Winning Space Race with Data Science

Mohan CK
26-09-2023

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- Exploratory Data Analysis with Data Visualization
- Exploratory Data Analysis with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

Summary of all results

- Exploratory Data Analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Introduction

Project background and context

- SpaceX is the most successful company of the commercial space age, making space travel affordable. The company 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 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. Based on public information and machine learning models, we are going to predict if SpaceX will reuse the first stage.

Problems we want to find answers

- How do variables such as payload mass, launch site, number of flights, and orbits affect the success of the first stage landing?
- Does the rate of successful landings increase over the years?
- What is the best algorithm that can be used for binary classification in this case?

Section 1

Methodology

Methodology

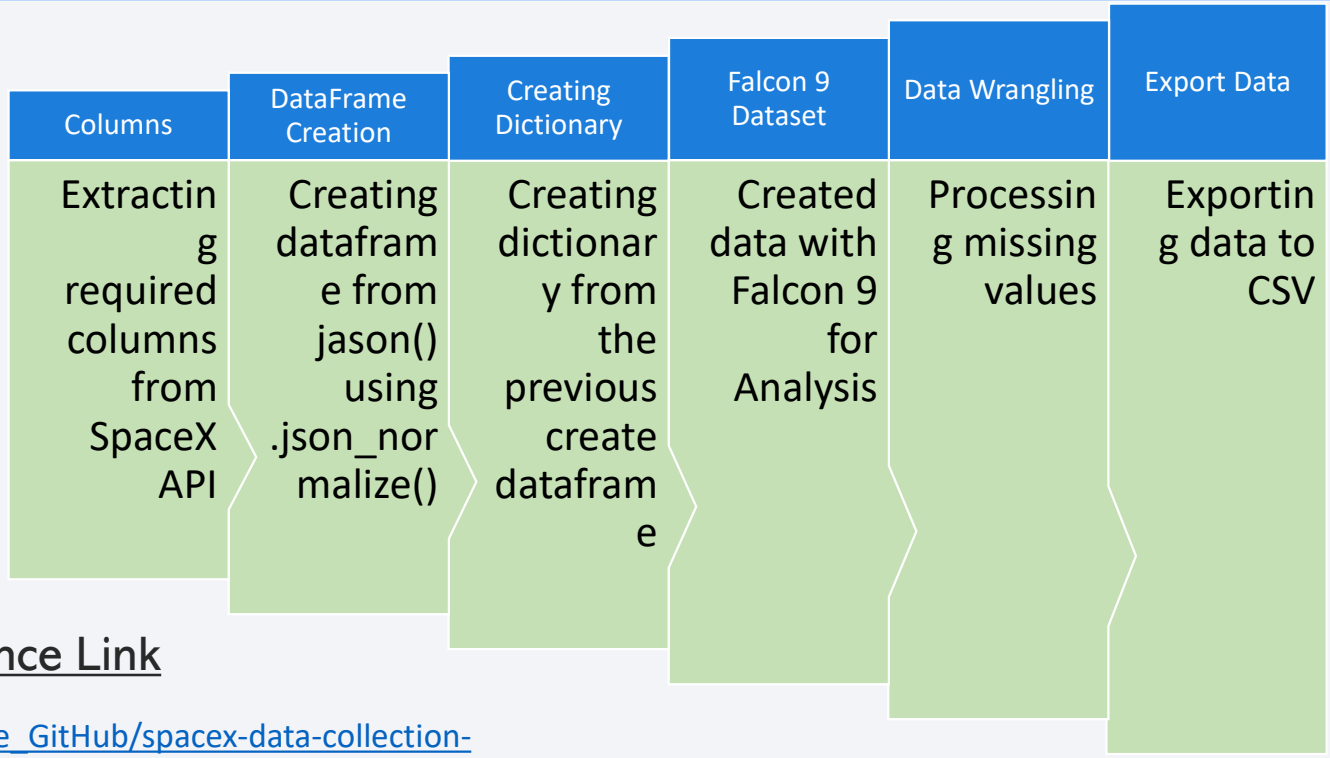
Executive Summary

- Data collection methodology:
  - Data were collected from SpaceX API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Filter required data
  - Finding and Updating missing 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
  - Build couple of classification models (KNN, Decision Tree, etc.,) to find best model.

Data Collection

- Describe how data sets were collected.
  - Data were collected from SpaceX API
- You need to present your data collection process use key phrases and flowcharts
  - With defined function extracted required columns (as listed below) from the datasets through SpaceX API.
  - Column names – rocket, launchpad, payloads_mass_kg, orbit, core, block, reuse_count, serial, flight, grindfins, reused, legs, landpad, etc.,

Data Collection – SpaceX API



- Reference Link

Data_science_GitHub/spacex-data-collection-api.ipynb at main · Mohanck19/Data_science_GitHub

Data Collection - Scraping

Requests	• Using Requests method, extracted the html codes with test from of Wikipedia
Beautifulsoup	• Using Beautifulsoup object extract and formatted html text
Findall()	• Using Findall() function extracted table and their contents
Extract column Names	• Extracted column names using arrays.
Create Empty DataFrame	• Created Dataframe to read to all table rows and definitions (Cells)

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.

```
# use requests.get() method with the provided static_url
# assign the response to a object
falcon9_page = requests.get(static_url).text
```

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(falcon9_page, 'html5lib')
```

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

```
# Use soup.title attribute
print(soup.title)
```

```
headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]

def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll < lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)

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

- Reference Link
- Mohanck19/Data_science_GitHub: Creating this space to practice code collaboration

Data Wrangling

- Describe how data were processed

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship. We mainly convert those outcomes into Training Labels with “1” means the booster successfully landed, “0” means it was unsuccessful

- Data_science_GitHub/Spacex-data_wrangling.ipynb at main · Mohanck19/Data_science_GitHub

TASK 1: Calculate the number of launches on each site

The data contains several Space X launch facilities: Cape Canaveral Space Launch Complex 40 **VAFB SLC 4E** , Vandenberg Ai Space Launch Complex 4E **(SLC-4E)**, Kennedy Space Center Launch Complex 39A **KSC LC 39A** .The location of each Launch 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
LS_ValueCounts = df['LaunchSite'].value_counts()
print(LS_ValueCounts)
```

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

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
  - Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit Type vs. Success Rate, Flight Number vs. Orbit Type, Payload Mass vs Orbit Type and Success Rate Yearly Trend
  - Scatter plots show the relationship between variables. If a relationship exists, they could be used in machine learning model.
  - Bar charts show comparisons among discrete categories. The goal is to show the relationship between the specific categories being compared and a measured value.
  - Line charts show trends in data over time (time series).
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

Data_science_GitHub/module_2-eda-dataviz.ipynb at main · Mohanck19/Data_science_GitHub

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'CCA'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date when the first successful landing outcome in ground pad was achieved
  - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster versions which have carried the maximum payload mass
  - Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015
  - Ranking 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
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
- Data_science_GitHub/EDA_SQL_completed.ipynb at main · Mohanck19/Data_science_GitHub

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

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)

- Summarize how you built, evaluated, im
  performing classification model
- You need present your model developn
  flowchart
- Add the GitHub URL of your completed
  reference and peer-review purpose
- [Data_science_GitHub/SpaceX_M… Analysis.ipynb at main · Mohanck19/Data_science_GitHub]

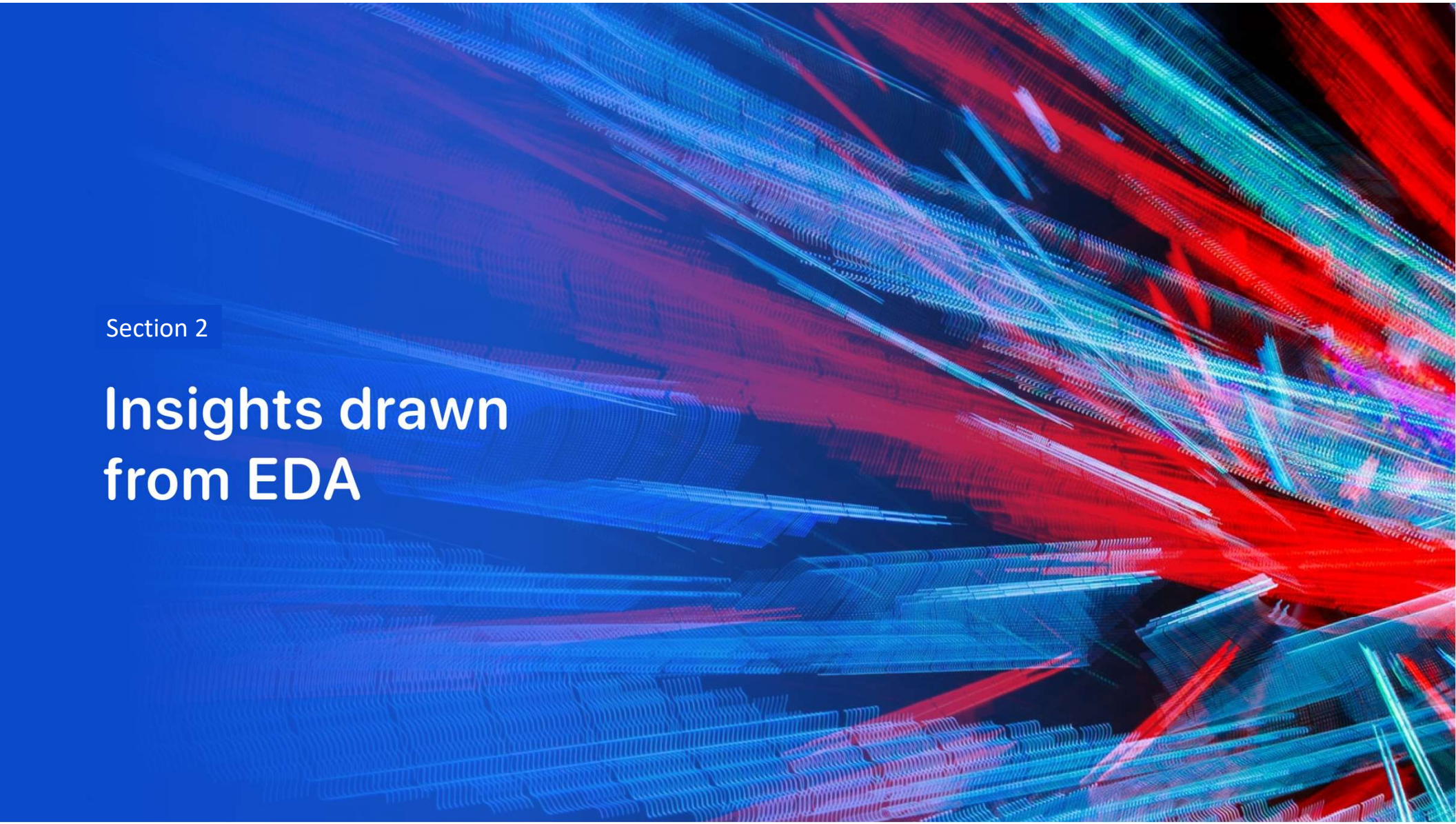
```
from js import fetch
import io

URL1 = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_pa
resp1 = await fetch(URL1)
text1 = io.BytesIO((await resp1.arrayBuffer()).to_py())
data = pd.read_csv(text1)
```

```
data.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Blo
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	

Results

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



Section 2

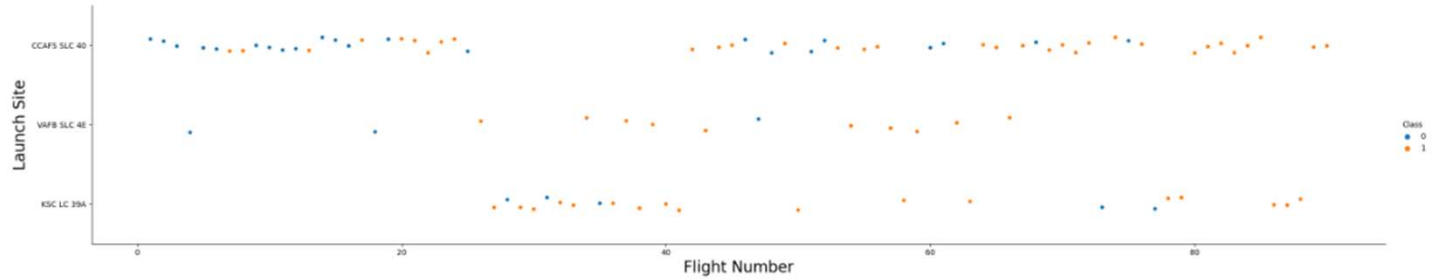
Insights drawn from EDA

Flight Number vs. Launch Site

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

- Show the screenshot of the scatter plot with explanations

```
[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

- Show a scatter plot of Payload vs. Launch Site

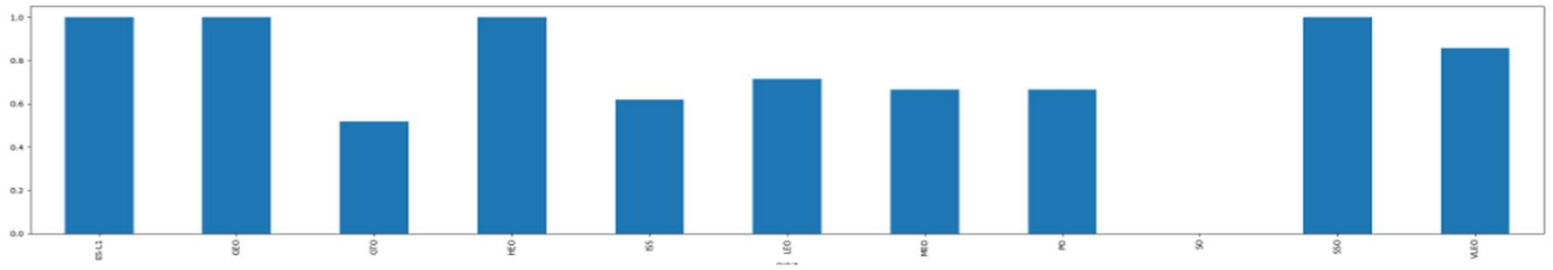
- Show the screenshot of the scatter plot with explanations

[6]:
```
# 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
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass", 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
- Show the screenshot of the scatter plot with explanations

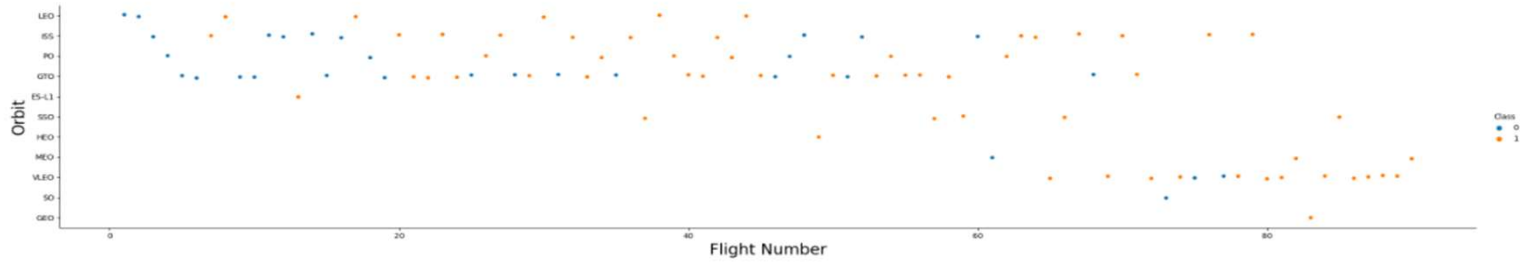
```
# HINT use groupby method on Orbit column and get the mean of Class column
df_success_grp = df.groupby(['Orbit'])['Class'].mean()
df_success_grp.plot(kind='bar')
plt.show()
```



Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations

```
[11]: # 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

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations

```
5]:
# 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("PayloadMass", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```

Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations

All Launch Site Names

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

Task 1

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

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

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

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

Task 2

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

```
%sql select * from SPACEXTBL where Launch_Site like '%CCA%' limit 5
```

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

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

Total Payload Mass

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

Task 3

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

```
%sql select Customer, sum(PAYLOAD_MASS__KG_) AS PAYLOAD_MASS FROM SPACEXTBL where Customer = "NASA (CRS)" group by Customer
```

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

Customer	PAYLOAD_MASS
NASA (CRS)	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

Task 4

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

```
11]: %sql select Booster_Version, avg(PAYLOAD_MASS__KG_) AS PAYLOAD_MASS FROM SPACEXTBL  where Booster_Version like '%F9 v1.1%'
 * sqlite:///my_data1.db
Done.
```

11]:

Booster_Version	PAYLOAD_MASS
F9 v1.1 B1003	2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

Task 5

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

Hint:Use min function

```
%sql select distinct(Landing_Outcome), min(Date) as Date from SPACEXTBL WHERE Landing_Outcome = "Success (ground pad)"
```

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

Landing_Outcome	Date
Success (ground pad)	2015-12-22

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

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, PAYLOAD_MASS__KG_ from SPACEXTBL WHERE Landing_Outcome = "Success (drone ship)" and PAYLOAD_MAS
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

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

Task 7

List the total number of successful and failure mission outcomes

```
%sql select Landing_Outcome, count(Landing_Outcome) as count from SPACEXTBL where Landing_Outcome like '%Success%' or Landir
```

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

Landing_Outcome	count
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Success	38
Success (drone ship)	14
Success (ground pad)	9

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

Task 8

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

```
%sql select Booster_Version, max(PAYLOAD_MASS__KG_) as PAYLOAD_MASS from SPACEXTBL
```

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

Booster_Version	PAYLOAD_MASS
F9 B5 B1048.4	15600

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

- Present your query result with a short explanation here

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, strftime('%m', datetime(date, 'unixepoch')) as monthname, substr(Date, 6,2) as month, Landing_Outcome, Boo
```

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

Date	monthname	month	Landing_Outcome	Booster_Version	Launch_Site
2015-10-01	None	10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	None	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-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

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.

[44]:
```
%sql select date, Landing_Outcome, count(Landing_Outcome) as count from SPACEXTBL where date >='2010-06-04' and date <= '201
```

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

[44]:

Date	Landing_Outcome	count
2015-12-22	Success (ground pad)	5
2016-08-04	Success (drone ship)	5
2015-10-01	Failure (drone ship)	5
2018-05-12	Failure	3
2010-04-06	Failure (parachute)	2

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

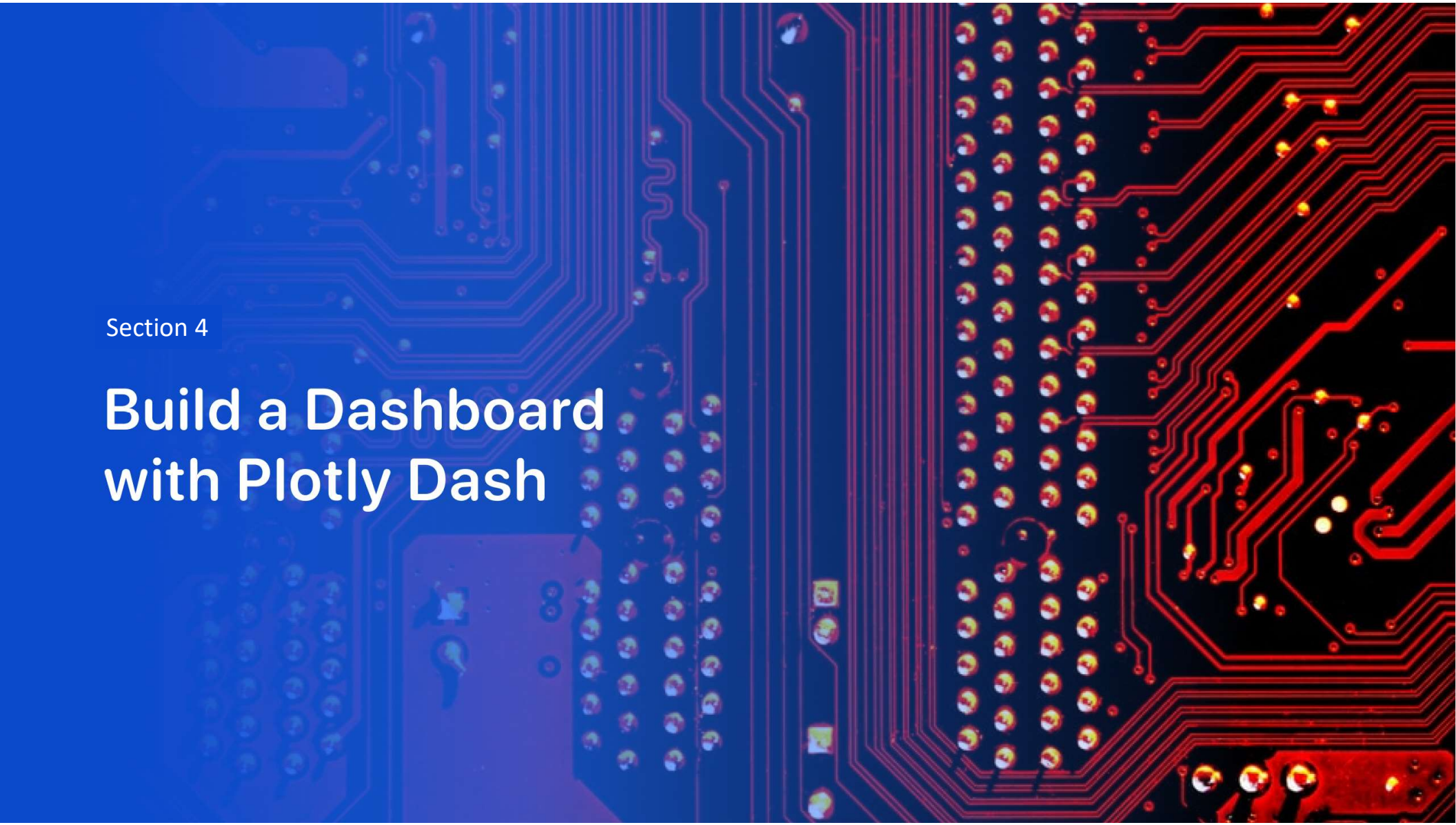
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

<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>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

<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)

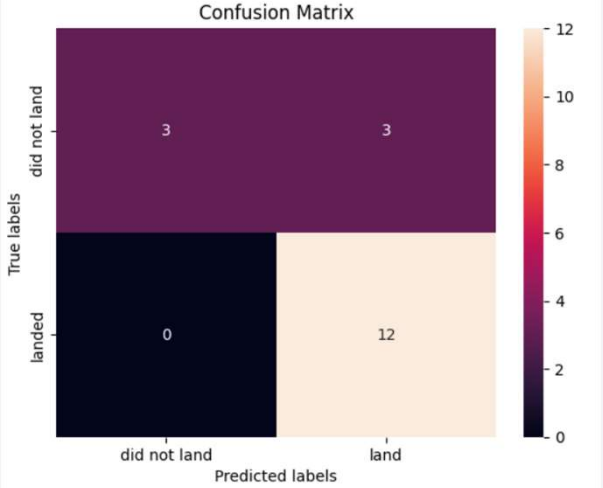
Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy
- Best model is KNN with highest accuracy socre of 0.8875

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

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



Conclusions

- Point 1
- Point 2
- Point 3
- Point 4
- …

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!