IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dikeola Ogunmola
16th September 2024

Outline

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

Executive Summary

- Data Collection with Web-Scraping and APIs
- Data Wrangling and Exploration Python
- Insights from Analysis
- Recommendations

Introduction

- SPACE X
- Rocket Launches; Successes and Failures
- The goal is to predict the success of subsequent launches

Section 1

Methodology

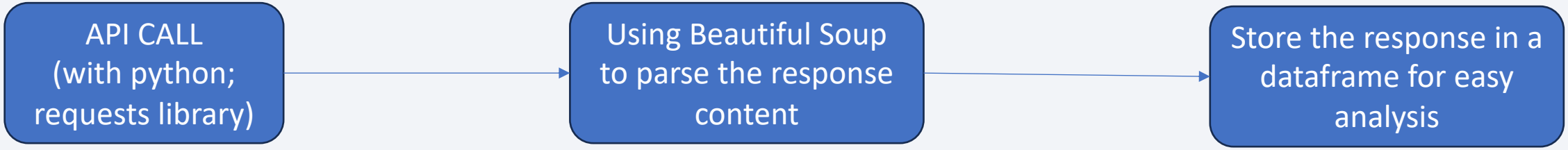
Methodology

Executive Summary

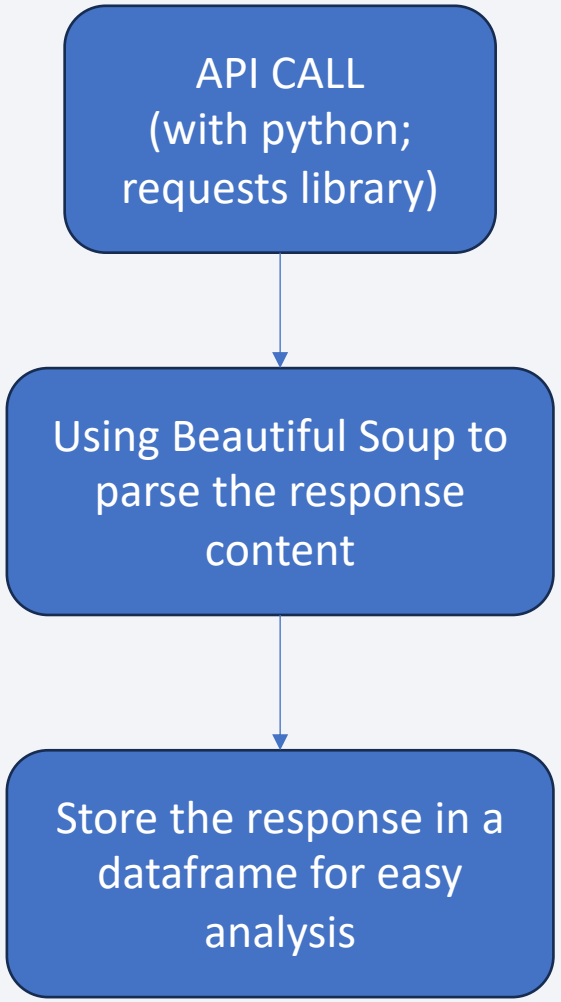
- Data collection methodology:
  - Web-Scraping
  - API
- Perform data wrangling
  - Data Preprocessing with Python
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data was collected using API calls; through Python programming language. Then the needed information was retrieved with the BeautifulSoup python library.



Data Collection – SpaceX API

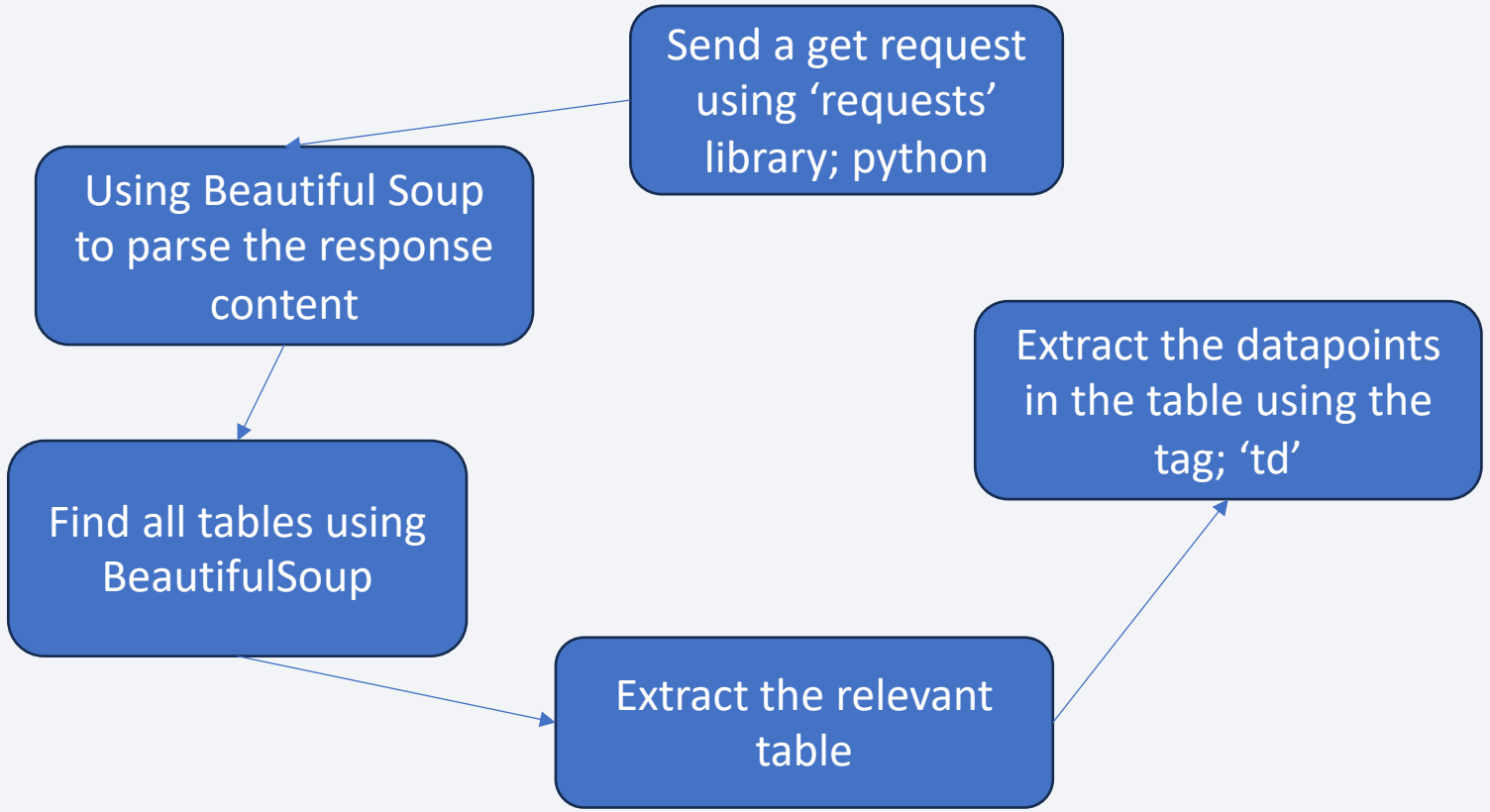


- GitHub URL : https://github.com/Dikeola/IBM-Data-Science-Professional-Certificate/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- GitHub URL:

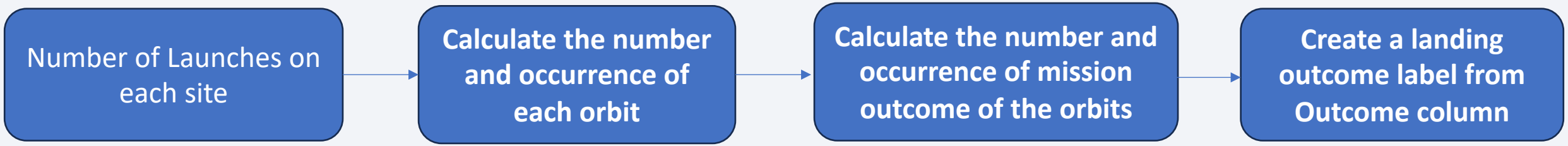
https://github.com/Dikeola/IBM-Data-Science-Professional-Certificate/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Data Preprocessing done with python
- GitHub URL

https://github.com/Dikeola/IBM-Data-Science-Professional-Certificate/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

EDA with SQL

- %sql select distinct(Launch_Site) from SPACEXTABLE
- %sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
- %sql select sum(PAYLOAD_MASS__KG_) as total_payload_mass from SPACEXTABLE where Customer='NASA (CRS)'
- %sql select avg(PAYLOAD_MASS__KG_) as average_payload_mass from SPACEXTABLE where Booster_Version='F9 v1.1'
- %sql select min(Date) from SPACEXTABLE where Landing_Outcome like '%Success%'
- %%sql

SELECT Booster_VersionFROM SPACEXTABLE

WHERE Landing_Outcome NOT LIKE '%ilure%' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000 AND Landing_Outcome LIKE '%drone%'

EDA with SQL

- %%sql

select Landing_Outcome,count(Landing_Outcome)

from SPACEXTABLE

group by Landing_Outcome%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5

- %%sql

select Booster_Version from SPACEXTABLE

where PAYLOAD_MASS__KG_ =(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)

- %%sql

select substr(Date, 6,2) as month, Landing_Outcome, Booster_Version,Launch_Site

from SPACEXTABLE

where Landing_Outcome like '%ilure%' and Landing_Outcome like '%drone%'

and substr(Date,0,5)='2015'

EDA with SQL

- %%sql

```
select Date, Landing_Outcome, count(Date) as countt
    from SPACEXTABLE
where Date between '2010-06-04' and '2017-03-20
group by Date
and Landing_Outcome in ('Failure (drone ship)', 'Success (ground pad)')
```

- GitHub URL:

https://github.com/Dikeola/IBM-Data-Science-Professional-Certificate/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

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

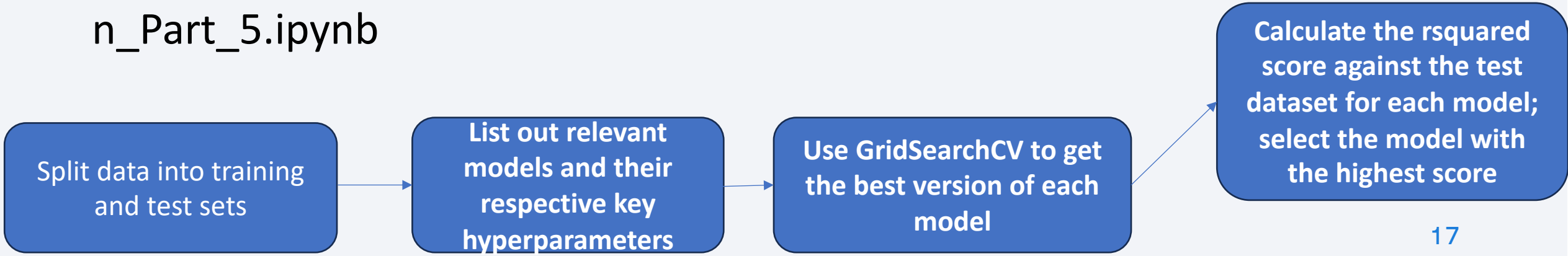
Build a Dashboard with Plotly Dash

- 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)

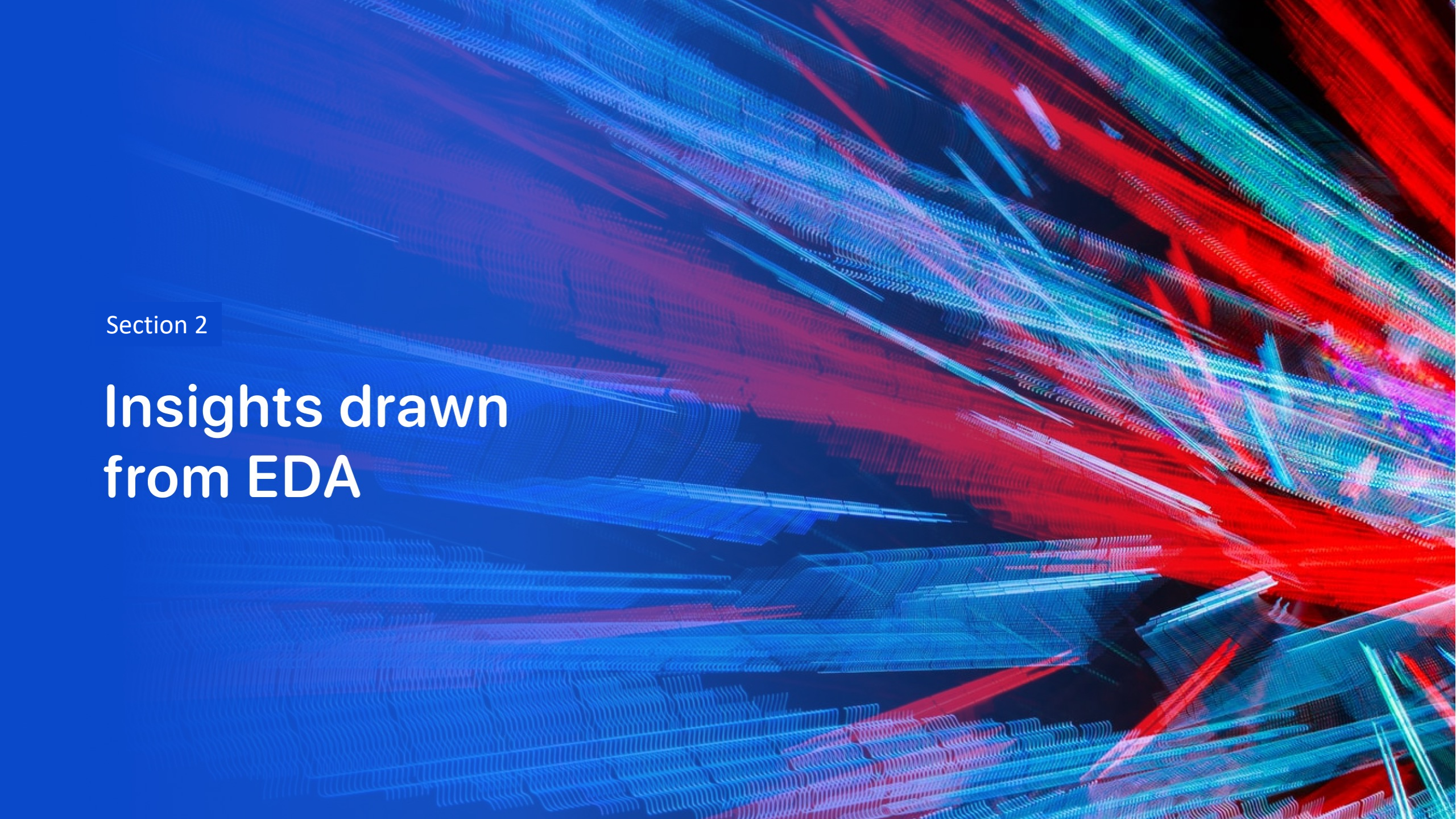
- Made a selection of classification models and used them along with key parameters in GridSearchCV for getting the best performing model with corresponding parameters.
- GitHub URL

https://github.com/Dikeola/IBM-Data-Science-Professional-Certificate/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

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

- Show the screenshot of the scatter plot with explanations

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations

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

Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

- Show the screenshot of the scatter plot with explanations

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations

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

LAUNCH SITE NAMES

1. CCAFS LC-40
2. VAFB SLC-4E
3. KSC LC-39A
4. CCAFS SLC-40

- %sql select distinct(Launch_Site) from SPACEXTABLE

Distinct; is used to select the unique Launch Sites in the table

Launch Site Names Begin with 'CCA'

- %sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5

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

The where clause matches site names that start with ‘CCA’ followed by anything; space, numbers,characters or CCA itself.

Total Payload Mass

- %sql select sum(PAYLOAD_MASS__KG_) as total_payload_mass from SPACEXTABLE where Customer='NASA (CRS)'

- **total_payload_mass –** 45596

- Filter the records with 'where Customer='NASA (CRS)'. The Sum function is used to add up the entries In a numerical column

Average Payload Mass by F9 v1.1

- %sql select avg(PAYLOAD_MASS__KG_) as average_payload_mass from SPACEXTABLE where Booster_Version='F9 v1.1’

- **average_payload_mass** - 2928.4

First Successful Ground Landing Date

- %sql select min(Date) from SPACEXTABLE where Landing_Outcome like '%Success%'

- **First successful Ground Landing Date** - 2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- %%sql

```
SELECT Booster_Version

FROM SPACEXTABLE

WHERE Landing_Outcome NOT LIKE
'%ilure%'

  AND PAYLOAD_MASS__KG_ > 4000

  AND PAYLOAD_MASS__KG_ < 6000

  AND Landing_Outcome LIKE '%drone%'
```

Booster_Version

1.F9 FT B1022

2.F9 FT B1026

3.F9 FT B1021.2

4.F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- %%sql

```
select Landing_Outcome,count(Landing_Outcome)
from SPACEXTABLE
group by Landing_Outcome
```

Landing_Outcome	count(Landing_Outcome)
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	21
No attempt	1
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Boosters Carried Maximum Payload

- %%sql

select Booster_Version from SPACEXTABLE

where PAYLOAD_MASS__KG_ =(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)

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

- %%sql

```
select substr(Date, 6,2) as month, Landing_Outcome,
Booster_Version,Launch_Site

from SPACEXTABLE

where Landing_Outcome like '%ilure%' and Landing_Outcome like '%drone%'

and substr(Date,0,5)='2015'
```

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

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

- %%sql

```
select Date, Landing_Outcome, count(Date) as countt
    from SPACEXTABLE
where Date between '2010-06-04' and '2017-03-20
group by Date
and Landing_Outcome in ('Failure (drone ship)', 'Success (ground pad)')
```


Section 3
Launch Sites
Proximities Analysis

<Folium Map Screenshot 1>

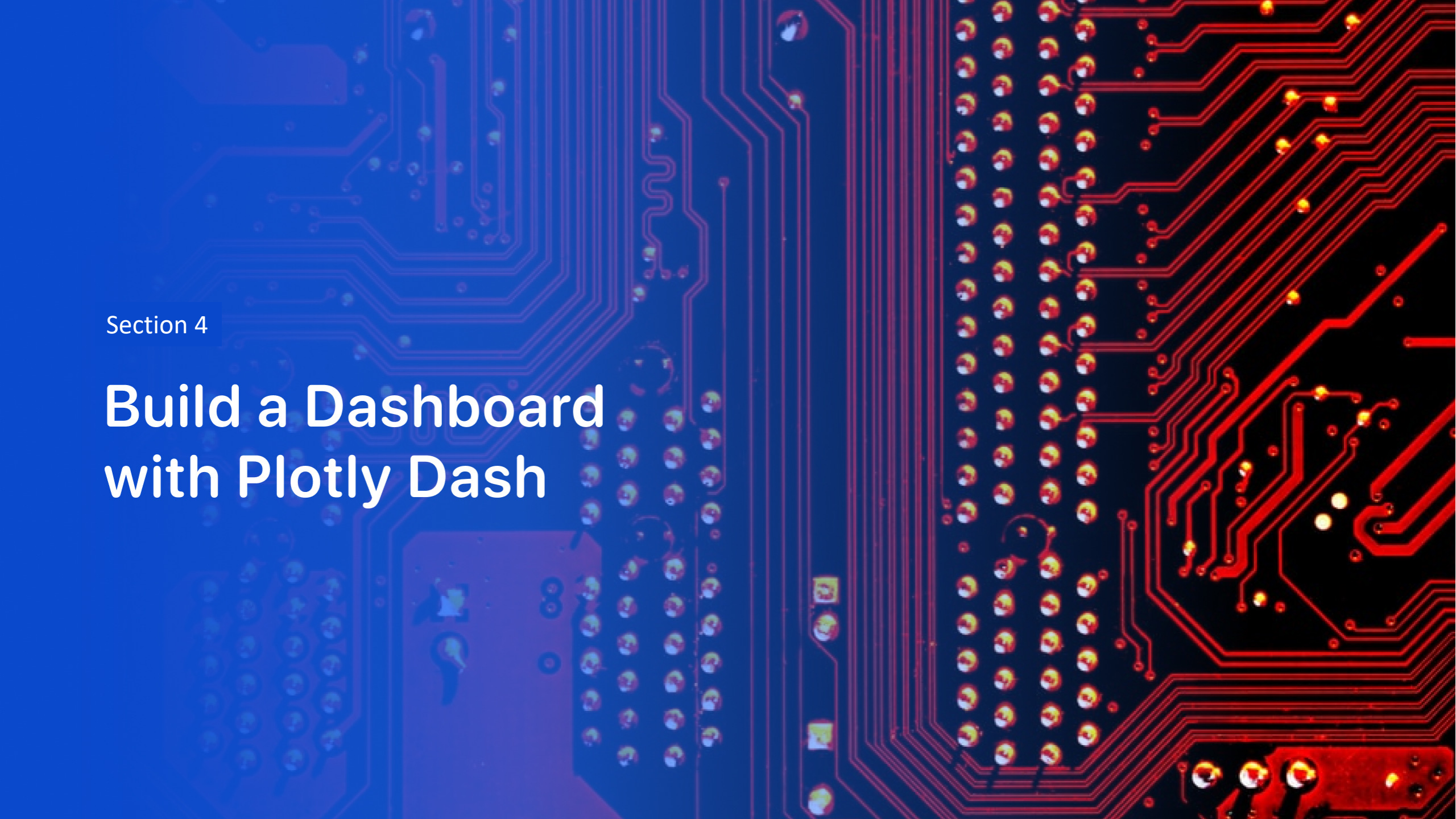
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

<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>

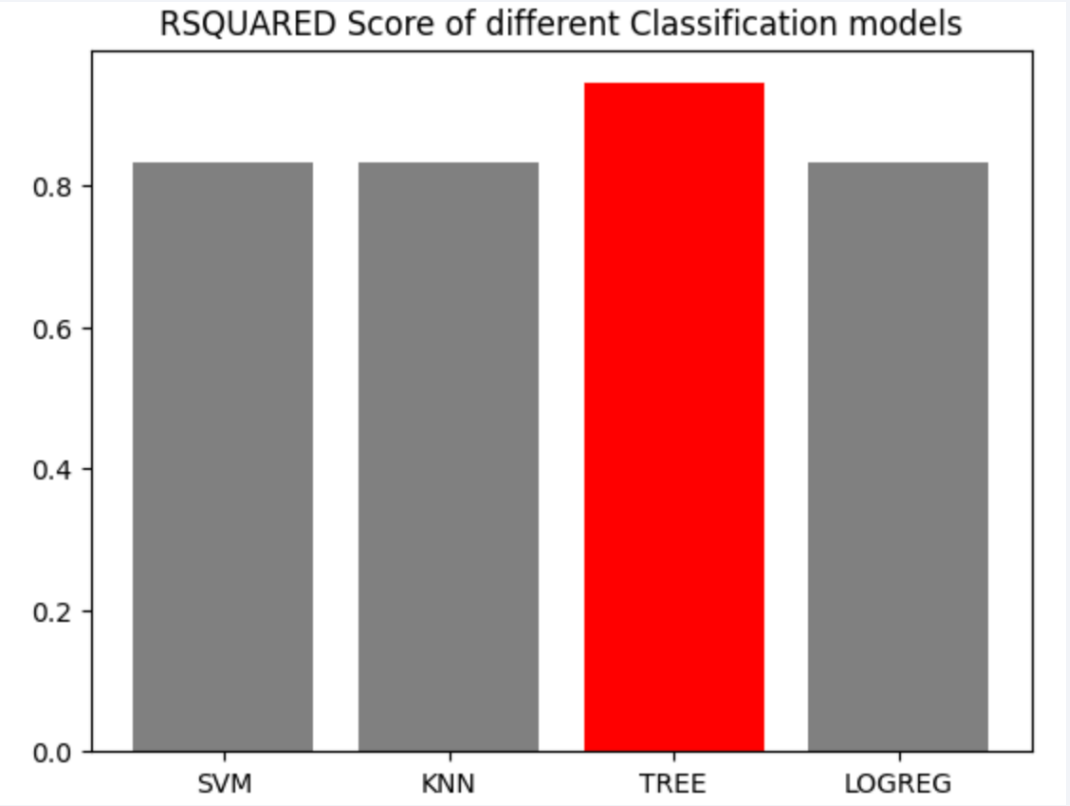
- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version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

Section 5

Predictive Analysis (Classification)

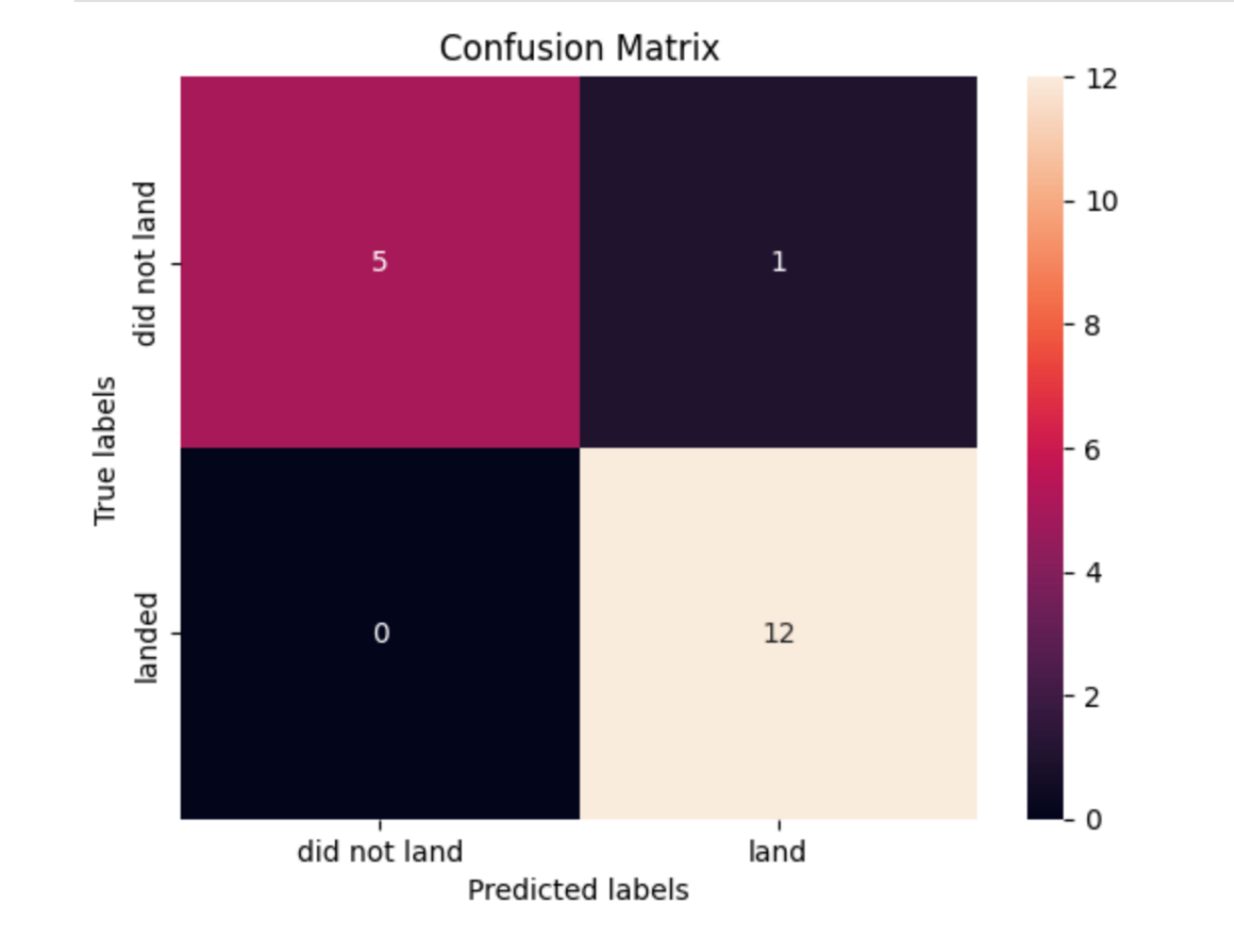
Classification Accuracy

- Decision Tree Classifier has the highest accuracy



Confusion Matrix

- Decision Tree Confusion Matrix



Conclusions

- Decision Tree Classifier is the model of choice

- 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!