Winning Space Race with Data Science

<Vadim Savenkov>
<30-Jul-2022>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - EDA and Data Visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard using Plotly Dash
  - Predictive Analysis(Classification)
- Summary of all results
  - EDA Results
  - Interactive Analytics
  - Predictions

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings because SpaceX can reuse the first stage.
- Problems you want to find answers
  - The objective is to predict if the first stage of the SpaceX Falcon 9 will land successfully

Section 1

Methodology

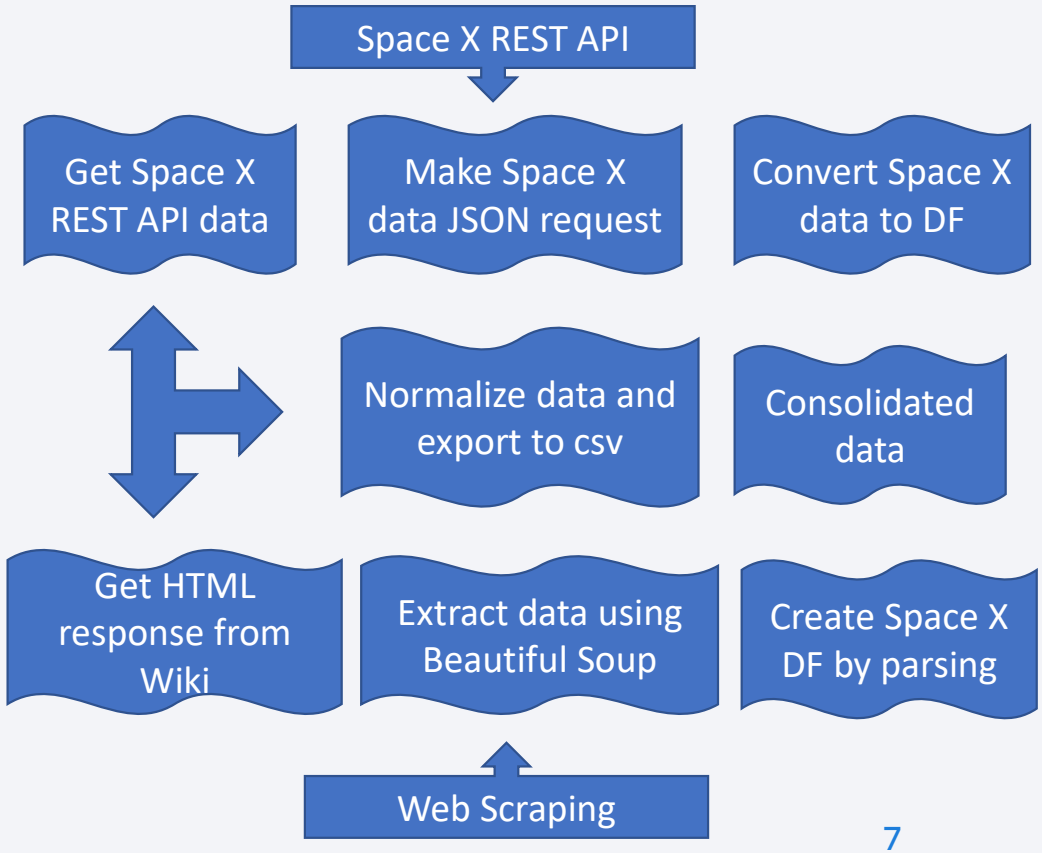
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping From Wikipedia
- Perform data wrangling
  - One Hot Encoding data field for Machine Learning and data cleaning of null values and irrelevant columns
- 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 the best Hyperparameter for SVM, Classification Trees, and Logistic Regression

Data Collection Flow Chart

- SpaceX launch data that is gathered from the SpaceX REST API.
- This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/.
- Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using Beautiful Soup.



Data Collection – SpaceX API

- Get requests to the SpaceX API and to do some basic data wrangling and formatting:
  - Request to the SpaceX API
  - Clean the requested data
- Check the URL of the completed SpaceX API calls notebook as an external reference and for peer-review purposes:

https://github.com/vadimsavenkov/Applied_DS_Capstone/blob/main/.ipynb_checkpoints/spacex-data-collection-api-checkpoint.ipynb

- Request rocket launch data from SpaceX API with the following URL:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```

- To make the requested JSON results more consistent, use the static response object:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.
```

- Use the json_normalize method to convert the JSON result into a data frame:

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

- Get the API again to get information about the launches using the following IDs: rocket, payloads, launchpad, and cores.
- Use Global Variables

```
getBoosterVersion(data)   getLaunchSite(data)
getPayloadData(data)      getCoreData(data)
```

- Construct a new dataset and combine columns into the dictionary:

```
launch_dict = {'FlightNumber': list(data['flight_number'])
```

- Create a data frame from a dictionary:

```
df = pd.DataFrame(data=launch_dict)
```

- Filter data frame to only include Falcon 9 launches:

```
data_falcon9 = df.loc[df['BoosterVersion']!='Falcon 1']
```

- Clean data frame and export to csv:

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection – Web Scraping

- Web scrap Falcon 9 launch records with Beautiful Soup:
  - Extract a Falcon 9 launch records HTML table from Wikipedia
  - Parse the table and convert it into a Pandas data frame
- Check the URL of the completed web scraping notebook as an external reference and for peer-review purposes:

https://github.com/vadimsavenkov/Applied_DS_Capstone/blob/main/.ipynb_checkpoints/webscraping-checkpoint.ipynb

- Perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_

page = requests.get(static_url)
page.status_code
```

- Create a Beautiful Soup object from the HTML response

```
soup = BeautifulSoup(page.text, 'html.parser')
soup.title
```

- Extract all column/variable names from the HTML table header

```
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
print(first_launch_table)
```

- Create a data frame by parsing the launch HTML tables and filling in the parsed launch record values into the dictionary

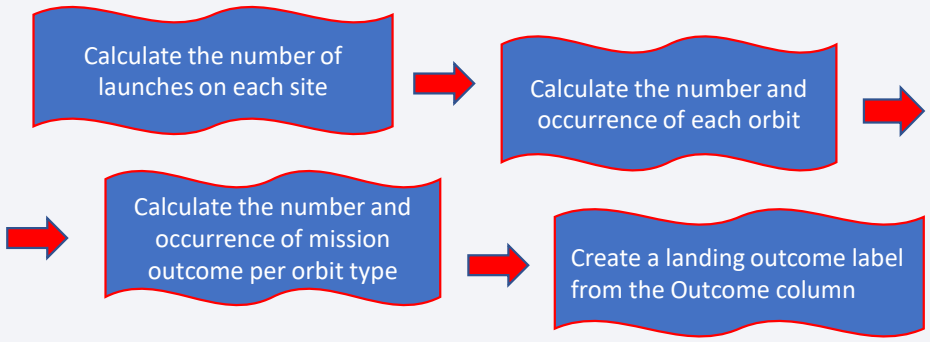
```
launch_dict= dict.fromkeys(column_names)
```

- Create a data frame and export it to CSV

```
df=pd.DataFrame(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

- Perform exploratory Data Analysis and determine Training Labels
  - Exploratory Data Analysis
  - Determine Training Labels
- Check the URL of the completed data wrangling notebook as an external reference and for peer-review purposes:
- *https://github.com/vadimsavenkov/Applied_DS_Capstone/blob/main/.ipynb_checkpoints/spacex-data-wrangling-checkpoint.ipynb*



- Load Space X dataset

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-
```

- Use the method value_counts for column LaunchSite to determine the number of launches on each site

```
df['LaunchSite'].value_counts()
```

- Use the method value_counts to determine the number and occurrence of each orbit for the column Orbit

```
df['Orbit'].value_counts()
```

- Use the method value_counts on the column Outcome to determine the number of landing_outcomes

```
landing_outcomes = df['Outcome'].value_counts()
```

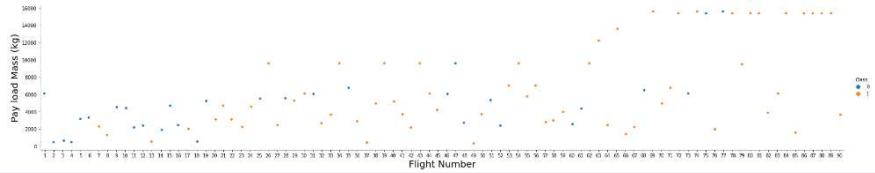
- Using the Outcome, create a list where the element is zero if the corresponding row in Outcome is in the set bad_outcome; otherwise, it's one. Then assign it to the variable landing_class and export to csv

```
landing_class = df['Outcome'].replace({'False Ocean': 0, 'False ASDS': 0,
'None None': 0, 'None ASDS': 0, 'False RTLS': 0, 'True ASDS': 1,
'True RTLS': 1, 'True Ocean': 1}, inplace = True)
df['Outcome'] = df['Outcome'].astype(int)
```

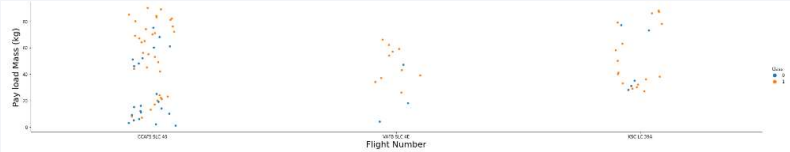
```
df.to_csv("dataset_part_2.csv", index=False)
```

EDA with Data Visualization

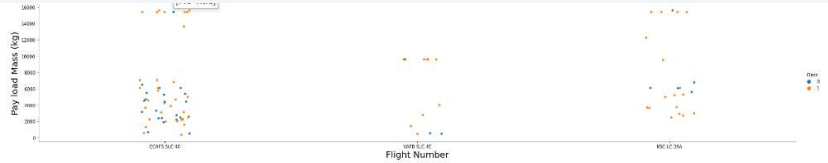
Pay Load Mass vs. Flight Number



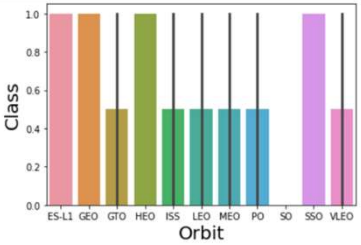
Relationship between Flight Number and Launch Site



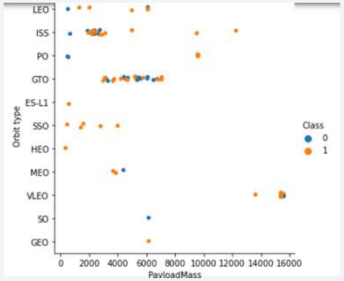
Relationship between Payload and Launch Site



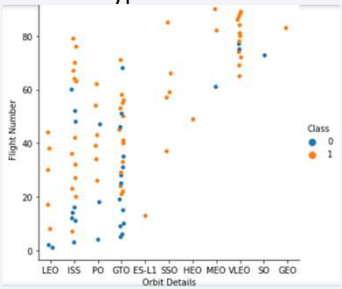
Bar chart for the success rate of each orbit



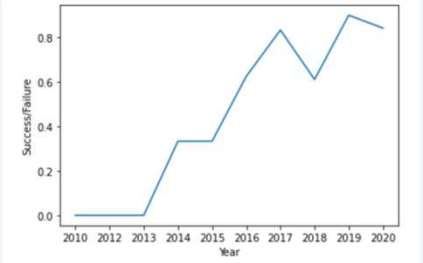
Relationship between Payload and Orbit type



Relationship between Flight Number and Orbit type



Get the average launch success trend



- Check the URL of the completed data visualization notebook as an external reference and for peer-review purposes

https://github.com/vadimsavenkov/Applied_DS_Capstone/blob/main/.ipynb_checkpoints/Exploratory%20Data%20Analysis%20Using%20Matplotlib-checkpoint.ipynb

EDA with SQL

Performed SQL queries:

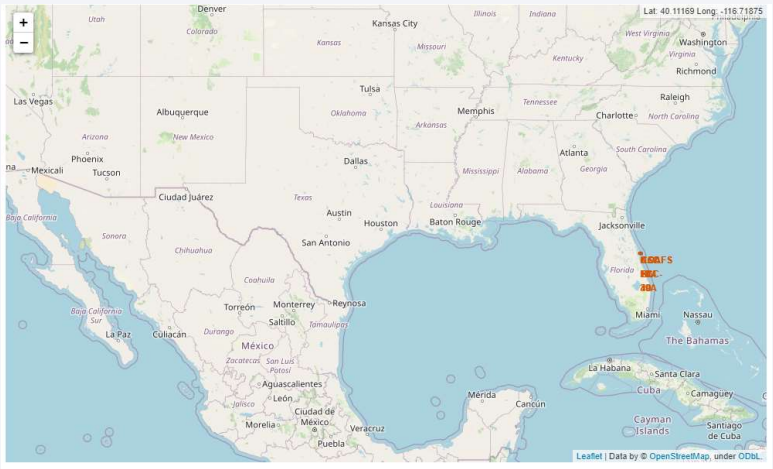
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in the ground pad was achieved
- 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 total number of successful and failed mission outcomes
- 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, land launch_site for the months in the year 2015
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

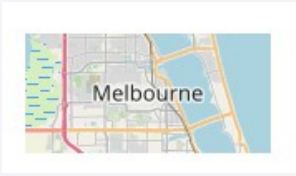
Check the URL of the completed EDA with SQL notebook as an external reference and for peer-review purposes:

https://github.com/vadimsavenkov/Applied_DS_Capstone/blob/main/.ipynb_checkpoints/eda-sqllite-checkpoint.ipynb

Build an Interactive Map with Folium

- Markers, circles, lines, etc. were added to a folium map to analyze geographical patterns about launch sites and check the proximity to various locations



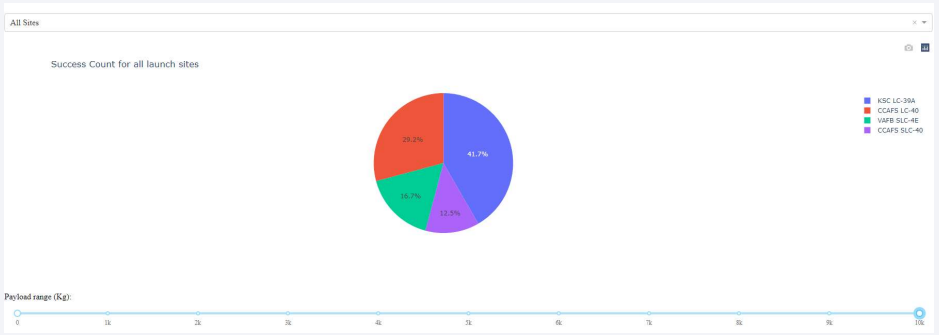


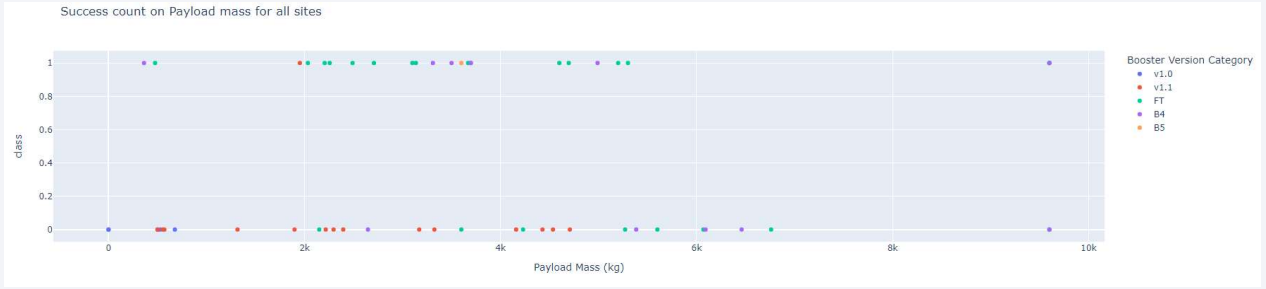
Check the URL of the completed interactive map with Folium notebook as an external reference and for peer-review purposes:

https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/d00ac88b-484b-4c72-b999-64fc3d1b7d5c/view?access_token=a3c8bba52db1e2492850c95455973df3339af6c7360e242dfbc7ca190c03fe4a

Build a Dashboard with Plotly Dash

- A success count interactive dashboard using records for 4 SpaceX launch sites was created
  - ➢ KSC LC 39A had highest number of successful among other sites

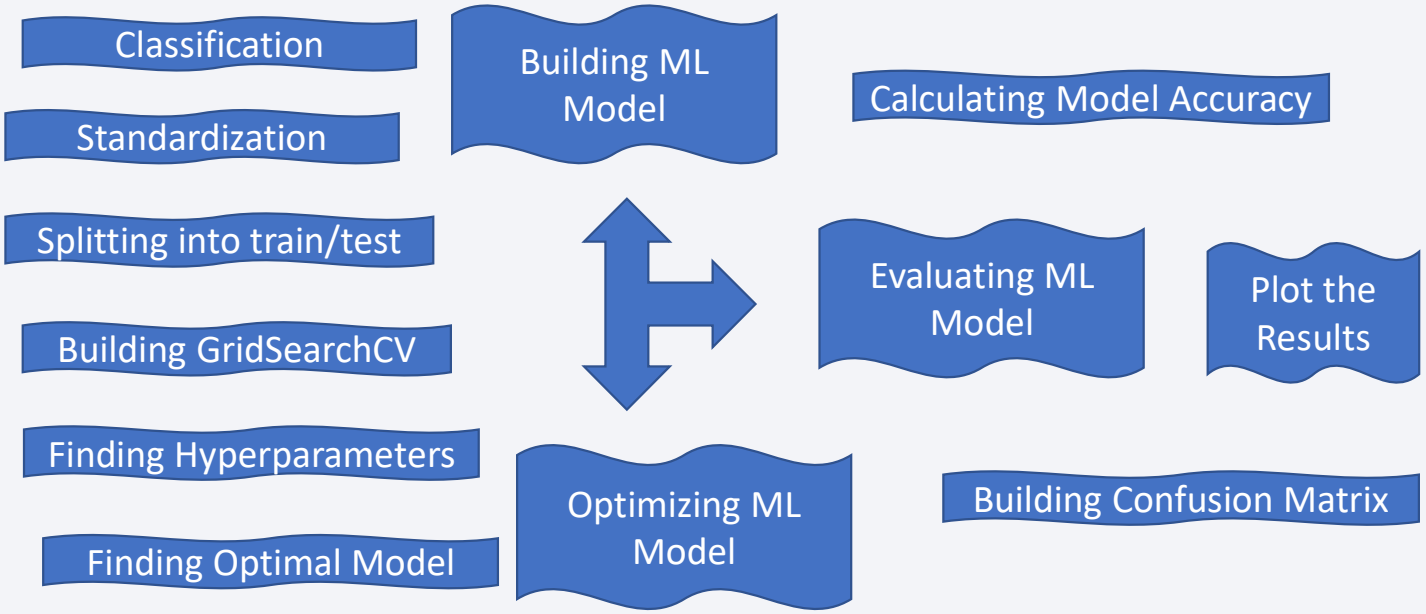




- Check the URL of the completed interactive dashboard with Plotly Dash notebook as an external reference and for peer-review purposes:

https://github.com/vadimsavenkov/Applied_DS_Capstone/blob/main/.ipynb_checkpoints/spacex_DB_app-checkpoint.ipynb

Predictive Analysis (Classification)

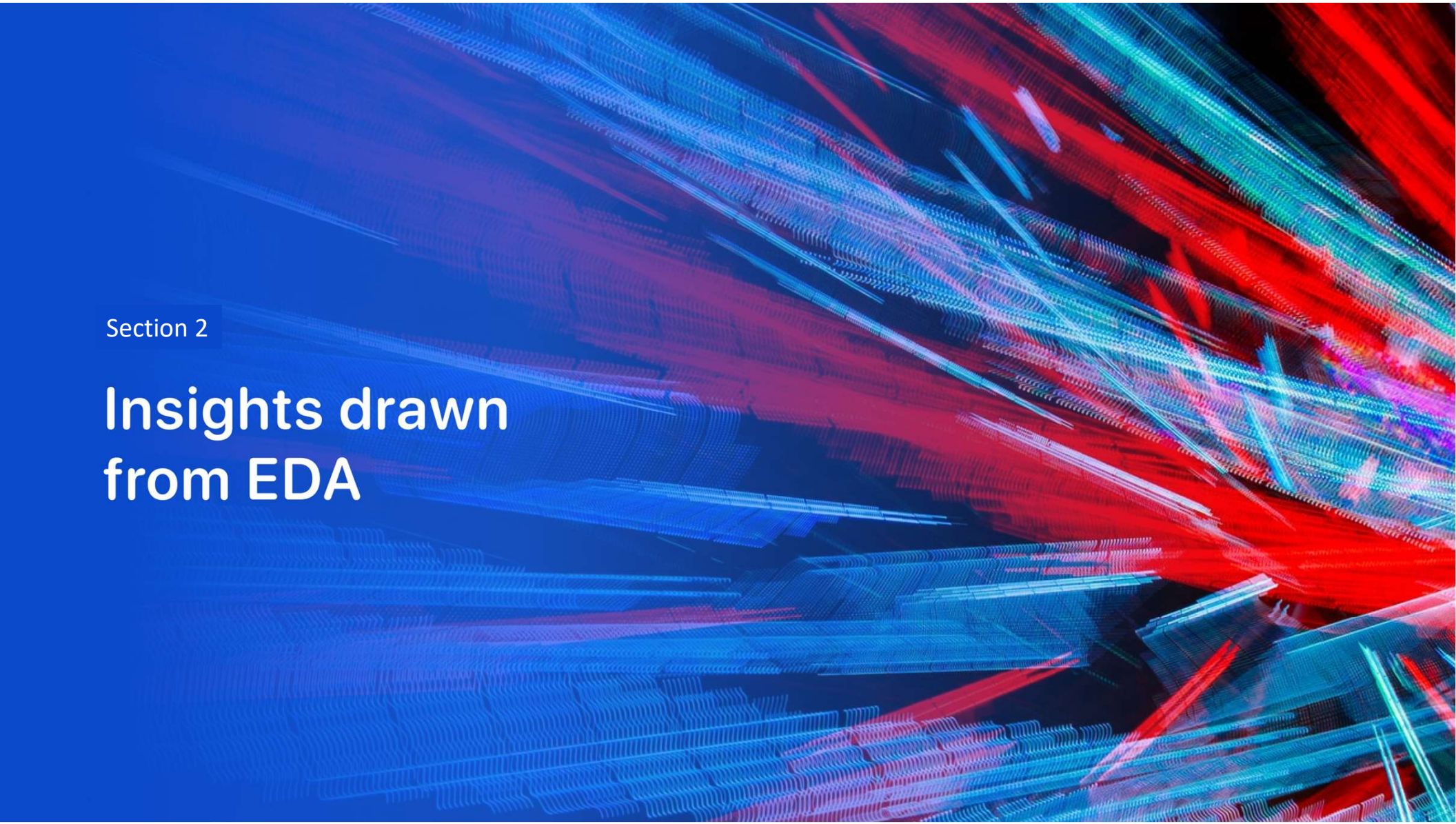


- Check the URL of the completed predictive analysis notebook as an external reference and for peer-review purposes:

https://github.com/vadimsavenkov/Applied_DS_Capstone/blob/main/.ipynb_checkpoints/SpaceX_Machine%20Learning%20Prediction-checkpoint.ipynb

Results

- Decision Tree model performed the best for the training data set with a score of 0.87.
- Low weighted payloads perform better than heavier payloads.
- The success rates for SpaceX launches shows increase with time and it will continue to grow.
- KSC LC 39A launch site had the highest success count among all the other sites.
- Orbit GEO, HEO, SSO, ES L1 has the best Success Rate.

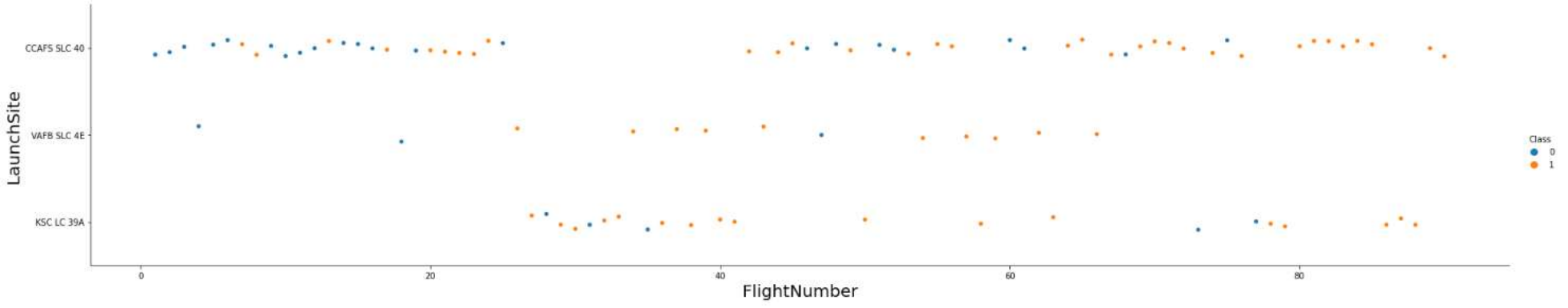


Section 2

Insights drawn from EDA

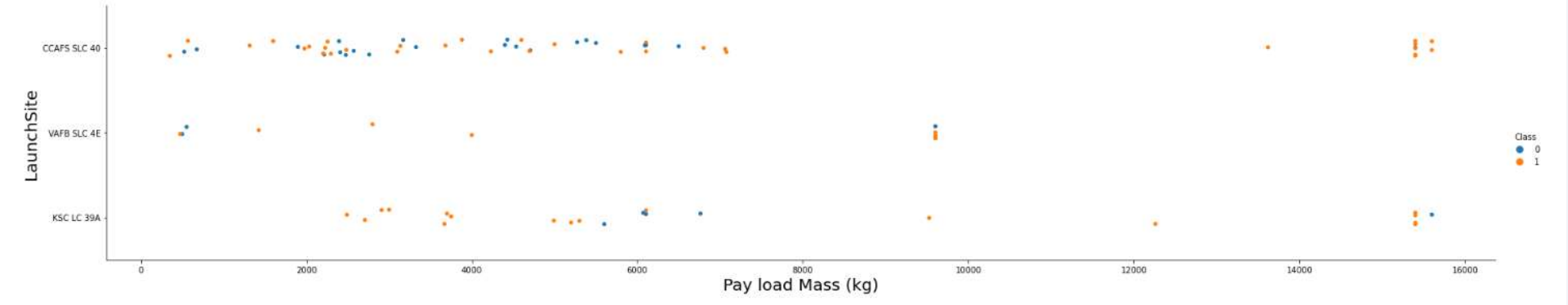
Flight Number vs. Launch Site

- ***The success rate is increasing with increasing the number of flights. The highest number of flights has been performed at CCAFS SLC 40.***



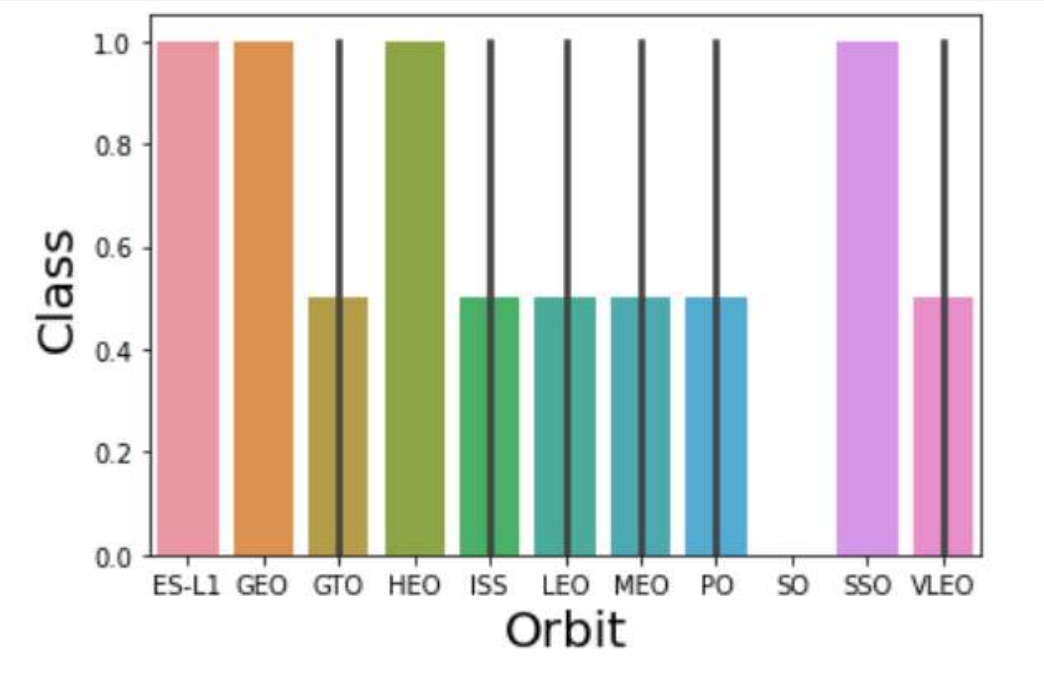
Payload vs. Launch Site

- ***The success rate is increasing with increased payload mass. The highest number of flights has been performed at CCAFS SLC 40.***



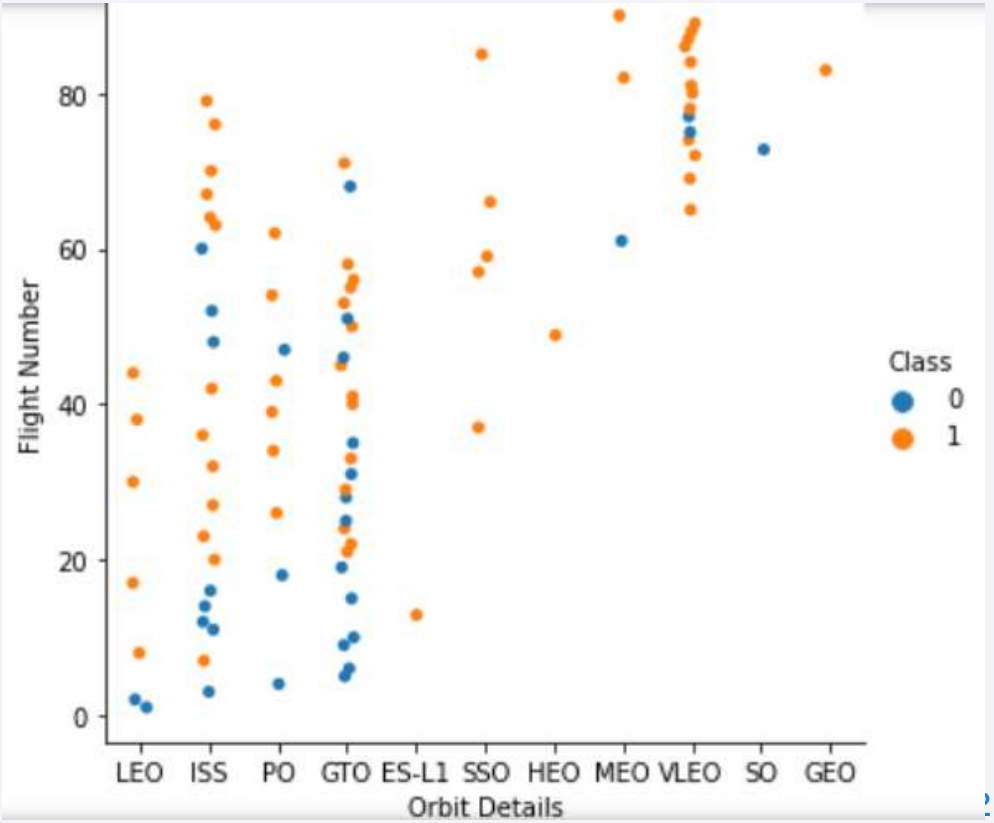
Success Rate vs. Orbit Type

- ***Success rate for ES-L1, GEO, HEO, and SSO orbit types are the highest. SO has never had any successful launches.***



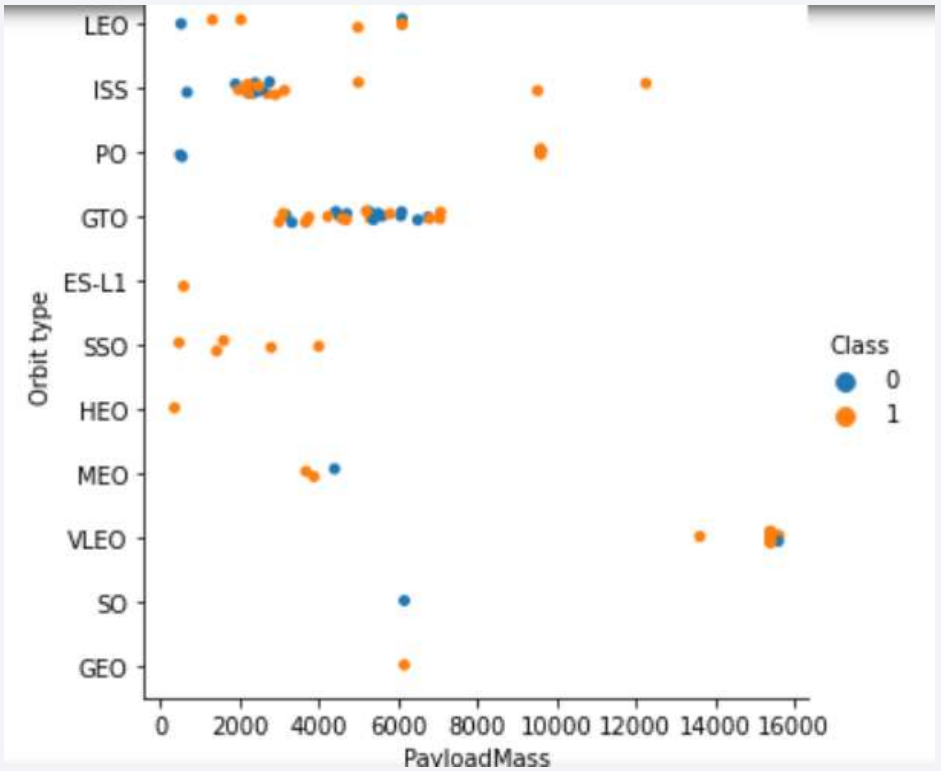
Flight Number vs. Orbit Type

- ***SSO Orbit type has never failed. The success rate for the majority of orbit types has increased with an increased number of flights***



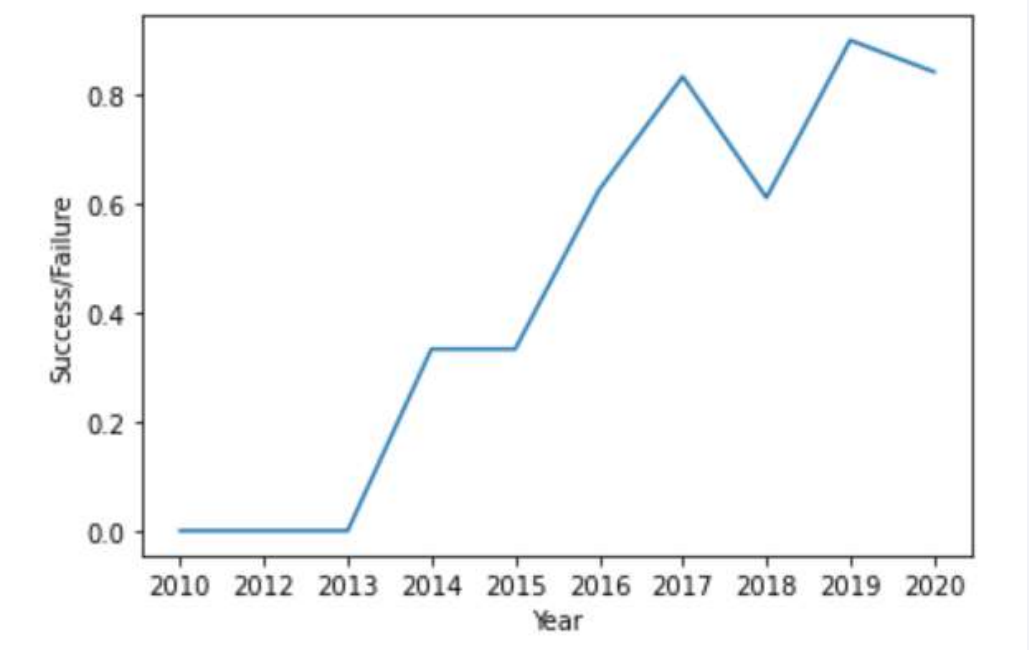
Payload vs. Orbit Type

- *The payload mass greater than 8000 kg shows a high success rate PO and ISS orbit types.*
- *The majority of launches are between 4000 and 8000 payload mass.*
- *The more heavily used orbit type is GTO*



Launch Success Yearly Trend

- ***The yearly average success rate keeps growing with time and supposedly continue to develop and grow.***



All Launch Site Names

- ***The query to display the names of the unique launch sites in the space mission is as follows:***

```
%sql select distinct(Launch_Site) from SPACEXTBL;
```

- ***To get unique site names we use Distinct keyword:***

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

Launch Site Names Begin with 'CCA'

- ***The query to find 5 records where launch sites begin with `CCA` is as follows:***

```
%sql select Launch_Site from SPACEXTBL where (Launch_Site) LIKE 'CCA%' LIMIT 5
```

- ***In order to get 5 records we use the LIMIT keyword:***

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- ***The query to calculate the total payload carried by boosters from NASA is as follows:***

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL where Customer = 'NASA (CRS)';
```

- ***In order to get the total payload we use SUM function keyword:***

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

- ***The query to calculate the average payload mass carried by booster version F9 v1.1 is as follows:***

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL where Booster_Version = 'F9 v1.1';
```

- ***In order to get average payload mass we use AVG function keyword:***

Average Payload Mass by Booster Version F9 v1.1
2928

First Successful Ground Landing Date

- ***The query to find the dates of the first successful landing outcome on the ground pad is as follows:***

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

- ***In order to get first successful outcome date we use MIN function keyword:***

First Succesful Landing Outcome in Ground Pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- ***The query to get the list of the names of boosters that have successfully landed on a drone ship and had payload mass greater than 4000 but less than 6000 is as follows:***

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

- ***In order to get successfully landed boosters we use keyword BETWEEN 4000 and 6000:***

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

Total Number of Successful and Failure Mission Outcomes

- The queries to calculate the total number of successful and failed mission outcomes are as follows:

```
%sql select count(Mission_Outcome) as 'Successful Mission' from SPACEXTBL where Mission_Outcome LIKE 'Success%';
```

```
%sql select count(Mission_Outcome) as 'Failure Mission' from SPACEXTBL where Mission_Outcome LIKE 'Failure%';
```

- In order to get success and failure mission outcomes we use COUNT and LIKE “Success%” and “Failure%” keywords:

Successful Mission
100

Failure Mission
1

Boosters Carried Maximum Payload

- ***The query to list the names of the booster which have carried the maximum payload mass is as follows:***

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL \
where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

- ***We use a subquery here to get boosters with maximum payload:***

boosterversion
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

- ***The query to list the failed landing_outcomes in drone ship, their booster versions, and launch site names for the year 2015 is as follows:***

```
%sql select substr(Date, 4, 2) as MONTH, Mission_Outcome, Booster_Version, Launch_Site from SPACEXTBL \
where substr(Date,7,4)='2015' and Landing_Outcome = 'Failure(drone ship)';
```

- ***We use Date keyword here to choose between month and year:***

MONTH	booster_version	launch_site
1	F9 v1.1 B1012	CCAFS LC-40
4	F9 v1.1 B1015	CCAFS LC-40

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

- ***The query to rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20, in descending order is as follows:***

```
%sql select Landing_Outcome as Landing Outcome, count(Landing_Outcome) as Count from SPACEXTBL \
where Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome \
ORDER BY count(Landing_Outcome) DESC;
```

- ***We use GROUP BY and ORDER BY keywords to consolidate the results:***

Landing Outcome	Total Count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

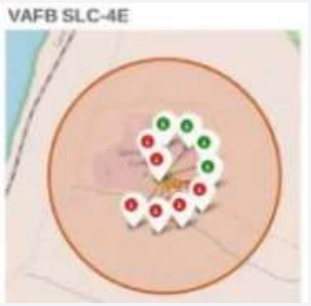
Launch Sites Proximities Analysis

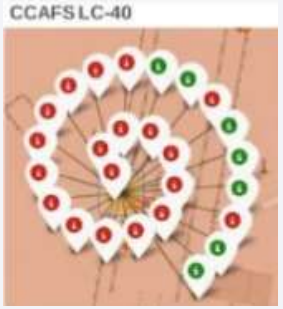
Launch Sites Location Markers – All Sites

- Launch sites are located on US coastline (Florida and California)

Color-Labeled Launch Outcomes

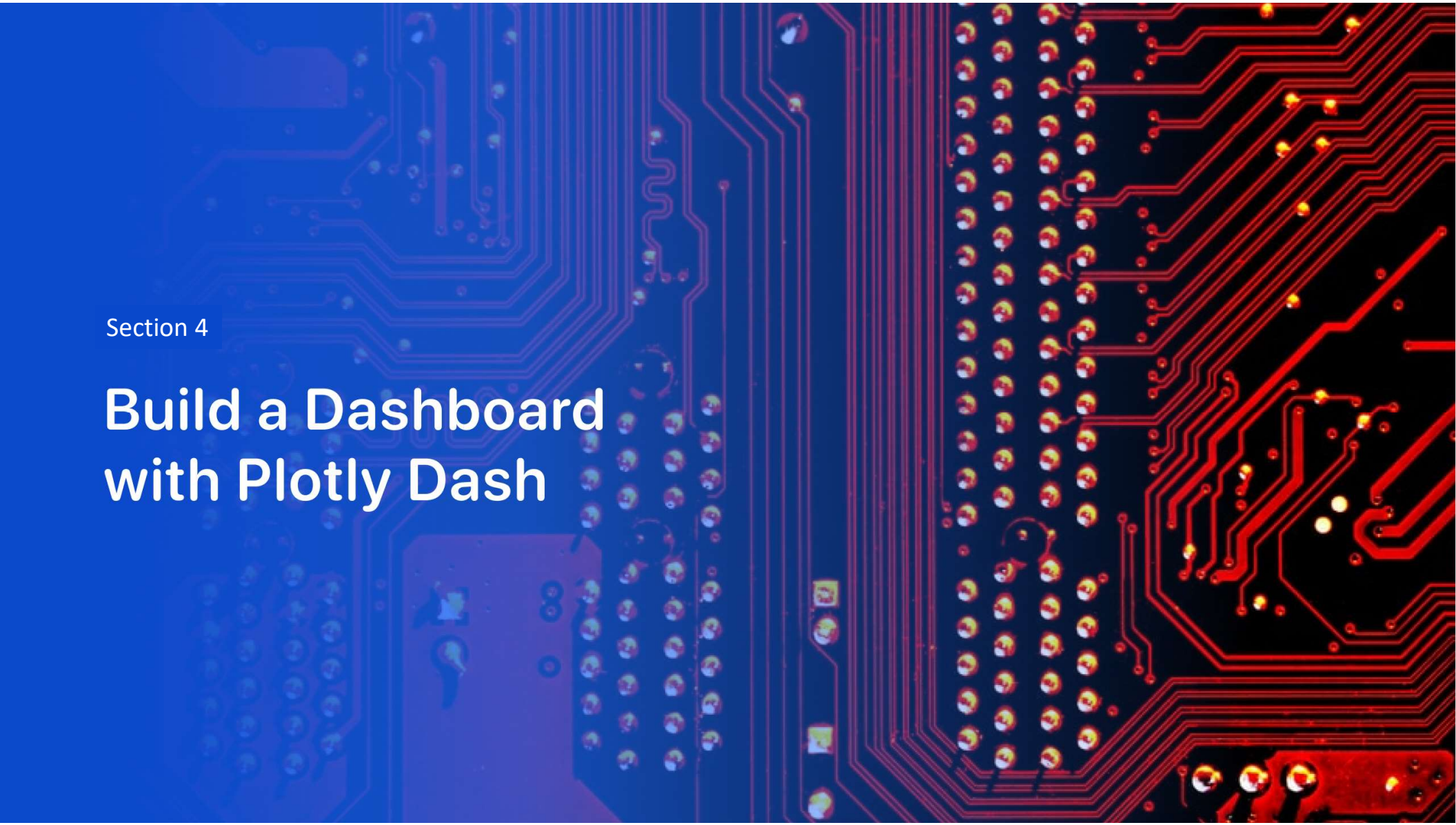
- Folium map illustrates success/failure launch outcomes for 2 sites.
- Green color defines successful outcome (Success)
- Red color defines unsuccessful outcome (Failure)





Launch Site Distance to Its Proximities

- The calculated distances from launch sites indicate that they are in its relatively closed proximity to railways, highways and costal line

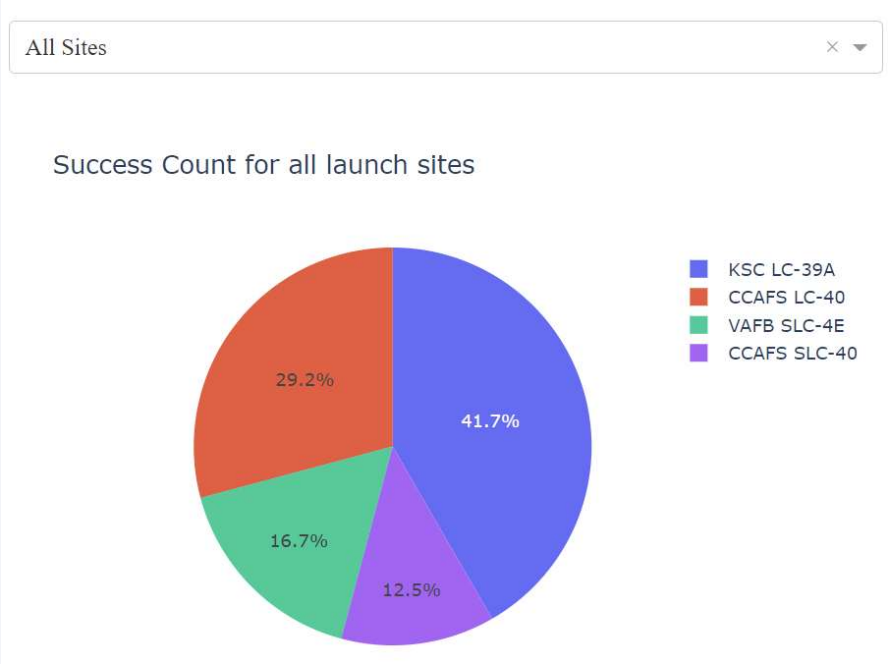


Section 4

Build a Dashboard with Plotly Dash

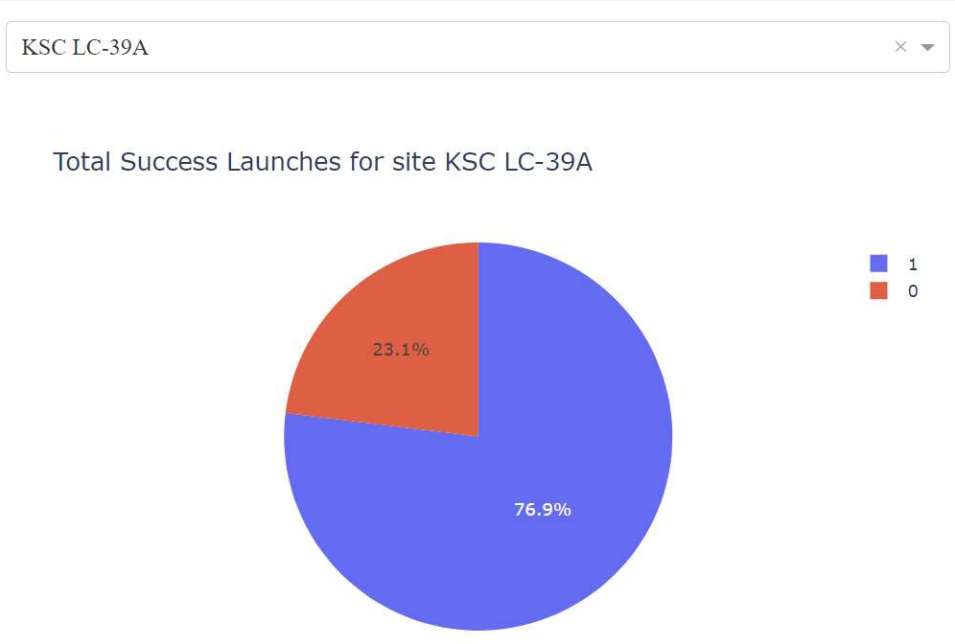
Launch Sites Success Count – All Sites

- KSCMLC-39A shows the highest success score of 41.7 %
- CCAFS LC-40 has the success score of 29.3 %
- CCAFS SLC-40 has the lowest success score of 12.5 %



Total Success Launches – Site KSC LC-39A

- KSC LC-39A has the highest successful number of launches of 76.9 %
- The highest success count in FT Booster version category
- The typical payload mass is 2000 – 6000 kg.



Success Count on Payload Mass – All Sites

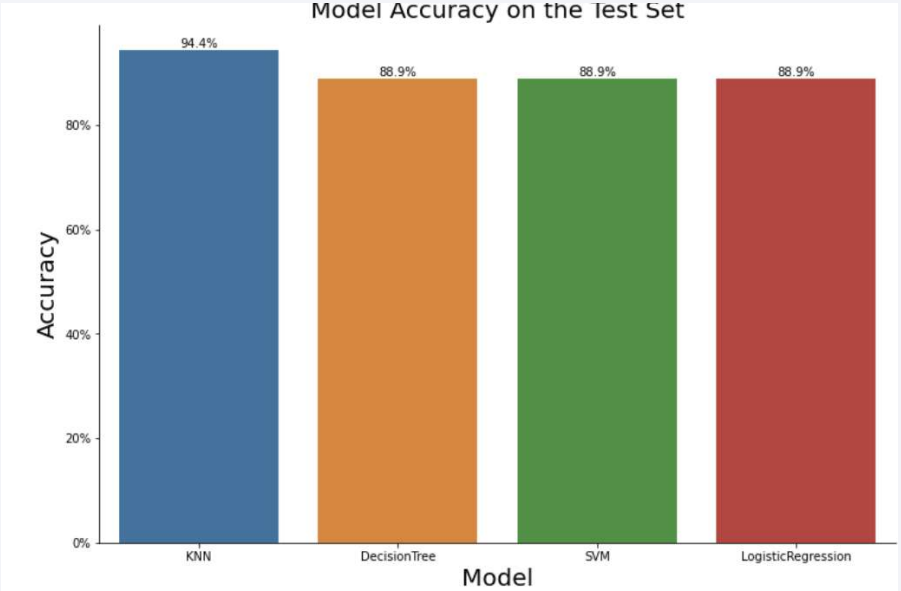
- The payload mass for all booster version category is between 1000 – 7000 kg.
- The highest success count belongs to FT and B4 booster version categories
- The payload mass for successful categories is 2000 – 5000 kg.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- *KNN model has the highest classification accuracy*

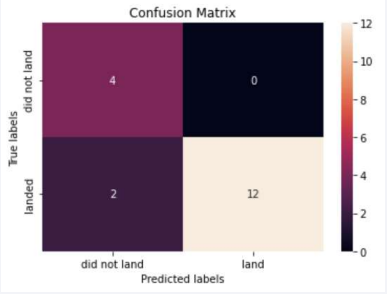


Confusion Matrix

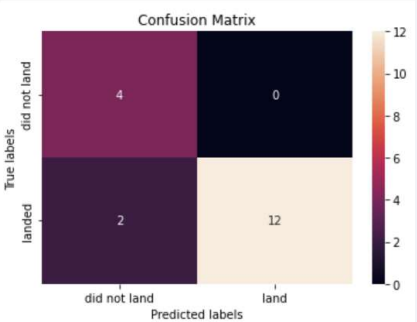
- Check out the confusion matrix for the best performing model

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

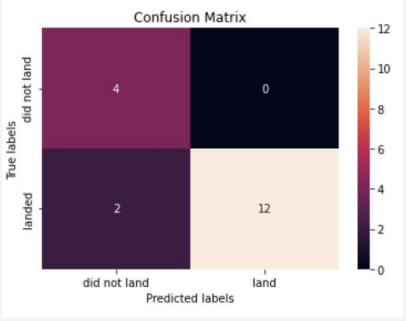

```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```




```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```




```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Launch site KSC LC-39A has a highest success rate
- The payload range 2000 – 5000 kg is more typical for successful launches
- Decision Tree model performed the best for the training data set with a score of 0.87.
- KNN model showed high accuracy of 94 % on test data set.

Appendix

- The repo for this capstone can be found here:

https://github.com/vadimsavenkov/Applied_DS_Capstone

Thank you!