Winning Space Race with Data Science

Wai Tak Wong
May 15, 2022

Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

Executive Summary

- Summary of methodologies
  - Data collection methodology
  - Perform data wrangling
  - EDA with SQL
  - EDA with Data Visualization
  - Build an Interactive Map with Folium
  - Build 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

- Background
  - 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 is because SpaceX can reuse the first stage.
  - In this project, we'll examine the landing result of SpaceX and determine the cost of a launch to verified what SpaceX claimed.
  - The cost of a rocket launch from SpaceX can be a reference to determine whether an alternate company is worthy to bid against SpaceX or not.
- Target Issues
  - The factors that the rocket landed successfully
  - What conditions that make the landing a better result.

Section 1

Methodology

Methodology

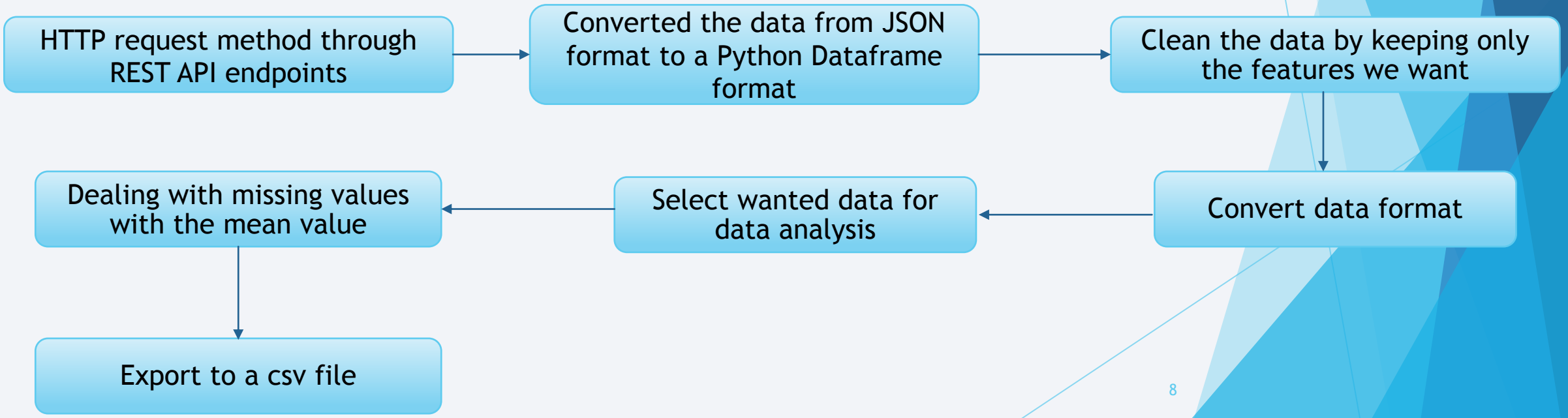
- Perform data collection by using Web Scraping technology.
- Perform data wrangling to find the mission outcome for each launch and label the outcome result.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Using category plot, bar chart, scatter chart and line chart to show the correlation and pattern of the data.
- Perform interactive visual analytics using Folium and Plotly Dash
  - Using Folium to mark each launch site with success and failure count on a map. Mark and measure the launch site to city, highway and railway.
  - Using Plotly Dash to show the success and failure situation for the launch site.
- Perform predictive analysis using classification models
  - Show how to build, tune, evaluate classification models

Data Collection

- Data collection methodology:
  - The data is retrieved by HTTP request method through REST API endpoints provided by SpaceX. The response is in JSON format.
  - The data is converted from JSON format to a Python Dataframe format by using the json_normalize() function.
  - Clean the data
    - Convert data format
    - Select wanted data for data analysis
    - Dealing with missing values with the mean value
  - Export the data to a file named ‘dataset_part_1.csv’

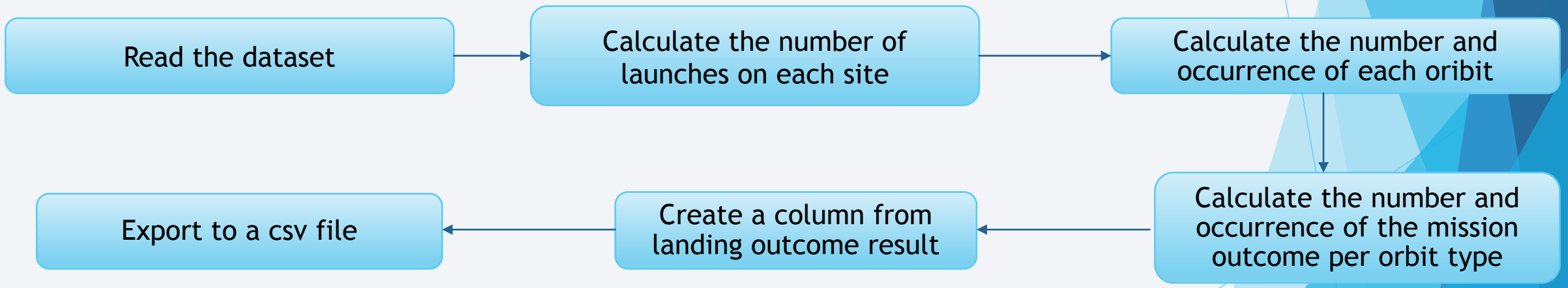
Data Collection – SpaceX API

- GitHub URL of the completed SpaceX web scraping API calls notebook
  - https://github.com/wtwong316/ads_capstone/blob/master/jupyter-labs-spacex-data-collection-api.ipynb
- Web scraping process flowcharts



Data Wrangling

- GitHub URL of the completed data wrangling process notebook
  - https://github.com/wtwong316/ads_capstone/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb
- Web scraping process flowcharts



Data Wrangling

- How data was processed
  - Using value_counts() function of DataFrame and apply to the column ‘LauchSite’, we see that CCAFS LC-40, has 55 launch count.
  - Using the same method and apply to the column ‘Oribit’, we see that GTO has 27 launch count
  - Using the same method and apply to the column ‘Outcome’, we see that the column is a combination of outcome and mission. Outcomes marked ‘None’ and ‘Failure’ mean not successful.
  - We create a column, named landing class, and set the value to 1 for successful launch and 0 for others.
  - Export the file to ‘dataset_part_2.csv’

EDA with SQL

- GitHub URL of the completed EDA with SQL notebook
  - https://github.com/wtwong316/ads_capstone/blob/master/eda-sql.ipynb
- Summary of the SQL queries performed
  - 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 ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

EDA with SQL

- Summary of the SQL queries performed (cont'd)
  - List the total number of successful and failure mission outcome
  - List the names of the booster_versions which have carried the maximum payload mass.
  - List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 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

EDA with Data Visualization

- GitHub URL of the completed EDA with data visualization notebook
  - https://github.com/wtwong316/ads_capstone/blob/master/jupyter-labs-eda-dataviz.ipynb
- Summary of the charts being used to plot
  - Category plot for flight number versus launch site and overlay the outcome of the launch to see the correlation between the launch site and flight number for successful launch.
  - Category plot for payload mass versus launch site and overlay the outcome of the launch to see the correlation between the launch site and pay load mass for successful launch
  - Bar chart for success rate of each orbit to find the higher successful launch

EDA with Data Visualization

- Summary of the charts being used to plot (cont'd)
  - Scatter plot for the flight number and each orbit to see the correlation between the orbit and the flight number for successful launch.
  - Scatter chart for the payload mass and each orbit to see the correlation between the orbit and pay load mass for successful launch.
  - Line chart for the yearly successful launch trend to see the yearly trend of successful launch

Build an Interactive Map with Folium

- GitHub URL of the completed Interactive Map with Folium notebook
  - https://github.com/wtwong316/ads_capstone/blob/master/lab_jupyter_launch_site_location.ipynb
- Summary of the created map objects added to a folium map
  - A Circle map object to add a highlighted circle area on a specific coordinate for each launch site.
  - A mark map object to specify the text label on a specific coordinate for each launch site.
  - A cluster of mark map object with an icon to specify the success and failure of the launch.

Build an Interactive Map with Folium

- Summary of the created map objects added to a folium map (cont'd)
  - A mark map object to specify the distance measurement to the proximities such as city, railway, highway and coastline
  - A PolyLine map object to draw a line from the launch site to the proximities

Build a Dashboard with Plotly Dash

- GitHub URL of the completed Dashboard with Plotly notebook
  - https://github.com/wtwong316/ads_capstone/blob/master/spacex_dash_app.py
- Summary of the plots/graphs and interactions added to the dashboard
  - A Circle

Predictive Analysis (Classification)

- GitHub URL of the completed Dashboard with Plotly notebook
  - https://github.com/wtwong316/ads_capstone/blob/master/SpaceX_Machine Learning Prediction_Part_5.ipynb
- Summary of how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart

Section 2

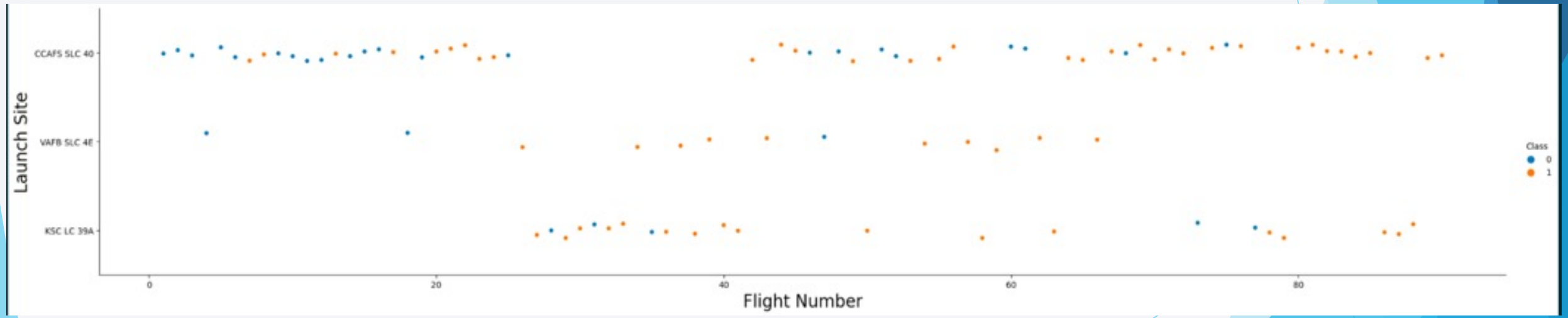
Insights drawn from EDA

EDA With Visualization Results

- A scatter plot of Flight Number vs. Launch Site
- A scatter plot of Payload vs. Launch Site
- A bar chart for the success rate of each orbit type
- A scatter plot of Flight number vs. Orbit type
- A scatter plot of Payload mass vs. Orbit type
- A line chart of yearly average success rate

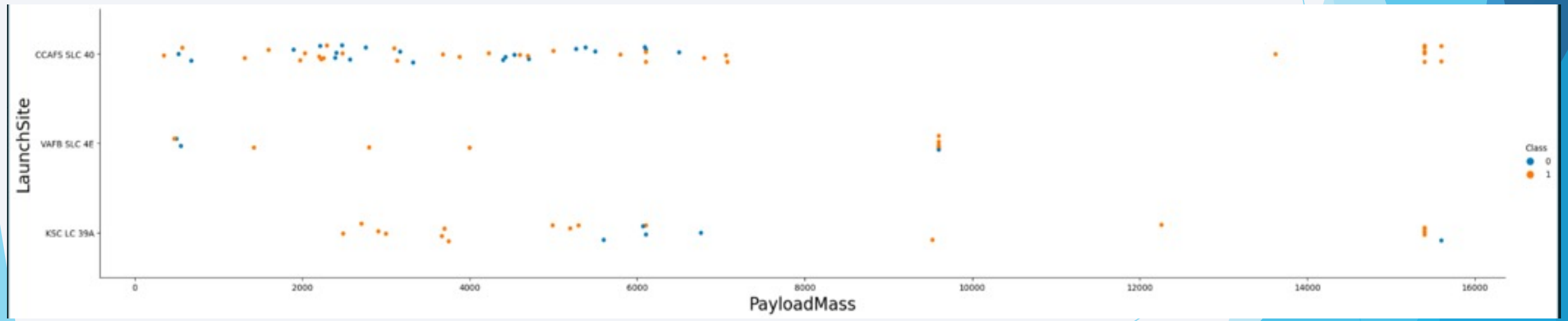
Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- Explanation of the scatter plot
  - The higher the flight number, the higher the successful probability in each launch site.



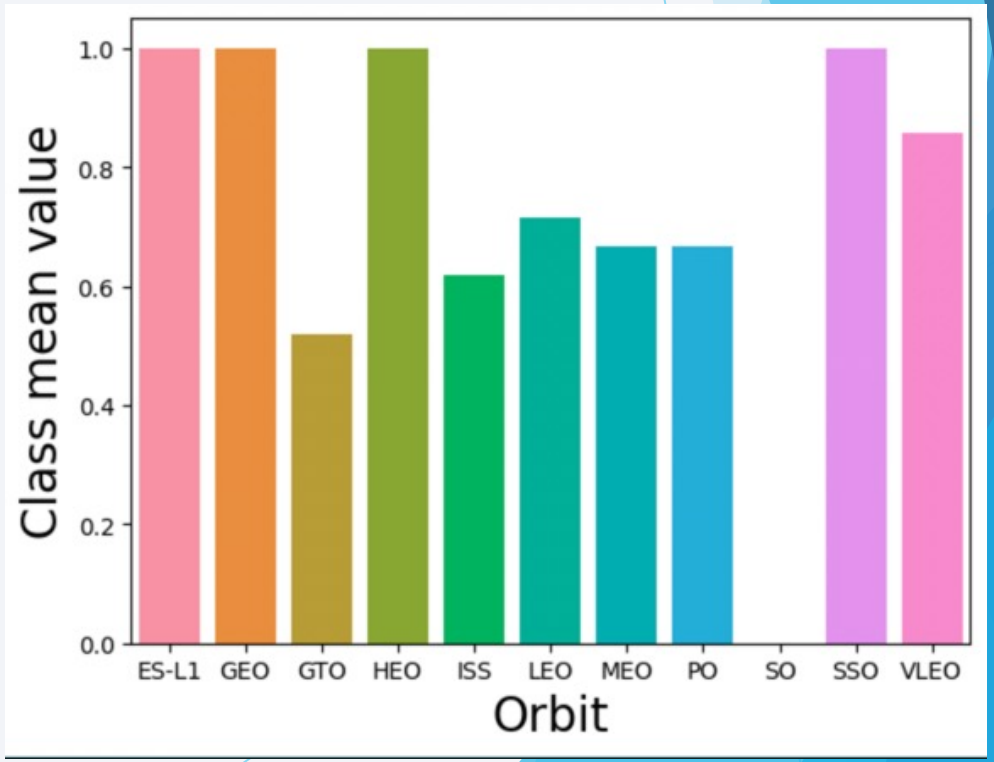
Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- Explanation of the scatter plot
  - The higher the flight number, the higher the successful probability in each launch site.
  - In launch site VAFB-SLC, there are no rockets launched for heavy payload mass (greater than 10000).



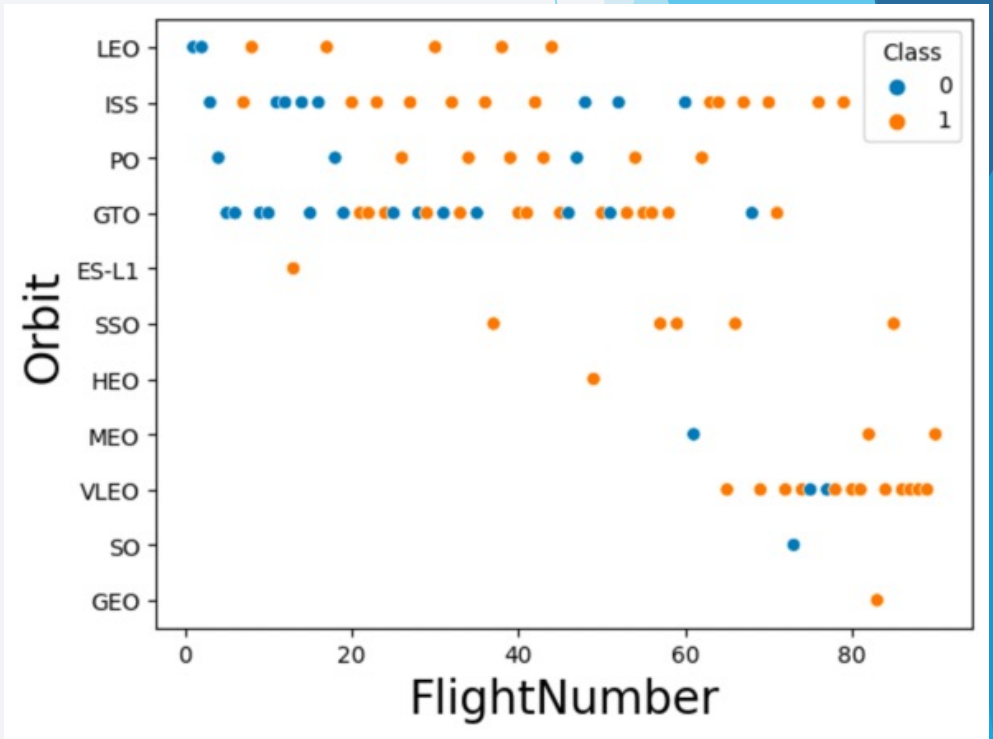
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- Explanation of the bar chart
  - Orbit ES-L1, GEO, HEO and SSO has 100% success rate.
  - Orbit SO has no successful launch



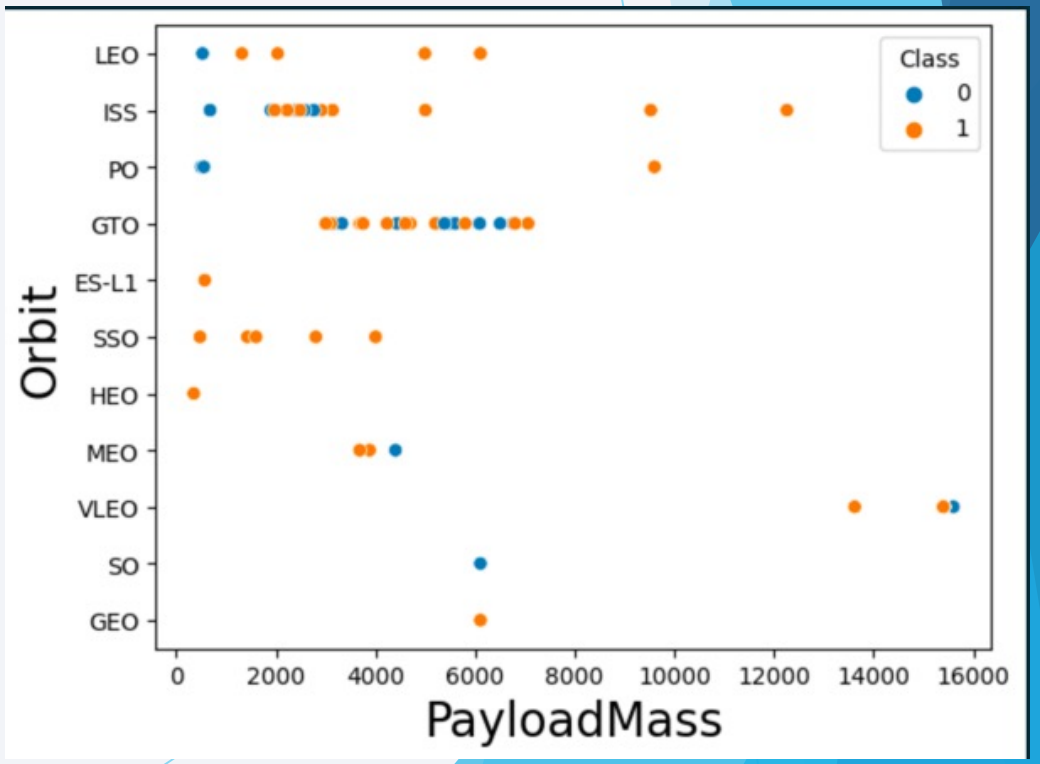
Flight Number vs. Orbit Type

- Show a scatter plot of Flight number vs. Orbit type
- Explanation of the scatter plot
  - The higher the flight number, the higher the successful probability in each orbit except orbit GTO and SO.



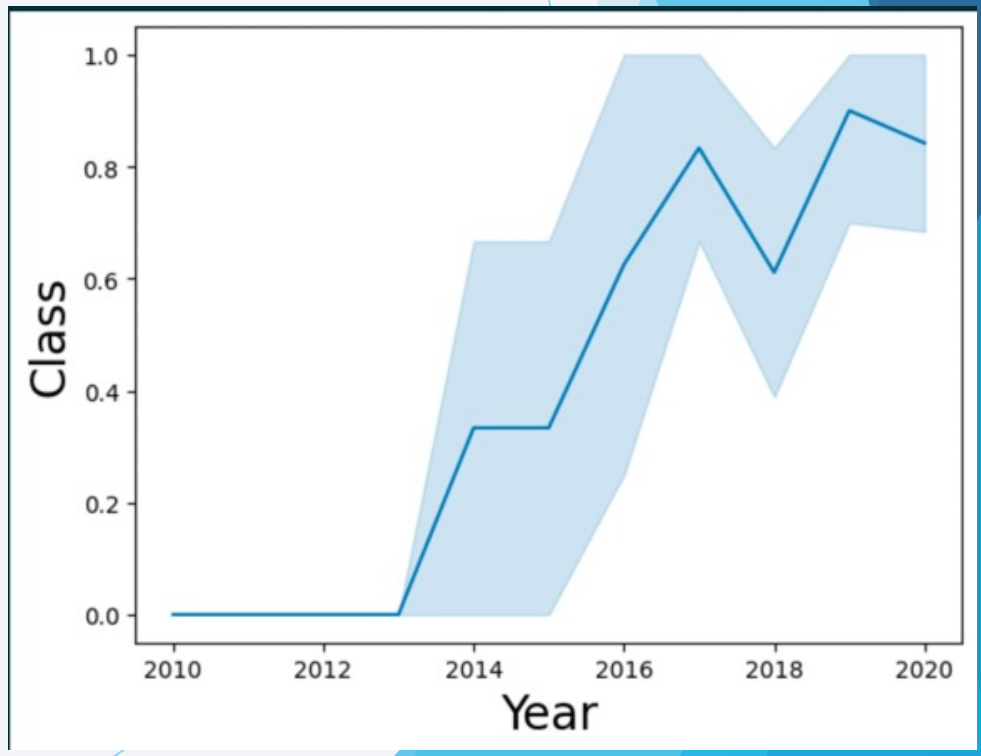
Payload Mass vs. Orbit Type

- Show a scatter plot of Payload mass vs. Orbit type
- Explanation of the scatter plot
  - With higher payloads. the successful landing or positive landing rate are more for orbit PO, LEO, and ISS.
  - For orbit GTO, we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) occurred randomly when payload mass is increased.



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Explanation of the line chart
  - The successful launch rate since 2013 kept increasing till 2017.
  - The successful launch rate is up and down during 2018 and 2020



EDA SQL Results

- Find the names of the unique launch sites
- Find 5 records where launch sites begin with `CCA`
- The total payload carried by boosters from NASA CRS
- The average payload mass carried by booster version F9 v1.1
- First Successful Ground Landing Date
- Successful Drone Ship Landing with Payload between 4000 and 6000
- Total Number of Successful and Failure Mission Outcomes
- Boosters Carried Maximum Payload
- List the failed landing_outcomes in drone ship
- Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

All Launch Site Names

- Find the names of the unique launch sites
- Explanation of the query result
  - There are four unique launch sites which are CCAFSLC-40, VAFB SLC-4E, KSC LC-39A and CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Explanation of the query result
  - All 5 records listed are from the Launch sit CCAFSLC-40

Task 2

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

```
%sql SELECT * FROM SPACEXTBL where LAUNCH_SITE Like '%CCA%' limit 5
```

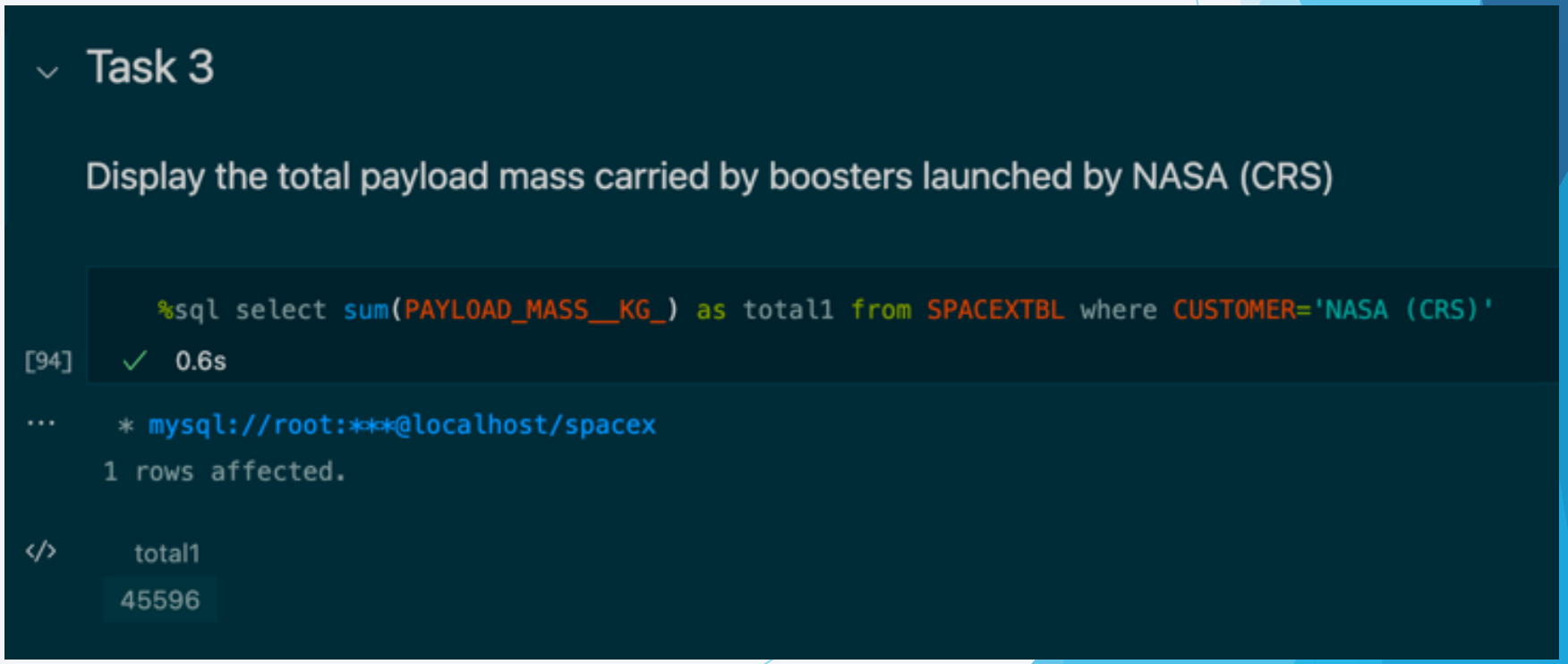
[43] ✓ 0.2s

```
* mysql://root:***@localhost/spacex
5 rows affected.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAY
4/6/10	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	
8/12/10	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	
22-05-2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	
8/10/12	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	
1/3/13	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	

Total Payload Mass

- Calculate the total payload carried by boosters from NASA CRS
- Explanation of the query result
  - The total payload mass launched from NASA CRS is 45596 KG



Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Explanation of the query result
  - The average payload mass carried by booster version F9 v1.1 is 14642 KG

Task 4

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

```
%sql select sum(PAYLOAD_MASS__KG_) as total2 from SPACEXTBL where BOOSTER_VERSION='F9 v1.1'
```

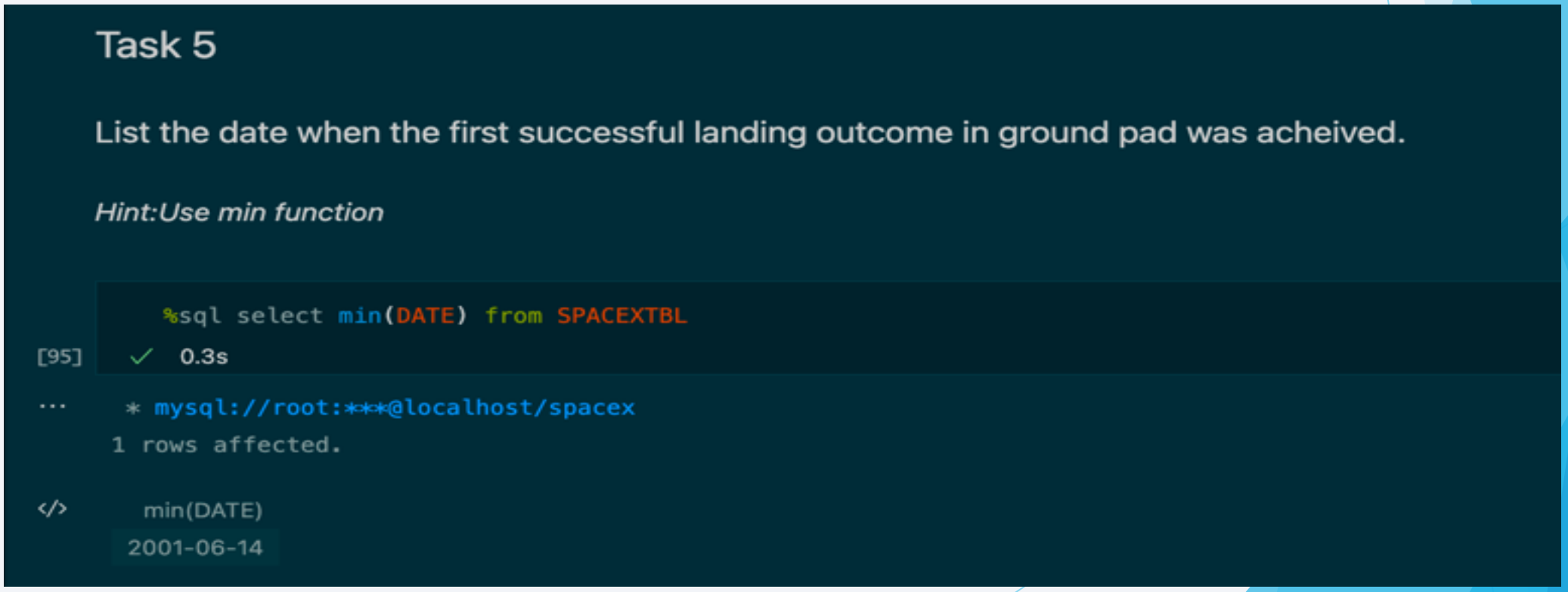
[45] ✓ 0.6s

```
* mysql://root:***@localhost/spacex
1 rows affected.
```

total2
14642

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Explanation of the query result
  - The first successful landing date is 2001-06-14



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

- The higher the flight are 6 booster version for the conditions above

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 from SPACEXTBL where PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 60000 and LANDING_OUTCOME='Success (d
```

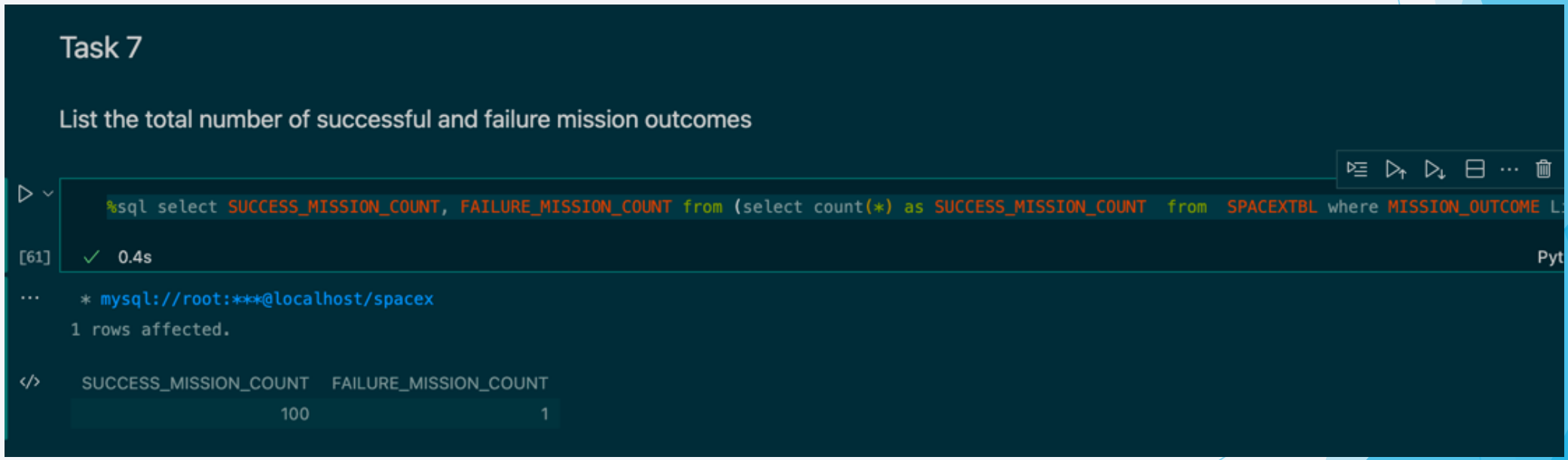
[98] ✓ 0.6s

```
 * mysql://root:***@localhost/spacex
7 rows affected.
```

BOOSTER_VERSION
F9 FT B1022
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1036.1
F9 B4 B1041.1
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- There are 100 Successful mission and 1 failure mission



Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- There are 12 Booster Version carried the same maximum payload mass

Task 8

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

```
%sql select * from (select Booster_Version, MAX(PAYLOAD_MASS__KG_) AS MAX_PAYLOAD_MASS  FROM SPACEXTBL GROUP BY Booster_Version  ORDER BY MAX_PAYLO
```

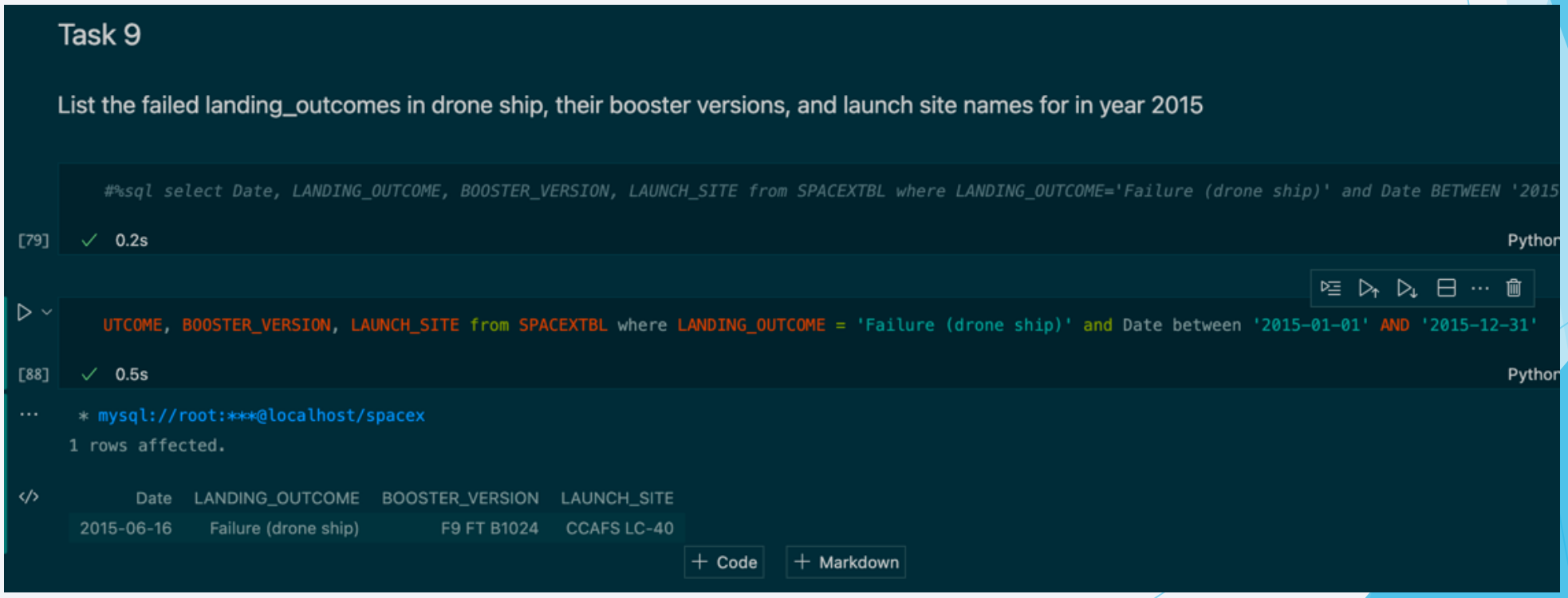
[104] ✓ 0.8s Python

```
* mysql://root:***@localhost/spacex
12 rows affected.
```

Booster_Version	MAX_PAYLOAD_MASS
F9 B5 B1048.4	15600
F9 B5 B1049.7	15600
F9 B5 B1048.5	15600
F9 B5 B1060.3	15600
F9 B5 B1051.6	15600
F9 B5 B1058.3	15600
F9 B5 B1051.4	15600
F9 B5 B1056.4	15600
F9 B5 B1060.2	15600
F9 B5 B1051.3	15600
F9 B5 B1049.5	15600
F9 B5 B1049.4	15600

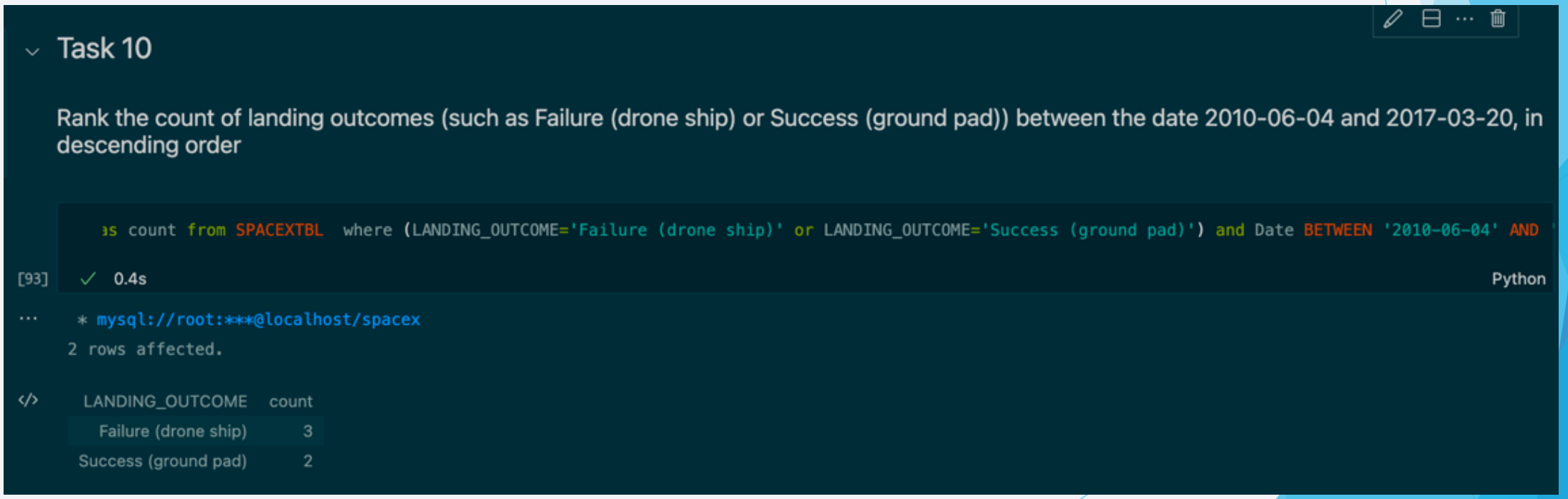
2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- There are 1 failed landing_outcomes in drone ship in year 2015



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
- There are 3 Failed launches of drone ship and 2 successful ground pad



Section 3

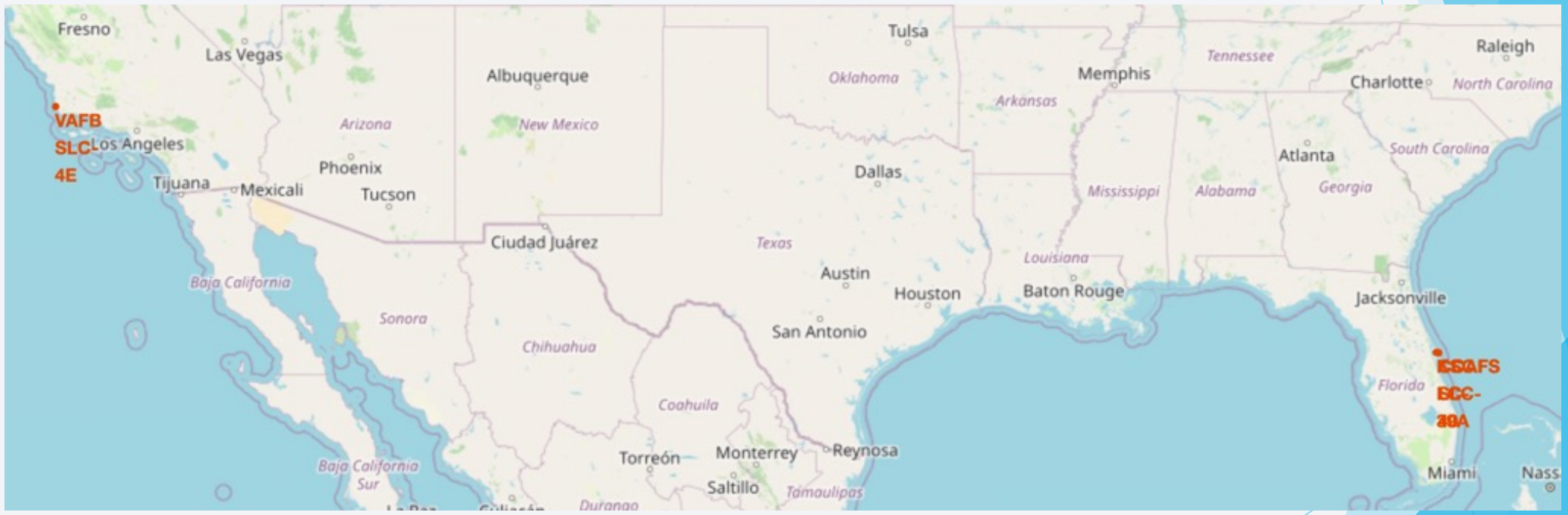
Launch Sites Proximities Analysis

Interactive Map with Folium Results

- All launch site's location
- Finding on the launch site's location
- Launch site location with total count, success and failure launch count
- Success and Failure Launch Statistics on Launch Site
- Launch site and its proximities
  - Coastline
  - City
  - Railway
  - Highway

Launch Site Location Analysis with Folium

- All launch site's location



Launch Site Location Analysis with Folium

- Finding on the launch site's location
  - Launch site KSC LC-39, CCAFS LC-40, CCAFS SLC-40 are quite near
  - Launch site CCAFS LC-40 and CCAFS SLC-40 are almost overlapped

Success and Failure Launch Statistics on Launch Site

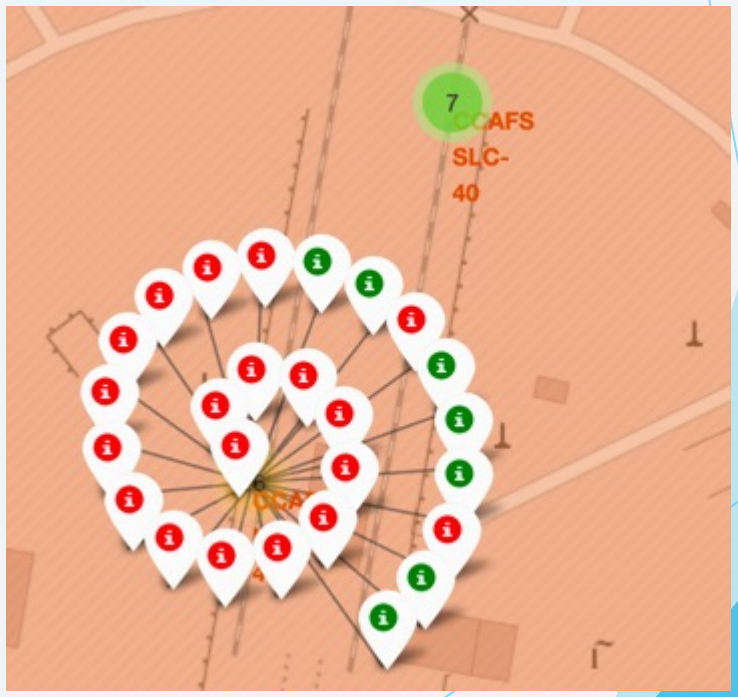
- Launch site location with total count, success and failure launch count

Success and Failure Launch Statistics on Launch Site

- Finding on the launching number with the launch site location
  - Eastern launch site got more launches

Success and Failure Launch Statistics on Launch Site

- Finding on the launching result with the launch site location (cont'd)
  - Launch site CCAFS LC-40 has 26 launches which is the most.

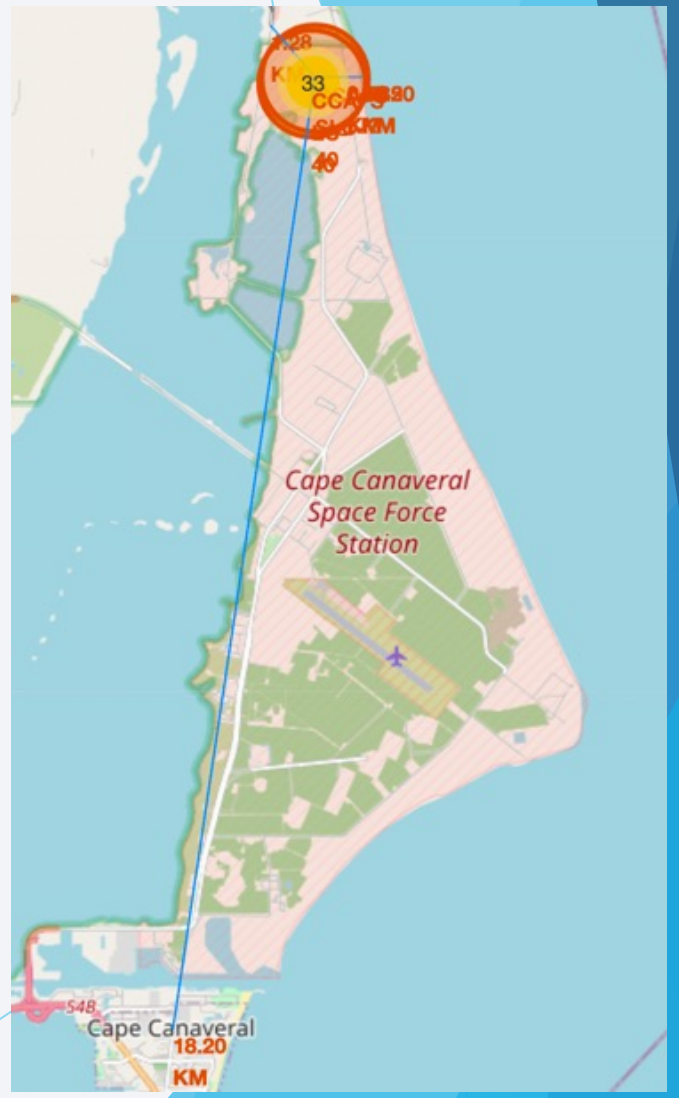


Launch site and its proximities

- Finding on the launching and its proximities (use launch site CCAFS SLC-40 as an example)
  - The distance between Launch site CCAFS SLC-40 and nearest coastline is 0.8499 KM

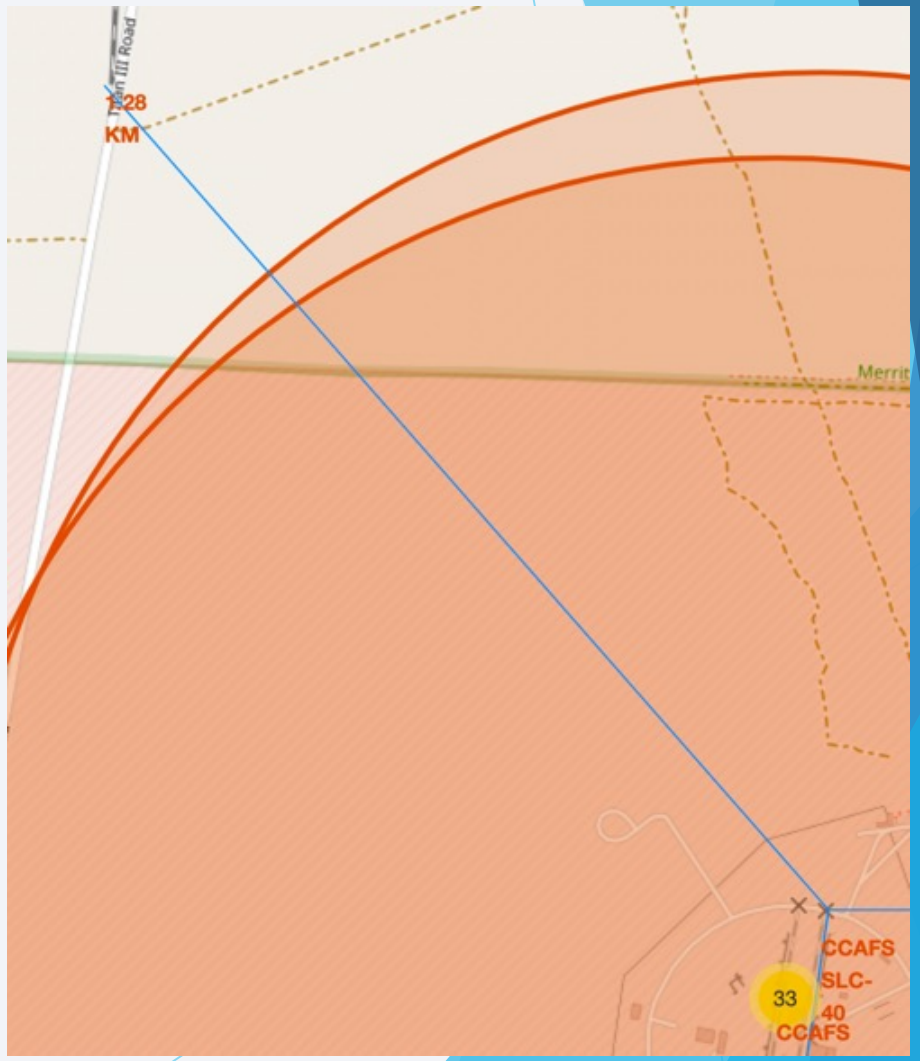
Launch site and its proximities

- Finding on the launching and its proximities (use launch site CCAFS SLC-40 as an example)
  - The distance between Launch site CCAFS SLC-40 and nearest city, Cape Canaveral, is 18.2 KM



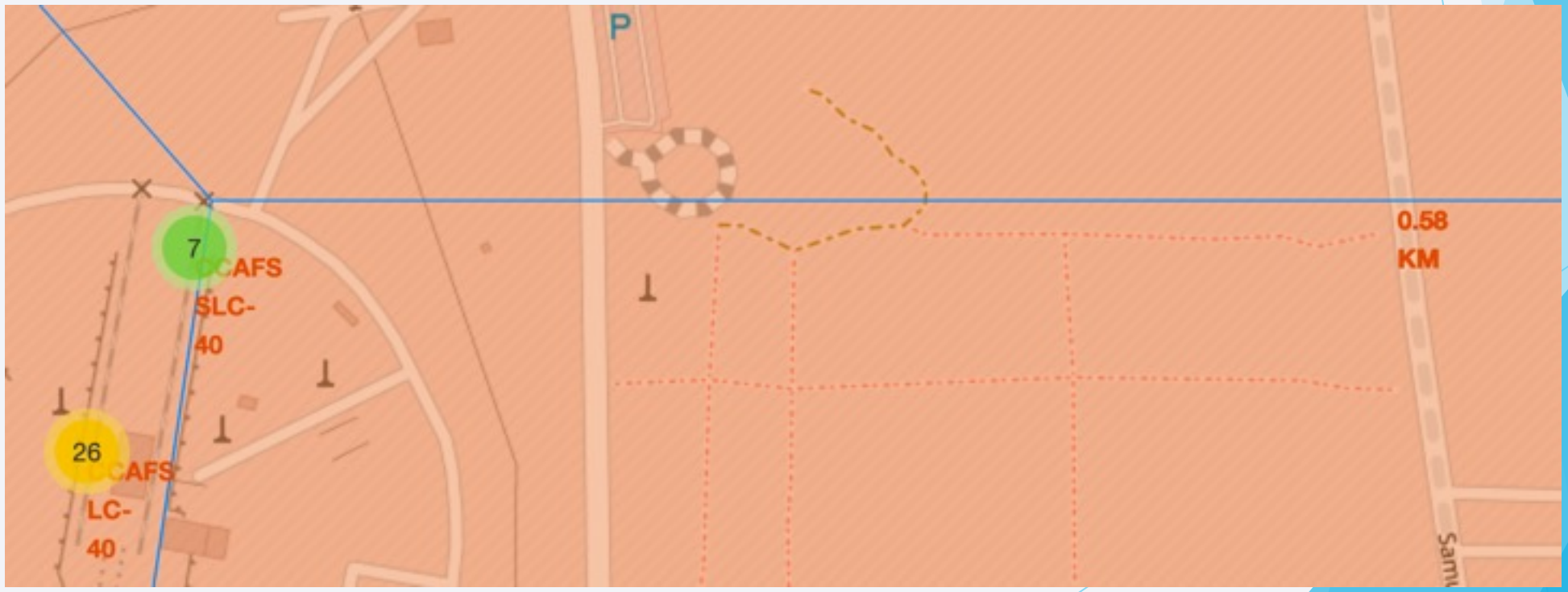
Launch site and its proximities

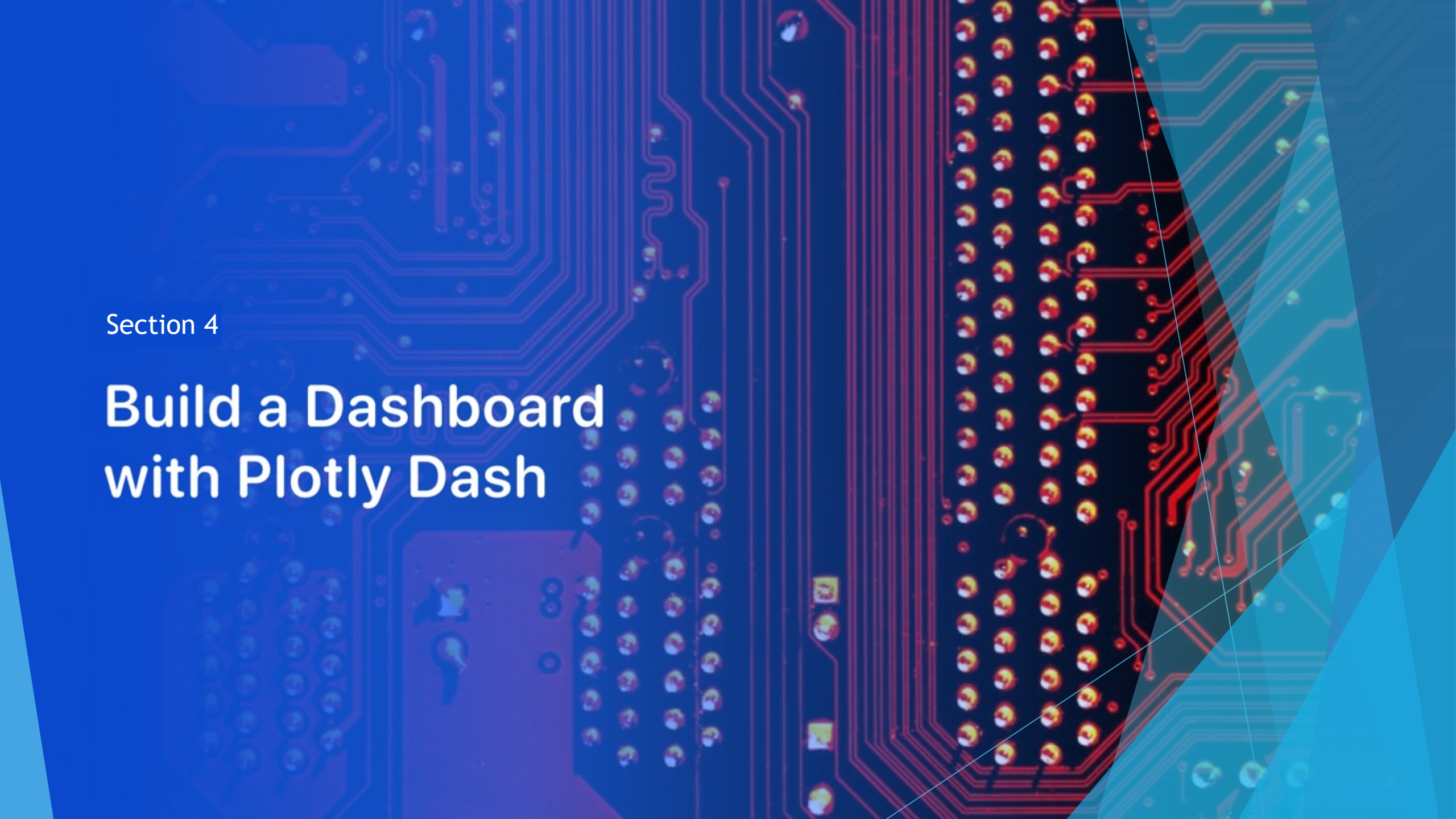
- Finding on the launching and its proximities (use launch site CCAFS SLC-40 as an example)
  - The distance between Launch site CCAFS SLC-40 and the nearest railway is 1.28 KM



Launch site and its proximities

- Finding on the launching and its proximities (use launch site CCAFS SLC-40 as an example)
  - The distance between Launch site CCAFS SLC-40 and the nearest highway is 0.58 KM



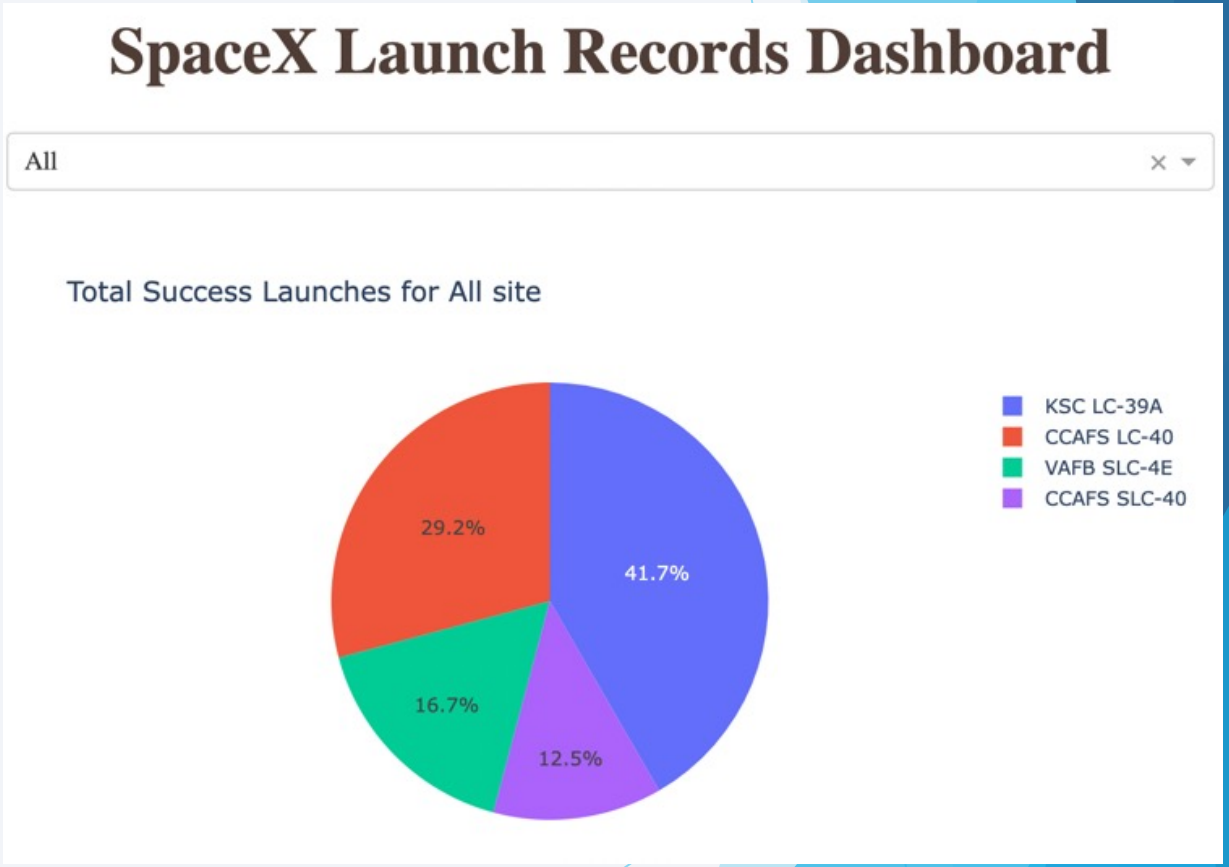

Section 4
Build a Dashboard with Plotly Dash

Plotly Dash Dashboard Results

- Success Launches for All SpaceX Launch Sites
- Highest Launch Success Ratio of SpaceX
- Payload range vs. Launch Outcome Correlation
- Booster version vs. Launch Outcome Correlation

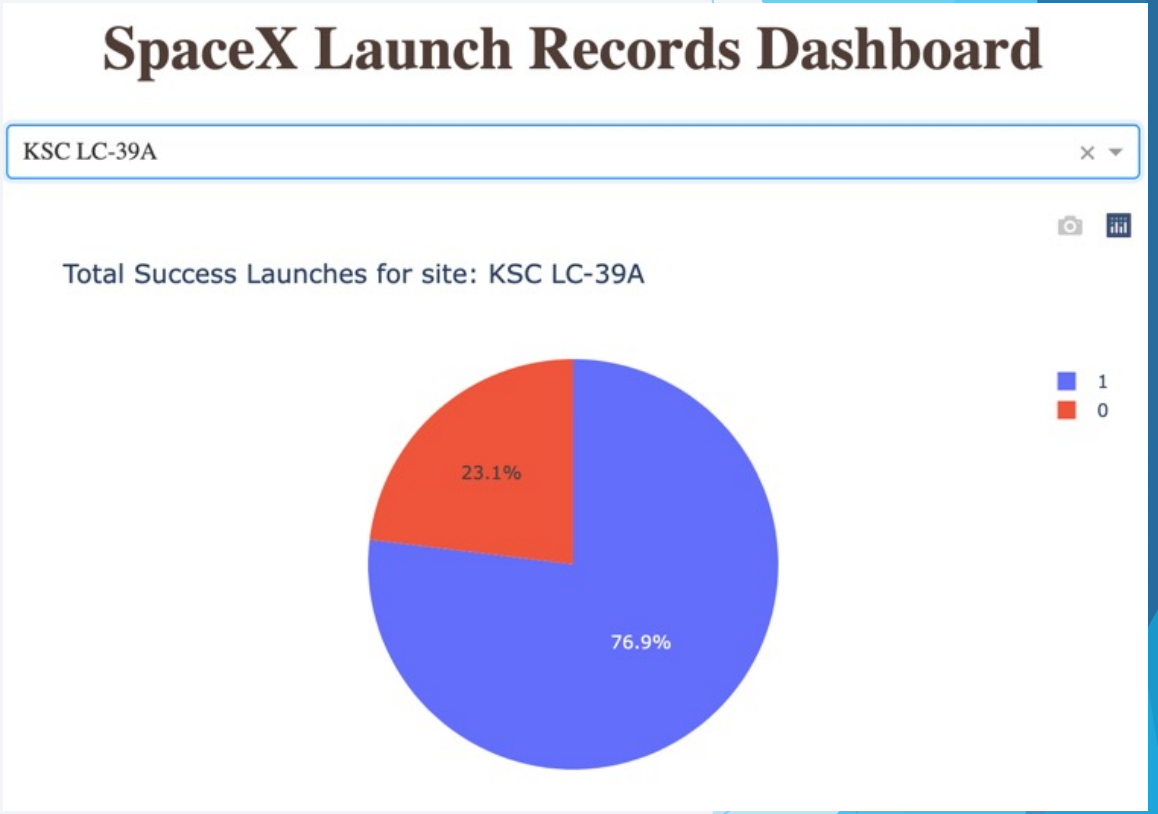
Success Launches for All SpaceX Launch Sites

- The screenshot of launch success count for all sites in a piechart.
- Explanation of the important elements and findings on the screenshot
  - Launch site KSC LC-39A has the highest number of successful launch
  - Launch site CCAFS SLC-40 has the lowest number of successful launch



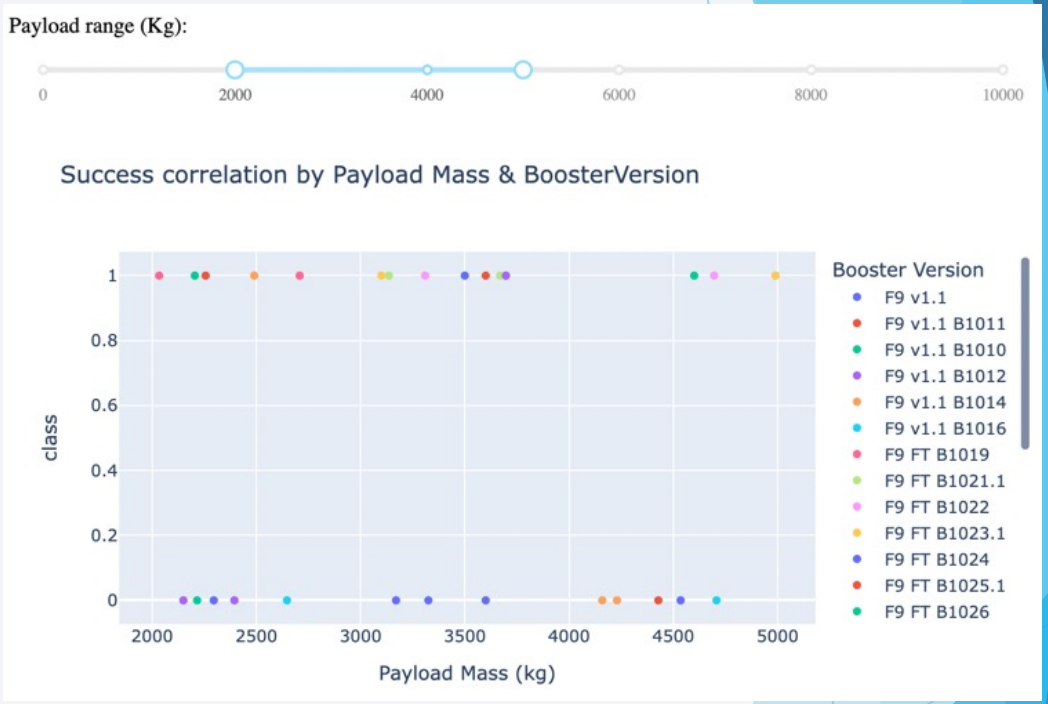
Highest Launch Success Ratio of SpaceX

- The screenshot of the highest launch success ratio (Site KSC LC-39A) in a pie chart.
- Explanation of the important elements and findings on the screenshot
  - Launch site KSC LC-39A has 76.9% of successful launch and 23.1% for failure



Payload vs. Launch Outcome Correlation

- The screenshot of the Payload vs. Launch Outcome scatter plot for all sites
- Explanation of the important elements and findings on the screenshot
  - The payload range between 2000 to 5000 have the largest success rate

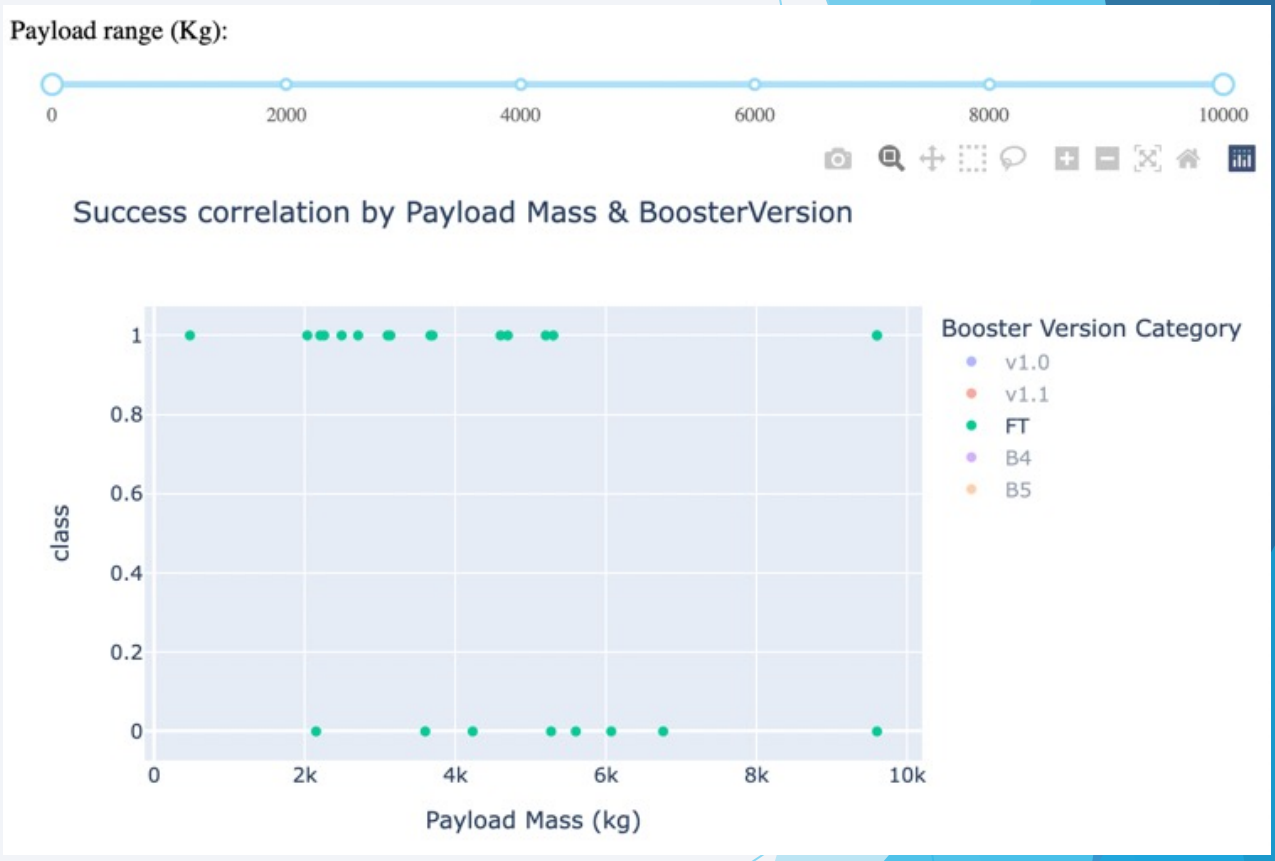


Payload vs. Launch Outcome Correlation

- Explanation of the important elements and findings on the screenshot
  - The payload range between 6000 to 7000 have the lowest success rate

Payload vs. Launch Outcome Correlation

- Explanation of the important elements and findings on the screenshot
  - The booster version FT has the largest success rate



Section 5

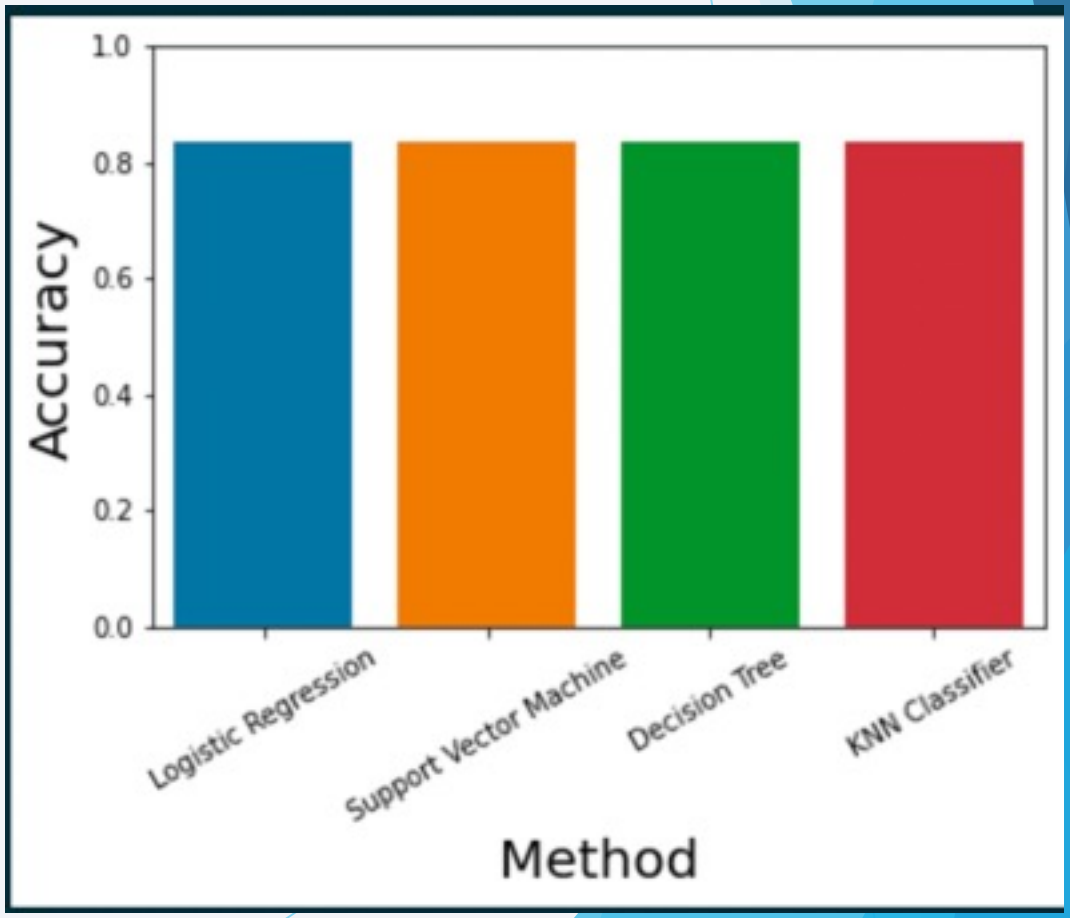
Predictive Analysis (Classification)

Predictive Analysis (classification) ResultsResults

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

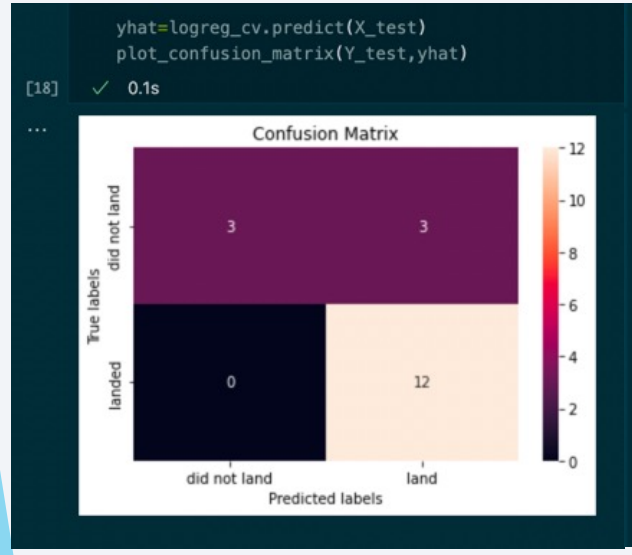
Classification Accuracy

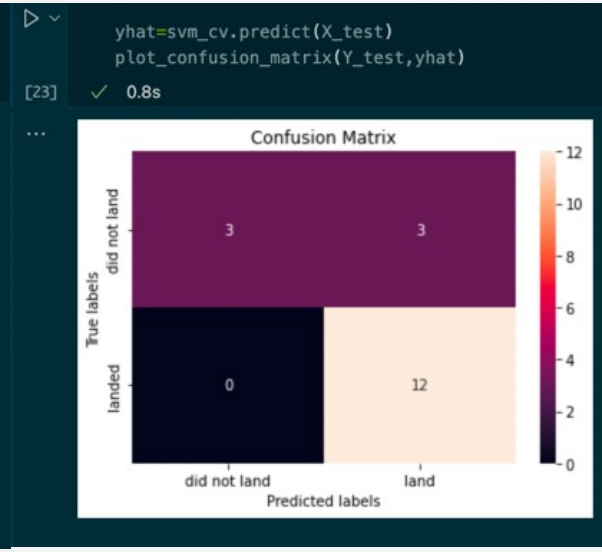
- Visualize the accuracy for all built classification models in a bar chart
  - All models have the same accuracy, 83.3% for the test data.

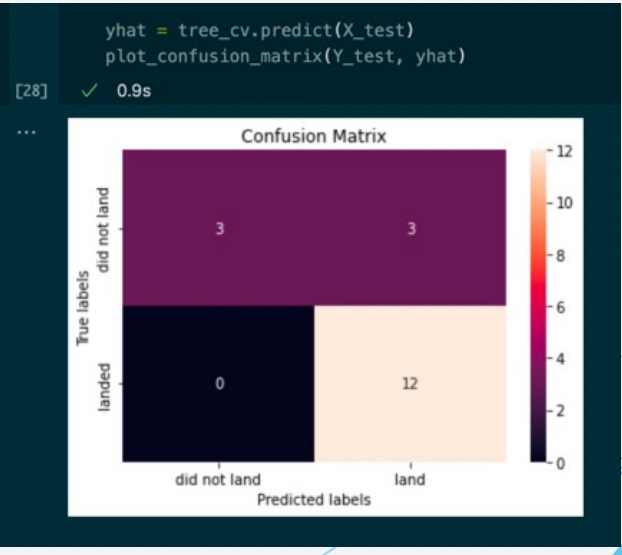


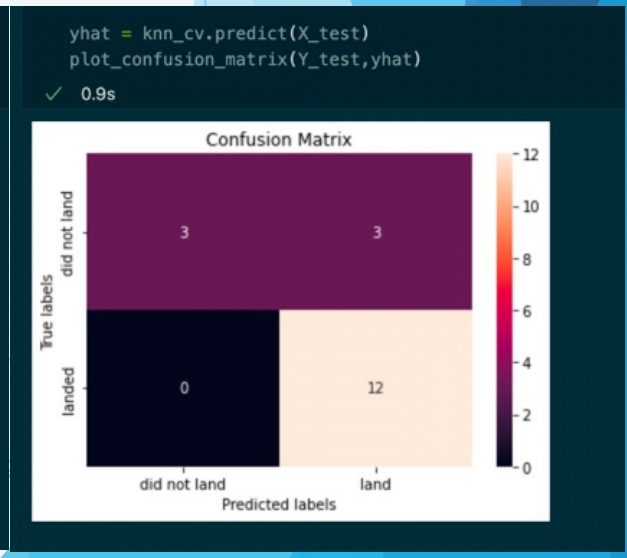
Confusion Matrix

- Since all the models have the same accuracy, all the confusion matrix are shown









Conclusions

- Although the accuracy for the training data are different from the models, the prediction accuracy is the same for all models. Most possibly is the test data size are small, only 18.

```
    print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
    print("accuracy :",logreg_cv.best_score_)
[16] ✓ 0.2s
...  tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
     accuracy : 0.8464285714285713
```

```
    print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
    print("accuracy :",svm_cv.best_score_)
[ ]
...  tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
     accuracy : 0.8482142857142856
```

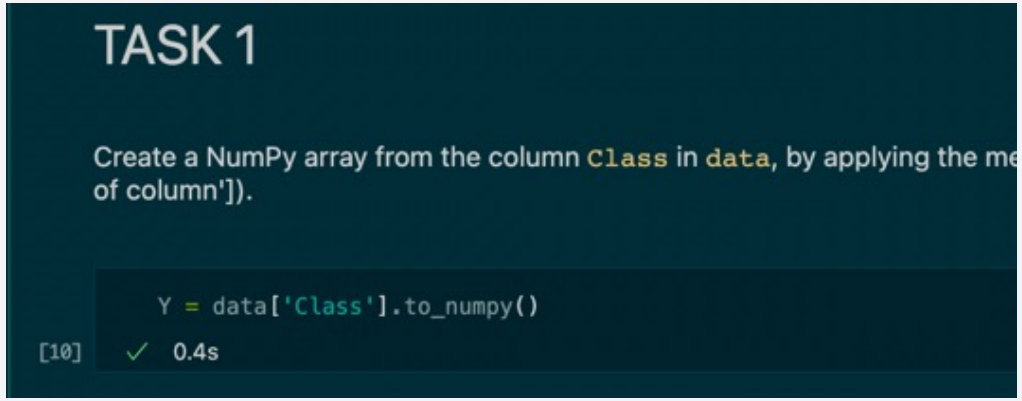
Conclusions

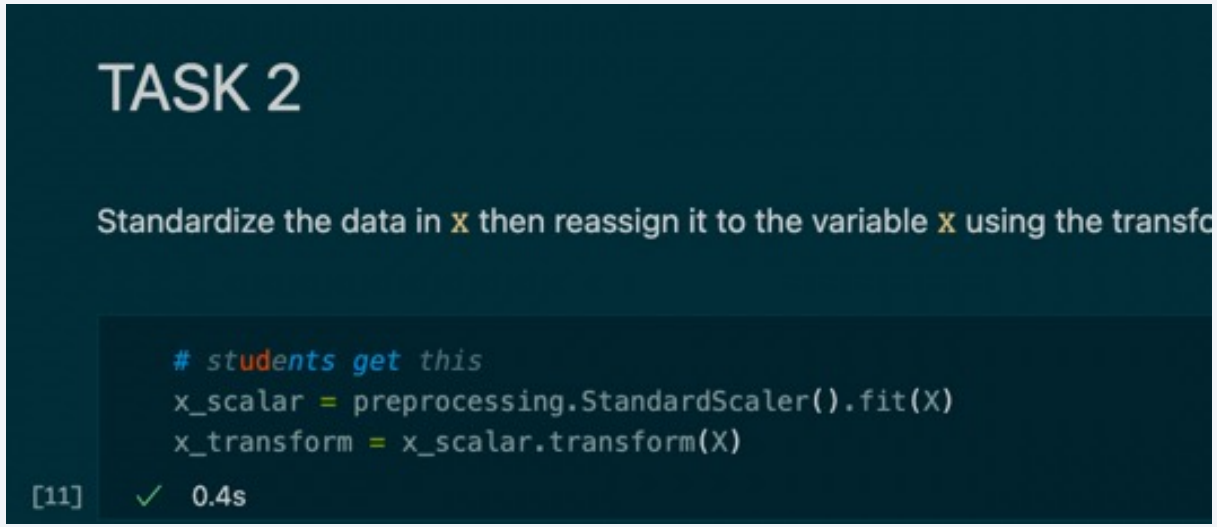
```
    print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
    print("accuracy :",tree_cv.best_score_)
[26]  ✓ 0.3s

...  tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt',
     'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'}
     accuracy : 0.8767857142857143
```

```
    print("tuned hpyerparameters :(best parameters) ",knn_cv.best_params_)
    print("accuracy :",knn_cv.best_score_)
[31]  ✓ 0.2s

...  tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
     accuracy : 0.8482142857142858
```

Appendix

- Python code snippets



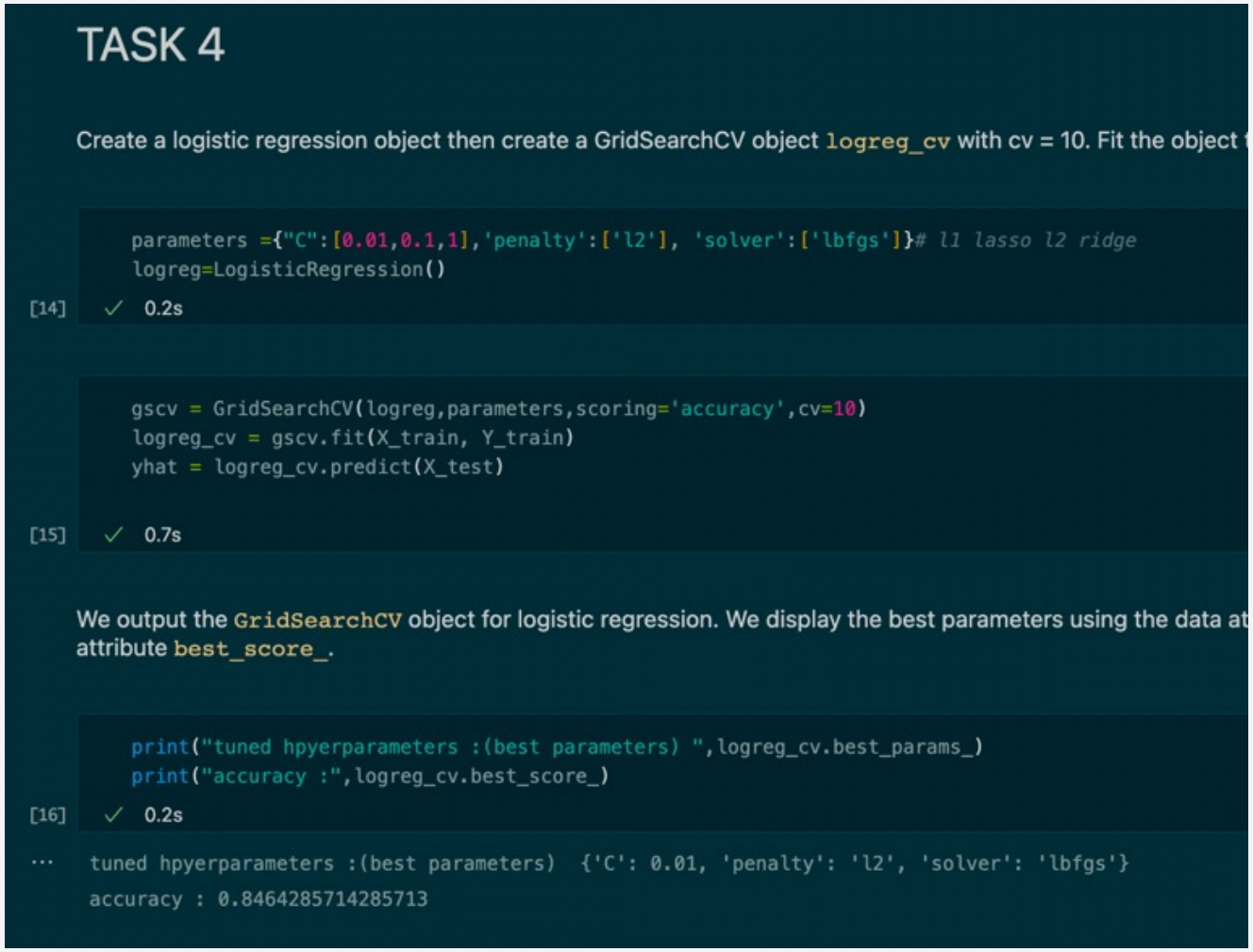


Appendix (cont'd)

- Python code snippets

Appendix (cont'd)

- Python code snippets



TASK 4

Create a logistic regression object then create a GridSearchCV object `logreg_cv` with cv = 10. Fit the object

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
logreg=LogisticRegression()
```

[14] ✓ 0.2s

```
gscv = GridSearchCV(logreg,parameters,scoring='accuracy',cv=10)
logreg_cv = gscv.fit(X_train, Y_train)
yhat = logreg_cv.predict(X_test)
```

[15] ✓ 0.7s

We output the `GridSearchCV` object for logistic regression. We display the best parameters using the data at attribute `best_score_`.

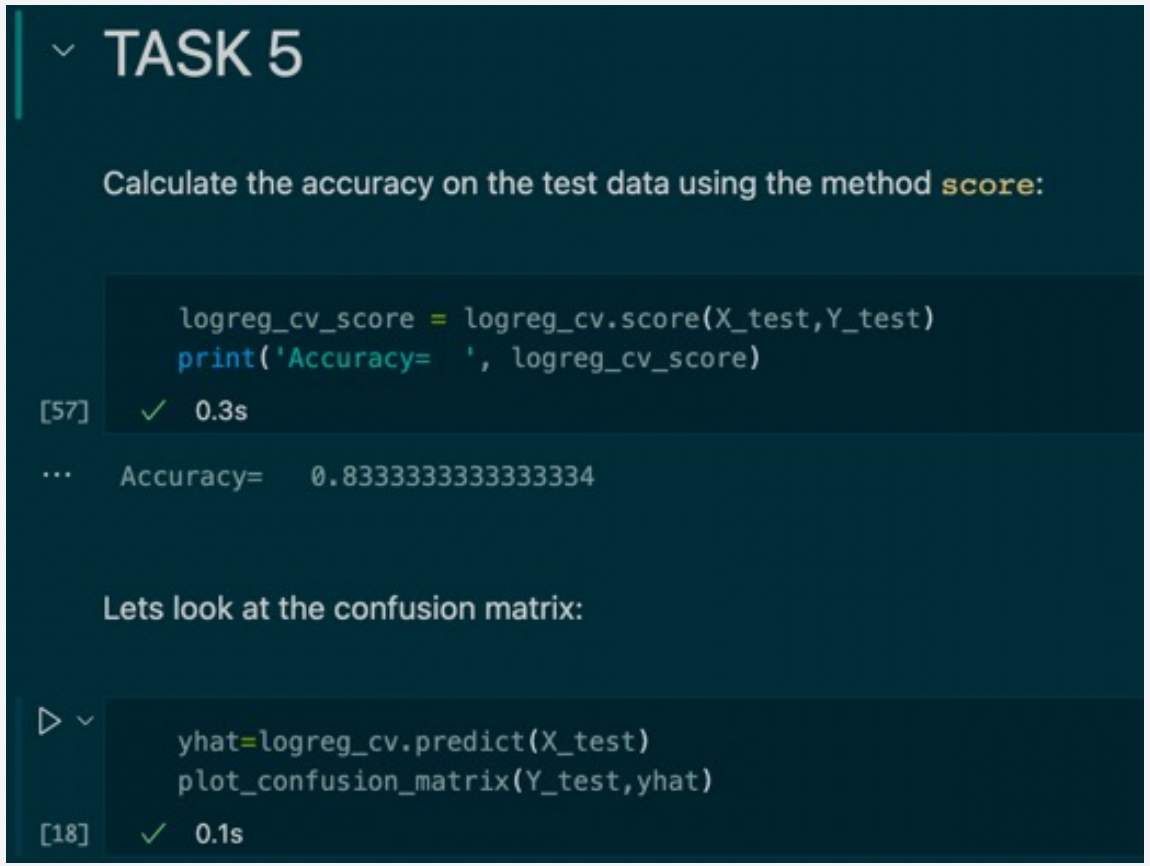
```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

[16] ✓ 0.2s

```
tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

Appendix (cont'd)

- Python code snippets



Appendix (cont'd)

- Python code snippets

TASK 6

Create a support vector machine object then create a `GridSearchCV` object `svm_cv` wi

```
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
```

[52]

```
gscv = GridSearchCV(svm, parameters,scoring='accuracy', cv=10)
svm_cv = gscv.fit(X_train, Y_train)
```

[53]

```
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

[54]

Appendix (cont'd)

- Python code snippets

Appendix (cont'd)

- Python code snippets

TASK 8

Create a decision tree classifier object then create a `GridSearchCV` object t

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

[59]

```
gscv = GridSearchCV(tree, parameters, scoring='accuracy', cv=10)
tree_cv = gscv.fit(X_train, Y_train)
```

Appendix (cont'd)

- Python code snippets

TASK 9

Calculate the accuracy of tree_cv on the test data using the method `score`:

```
[72]
tree_cv_score = tree_cv.score(X_test,Y_test)
print('Accuracy=  ', tree_cv_score)
```

```
...  Accuracy=   0.8888888888888888
```

We can plot the confusion matrix

```
[64]
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test, yhat)
```

Appendix (cont'd)

- Python code snippets

TASK 10

Create a k nearest neighbors object then create a `GridSearchCV` object `knn_cv` with cv = 10. Fit th

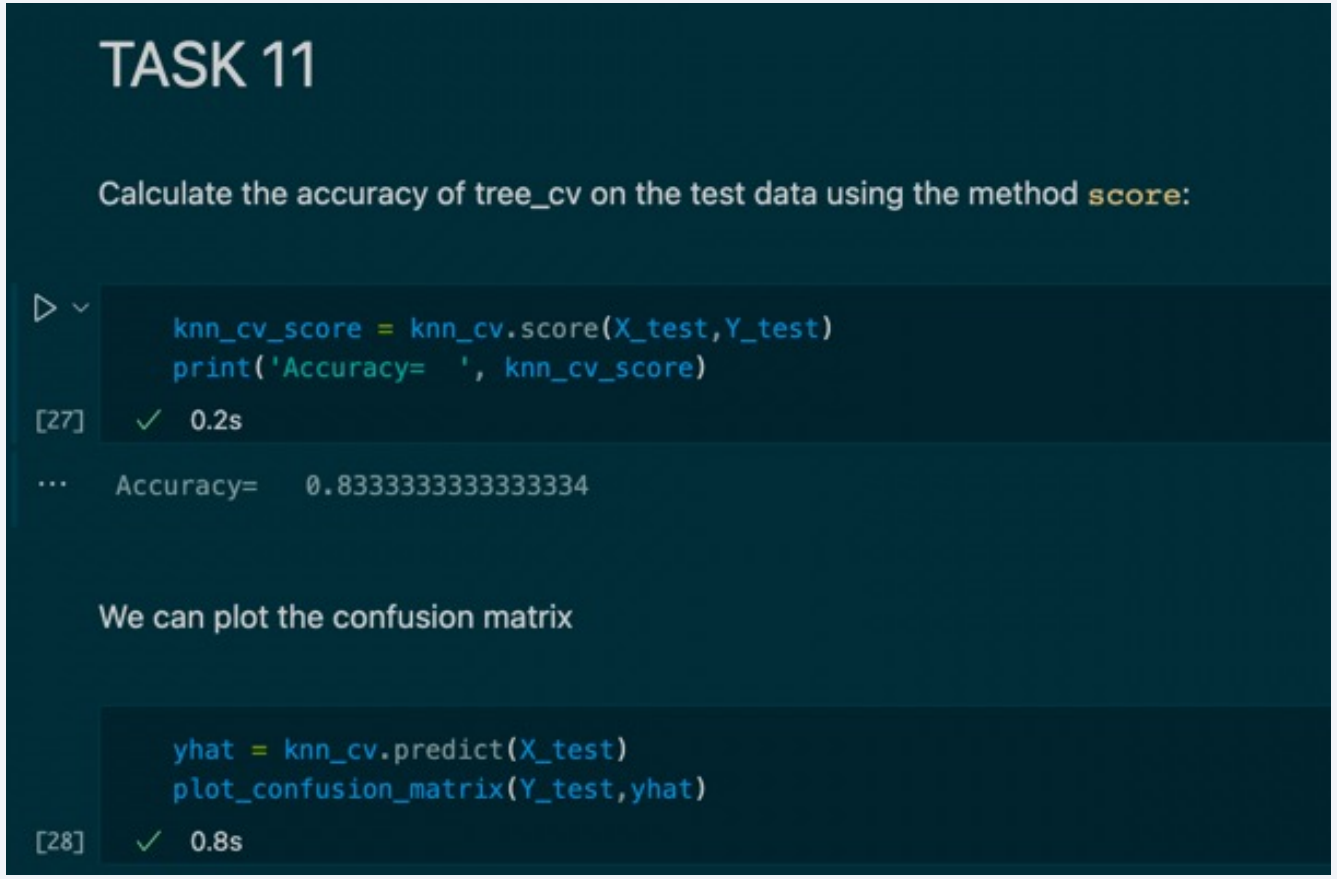
[65]
```
parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'p': [1,2]}

KNN = KNeighborsClassifier()
```

[66]
```
gscv = GridSearchCV(KNN, parameters, scoring='accuracy', cv=10)
knn_cv = gscv.fit(X_train, Y_train)
```

[67]
```
print("tuned hpyerparameters :(best parameters) ",knn_cv.best_params_)
print("accuracy :",knn_cv.best_score_)
```

Appendix (cont'd)

- Python code snippets



Appendix (cont'd)

- Python code snippets

TASK 12

Find the method performs best:

```
algorithms = {'Logistic Regression': logreg_cv_score, 'Support Vector Machine': svm_cv_score, 'Decision Tree': tree_cv_score, 'KNN Classifier': knn_cv_score}
best_algorithm = max(algorithms, key=algorithms.get)
print("Best algorithm is %s and the score is %s" % (best_algorithm, algorithms.get(best_algorithm)))
```

[29] ✓ 0.2s

```
Best algorithm is Logistic Regression and the score is 0.8333333333333334
```

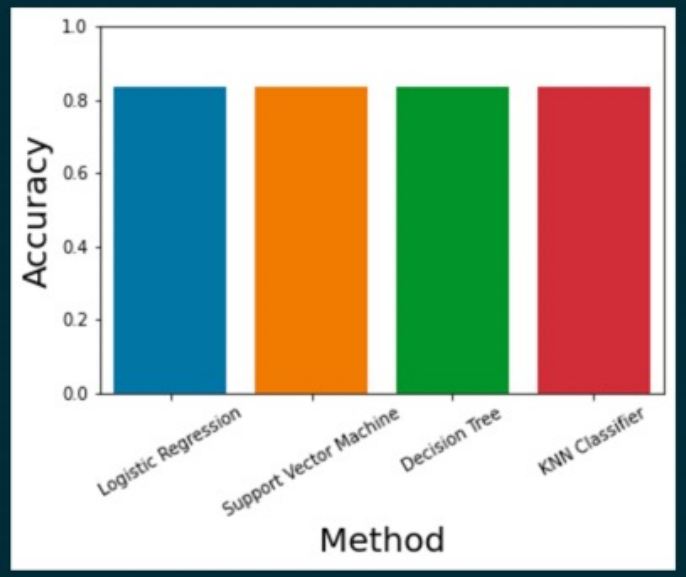
Appendix (cont'd)

- Python code snippets

```
# plot bar chart for the prediction result
df = pd.DataFrame({'Method': list(algorithms.keys()), 'Accuracy': list(algorithms.values())})
sns.barplot(x='Method', y='Accuracy', data=df)
plt.xlabel("Method",fontsize=20)
plt.ylabel("Accuracy",fontsize=20)
plt.xticks(rotation=30)
plt.ylim(0,1)
```

[35] ✓ 0.7s

```
(0.0, 1.0)
```



Thank you!