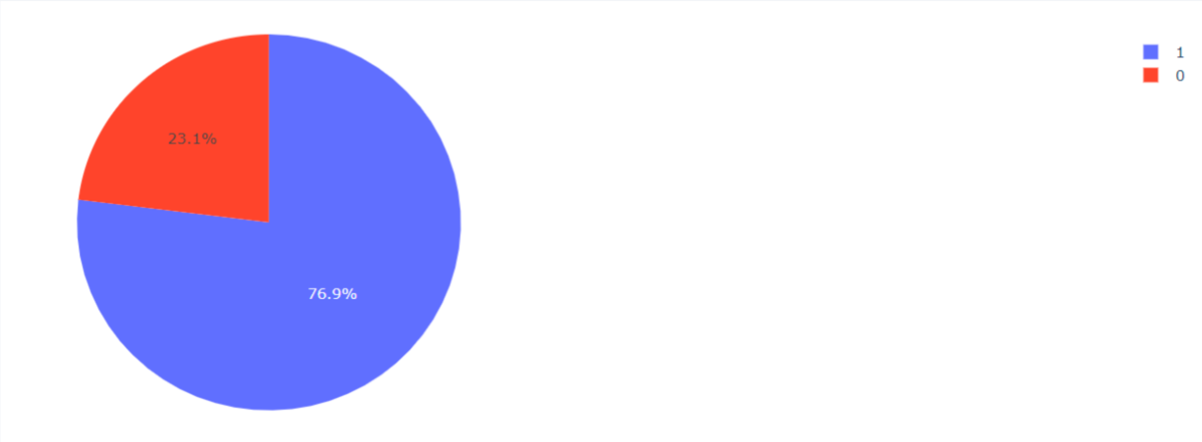
Build a Dashboard with Plotly Dash

Share of Launch Sites

- KSC LC-39A makes up 41.7%

Launch Success of KSC LC-39A

- Highest Success rate of all sites
- 10 successful launches, 3 failures



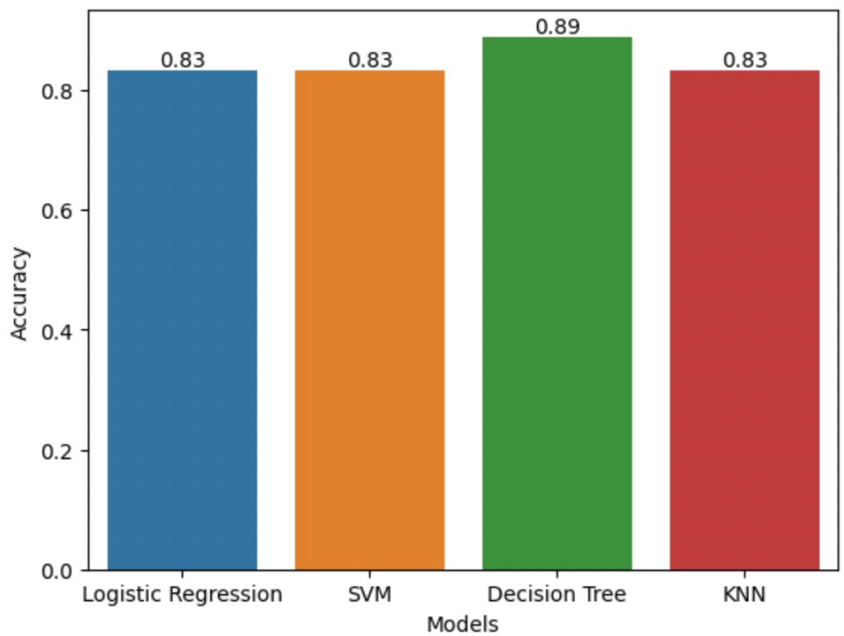
Payload Mass and Success

- Payloads between 2k kg and 5k kg have highest success rates

Section 5

Predictive Analysis (Classification)

Classification Accuracy



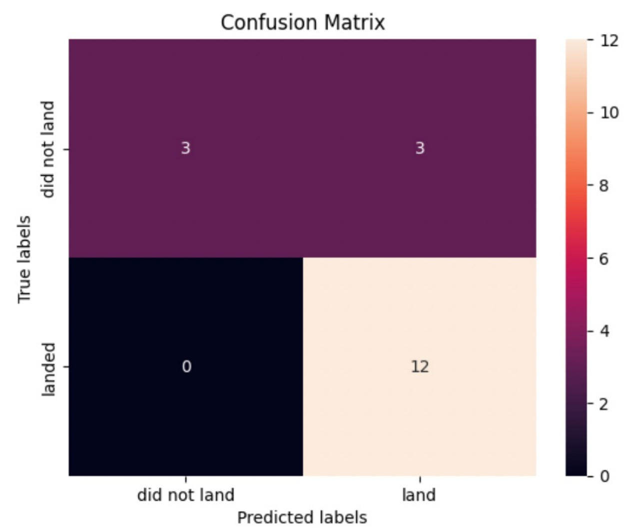
Answer: Decision Tree Classifier score the best of all the models with an accuracy of 88.88% on the unseen test dataset.

Confusion Matrix

- All confusion matrices were identical

```
Accuracy: 0.8333
Jaccard score: 0.8000
Precision: 0.8000
Recall: 1.0000
Avg F1-score: 0.8148
```

```
plot_confusion_matrix(Y_test,knnyhat)
```



Conclusions

- All models performed similarly, with the decision tree model slightly ahead of the rest
- Launch success increases over time
- ES-L1, GEO, HEO, SSO launch sites have 100% success rates
- The higher the payload mass, the higher the success rate
- Most successful launch site was KSC LC-39A
- Most of the launch sites are near the equator and all are near the coast


Thank you!