Space X Falcon 9 First Stage Landing Prediction Project

Kaya CANER
Data Scientist, SpaceY
22.11.2023

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- Can we find a pattern and make predictions if our missions will be successful or not according to variables?(like Payload mass, launching site, etc...)
- Methodology
  - Python programming language and related libraries are used(Methodology for details)
  - Steps are applied as explained in details
    - Data collection and webscraping
    - Data wrangliing
    - Exploratory Data Analysis
    - SQL queries
    - Data visualization
    - Machine learning models
    - Build an interactive Dash App
- Relationship between Payload mass, orbit, flight number, launch site and landing success rate is found
- Machine learning models are built with 0.86 accuracy score
- It's concluded that data science methods can be developed for further mission plans

INTRODUCTION

- As a developing company SpaceY, It is crucial to minimize the risks of missions.
- Aim of this project is obtaining information and insights to determine the cost of each launch using data science perspective.
- Can we find a pattern and make predictions if our missions will be successful or not according to variables?(like Payload mass, launching site, etc...)

METHODOLOGY

- Python programming language and related libraries are used for analysis.

- JupyterLab environment is used for;
  - Data collection and webscraping
  - Data wrangling
  - Exploratory Data Analysis
  - SQL queries
  - Data visualization
  - Machine learning models
- Skills Network Labs virtual environment is used for;
  - Build an interactive Dash App

METHODOLOGY

Pandas Library:

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.[1]

Pandas is one of the major tool that is useful on every steps of project.

It is used completely or partially for:

- Data collection and webscraping
- Data wrangling
- Exploratory Data Analysis
- SQL queries
- Data visualization
- Machine learning models
- Build an interactive Dash App

METHODOLOGY

NumPy

- Numpy: The fundamental package for scientific computing with Python[2]. Used in this project on steps;
  - Data collection
  - Data wrangling
  - Exploratory Data Analysis
  - Data visualization
  - Machine learning models
- Matplotlib and Seaborn: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python[3]. Seaborn is a Python data visualization library based on matplotlib. Both used in this project on for;
  - Data visualization

METHODOLOGY

- Folium: Folium makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map[4]. Used in this project for;
  - Data visualization
- Dash: Dash is the original low-code framework for rapidly building data apps in Python[5]. Used in this project for;
  - Build an interactive Dash App
- Scikit-learn: Scikit-learn is simple and efficient tools for predictive data analysis and machine learning[6]. Used in this project for;
  - Machine learning models

METHODOLOGY

- Open-source Space X data is used.
  - Space X Data API is used to obtain data(7)
  - Also it can be obtained on Wikipedia(8) by webscraping.
- After filtering relevant data, dataframe is obtained shown below.
  - Date, Booster version, Payload mass, Launch site, etc. can be seen.

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	11	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	12	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

METHODOLOGY

Data Wrangling:

- It is detected that 5 rows are missing in PayloadMass column and 26 rows are missing in LaunchPad column.
- It is decided to fill missing PayloadMass values with mean value which is 6123.55 kg.
- Missing LaunchPad values are not expected to have major importance. If it seems necessary later, it can be reassessed.

```
data_falcon9.isnull().sum()
```

```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        5
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

METHODOLOGY

Exploratory Data Analysis

We perform some exploratory data analysis to understand data, to get some insights and find patterns for predictive analysis using Pandas. Some queries with SQL also is shared in Results part for same reasons.

- Launch sites and number of launches are obtained. As seen, "CCAFS SLC 40" is the most used one. Location of launch sites are shown in Results part.

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

METHODOLOGY

Exploratory Data Analysis

- How many missions are aimed to which orbits listed below:

```
GTO       27
ISS       21
VLEO      14
PO         9
LEO        7
SSO        5
MEO        3
ES-L1      1
HEO        1
SO         1
GEO        1
Name: Orbit, dtype: int64
```

GTO, (A geosynchronous orbit is a high Earth orbit that allows satellites to match Earth's rotation.) is orbit aimed most.[9]

METHODOLOGY

Exploratory Data Analysis

- Landing outcomes show the location of landing and if the landing is successful or not.

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

- True ASDS : Successful landing to drone ship.
- None None, None ASDS: Failure to land.
- True RTLS: Successful landing to ground pad.
- False ASDS: Unsuccessfully landed to drone ship.
- True Ocean: Successful landing to ocean.
- False RTLS: Unsuccessfully landed to ground pad.

METHODOLOGY

Exploratory Data Analysis

- Classifying landing outcomes numerically is very important for using this data for machine learning algorithms. That's why successful landings are assigned 1 and unsuccessful landings are assigned 0. New column is created called "Class".
- Also, success rate of landing is obtained easily which is %66,7.

```
df["Class"].mean()
```

```
0.6666666666666666
```

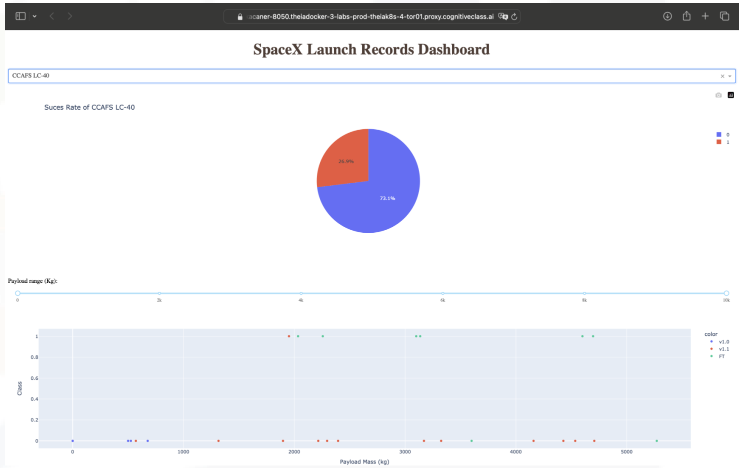
ıchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
ꜰS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
ꜰS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
ꜰS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
ꜰB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
ꜰS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

METHODOLOGY

Data Visualization and Interactive Visual Analytics:

- Relatonship between Payload mass, launch site, orbit, years and success of landing is explored by using matplotlib and seaborn and shared at results section.
- Success of landing vs payload mass and launch site relationships are showed with Dash app and shared at results section(Figures are just shown as an example below).

```python
# TASK 2:
# Add a callback function for `site-dropdown` as input, `success-pie-chart` a
@app.callback(
    Output(component_id="success-pie-chart", component_property="figure"),
    Input(componen (parameter) entered_site: Any _property="value"),
)
def get_pie_chart(entered_site):
    filtered_df = spacex_df[spacex_df["Launch Site"] == entered_site]
    if entered_site == "All":
        fig = px.pie(
            spacex_df,
            values="class",
            names=spacex_df["Launch Site"],
            title="Total Success Launches by Site",
        )
        return fig
    else:
        fig_01 = px.pie(
            values=filtered_df["class"].value_counts(),
            names=filtered_df["class"].value_counts().index,
            title="Suces Rate of {}".format(entered_site),
        )
        return fig_01
```



METHODOLOGY

Machine Learning Predictions using Scikit-learn

- Machine learning algorithms work with numerical data. One-hot encoding technique is applied to change categorical data to binary data. Pandas get_dummies() method is used for this technique.
- Data is standartized using scikit-learn library.
- Logistic regression, support vector machine, decision tree, K nearest neighbor techniques are applied and scores are evaluated.

RESULTS

- Exploratory Data Analysis results that are made in SQL using SqlLite are shared below:

Task 1

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

[24]:
```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

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

[24]:

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

Task 2

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

[15]:
```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE "%CCA%" LIMIT 5
```

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

[15]:

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
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

RESULTS

- Exploratory Data Analysis results that are made in SQL using SqlLite are shared below:

Task 3

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

[19]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = "NASA (CRS)"
```

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

[19]:

SUM(PAYLOAD_MASS__KG_)
45596

Task 4

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

[20]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = "F9 v1.1"
```

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

[20]:

AVG(PAYLOAD_MASS__KG_)
2928.4

Task 5

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

Hint:Use min function

[26]:
```
%sql SELECT MIN(Date) from SPACEXTABLE WHERE Landing_Outcome = "Success (ground pad)"
```

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

[26]:

MIN(Date)
2015-12-22

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

[33]:
```
%sql SELECT Booster_Version, Landing_Outcome, PAYLOAD_MASS__KG_ FROM SPACEXTABLE \
WHERE Landing_Outcome = "Success (drone ship)"\
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

[33]:

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

RESULTS

- Exploratory Data Analysis results that are made in SQL using SqlLite are shared below:

Task 7

List the total number of successful and failure mission outcomes

[34]:
```
%sql SELECT Mission_Outcome, Count(Mission_Outcome) FROM SPACEXTABLE GROUP BY Mission_Outcome
```

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

[34]:

Mission_Outcome	Count(Mission_Outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Task 8

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

[37]:
```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE \
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

[37]:

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

RESULTS

- Exploratory Data Analysis results that are made in SQL using SqlLite are shared below:

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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[42]:
```
%sql SELECT Landing_Outcome, Booster_Version, Launch_Site, substr(Date, 6,2) AS Month, substr(Date,0,5) AS Year \
FROM SPACEXTABLE WHERE substr(Date,0,5)="2015" AND Landing_Outcome = "Failure (drone ship)"
```

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

[42]:

Landing_Outcome	Booster_Version	Launch_Site	Month	Year
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	10	2015
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	04	2015

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.

[43]:
```
%sql SELECT Landing_Outcome, Count(Landing_Outcome) from SPACEXTABLE GROUP BY Landing_Outcome \
having Date between "2010-06-04" and "2017-03-20" ORDER BY COUNT(Landing_Outcome) DESC
```

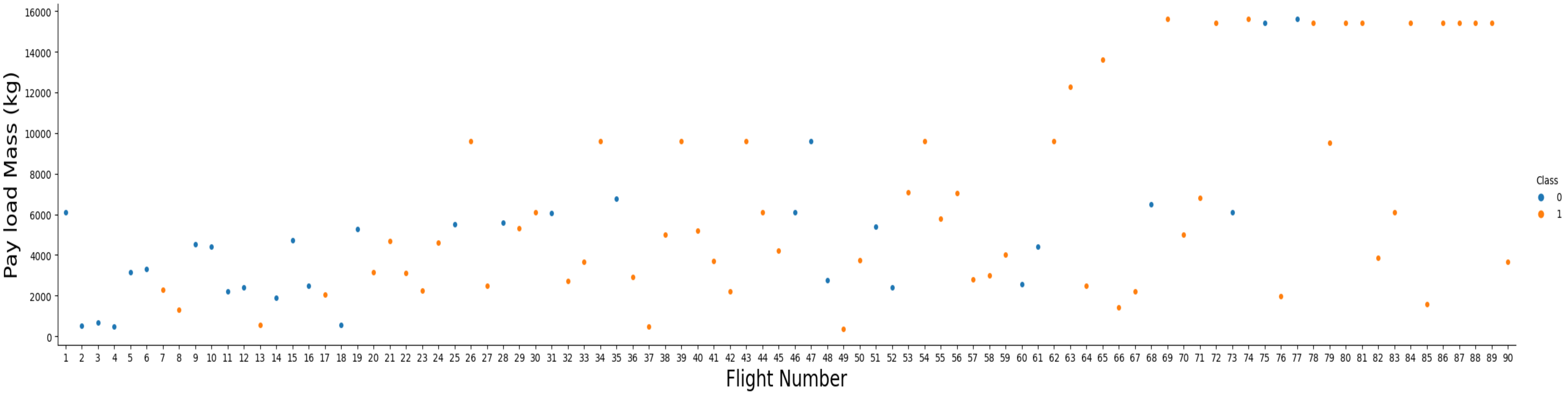
```
 * sqlite:///my_data1.db
Done.
```

[43]:

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

RESULTS

Payload Mass vs Flight Number

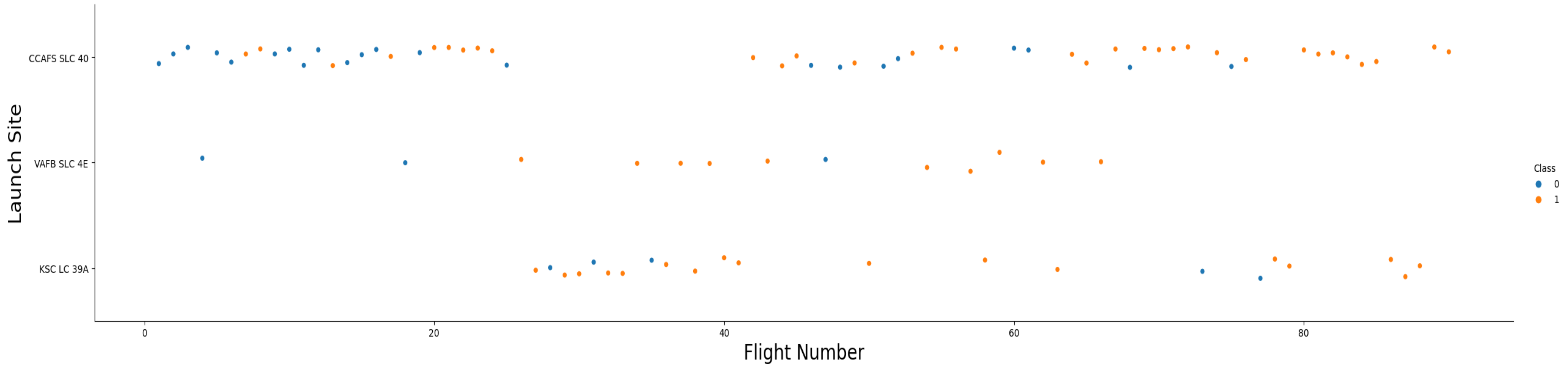


Seems like,

- Landing success increases with flight number increases.
- Payload mass increased over time.

RESULTS

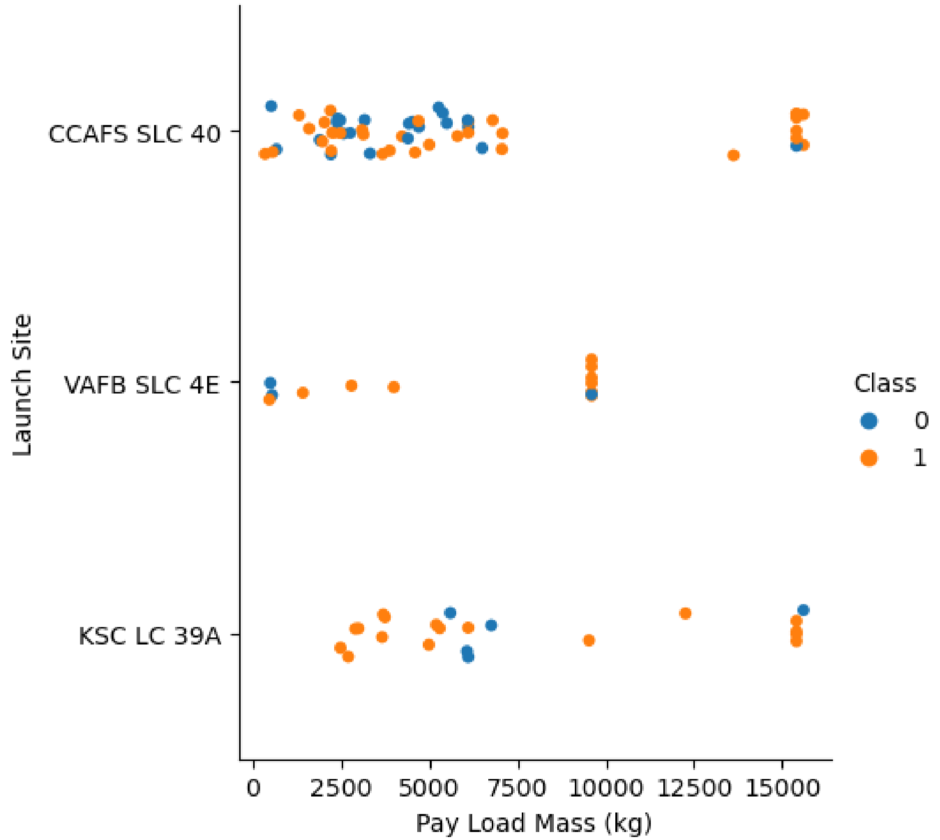
Launch Site vs Flight Number



Seems like, CCAFS SLC 40 is used heavily and VAFB SLC 4E is not used anymore.

RESULTS

Payload Mass vs Launch Site

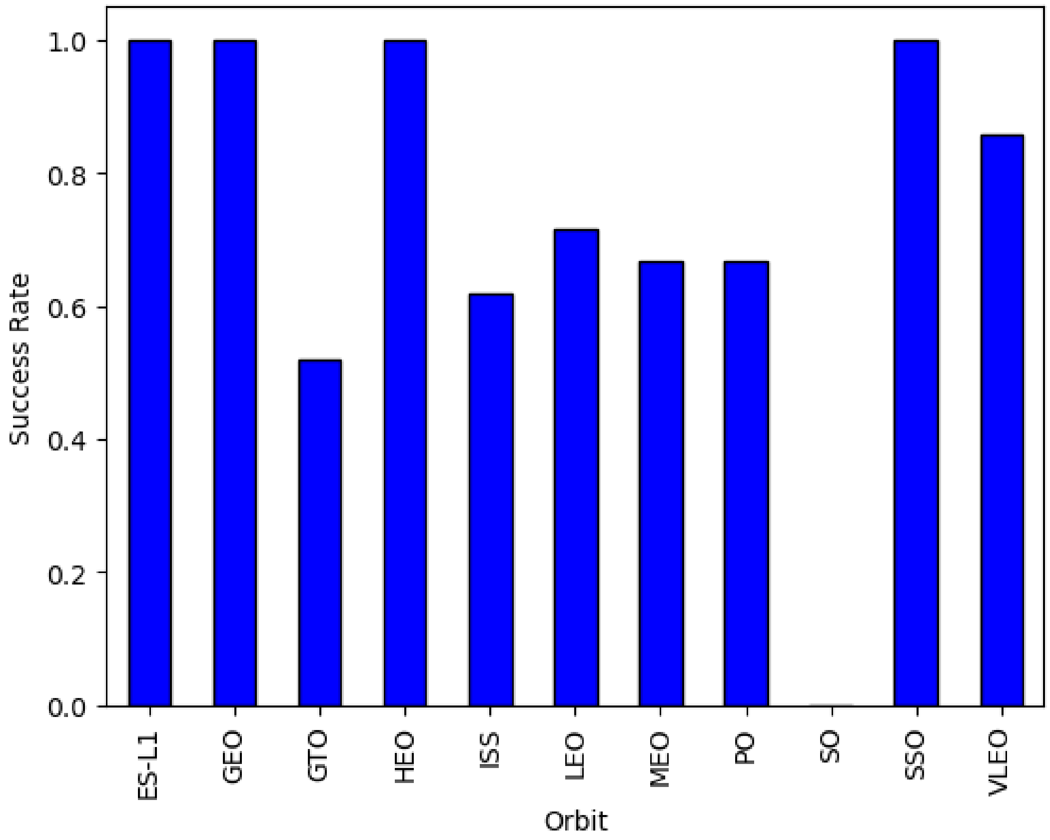


Seems like,

- VAFB SLC 4E is not used for heavy loads.
- Higher Pay Load Mass is related with more successful landing rates.

RESULTS

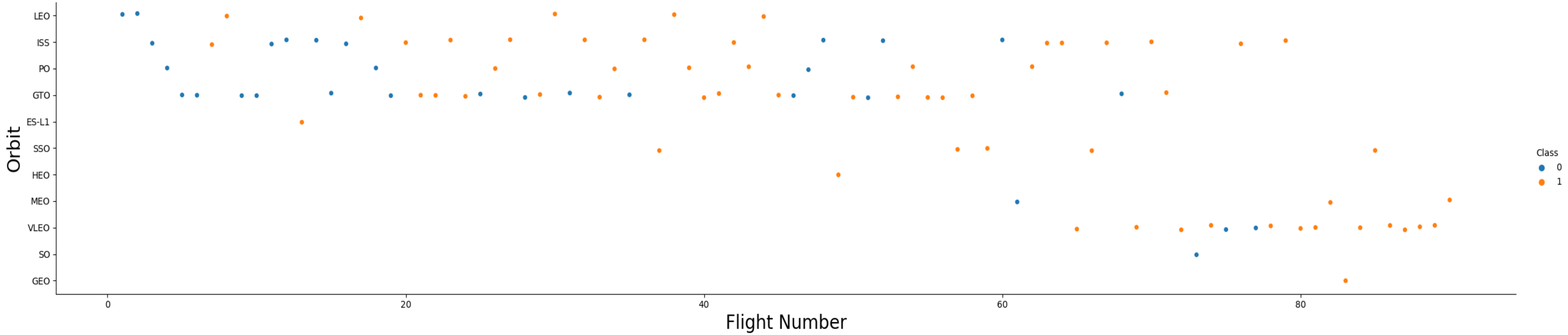
Orbit vs Success Rate



- Seems like, ES-L1, GEO, HEO, SSO missions have %100 success rate.

RESULTS

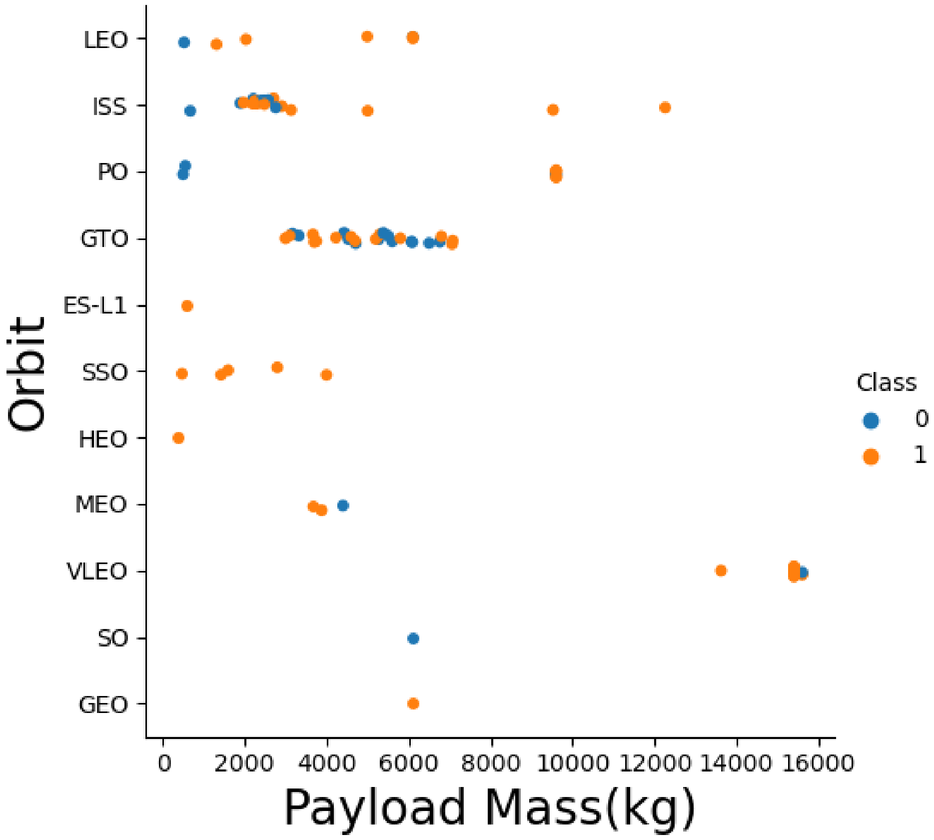
Flight Number vs Orbit



- Seems like, ES-L1, GEO, HEO, SSO missions which have %100 success rate, heavily done at higher flight numbers.

RESULTS

Payload Mass vs Orbit

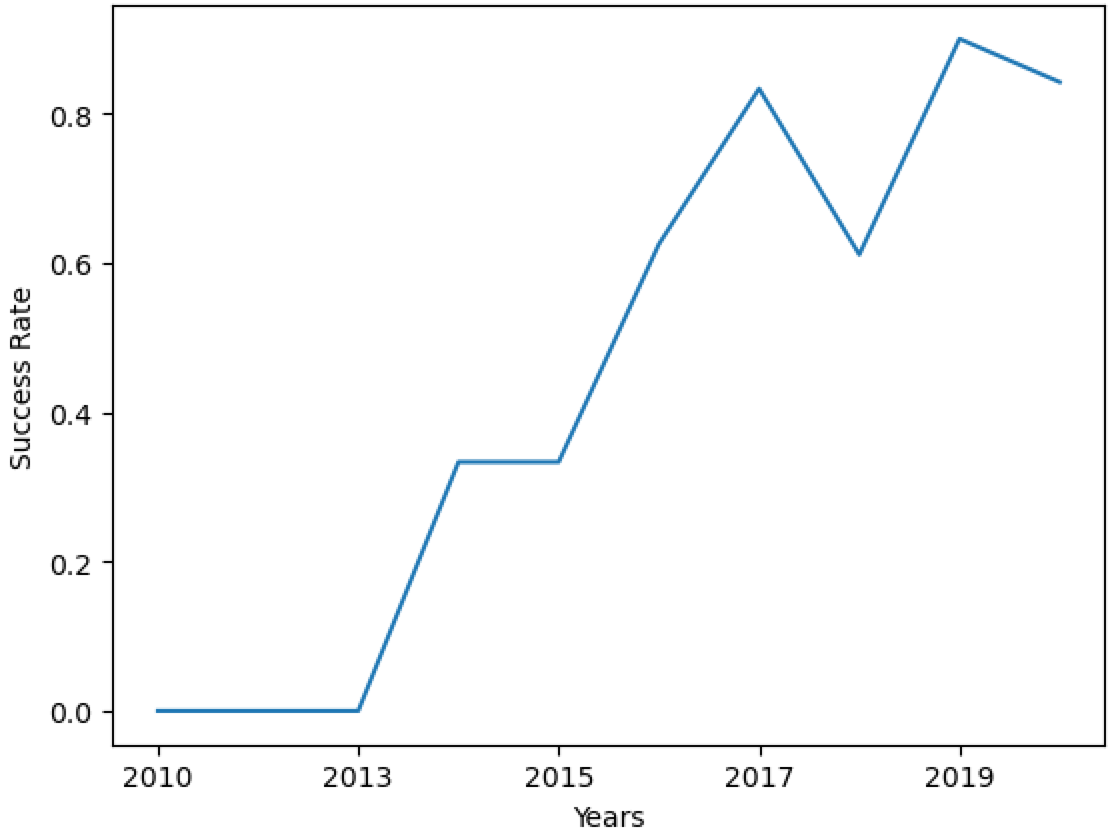


Seems like,

- VLEO missions has the heaviest Payload.
- For LEO, ISS and PO missions that have successful landing is more with heavy loads.
- ES-L1, SSO, HEO missions are made with lower Payload Mass and has %100 successful landing

RESULTS

Years vs Success Rate

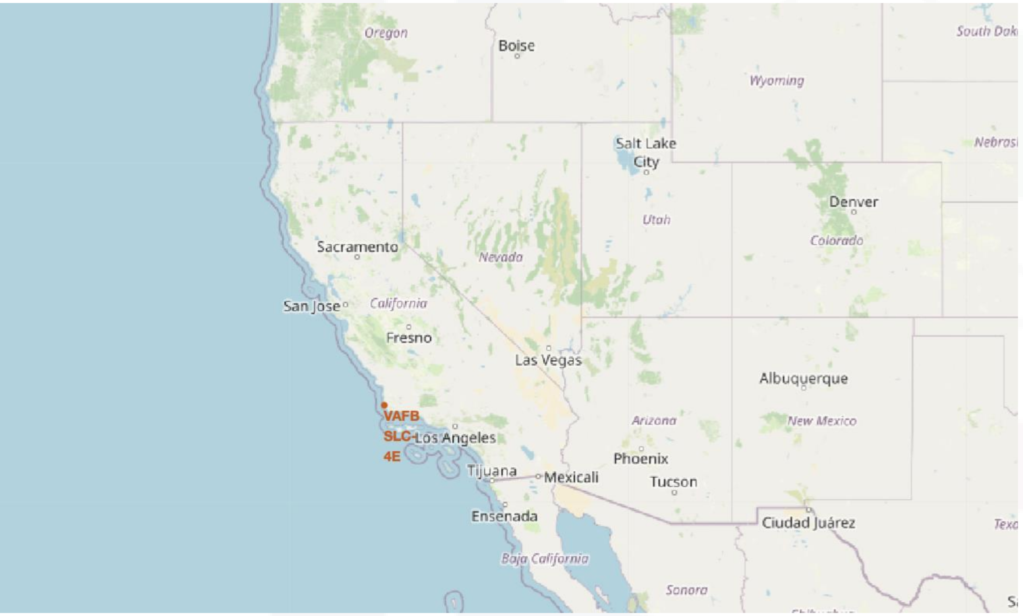


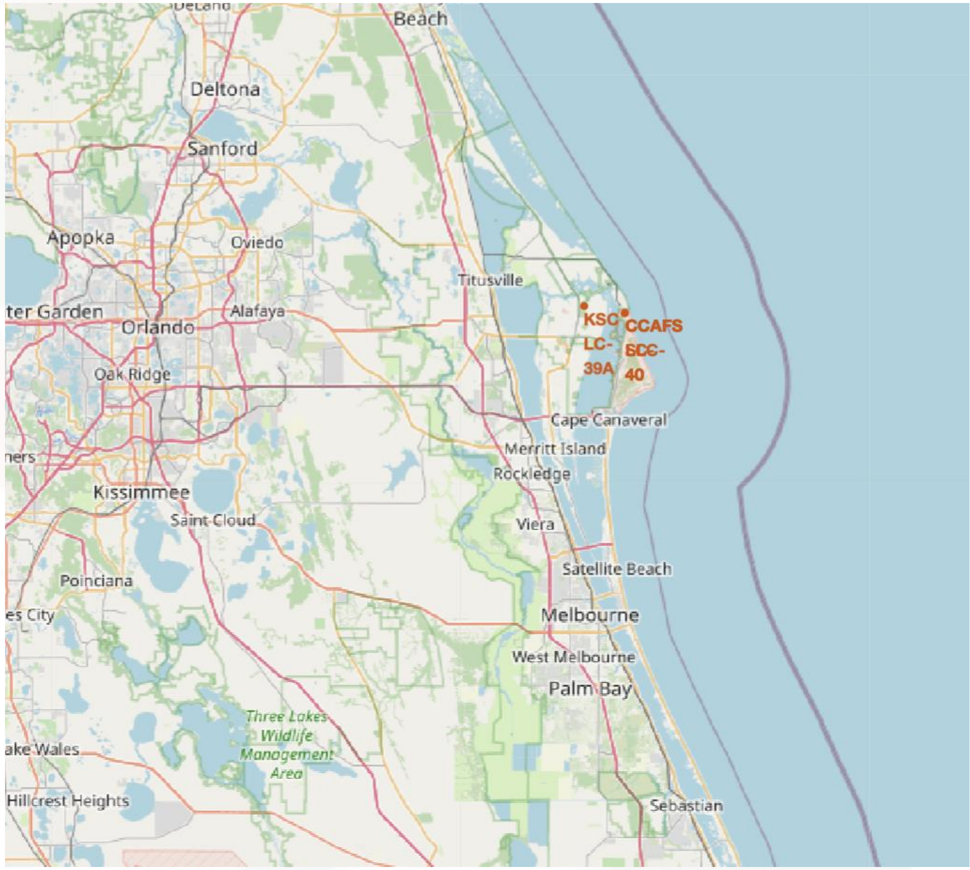
Seems like,

- Success rate increases over time.
- There is drop between 2017 and 2018, after 2019. However, general trend in general perspective is increasing.

RESULTS

Locations of Launch Sites

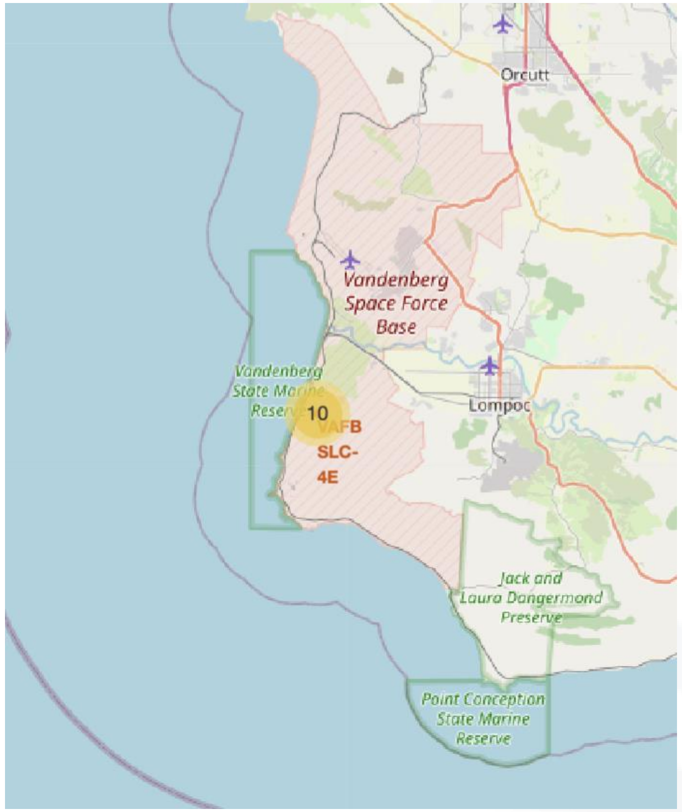


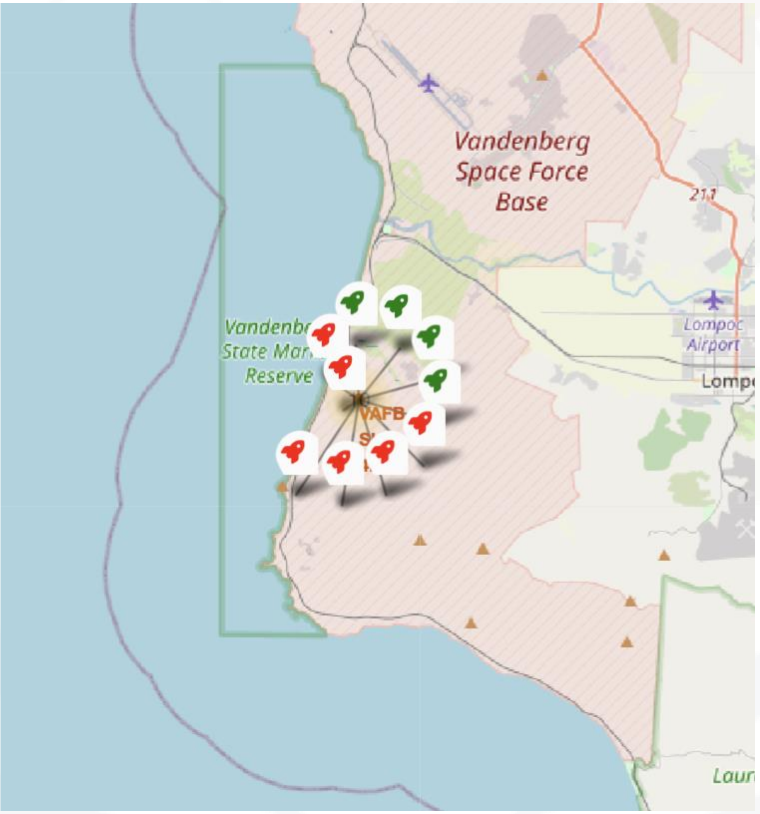


- VAFB-SLC-4E is in CALIFORNIA, US as signed on map.
- KSC LC-39A, CCAFS-SLC40 and CCAFS-LC40 is in FLORIDA, US as signed on map.

RESULTS

Launches from VAFB-SLC-4E

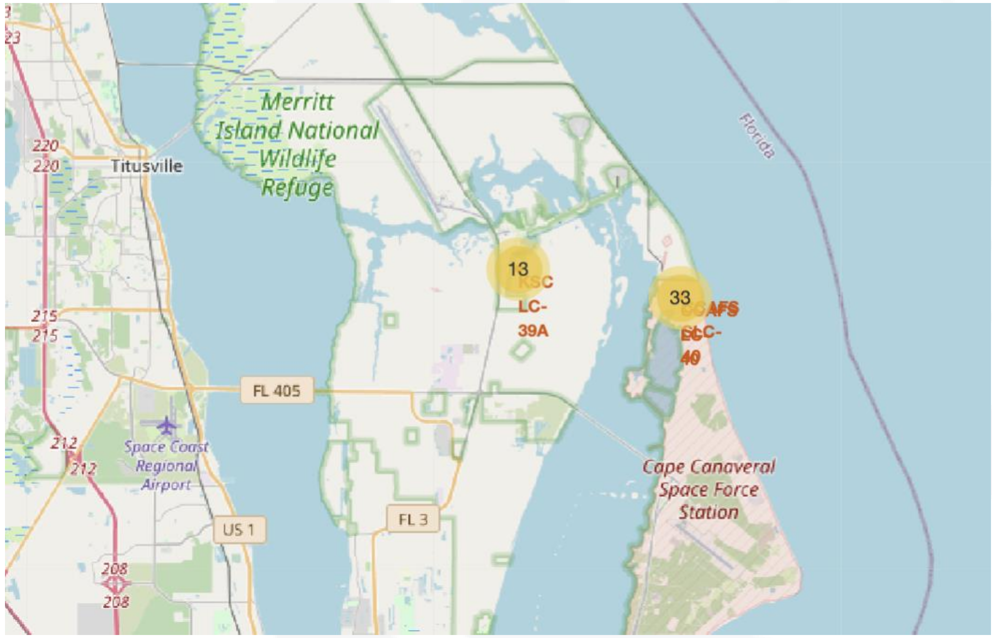


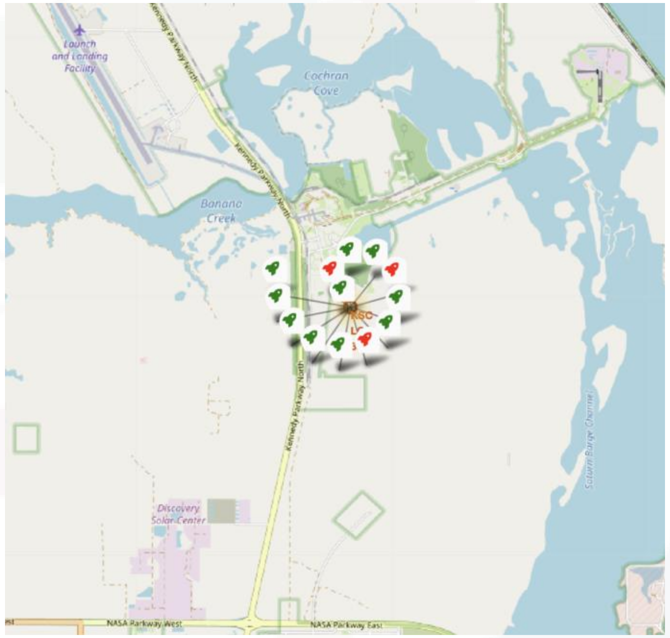


- There are 10 launches from VAFB-SLC-4E and 4 of them have successful landing.

RESULTS

Launches from KSC LC-39A, CCAFS-SLC40 and CCAFS-LC40





- 10 launches from KSC LC-39A out of 13 launches have successful landing.

- There are 13 launches from KSC LC-39A, 33 launches from CCAFS-SLC40 and CCAFS-LC40

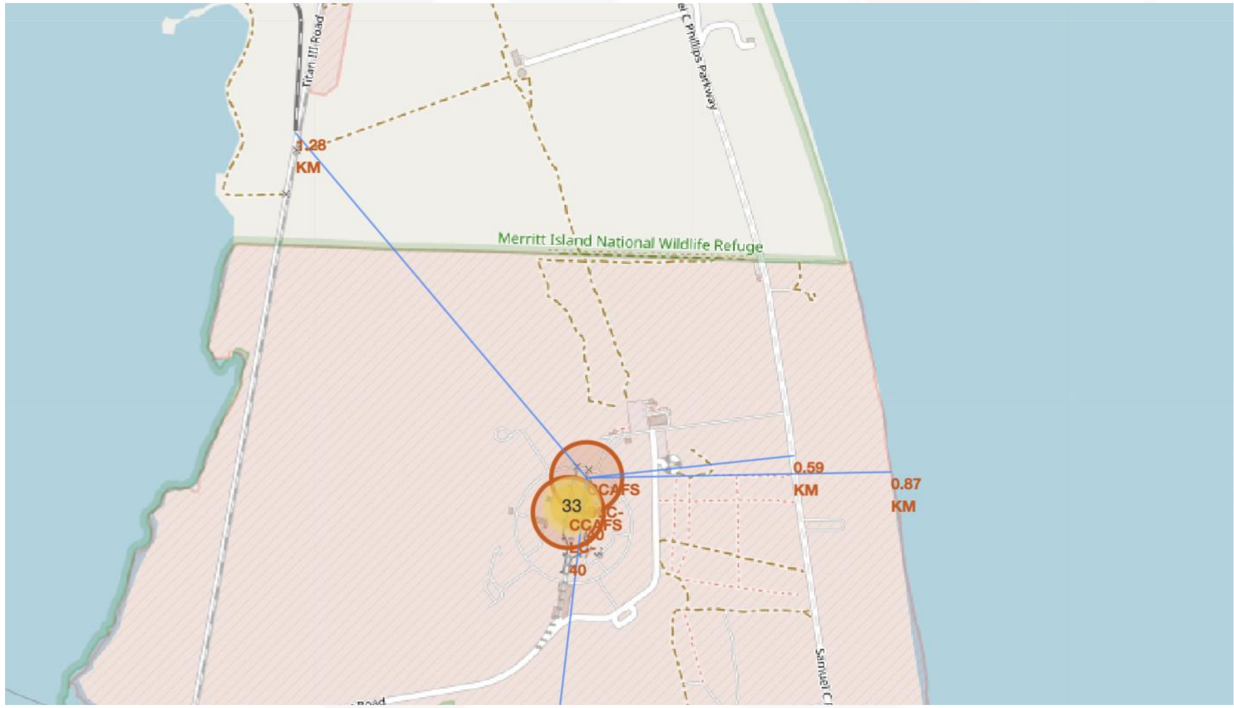
RESULTS

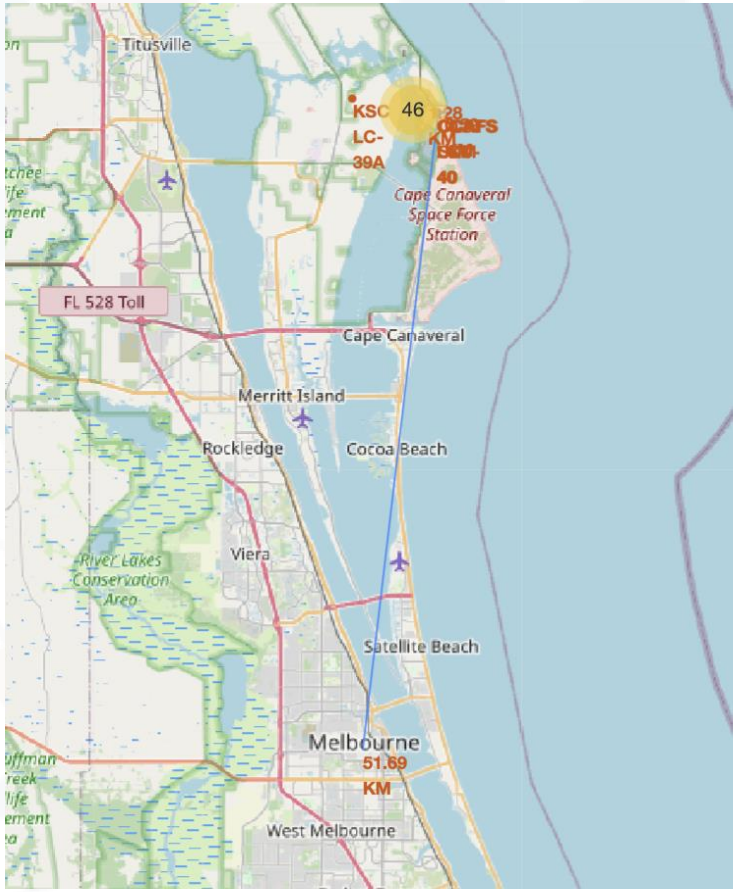
Launches from CCAFS-SLC40 and CCAFS-LC40

- There are 26 launches from CCAFS-LC40 and 7 of them have successful landing.
- There are 7 launches from CCAFS-SLC40 and 3 of them have successful landing.

RESULTS

Detailed location of CCAFS-SLC40 and CCAFS-LC40



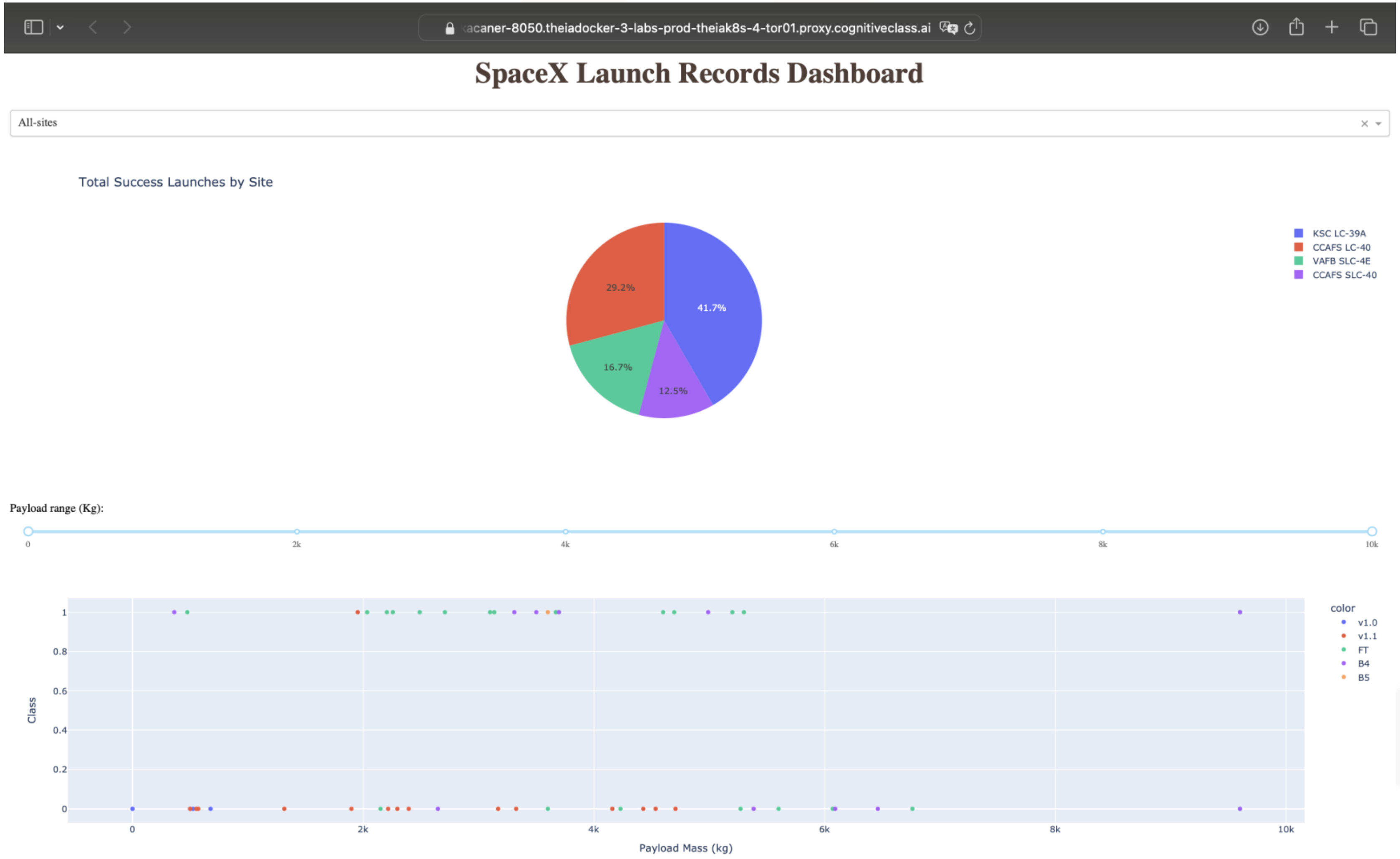


- CCAFS- SLC40 and CCAFS-LC40 launch sites are 0.87 km far from sea, 0.59 km far from highway, 1.28 km far from railway and 51.69 km far from city Melbourne

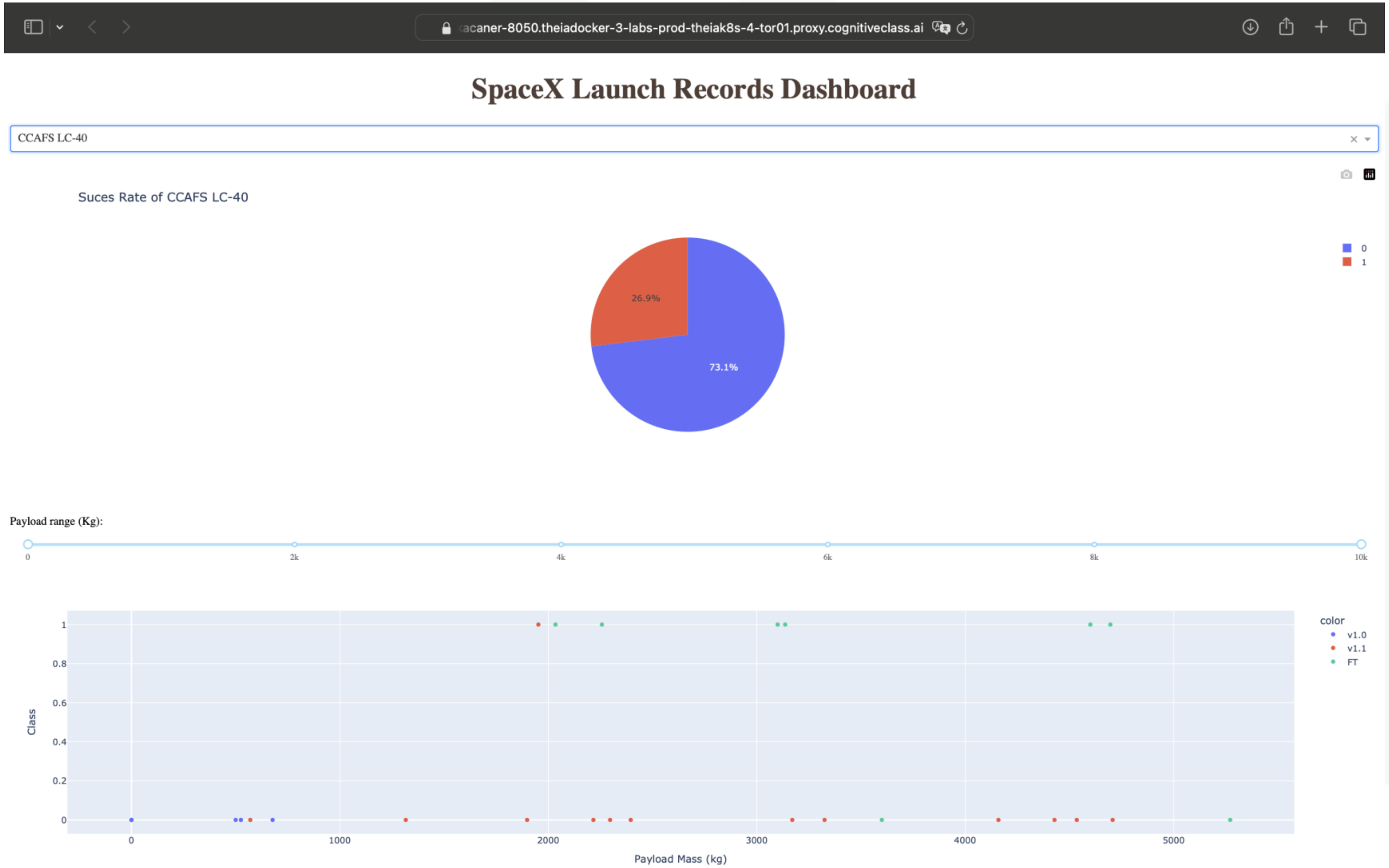
DASHBOARD

Dashboard app is built and successful landing rates according to launch site and payload mass can be observed in detail. For details look for appendix no:10 link

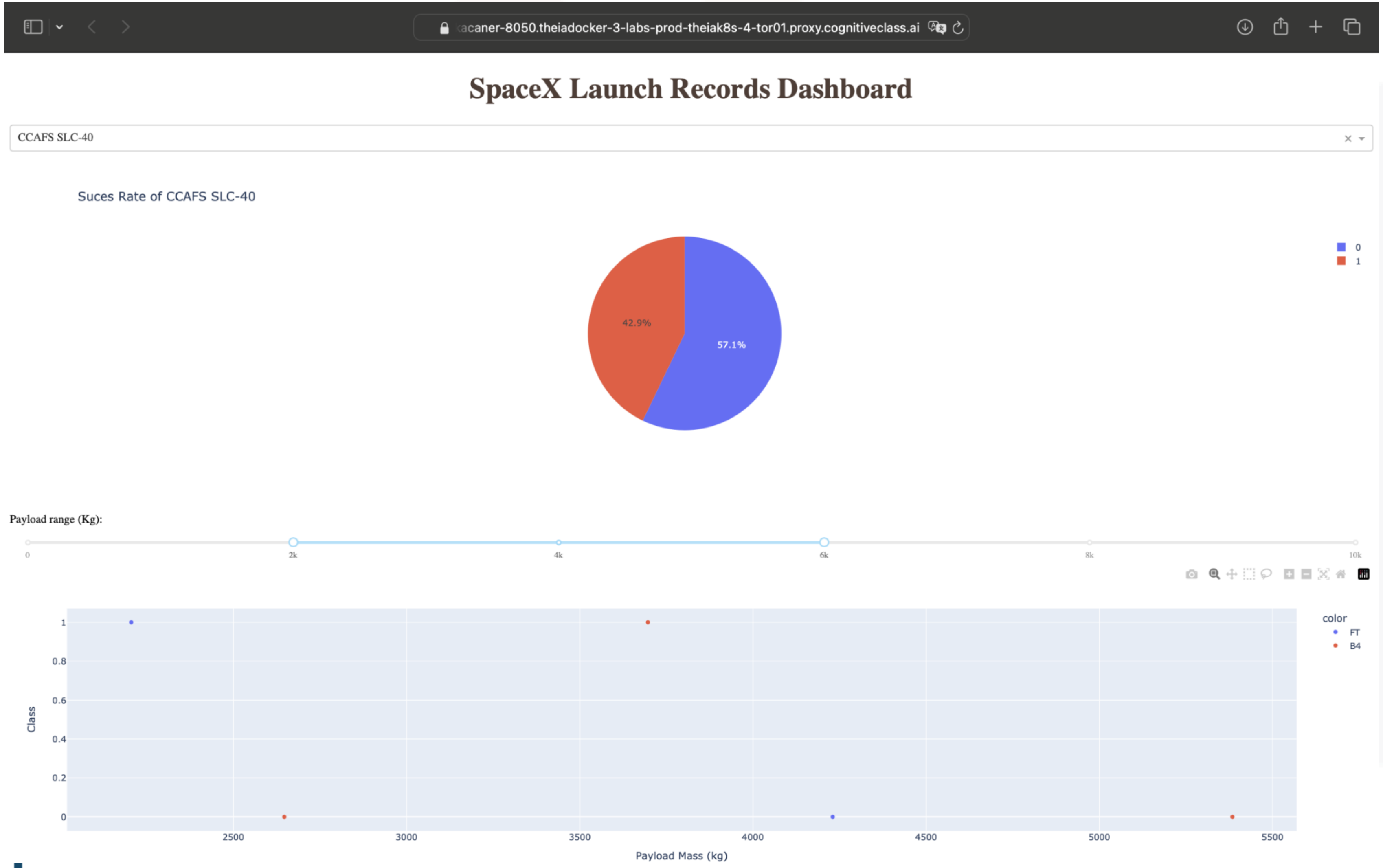
DASHBOARD TAB 1



DASHBOARD TAB 2



DASHBOARD TAB 3



RESULTS

As mentioned in methodology part, data is one-hot-encoded and standardized

```
[20]: parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()
logreg_cv = GridSearchCV(lr, parameters, cv=10)
logreg_cv.fit(X_train, Y_train)
```

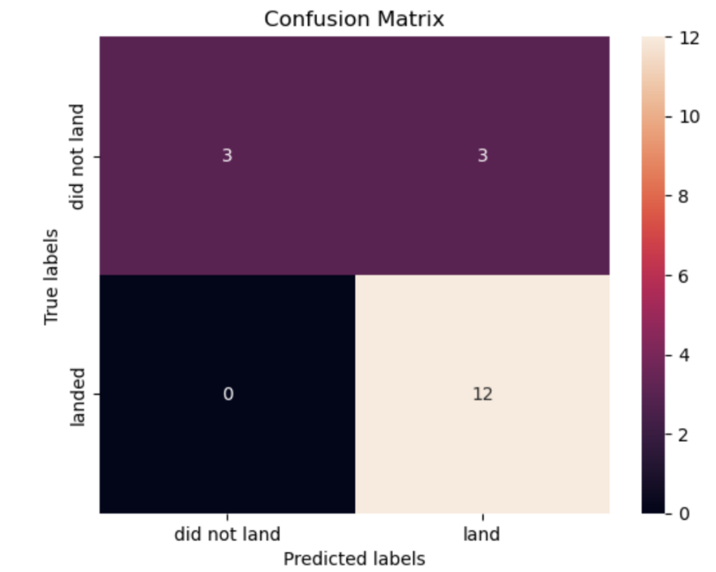
```
[20]: ▸ GridSearchCV
▸ estimator: LogisticRegression
    ▸ LogisticRegression
```

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

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

```
[22]: accuracy = logreg_cv.score(X_test, Y_test)
accuracy

[22]: 0.8333333333333334
```



- Logistic regression model is built with GridSearchCv method and best parameters are found. We can see best parameters and best score is 0.846.
- As seen in confusion matrix, there are 3 false positives.

RESULTS

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

```
[26]: svm_cv=GridSearchCV(svm, parameters, cv= 10)
      svm_cv.fit(X_train, Y_train)
```

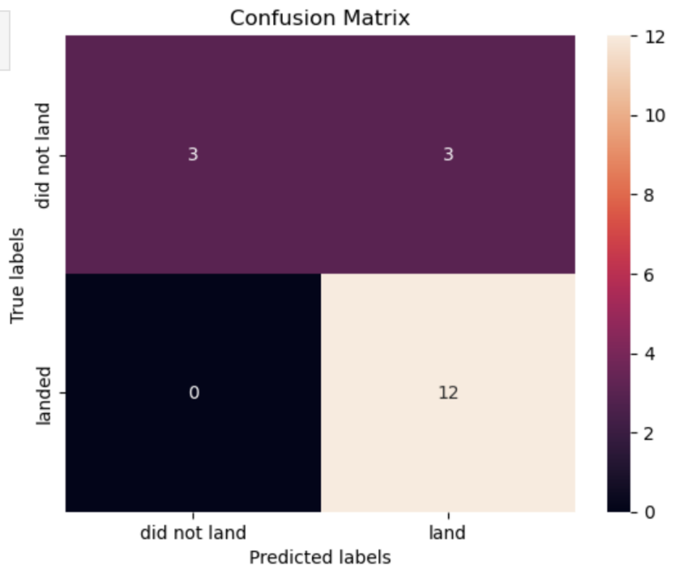
```
[26]: ▸ GridSearchCV
      ▸ estimator: SVC
          ▸ SVC
```

```
[27]: 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
```

```
[28]: accuracy = svm_cv.score(X_test, Y_test)
      accuracy

[28]: 0.8333333333333334
```



- Support vector machine model is built with GridSearchCv method and best parameters are found. We can see best parameters and best score is 0.848.
- As seen in confusion matrix, there are 3 false positives.

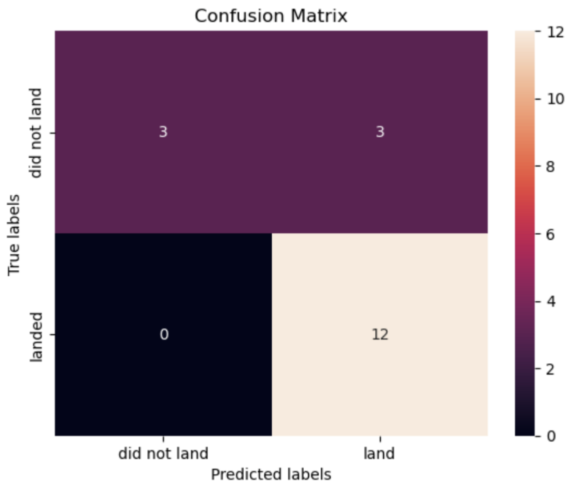
RESULTS

[33]:
```
tree_cv = GridSearchCV(tree, parameters, cv=10)
tree_cv.fit(X_train, Y_train)
```

[33]:
```
▸ GridSearchCV
▸ estimator: DecisionTreeClassifier
    ▸ DecisionTreeClassifier
```

[34]:
```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'spli
tter': 'random'}
accuracy : 0.8625
```



[35]:
```
accuracy = tree_cv.score(X_test, Y_test)
accuracy
```

[35]:
```
0.8333333333333334
```

- Decision tree model is built with GridSearchCv method and best parameters are found. We can see best parameters and best score is 0.8625.
- As seen in confusion matrix, there are 3 false positives.

RESULTS

```
[39]: knn_cv = GridSearchCV(KNN, parameters, cv=10)
      knn_cv.fit(X_train, Y_train)
```

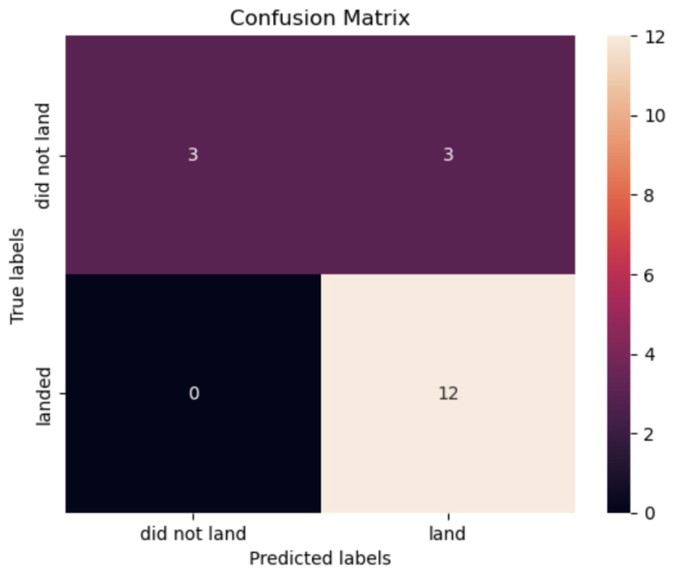
```
[39]: ▸ GridSearchCV
      ▸ estimator: KNeighborsClassifier
          ▸ KNeighborsClassifier
```

```
[41]: accuracy = knn_cv.score(X_test, Y_test)
      accuracy
```

```
[41]: 0.8333333333333334
```

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

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



- K nearest neighbor model is built with GridSearchCv method and best parameters are found. We can see best parameters and best score is 0.848.
- As seen in confusion matrix, there are 3 false positives.

RESULTS

```
[44]: scores = pd.DataFrame({"Method":["Logistic Regression", "Support Vector Machine", "Decision Tree", "K Nearest Neighbors"],
                            "Best Score":[logreg_cv.best_score_, svm_cv.best_score_, tree_cv.best_score_, knn_cv.best_score_]})
      scores
      #There are no major differences, confusion matrixes and test data scores are same. Decision tree performs better
      # according to best scores.
```

[44]:

	Method	Best Score
0	Logistic Regression	0.846429
1	Support Vector Machine	0.848214
2	Decision Tree	0.862500
3	K Nearest Neighbors	0.848214

DISCUSSION

- In view of these results,
  - When we create a mission in our company, we can make some decisions by using experience of Space X.
  - We can make decisions for increasing succesful landing rates.
  - These decisions may minimize the risks of failure.
  - Minimize the budget.
  - Has a potential to impact on growing faster.

OVERALL FINDINGS & IMPLICATIONS

Findings

- Landing success increases with flight number increases.
- Payload mass increased over time.
- CCAFS SLC 40 is used heavily.
- VAFB-SLC-4E is not used for heavy loads and for last missions.
- Higher payload mass missions have more successful landing rates.
- VLEO missions has the heaviest Payload.
- ES-L1, GEO, HEO, SSO orbit missions have %100 successful landing rates.
- ES-L1, SSO, HEO missions are made with lower Payload Mass and has %100 successful landing.
- Success had an increasing trend in general. However, there is drop between 2017-2018.
- Launch sites are close to sea, highway, railroads and city.
- Decision tree model that is built has 0.86 accuracy score.

OVERALL FINDINGS & IMPLICATIONS

Implications

- Landing success increases over time. It might be related with experience and engineering developments.
- Payload mass increased over time. It might also might be related with engineering developments or growing market. It should be researched to make decisions.
- If there is no obstacle to make heavy payload mass missions from engineering perspective, it can be considered.
- ES-L1, GEO, HEO, SSO orbit missions have %100 successful landing rates. These missions should be considered.
- If it is necessary to start with lower payload mass missions, ES-L1, SSO, HEO missions are made with lower Payload Mass and has %100 successful landing. It should be considered.
- Success had an increasing trend in general. However, there is drop between 2017-2018. The reason should be searched from engineering perspective.
- Launch sites are close to sea, highway, railroads and city. If a new launch site is planned to build. It should be considered.
- Decision tree model that is built has 0.86 accuracy score. Predictions should be made before mission plans.

CONCLUSION

- Besides engineering problems, strategies can be built according to Payload Mass, Launch Site, Orbit and other variables.
- Predictions will be made with Decision Tree Model according to company strategy.

APPENDIX

References:

- (1) https://pandas.pydata.org/
- (2) https://numpy.org/
- (3) https://matplotlib.org/
- (4) https://pypi.org/project/folium/
- (5) https://dash.plotly.com/
- (6) https://scikit-learn.org/stable/
- (7) https://docs.spacexdata.com/
- (8) https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- (9) https://www.space.com/29222-geosynchronous-orbit.html
- (10) https://github.com/kycnr/Final_Project_for_Coursera_DSP/tree/main