Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies
  - IBM Watson Studio enterprise solution for data scientists and data engineers
  - GitHub for Web-based access control, collaboration, and cloud storage
  - Jupyter notebook free software, open standards, and web services for interactive computing
  - SQL Structured Query Language for access and manipulate databases
  - Python language, Folium Lab, Plotly Dash for data programming processing and plots
- Summary of all results
  - Module 1 Understood IBM Cloud, Watson Studio, Data Collection with data scraping and wrangling
  - Module 2 Learned Exploratory Data Analysis with SQL and Visualization
  - Module 3 Exercised with Interactive Visual Analytics, Folium, Plotly Dash
  - Module 4 Exercised with Predictive Analysis with Lab
  - Module 5 Reviewed entire course and provided final presentation in Microsoft PowerPoint slides

Introduction

- Project background and context
  - Apply data science skills to analyze data from a space launch company
  - Collect relevant data from different WEB sites, practice SQL
  - Use basic statistics to analyze data, use software tools to visualize data
  - Assign the data into categorical or factorized groups
  - Build predictive data models
  - Review all the learned materials, develop final PowerPoint slides for presentation
- Problems you want to find answers
  - To discover and know commercial Data Science tools such as IBM Cloud Watson Studio, GitHub, SQL, Folium, Plotly etc.
  - To be familiar with Python commands for Data Science application

Section 1

Methodology

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

Methodology

Executive Summary

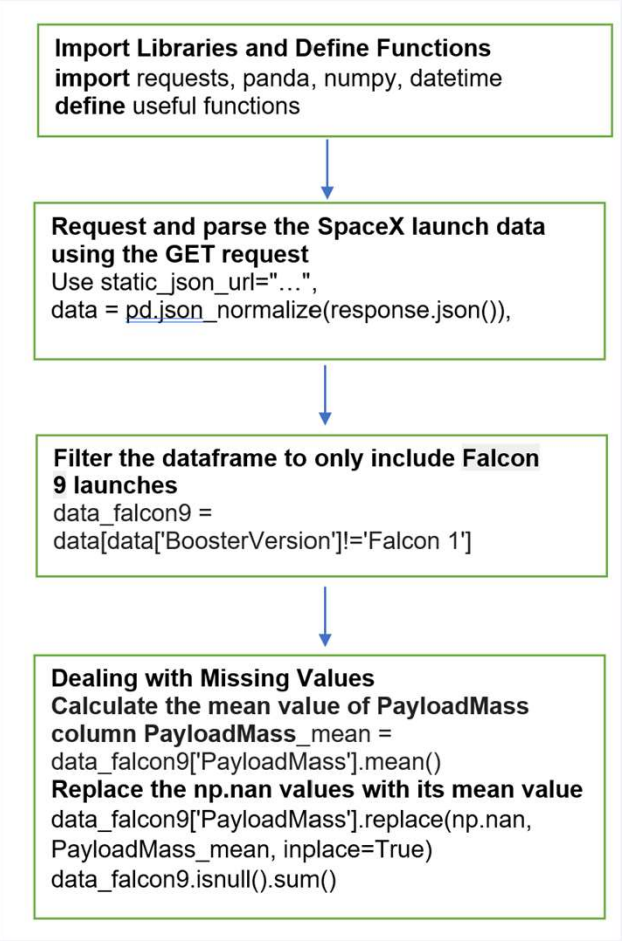
- Data collection methodology:
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
  - Import Python libraries Panda, Numpy, datetime; Define helper functions, use API `requests.get() method, create a beautiful object,` use .json() to decode data and turn it into a Pandas dataframe by .json_normalize(), fill missing data with mean() values.
- You need to present your data collection process use key phrases and flowcharts

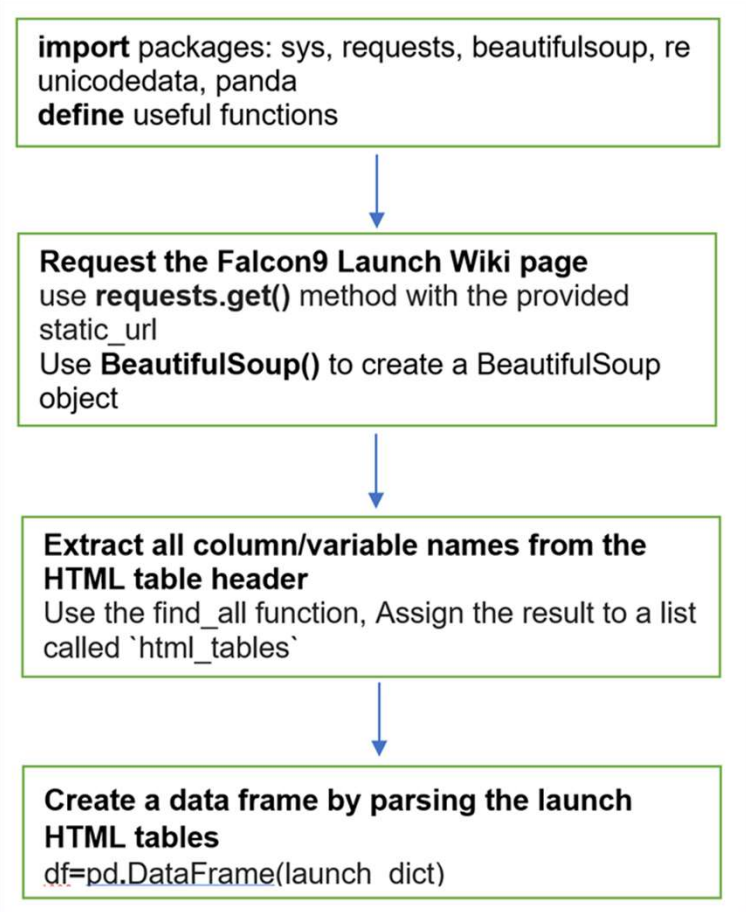
Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
- GitHub URL of the completed SpaceX API calls notebook [test_repository/Complete the Data Collection API Lab.ipynb at main · mygithub22021/test_repository]



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- Add the GitHub URL of the completed web scraping notebook
  test_repository/Data Collection WEB scraping Lab.ipynb at main · mygithub22021/test_repository



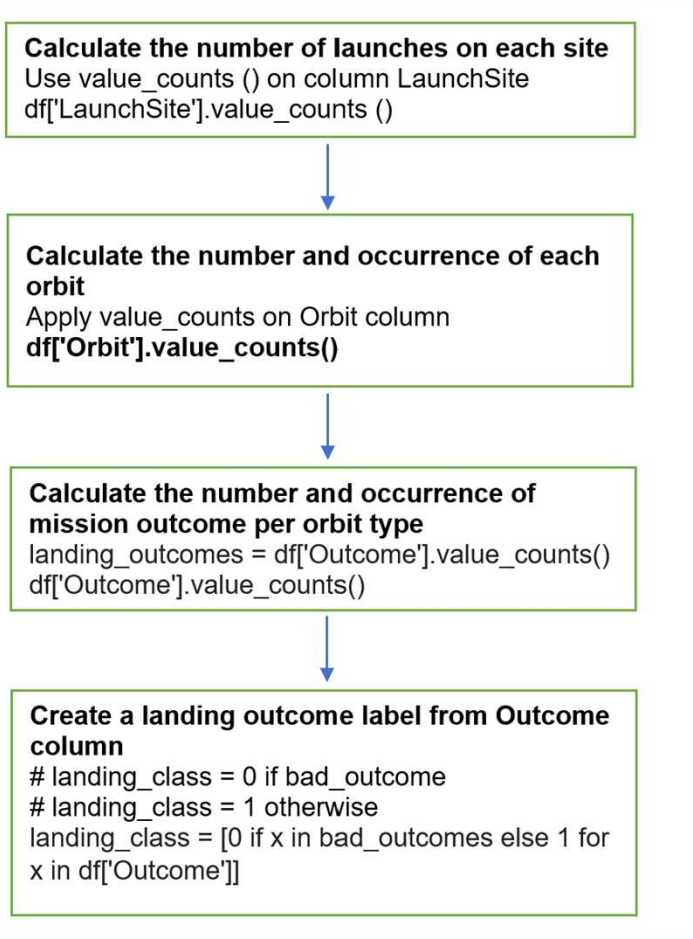
Data Wrangling

- Describe how data were processed

  Load Space X dataset, Identify and calculate the percentage of the missing values in each attribute, use value_counts() on column LaunchSite to calculate launches, occurrences and outcomes on each site and/or orbit, create landing outcome label 1 is success, 0 is fail

- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebook

  test_repository/Data Wrangling.ipynb at main · mygithub22021/test_repository



CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
  - 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. CCAFS LC-40 success 60 %, KSC LC-39A and VAFB SLC are 77%.
  - 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 value. Verify the Play load and sites.
  - Plot a bar chart on Orbit column and get the mean of Class. column to see success rate.
  - 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. Visualize the relationship between FlightNumber 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. Visualize the relationship between Payload and Orbit type.
  - Plot a line chart with x axis to be the extracted year and y axis to be the success rate for average success trend. Visualize the launch success yearly trend.
- Add the GitHub URL of your completed EDA with data visualization notebook

  test_repository/Exploratory Data Analysis with Visulization.ipynb at main · mygithub22021/test_repository

EDA with SQL

- **Using bullet point format, summarize the SQL queries you performed**
  - Download SPACEXDATASET to IBM DB2 database cloud, connect to DB2 dataset.
  - Write and execute SQL queries %SQL
  - Display launch sites, 5 records of KSC launch site
  - Display total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date where the first successful landing outcome in drone ship was achieved
  - List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes, the names of the booster_versions which have carried the maximum payload mass.
  - List the records which will display the month names, successful landing outcomes in ground pad
  - List booster versions, launch_site for the months in year 2017
  - Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.
- **Add the GitHub URL of your completed EDA with SQL notebook**
  - test_repository/Exploratory Data Analysis with SQL 12-20-2021.ipynb at main · mygithub22021/test_repository

Build an Interactive Map with Folium

- Created and added circles markers and lines to folium maps.
- Circles are for the locations to be found of launch sites on the map. Markers shows the launch site names and each site launch success or fail counts with color codes. Lines show the distances to the losest coast line and railway.
- GitHub URL of completed interactive map with Folium map

test_repository/Interactive Visual Analytics with Folium Lab (5).ipynb at main · mygithub22021/test_repository

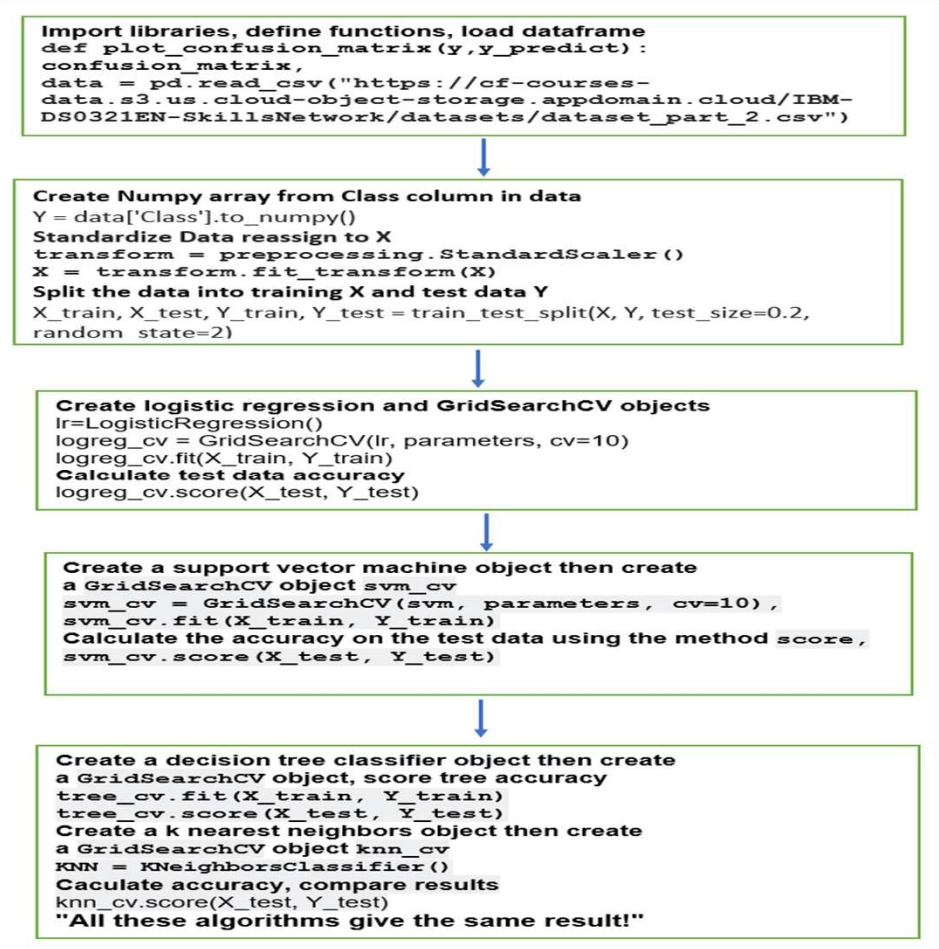
Build a Dashboard with Plotly Dash

- 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

Predictive Analysis (Classification)

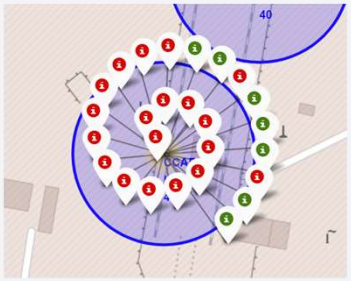
- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab

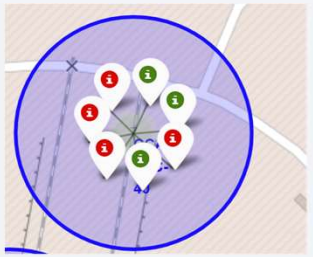
https://github.com/mygithub22021/test_repository/blob/main/Machine%20Learning%20Prediction%20Lab.ipynb

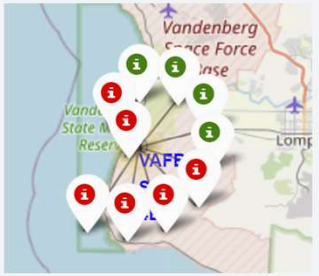


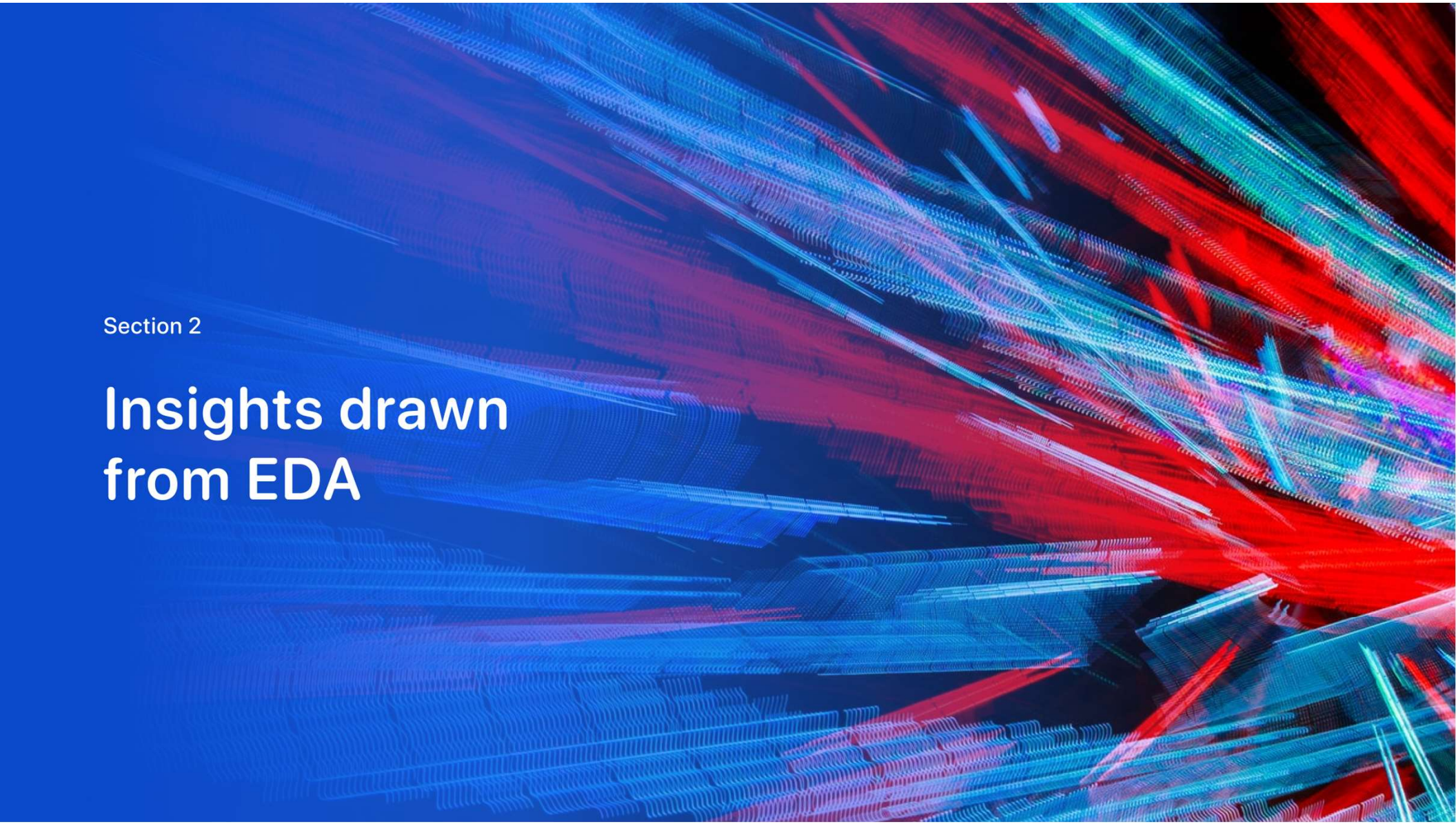
Results

- Exploratory data analysis results
  - **landing__outcome** **COUNT**
    - No attempt 7
    - Failure (drone ship) 2
    - Success (drone ship) 2
    - Success (ground pad) 2
    - Controlled (ocean) 1
    - Failure (parachute) 1
- Interactive analytics demo in screenshots
  - See left
- Predictive analysis results
  - All these algorithms give the same result!





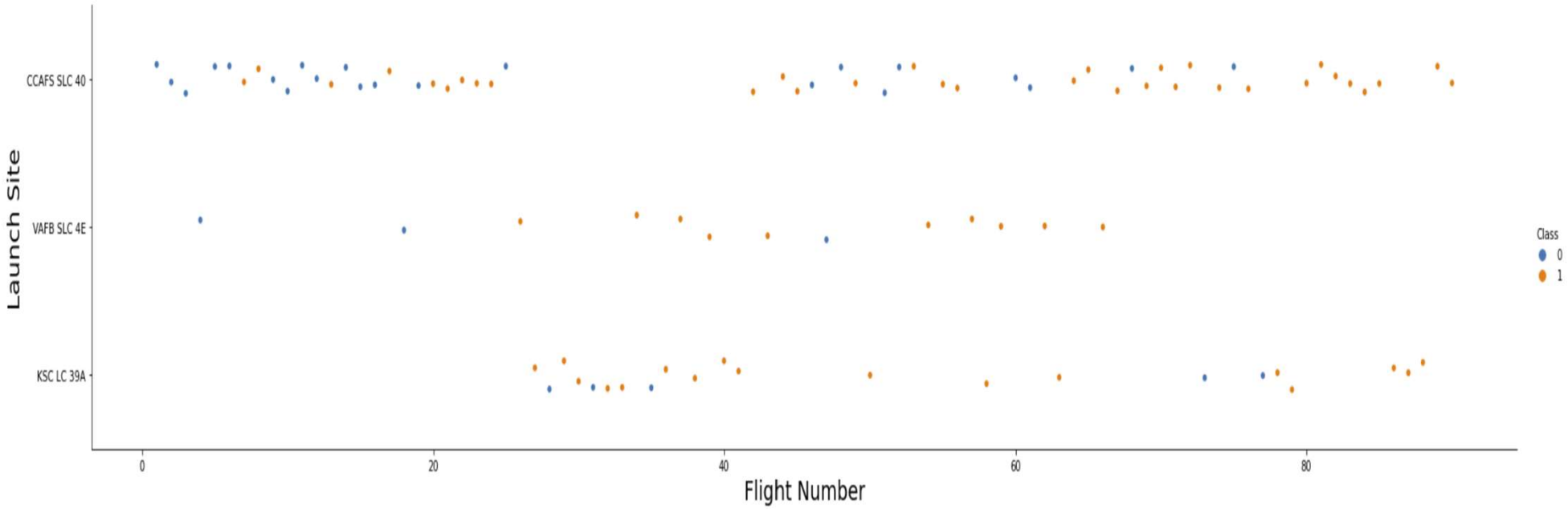




Section 2

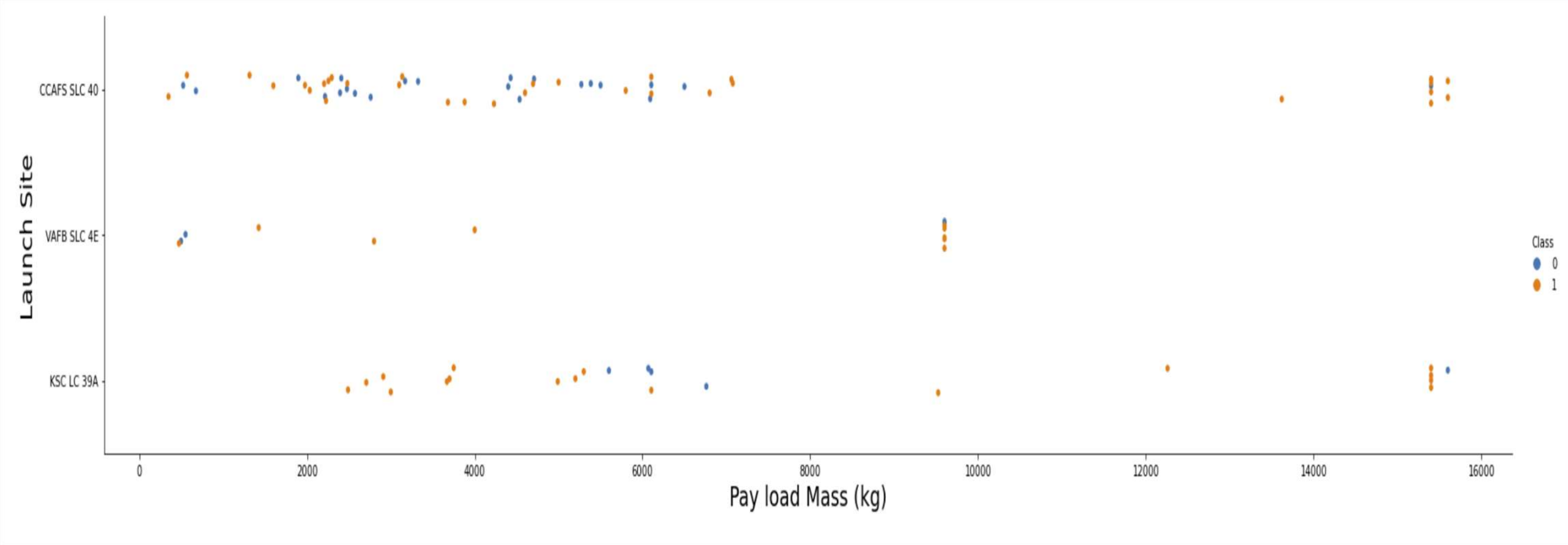
Insights drawn from EDA

Flight Number vs. Launch Site Scatter Plot



- **CCAFS LC-40, has a success rate of 60 %.**
- **KSC LC-39A and VAFB SLC 4E has a success rate of 77%.**

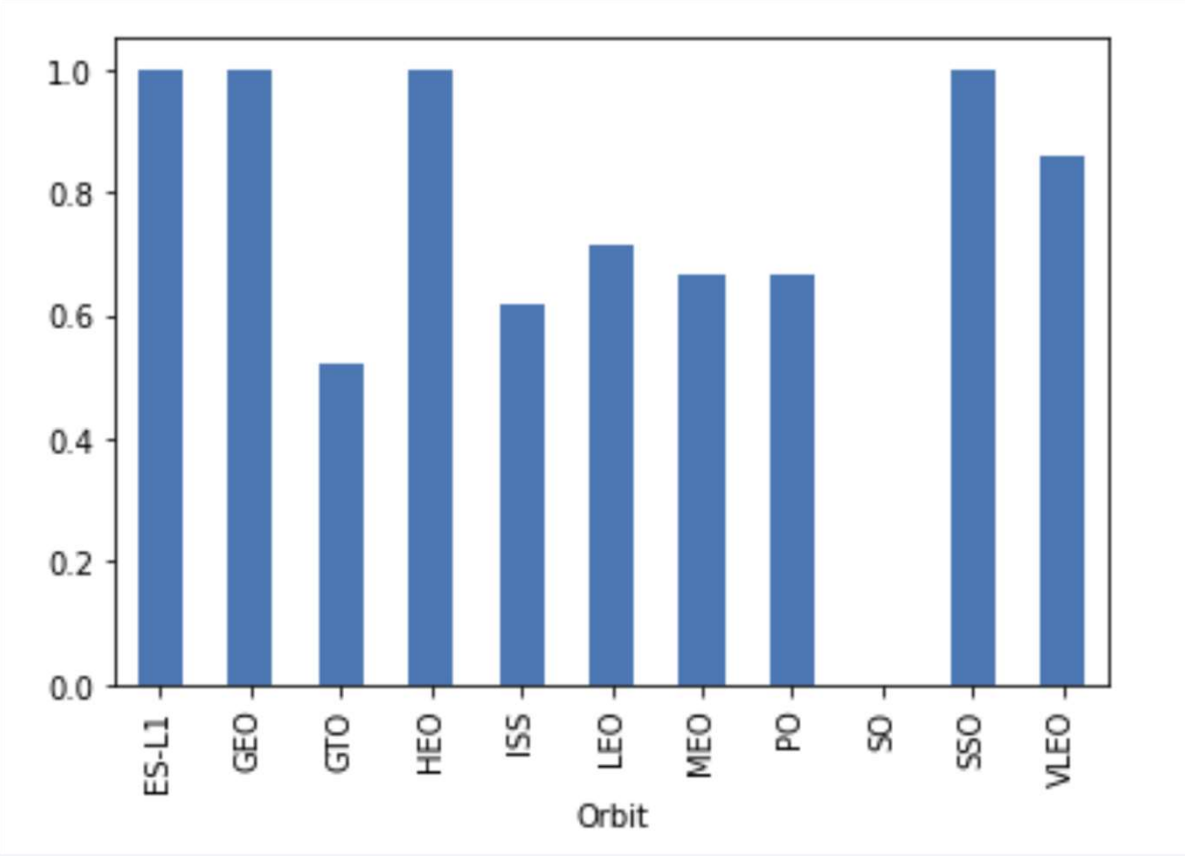
Payload vs. Launch Site



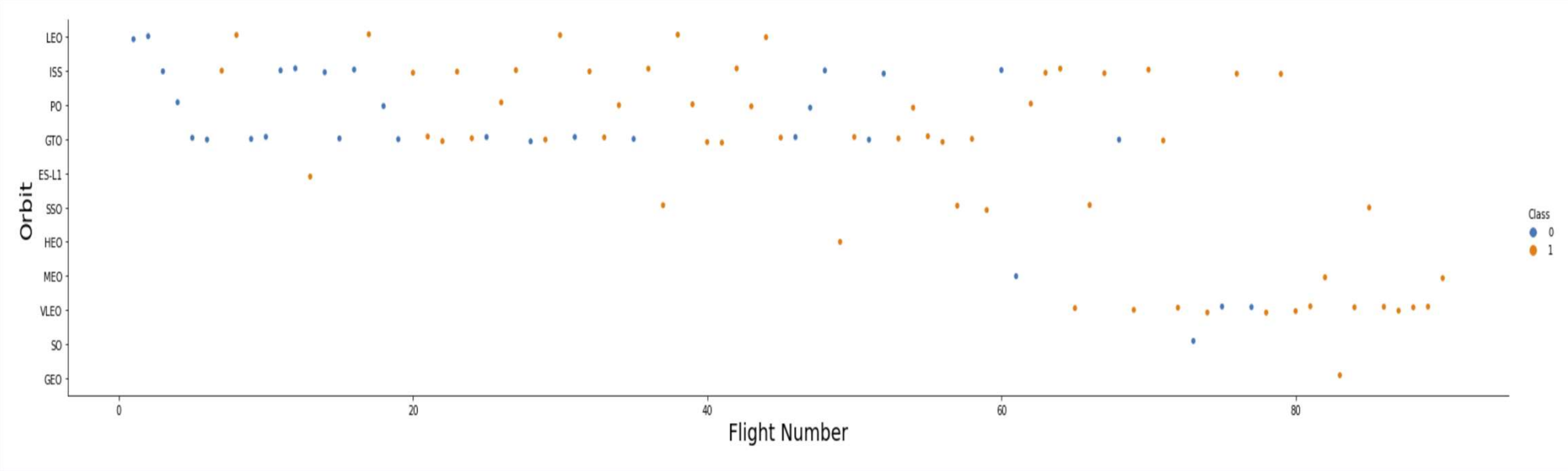
VAFB-SLC site did not launch for heavy payload mass greater than 10000kg.

Success Rate vs. Orbit Type

- Orbits ES-L1, GEO HEO and SSO have highest launch success rate of 100%.

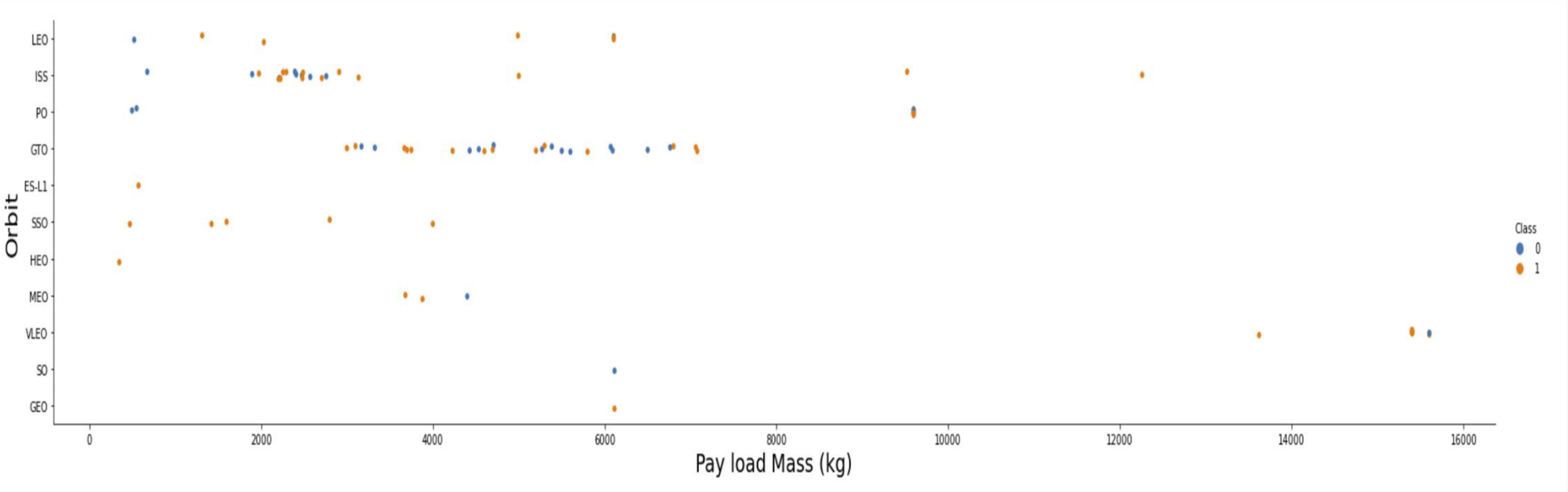


Flight Number vs. Orbit Type



LEO orbit landing success rate relates to the number of flights; GTO orbit does not have relationship between success rate and flight number.

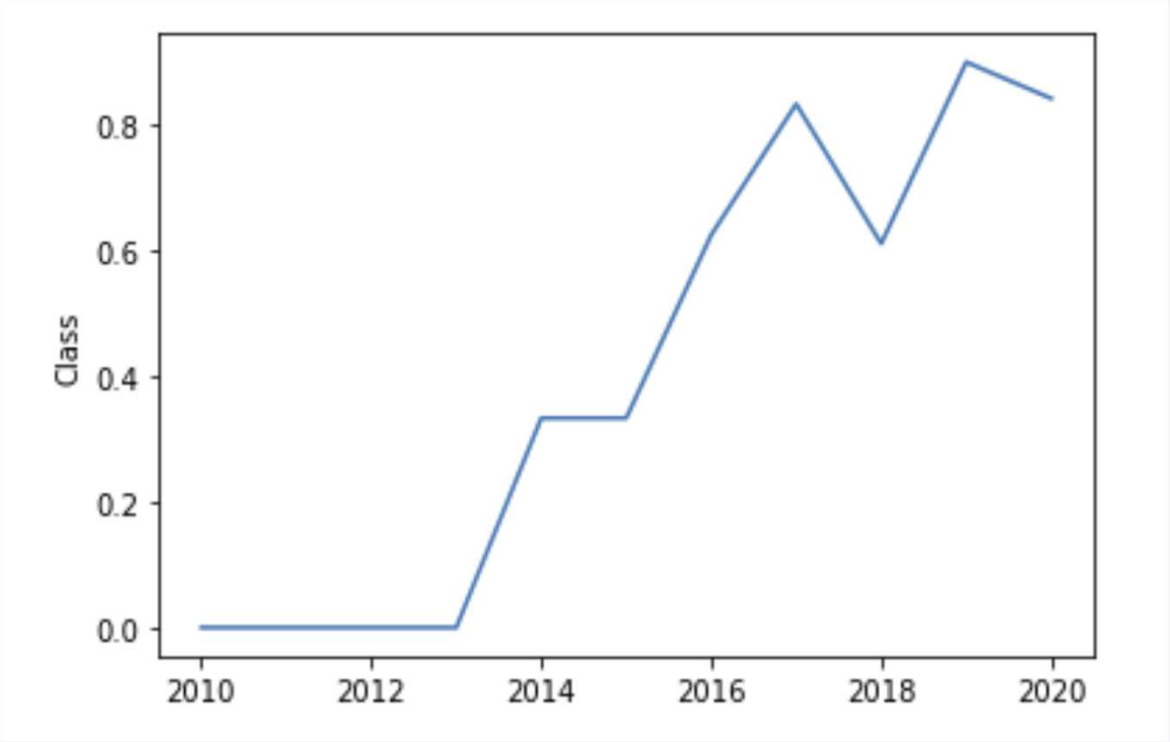
Payload vs. Orbit Type



With heavy payloads the successful landing rate are more for Polar, LEO and ISS orbits.
GTO orbit does not have relationship between success landing rate and payloads.

Launch Success Yearly Trend

Success rate since 2013 kept increasing, then stops in 2020.



All Launch Site Names

- Find the names of the unique launch sites
  - Launch sites are CCAFS LC-40, CCAFS SLC-40, KSC LC 39A and VAFB SLC 4E
- Present query result with a short explanation here
  - %%sql
  - select distinct(LAUNCH_SITE) from SPACEXDATATABLE
- Use %%sql format
- select LAUNCH_SITE from data that is stored in DB2 SPACEXTBL table.
- The output shows the LAUNCH SITES.

Launch Site Names Begin with 'CCA'

- Launch sites begin with CCA are CCAFS LC-40, CCAFS SLC-40

- %%sql
- SELECT * FROM SPACEXTABLE WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 50
  * bm_db_sa://wvx02184:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
- Use %%sql format, select data FROM db2 database SPACEXTBL table file with launch sith begin with CCA.

Total Payload Mass

```
numpy_array_PayloadMass = df['PayloadMass'].to_numpy()
Total_PayloadMass = sum(numpy_array_PayloadMass)
print('Total Payload Mass = ',Total_PayloadMass,'kg')
```

Total Payload Mass = 549446.3470588236 kg

- Convert df PayloadMass rows of data to an Array.
- Use sum() to add the data in the array.
- Print the sum of Payload Mass.

Average Payload Mass by F9 v1.1

```
%%sql
select avg(PAYLOAD_MASS__KG_) as "F9 v1.1_avg_payload_mass"
from SPACEXDATATABLE
where BOOSTER_VERSION = 'F9 v1.1'
group by BOOSTER_VERSION
```

- Average payload mass by F9 v1.1 is 3676 kg.
- Select average PAYLOAD_MASS_KG data by F9 v1.1 booster version

First Successful Ground Landing Date

```
%%sql
SELECT MIN(DATE) as " The first successful ground landing date"
FROM SPACEXTABLE
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

- The first successful ground landing date is 2017-01-05
- Use %%sql, select the minimum(the first) date of the success ground pad occurred. from SPACEXDATATABLE database file.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
select BOOSTER_VERSION,PAYLOAD_MASS__KG_,LANDING__OUTCOME from SPACEXDATATABLE
where PAYLOAD_MASS__KG_ between 4000 and 6000
and LANDING__OUTCOME = 'Success (drone ship)'
```

booster_version	payload_mass__kg_	landing__outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

- Select data in BOOSTER_VERSION, PAYLOAD_MASS_KG, LANDING_OUTCOME, use **where** to filter payload mass 4000 - 6000 kg landing outcome ‘success drone drop’ data.

Total Number of Successful and Failure Mission Outcomes

```
%%sql
select MISSION_OUTCOME, count(*) as count from SPACEXDATATABLE
group by MISSION_OUTCOME
*ibm_db_sa://csg08221:***@b0aebb68-94fa-46ec-a1fc-
1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bl
udb Done.
```

MISSION_OUTCOME

Success 44 Failure 1 (payload unclear)

- Select MISSION_OUTCOME from SPACEXTATABLE group outcome.

Boosters Carried Maximum Payload

```
%%sql
select BOOSTER_VERSION,PAYLOAD_MASS__KG_ from SPACEXDATATABLE
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXDATATABLE)
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600

- Select BOOSTER_VERSION, PAYLOAD_MASS_KG from SPACEXTABLE data file, filtered by maximum PAYLOAD_MASS_KG.

2017 Launch Records

```
%%sql
select  BOOSTER_VERSION,LAUNCH_SITE from SPACEXDATATABLE
where LANDING__OUTCOME = 'Success (ground pad)'
and year(DATE) = '2017’
```

DATE	booster_version	launch_site
2017-01-05	F9 FT B1032.1	KSC LC-39A
2017-03-06	F9 FT B1035.1	KSC LC-39A
2017-07-09	F9 B4 B1040.1	KSC LC-39A

- Select BOOSTER_VERSION, LAUNCH_SITE from SPACEXDATATABLE file
- Filter out LANDINF_OUTCOME is successful on ground pad and 2017 year-date.

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

```
%%sql
select DATE, LANDING__OUTCOME,count(*) as count
from SPACEXDATATABLE
where DATE between '2010-06-04'and '2017-03-20'
group by LANDING__OUTCOME
order by count desc
```

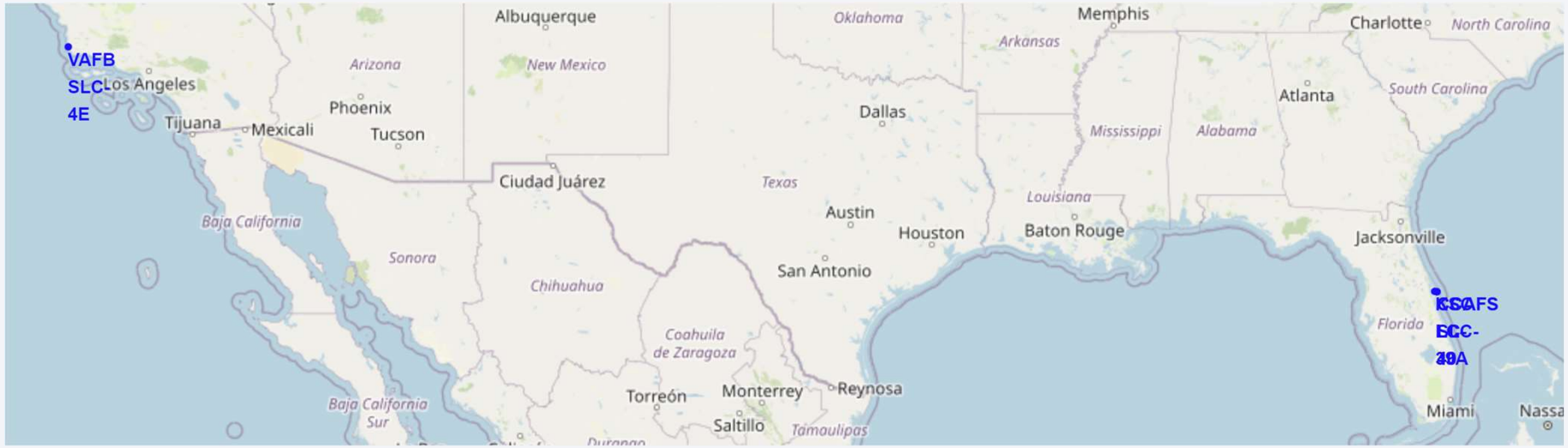
- Select DATE, LANDING_OUTCOME, COUNT() FROM SPACEXTABLE, filter dates 2010-6-4 to 2017-3-20 for landing outcome.

DATE	landing_outcome
2017-02-19	Success (ground pad)
2017-01-14	Success (drone ship)
2016-08-14	Success (drone ship)
2016-07-18	Success (ground pad)
2016-05-27	Success (drone ship)
2016-05-06	Success (drone ship)
2016-04-08	Success (drone ship)
2015-12-22	Success (ground pad)

Section 4

Launch Sites Proximities Analysis

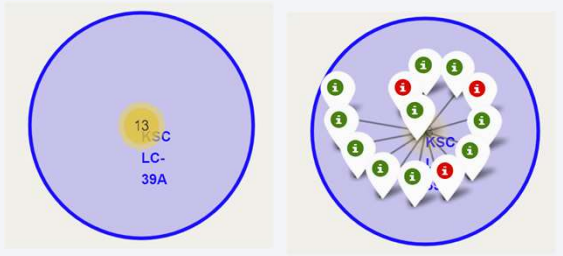
Mark all launch sites on a Folium map



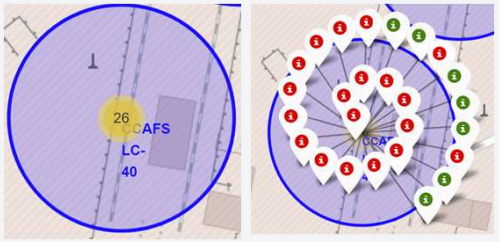
- On the Folium map, viewers can click in the launch sites, and find all four sites in USA. These sites have circles and markers for locations and names.
- All sites are close to coast lines, 1 at California, 3 at Florida.
- All site are close to equator line.

Mark the success/failed launches for each site on the map

California VASB SLC-4E



Florida KSC LC-39A

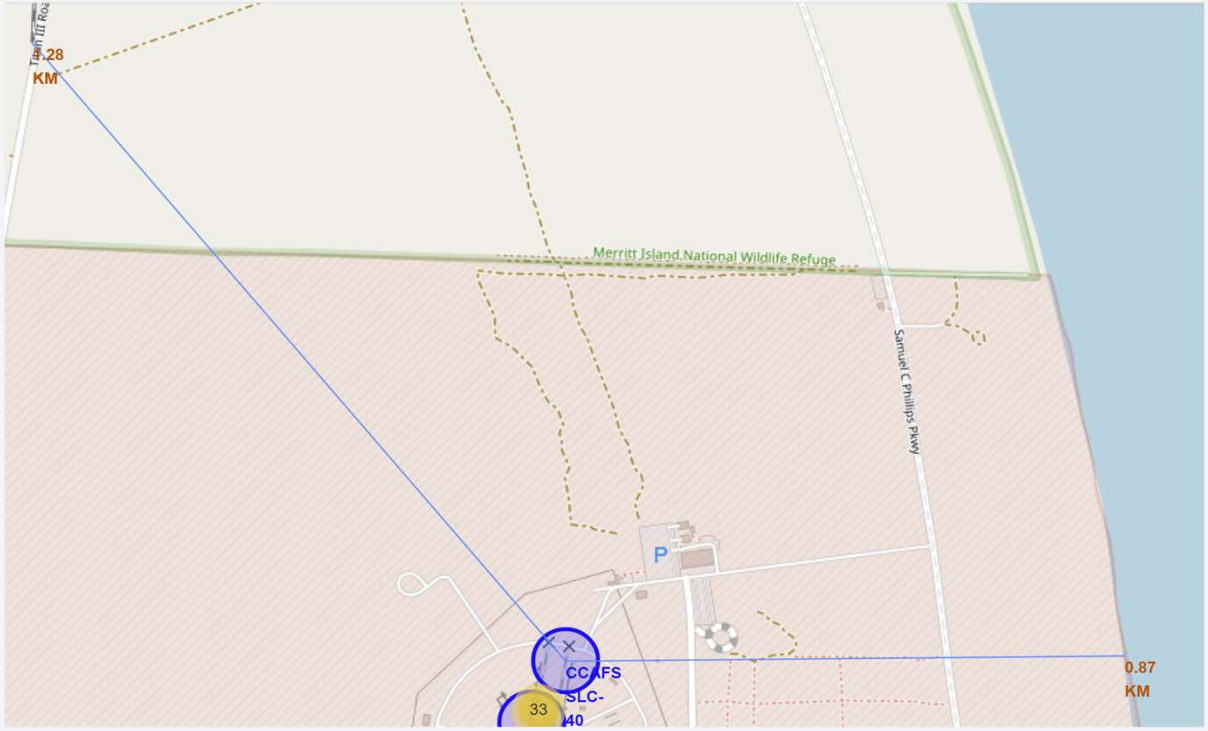


Florida CCAFS LC-40

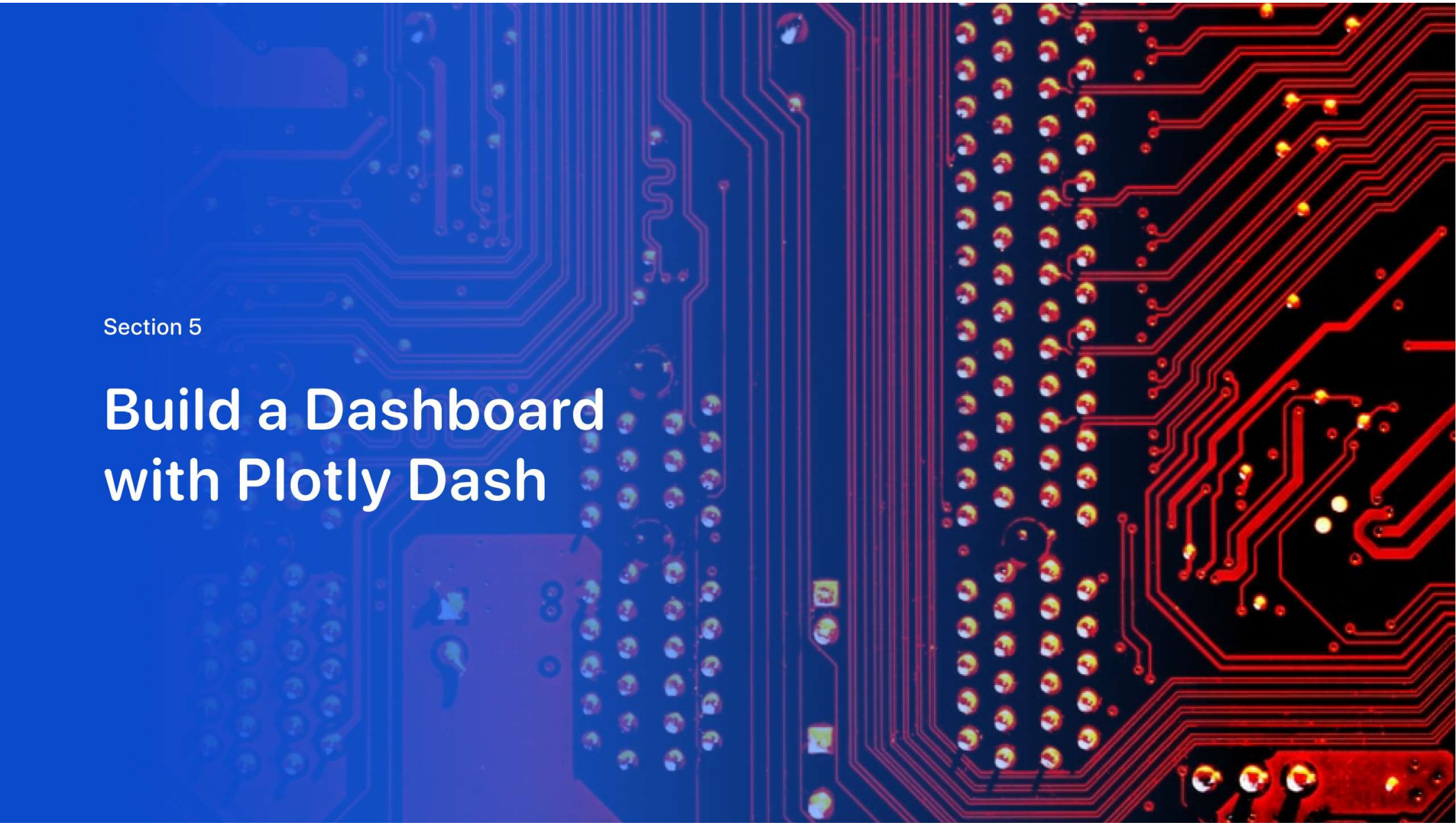
Florida CCAFS LC-40

- Mark 1 California and Florida sites for total launch counts and each success/failed with color codes. Green is success, red is failed.

Calculate the distances between a launch site to its proximities: the closest coastline and railway



- Calculate launch site Florida CCAFS SLC-40 to the closet distance to coastline is 0.87 KM, to railway is 1.28 KM. Folium map can find the information very effectively.



Section 5

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 pie chart
- 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.

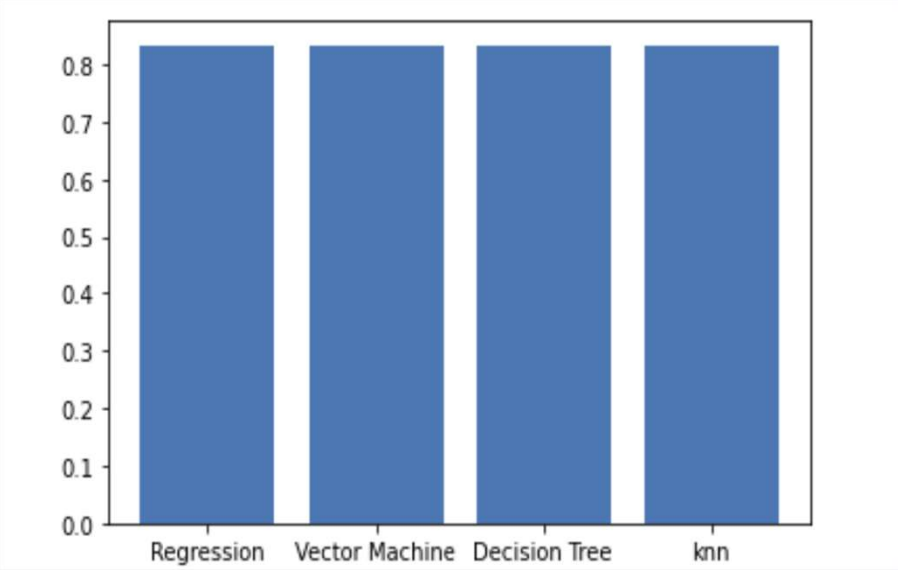
Section 6

Predictive Analysis (Classification)

Classification Accuracy

- Test data accuracies are practically the same.
- Use score for test data accuracy calculation
- logreg_cv.score(X_test, Y_test) svm_cv.score(X_test, Y_test) knn_cv.score(X_test, Y_test) tree_cv.score(X_test, Y_test)
- Create dictionary and plot bar chart

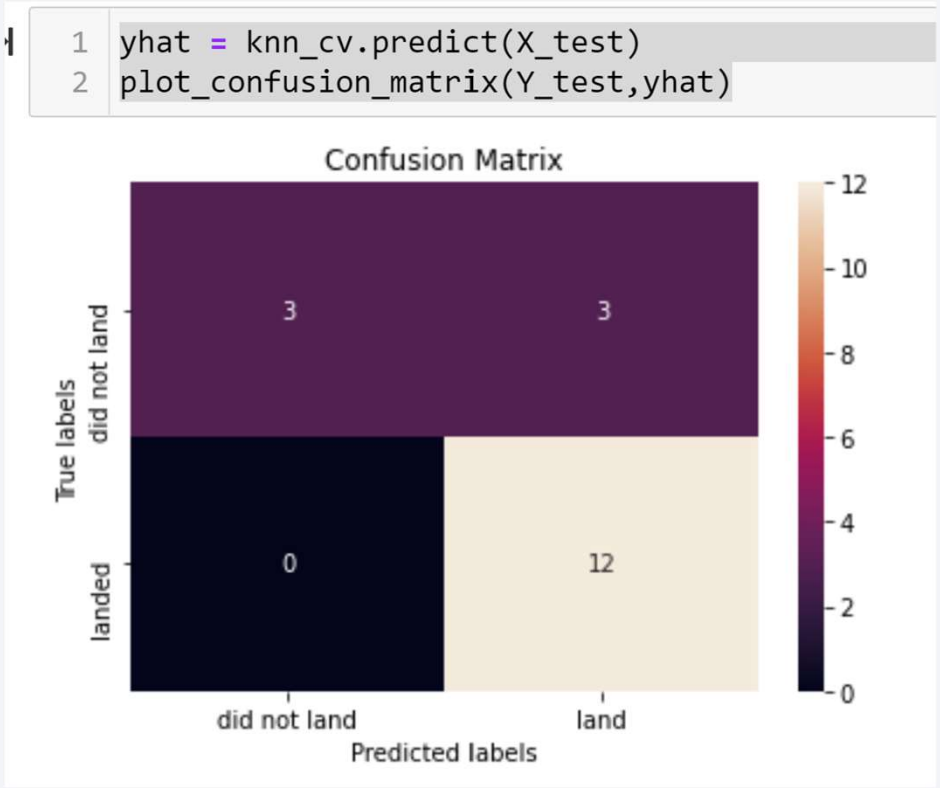
```
import matplotlib.pyplot as plt
test_data_accuracy = {'Regression': 0.8333333333333334
, 'Vector Machine': 0.8333333333333334,'Decision Tree': 0.8333333333333334, 'knn': 0.8333333333333334 }

names = list(test_data_accuracy.keys())
values = list(test_data_accuracy.values())

plt.bar(range(len(test_data_accuracy)), values, tick_label=names)
plt.show()
```



Confusion Matrix

- The test data accuracy is the same.
- Confusion matrix shows
- yhat = knn_cv.predict(X_test)
- plot_confusion_matrix(Y_test,yhat)
- Predict 3 did not land and 12 land correctly. The accuracy is 15/18 = 0.833333, the same as score calculation.

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



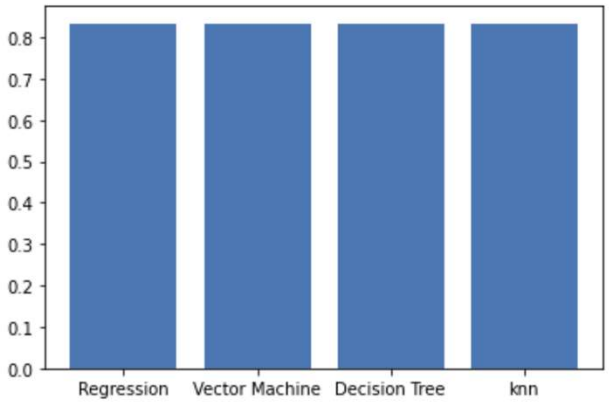
Conclusions

- IBM Watson Studio, Db2, Folium, Plotly dash are helpful.
- Data Science technologies are powerful.
- Set up environments is important and challenging for various tools.

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

```python
# Plot bar chart with test data accuracy from different built predictive models

import matplotlib.pyplot as plt

test_data_accuracy = {'Regression': 0.8333333333333334
, 'Vector Machine': 0.8333333333333334,'Decision Tree': 0.8333333333333334,'knn':0.8333333333333334 }
names = list(test_data_accuracy.keys())
values = list(test_data_accuracy.values())

plt.bar(range(len(test_data_accuracy)), values, tick_label=names)
plt.show()
```



Thank you!