SPACEX FALCON 9 LAUNCH ANALYSIS

Nousheen Ali

05 – OCT - 2021

CONTENTS

EXECUTIVE SUMMARY

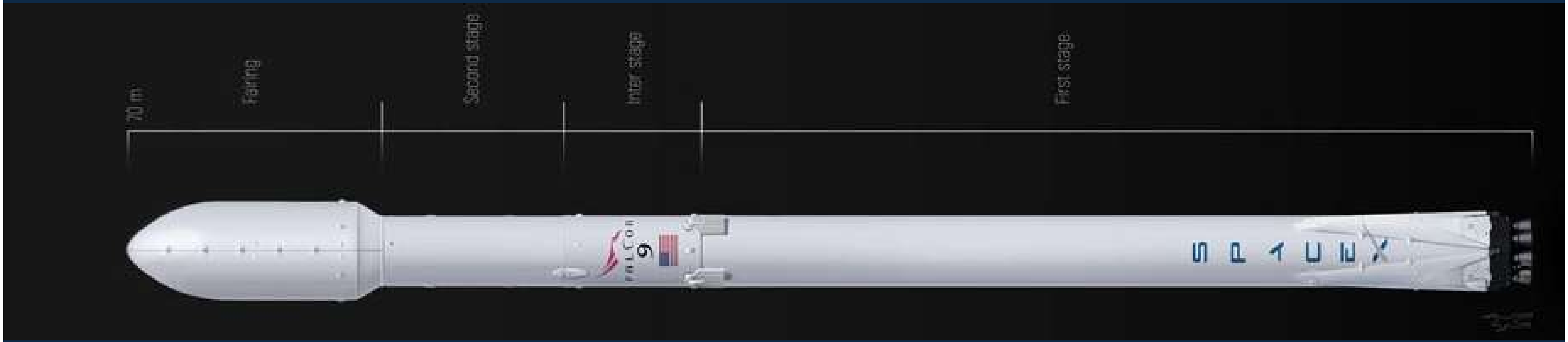
METHODOLGY

- Initially data was collected using REST API, Web scraping.
- The raw data was cleaned to obtain structured data through Data Wrangling.
- To obtain insights from the data, Exploratory Data Analysis was performed using SQL queries and Visualizations using graphs and charts.
- Interactive visual Analysis was performed using Folium Maps and Plotly Dashboard.
- Predictive Analysis was done to determine best Machine Learning Model.

RESULTS

- Exploratory Data Analysis Results.
- Interactive Visual Analysis Results.
- Predictive Analysis Results.

INTRODUCTION



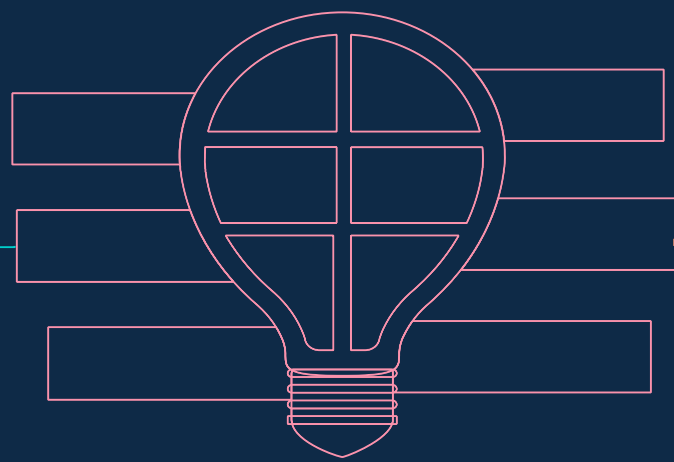
Space travel has always been a dream for the human race. In the last decade, however, this dream is turning to a reality with the help of companies such as SpaceX, Blue Origin and Virgin Galactic.

SpaceX has been the most successful in this field. SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars where other providers cost upward of 165 million dollars each. SpaceX Falcon 9 are able to provide relatively cheaper rates because it can recover the first stage of the rocket unlike its peers. The first stage of the rocket does most of the work and is much larger and expensive than the second stage.

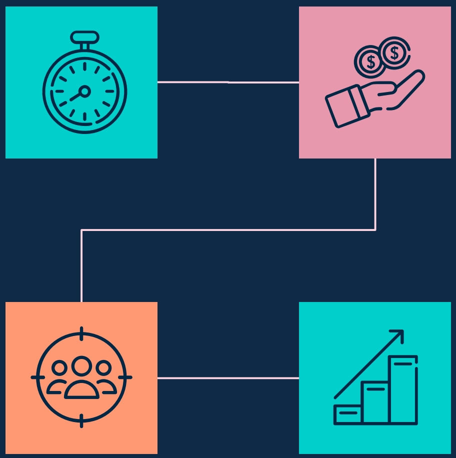
UNDERSTANDING THE PROBLEM

Which features affect the outcome of a launch?

Which Machine Learning Model gives the highest accuracy in predicting the success of future launches?



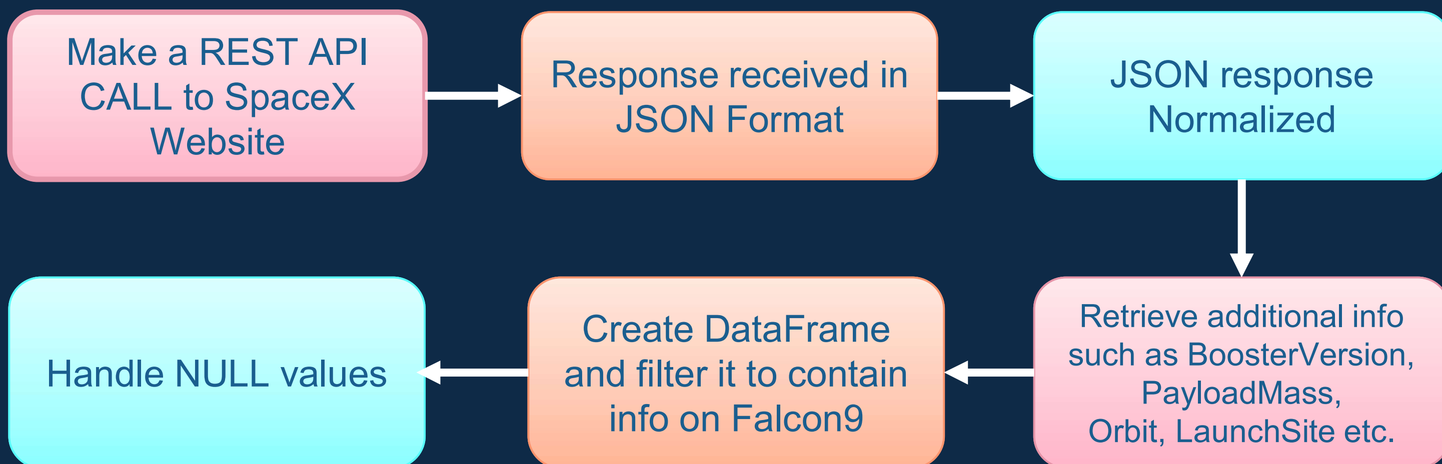
METHODOLOGY



DATA COLLECTION

1. Data collection using **SpaceX API**

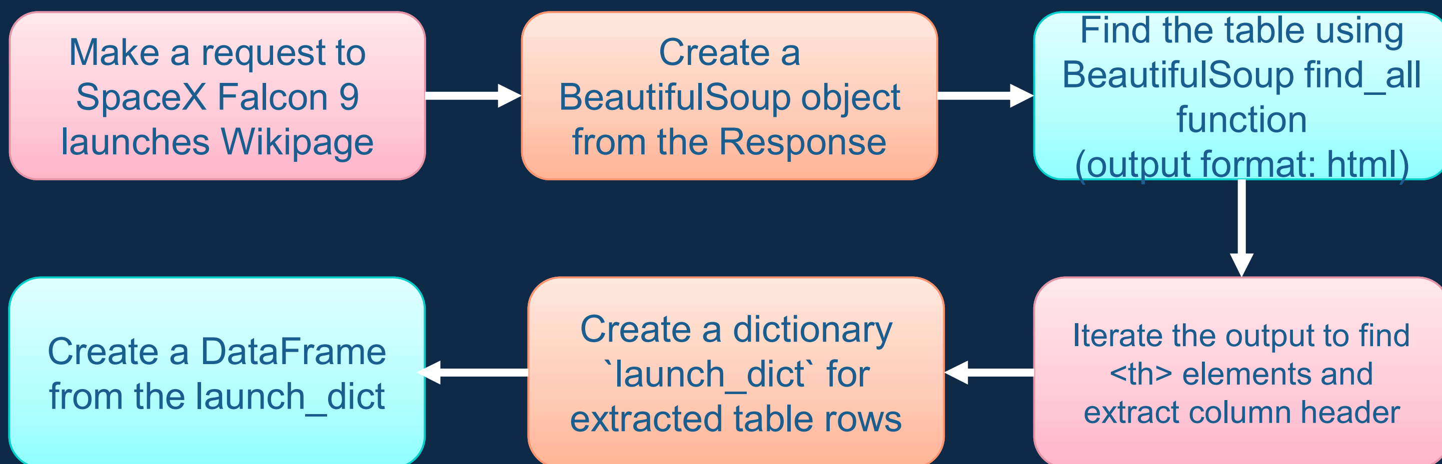
Representational State Transfer Application Program Interface (REST API/ simply API) refers to a set of protocols that a user can use to query a web service for data.



https://github.com/nousheenali/Coursera_Capstone/blob/master/Data%20Collection%20API.ipynb

2. Data collection using **Web Scraping**

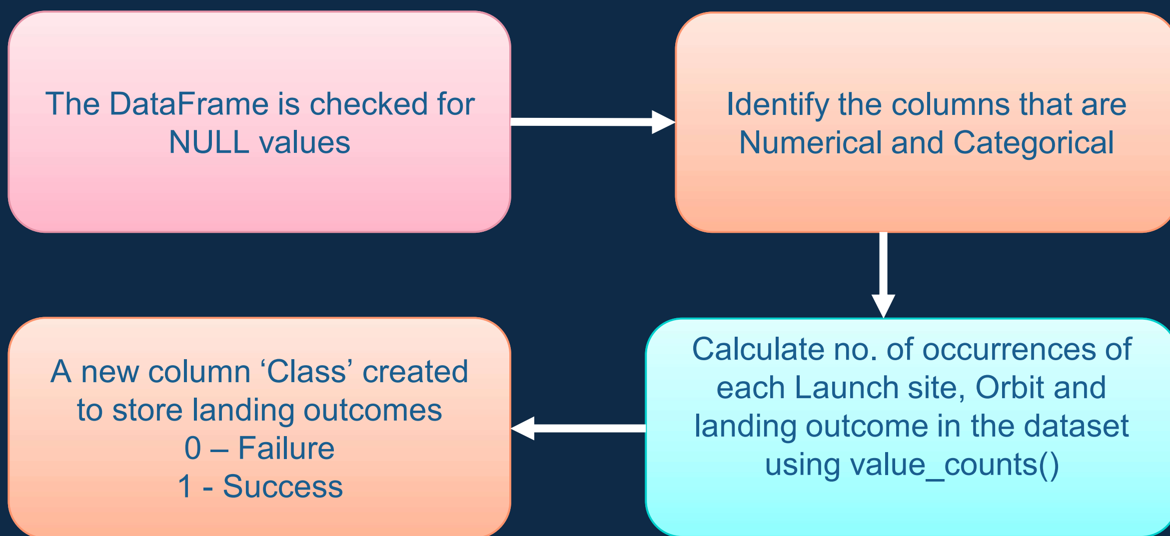
This is the process of extracting Data from a webpage. Once data is scarped it is exported into a more convenient format.



https://github.com/nousheenali/Coursera_Capstone/blob/master/Data%20Collection%20and%20Web%20Scraping.ipynb

DATA WRANGLING

The process of cleaning, structuring and enriching raw data into a desired format for better decision making in less time.



https://github.com/nousheenali/Coursera_Capstone/blob/master/Data%20Wrangling.ipynb

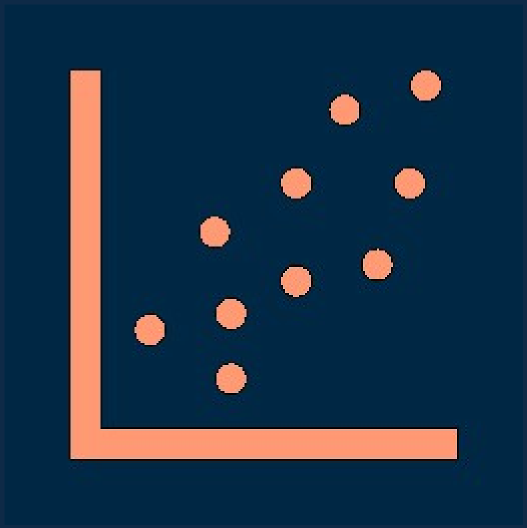
EXPLORATORY DATA ANALYSIS (EDA)

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

EDA with VISUALIZATION

Scatter Plot is used to find how the relation between two features will affect a certain outcome. The following relations were explored using scatter plot, to find its effect on the Launch Outcome(Class):

1. Flight Number vs Launch Site
2. Payload Mass vs Launch Site
3. Flight Number vs Orbit
4. Payload Mass vs Orbit



EDA with VISUALIZATION

A **Bar chart** performs a comparison of values across different subgroups of your data. In our case the bar graph was plotted to observe **Orbit Type vs Success Rate**

A **line plot** is a graph that shows frequency of data along a number line. Here we used it show the **Average Success Rate vs Years**

https://github.com/nousheenali/Coursera_Capstone/blob/master/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

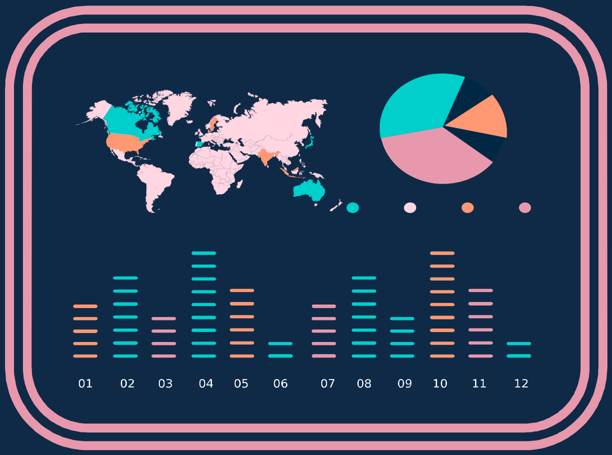
Data made available in .csv format is stored in a SQL table then exploratory analysis was performed and the following information was obtained using SQL queries.

1. *Unique launch site names.*
2. *Launch sites begin with 'CCA' (5 records)*
3. *Total payload mass carried by boosters launched by NASA (CRS)*
4. *Average payload mass carried by booster version F9 v1.1.*
5. *Date of first successful landing outcome in ground pad.*
6. *Names of the boosters which have success in drone ship (with 4000 < payload mass < 6000)*
7. *Total number of successful and failure mission outcomes*
8. *Names of the booster_versions that carried the maximum payload mass.*
9. *Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015*
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¶*

https://github.com/nousheenali/Coursera_Capstone/blob/master/EDA%20with%20SQL.ipynb

Interactive Visual Analytics

- Interactive Visual Analytics enables visualization, interaction, and automatic computation to facilitate insight generation from data.
- It presents data in such a manner that it is more understandable and appealing and helps to filter data in real time.
- To create a dashboard, we have used
  - Folium
  - Plotly Dash



Folium Map

We have used the Folium to

- Mark all Launch sites with the coordinates information from the dataset.
- Mark successful(green) and failed(red) launches on each site.
- Calculate and display distance between a particular launch site and a nearby location, with a polyline connecting both sites.

https://github.com/nousheenali/Coursera_Capstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Plotly Dash

Features included in the dashboard are:

- A Launch Site dropdown input component
- A pie chart that displays success rate based on the dropdown selection.
- A range slider to select payload mass.
- The scatter plot to show correlation between success rate and payload mass based on selection made on the slider and dropdown.

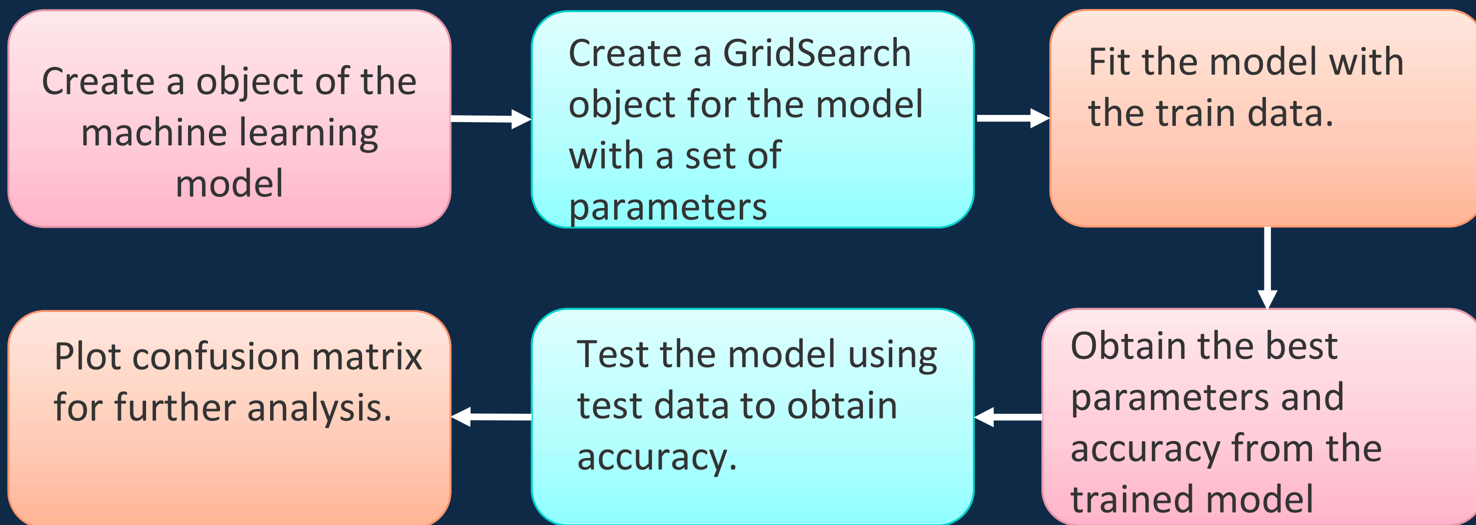
https://github.com/nousheenali/Coursera_Capstone/blob/master/spacex_dash_app.py

Predictive Analysis

Predictive Analysis makes predictions about future outcomes using historical data combined with statistical modeling and machine learning techniques.
Our assignment here is to build a model to predict if SpaceX Falcon 9 will land successfully or not.
The steps involve:

- Preprocessing: Create X and Y containing the input set and the output respectively. Standardize the X dataset.
- Train Test Split : Split X and Y to train and subsequently test the model.
- Model used : Logistic Regression, Support Vector Machine(SVM), Decision Tree Classifier, K Nearest Neighbour(KNN)
- For each of the models, we train the model and perform Gridsearch to find hyperparameters that helps the algorithm perform its best
- With the best hyperparameter values we find apt model with best accuracy using test data.

Developing a Machine Learning Model

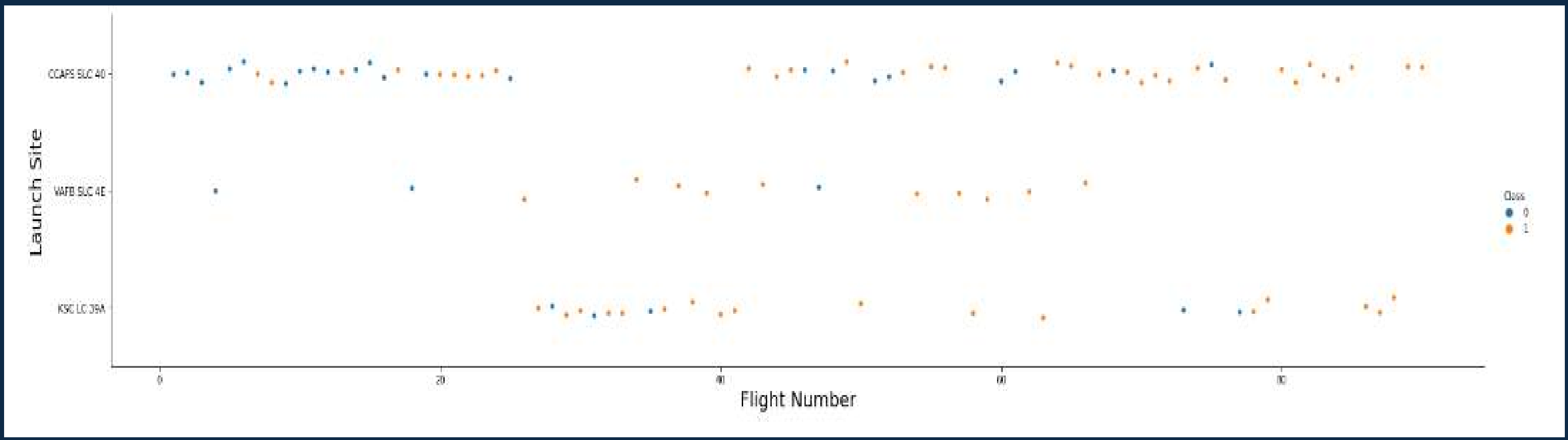


https://github.com/nousheenali/Coursera_Capstone/blob/master/Machine%20Learning%20Prediction.ipynb

RESULTS

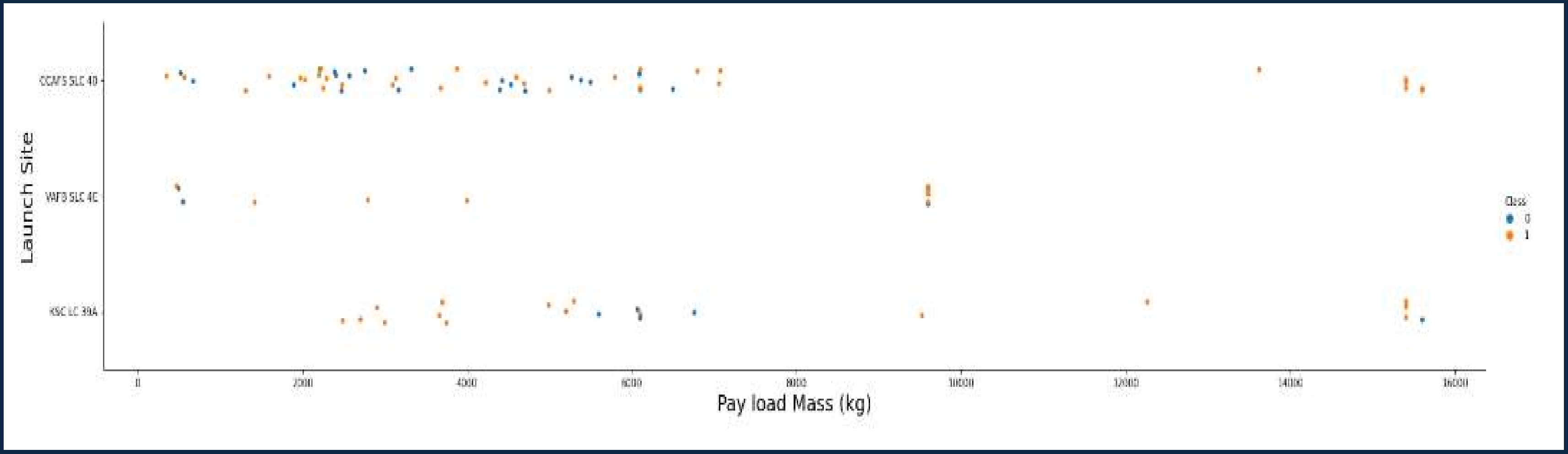
EXPLORARY DATA ANALYSIS – Using Visualization

Refer slides 11 and 12



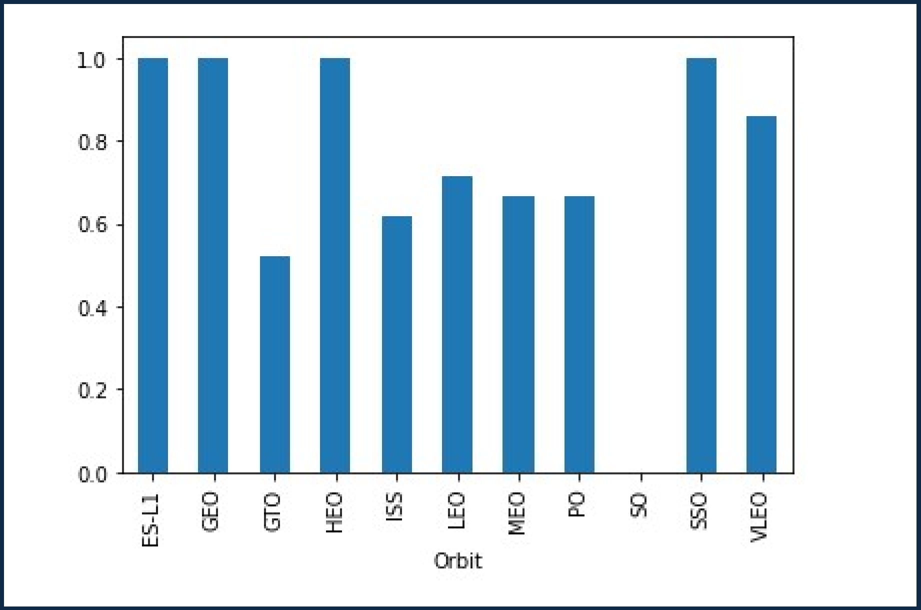
Flight Number vs Launch Site

As flight number increases , more successful launches occur for each site.



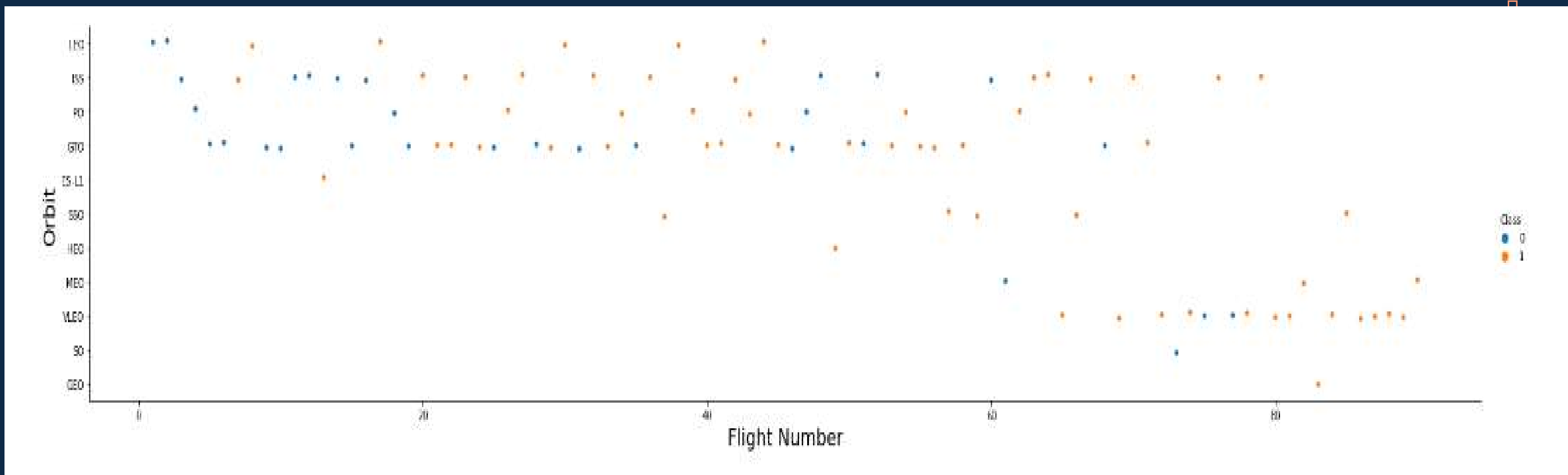
Flight Number vs Payload Mass

- As Payload Mass increases more success rate is higher for all 3 sites.
- For the site KSC LC 39A, success rate was high at for payload mass < 5000Kg



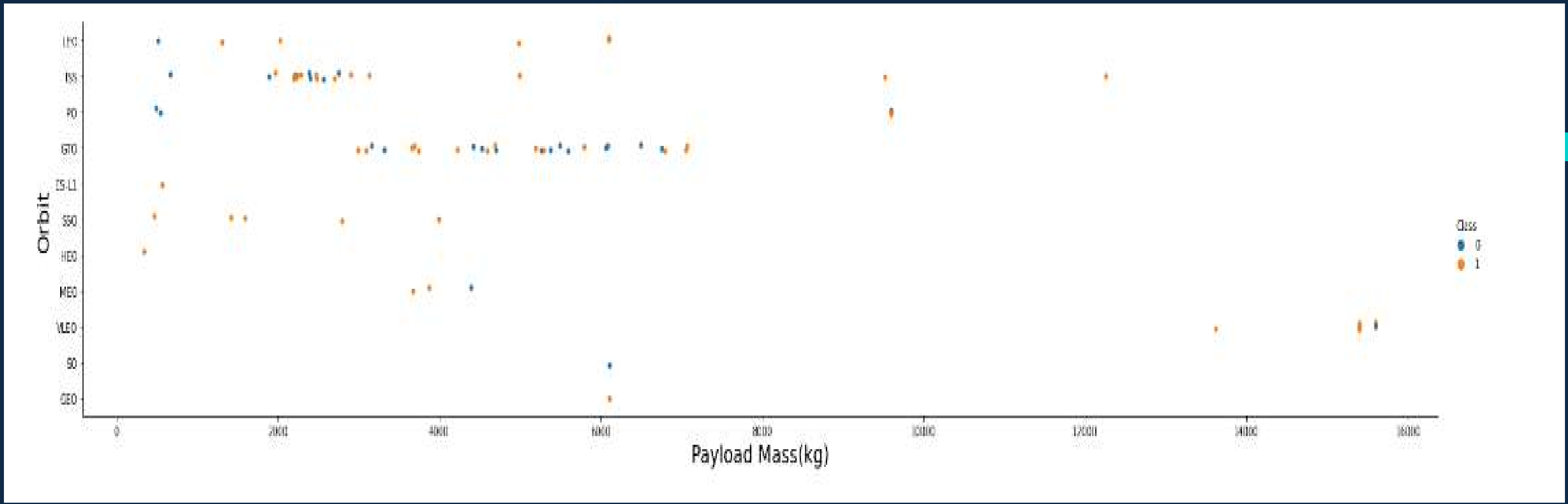
Orbit vs Success Rate

- GEO, HEO, SSO,ES-L1 orbits have 100% success rate
- GTO orbit has the lowest success rate of 50 %



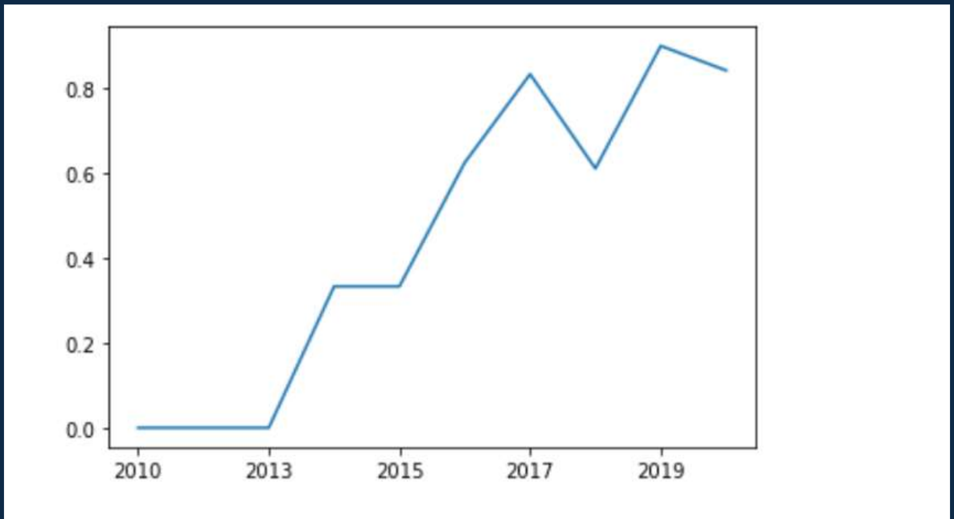
Flight Number vs Orbit

There doesn’t seem to be correlation between flight number and orbit.



Payload Mass vs Orbit

Payload Mass has different impacts on different orbits.



Average Success Rate vs Years

It is observed that the success rate since 2013 kept increasing till 2017. It showed a drop in year 2018 only to increase by 2019.

EXPLORARY DATA ANALYSIS – Using SQL

Refer slides 13

1. Unique Launch Sites

```
%%sql
select DISTINCT LAUNCH_SITE from SPACEXTBL;
```

Result:

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

2. Launch site names begin with `CCA`

```
%%sql
select * from SPACEXTBL
WHERE launch_site like 'CCA%'
LIMIT 5
```

Result:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

3. Total payload mass carried by boosters launched by NASA (CRS)

```
%%sql
select SUM(payload_mass__kg_) from SPACEXTBL
WHERE customer LIKE 'NASA (CRS)'
```

Result:

1
45596

4. Average payload mass carried by booster version F9 v1.1

```
%%sql
select AVG(payload_mass__kg_) from SPACEXTBL
where booster_version LIKE 'F9 v1.1'
```

Result:

1
2928

5. *Date of first successful landing outcome in ground pad.*

```
%%sql
select min(DATE) from SPACEXTBL
where landing__Outcome LIKE 'Success (ground pad)'
```

Result:

1
2015-12-22

6. Names of the boosters which have success in drone ship (with 4000 < payload mass < 6000)

```
%%sql
select booster_version from SPACEXTBL
WHERE landing__outcome LIKE 'Success (drone ship)'
AND payload_mass__kg_ BETWEEN '4000' and '6000'
```

Result:

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

7. Total number of successful and failure mission outcomes

```
%%sql
select COUNT(*) from SPACEXTBL
where landing__outcome LIKE 'Success%'
OR landing__outcome LIKE 'Failure%'
```

Result:

1
71

8. Names of the booster_versions that carried the maximum payload mass

```
%%sql
select booster_version from SPACEXTBL
WHERE payload_mass__kg_ = (select MAX(payload_mass__kg_) from SPACEXTBL)
```

Result:

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

9. Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
select booster_version, launch_site  from SPACEXTBL
WHERE landing__outcome LIKE 'Failure (drone ship)'
AND DATE LIKE '2015%'
```

Result:

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

```
%%sql

SELECT landing__outcome,COUNT(landing__outcome) AS Total FROM SPACEXTBL
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing__outcome ORDER BY Total DESC
```

Result:

landing__outcome	total
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Interactive Visual Analytics – Folium Map

Refer slides 14 and 15

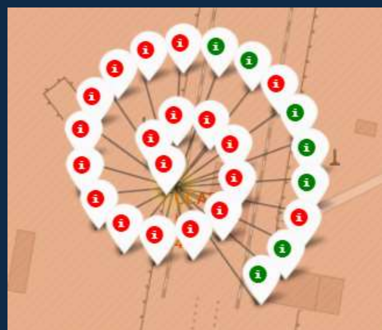
The map indicates all the launch sites using Folium markers.

Observations:

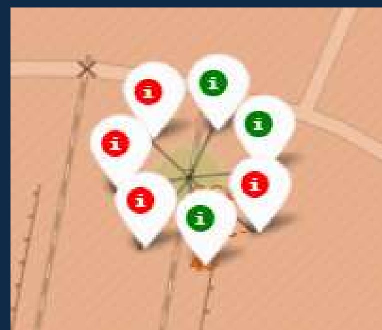
- Are all launch sites in proximity to the Equator line?
  They appear closer o the Tropic of cancer.

- Are all launch sites in very close proximity to the coast?
  Yes, all the sites are closer to the coast

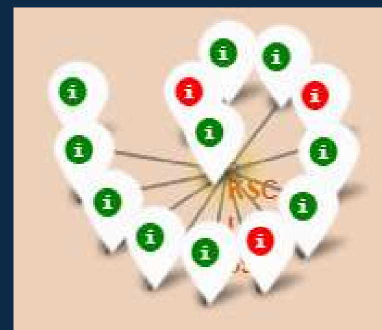
Successful and failed launches for each site



CCAFS LC-40



CCAFS SLC-40

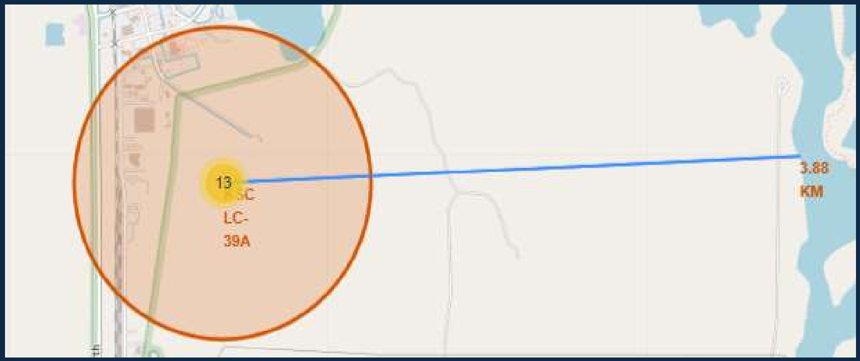


KSC LC-39A

VAFB SLC-4E

Here the failed launches have been indicated using RED and successful ones are indicated using GREEN. The color-labeled markers in marker clusters help easily identify which launch sites have relatively high success rates.

Distance Between a Launch Site and it’s Proximities



Distance from the site KSC LC-39A to a location on coastline : 3.88 KM

Distance from the site KSC LC-39A to the nearest railway : 0.72 KM
Distance from the site KSC LC-39A to the nearest highway: 0.84 KM

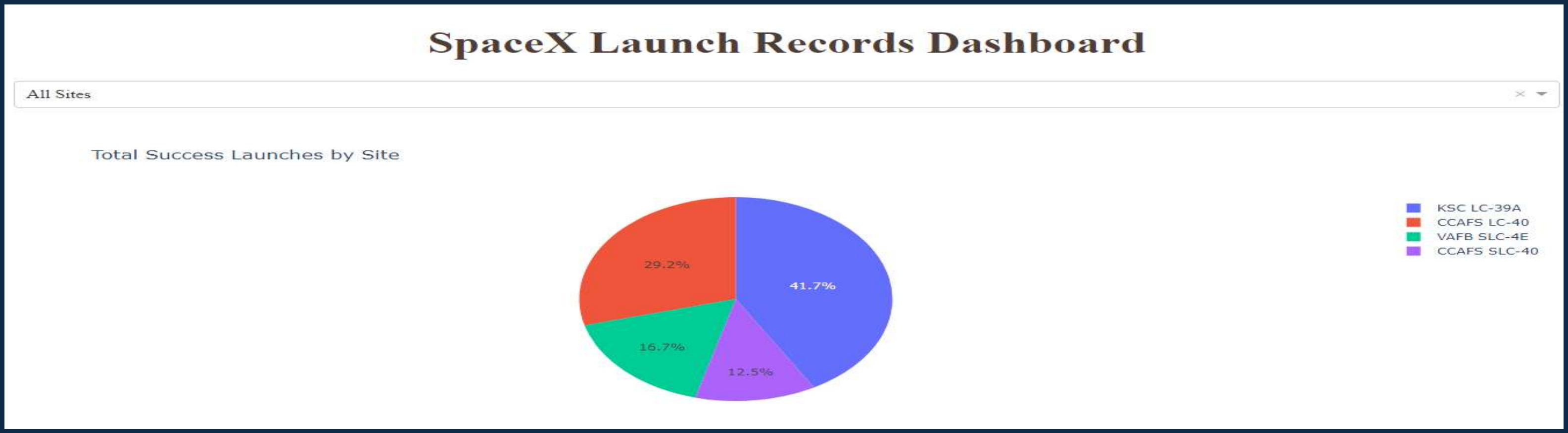
Distance from the site KSC LC-39A to the nearest city(Titusville): 16.28 KM

Observations

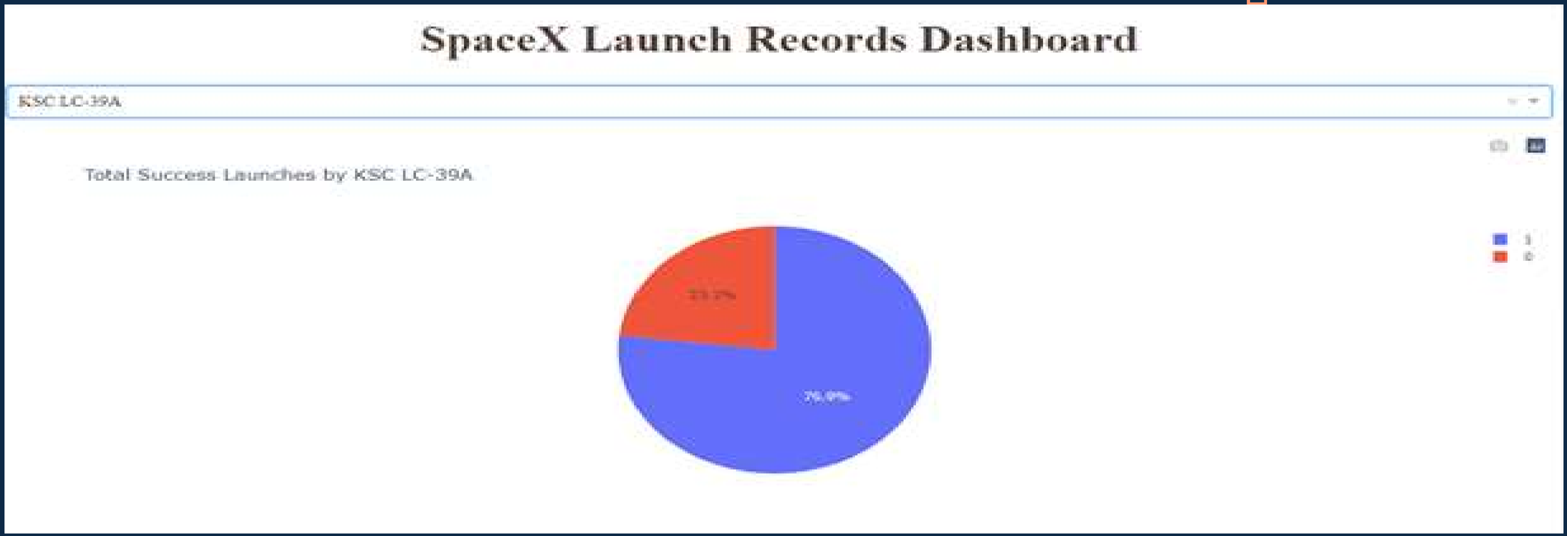
- Are launch sites in close proximity to railways?
  Yes, the closest railway line to site KSC LC-39A is within a kilometer.
- Are launch sites in close proximity to highways?
  Yes, the closest highway line to site KSC LC-39A is within a kilometer.
- Are launch sites in close proximity to coastline?
  Yes, all the launch sites are near the coastline.
- Do launch sites keep certain distance away from cities?
  Yes, all cities are at a considerable distance from the launch sites. Eg: Closest city (Titusville) is 16.28 KM away from the launch site KSC LC-39A.

Interactive Visual Analytics – Plotly Dash

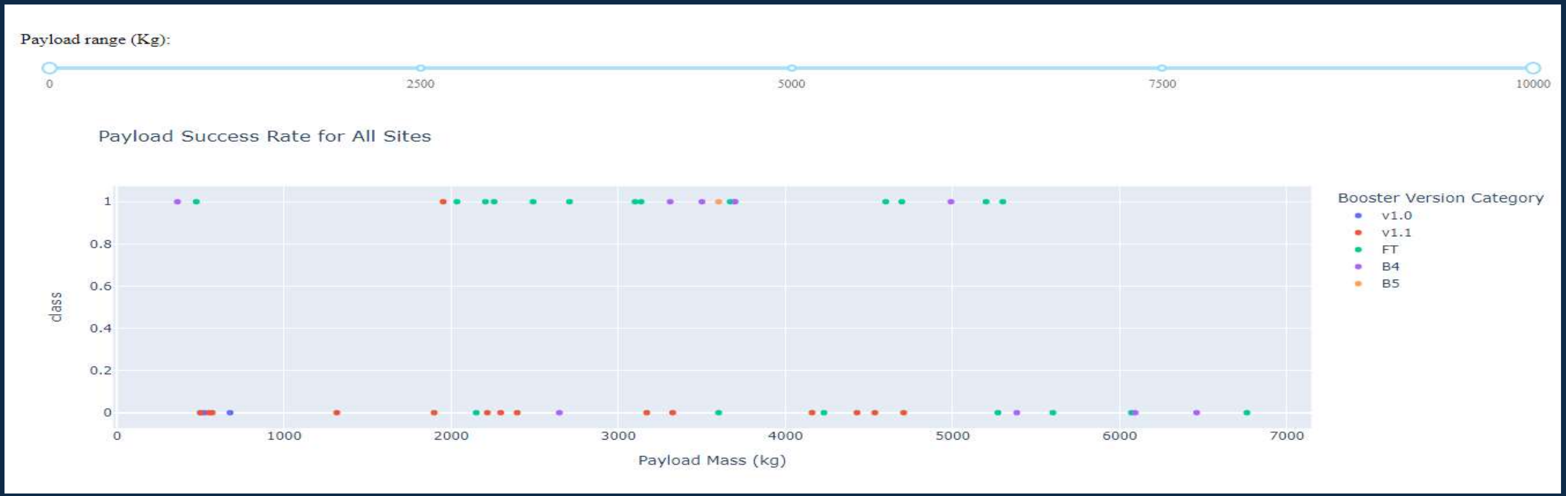
Refer slides 14 and 15



- The site KSC LC-39A has the highest rate of success at 41.7%.
- The site CCAFS SLC-40 has the lowest rate of success at 12.5%.



- If any one site is selected, the pie chart displays the percentage of successes and failures.
- The launch site with highest success ratio KSC LC-39A, has had 76.9% successful launches and 23.1% failures.



Payload Mass vs Class(success rate) scatter plot

- Beyond the payload mass of 5500kg, the launches have been failures. Booster versions FT and B4 have had the more successful launches below this range.
- Booster version V1.1 has mostly ended up in failure irrespective of the payload mass.

Predictive Analysis

Logistic Regression

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

```
svm_cv = GridSearchCV(svm,parameters)
svm_cv.fit(X_train, Y_train)
```

```
GridSearchCV(estimator=SVC(),
             param_grid={'C': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
       1.00000000e+03]),
                         'gamma': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
       1.00000000e+03]),
                         'kernel': ('linear', 'rbf', 'poly', 'rbf', 'sigmoid')})
```

```
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
```

- Using training data on Gridsearch object we obtain the best parameters.
  'C': 0.01'
  'penalty': 'l2'
  'solver': 'lbfgs'

- With the best parameters, logistic regression model gives an accuracy of 83.33% for test data.

```
logreg_cv.score(X_test,Y_test)
```

```
0.8333333333333334
```

Support Vector Machine (SVM)

```
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()
```

```
tree_cv = GridSearchCV(tree, parameters, cv =10)
tree_cv.fit(X_train, Y_train)
```

```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

- Using training data on Gridsearch object we obtain the best parameters.
  'C': 0.03162277660168379
  'gamma': 0.001
  'kernel': 'linear’

- With the best parameters, SVM model gives an accuracy of 83.33% for test data.

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

```
tuned hpyerparameters :(best parameters)  {'C': 0.03162277660168379, 'gamma': 0.001, 'kernel': 'linear'}
accuracy : 0.8342857142857142
```

```
svm_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

Decision Tree Classifier

```
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()
```

```
tree_cv = GridSearchCV(tree, parameters, cv =10)
tree_cv.fit(X_train, Y_train)
```

```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

Using training data on Gridsearch we obtain the best parameters.

'criterion': 'entropy' 'max_depth': 8
'max_features': 'auto' 'min_samples_leaf': 4
'min_samples_split': 5 'splitter': 'random'

With the best parameters, Decision Tree model gives an accuracy of 83.3% for test data.

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

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 8, 'max_features': 'auto', 'min_samples_leaf': 4, 'mi
n_samples_split': 5, 'splitter': 'random'}
accuracy : 0.875
```

```
tree_cv.score(X_test,Y_test)
```

```
0.8333333333333334
```

K Nearest Neighbor (KNN)

```
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()
```

```
knn_cv = GridSearchCV(KNN, parameters, cv =10)
knn_cv.fit(X_train, Y_train)
```

```
GridSearchCV(cv=10, estimator=KNeighborsClassifier(),
             param_grid={'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                         'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
                         'p': [1, 2]})
```

```
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
```

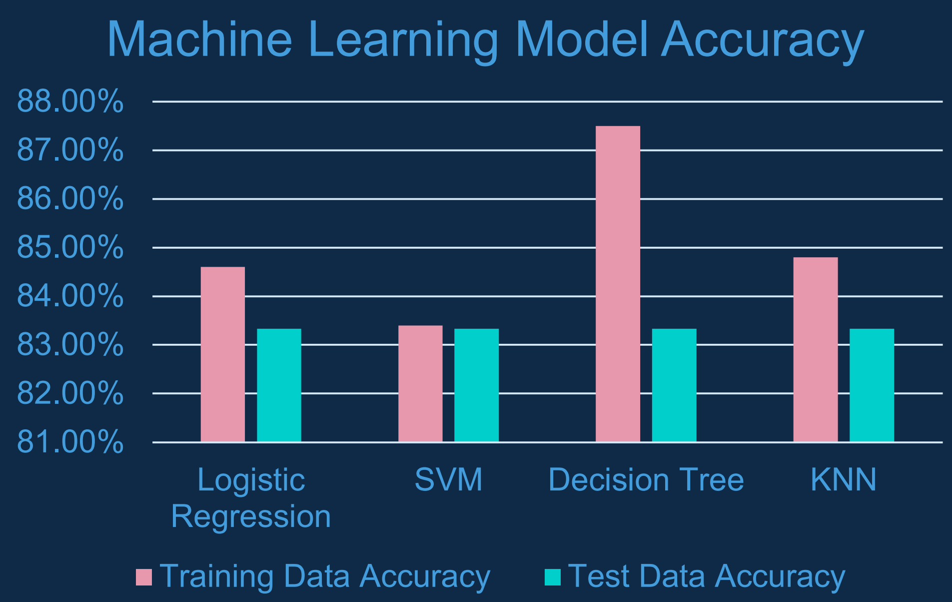
```
knn_cv.score(X_test,Y_test)
```

```
0.8333333333333334
```

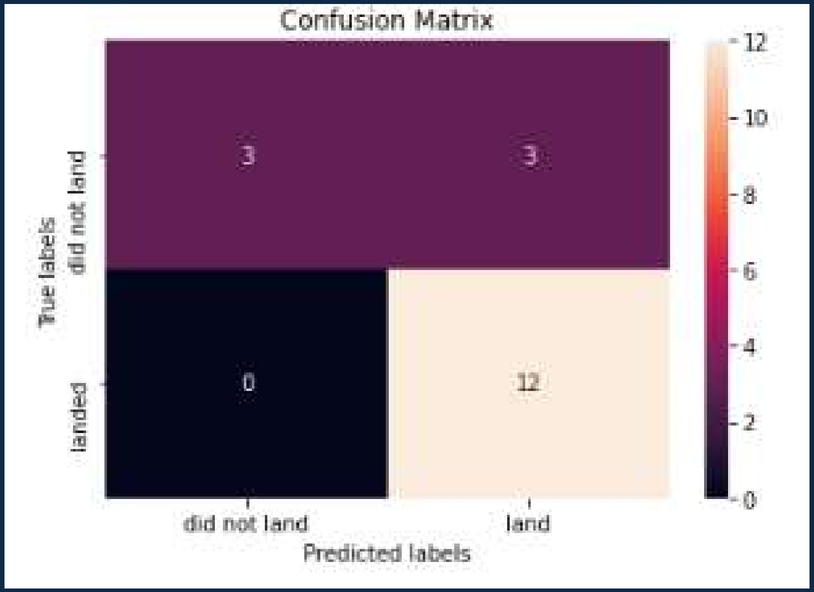
- Using training data on Gridsearch we obtain the best parameters.
  'algorithm': 'auto'
  'n_neighbors': 10
  'p': 1

- With the best parameters, KNN model gives an accuracy of 83.3% for test data.

Best Machine Learning Model



All the models perform equally well with an accuracy of 83.3% for test data.



All the models have yielded the same results in the confusion matrix.

Examining the confusion matrix, we see that the model can distinguish between the different classes. We see that the major problem is false positives.

CONCLUSION

- The features FlightNumber, PayloadMass, Orbit and LaunchSite influence the success/failure of the launch.
- All launch sites are located near the coastline away from cities.
- For booster versions, launches are more likely to be successful when payload mass is less than 6000kg.
- All the machine learning models(Logistic regression, SVM, Decision Tree and KNN) are equally good as they all provide the same accuracy of 83.3%.

THANK YOU!