Winning Space Race with Data Science

Cuneyt Harputluoglu
20.11.2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Predictive analysis

- Summary of all results
  - EDA results
  - Predictive analysis

Introduction

- Project background and context
  - Spacex advertises Falcon 9 rocket launches in its webpage, with a cost of 62M $ but competitors with a cost of 165M$/each.Success is because of reusable first stage module.
- Problems you want to find answers
  - Project task is to predict if the future first stage of the SpaceX Falcon9 rocket will land successfully or not

Section 1

Methodology

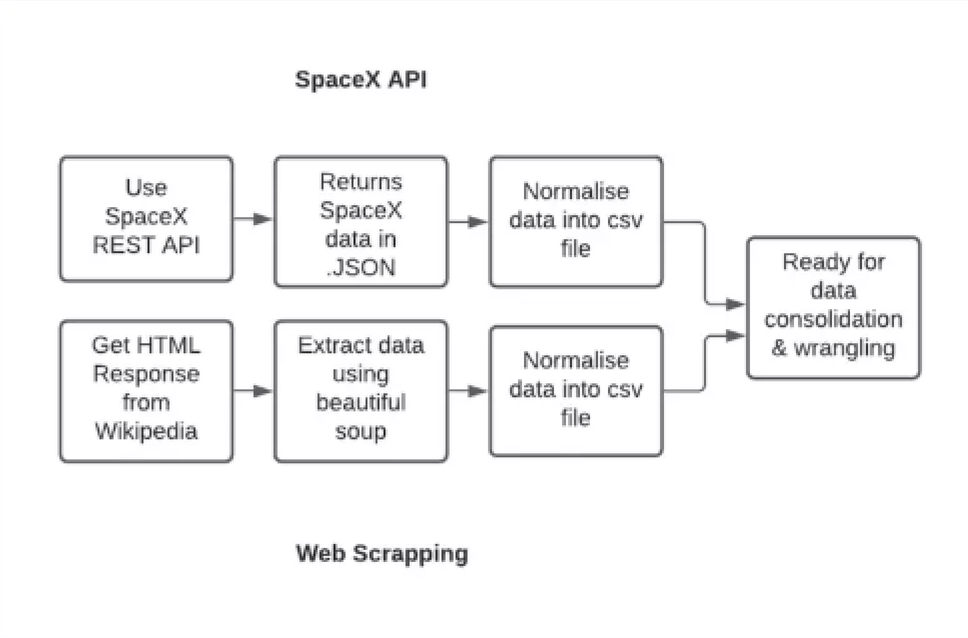
Methodology

Executive Summary

- Data collection methodology:
  - Data collected over SpaceX API
  - Webscraping over Wikipedia
- Perform data wrangling
  - One hot encoding data for ML and data cleaning,new field generation for ML
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic Reg., KNN, SVM and DecisionTree were performed to find the best predictor

Data Collection

- Data were collected from SpaceX API
- API gives many useful info about launches with rocket used,payload,launch specs, Landing specs and outcome data.
- REST API endpoints, or URL starts with «https://api.spacexdata.com/v4/»
- For webscraping, BeautifulSoup was used to scrape wikipedia.
- Process flow is illustrated at right.



Data Collection – SpaceX API

- Data preparation is explained on right side.

- https://github.com/cuneytharp/testrepo/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Request for connection to API and DF creation

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)

data = pd.json_normalize(response.json())
```

Dataset construction dictionary

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Filtering Falcon-9 data and filling nan values with mean

```
data_falcon9 = df[df["BoosterVersion"]=="Falcon 9"]

PayloadMass_mean = data_falcon9["PayloadMass"].mean()
data_falcon9["PayloadMass"].replace(np.nan , PayloadMass_mean,inplace=True)
# Replace the np.nan values with its mean value
data_falcon9.to_csv("dataset_part_1.csv",index=False)
```

Data Collection - Scraping

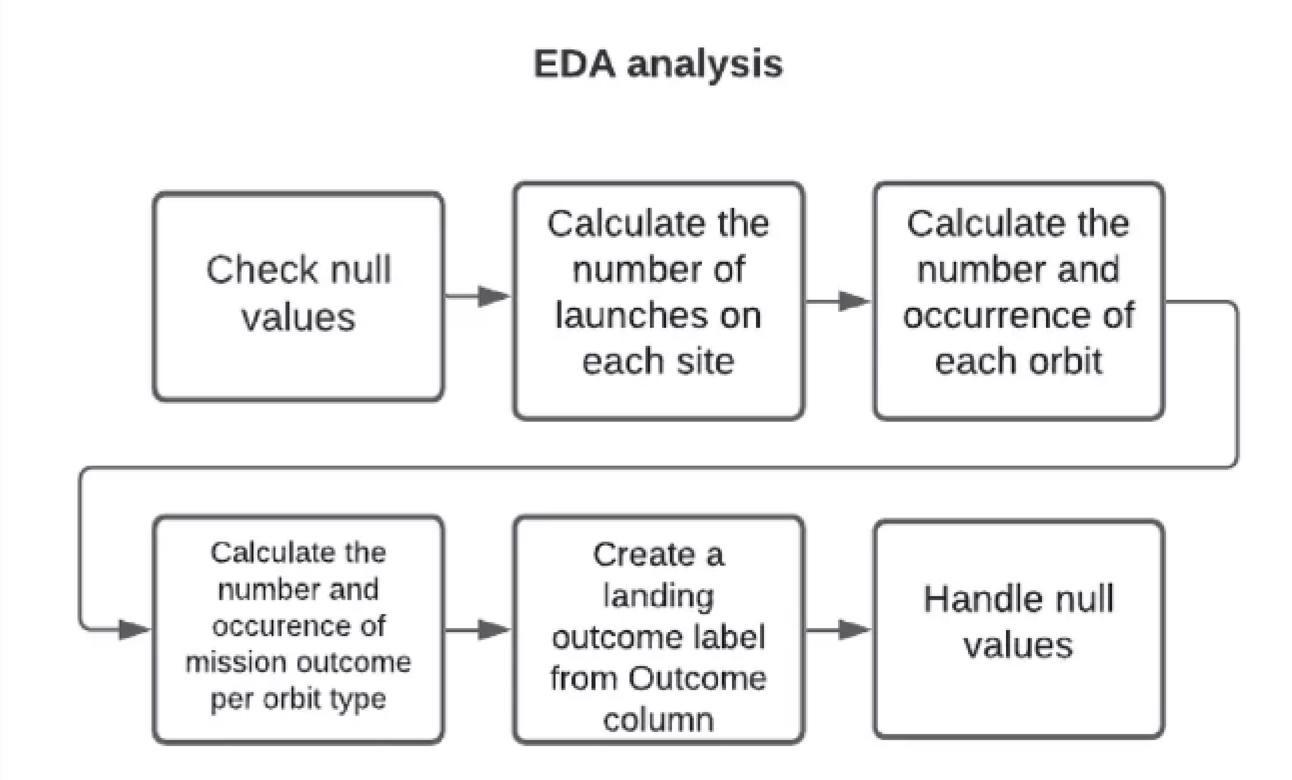
- Web Scraping from Wiki

- https://github.com/cuneytharp/testrepo/blob/main/jupyter-labs-webscraping.ipynb

1. Request and get response from HTML
2. Create BeautifulSoup object
3. Find Tables
4. Get column names
5. Create an empty dictionary
6. Append data to dictionary
7. Create df from dictionary
8. DF to .CSV

Data Wrangling

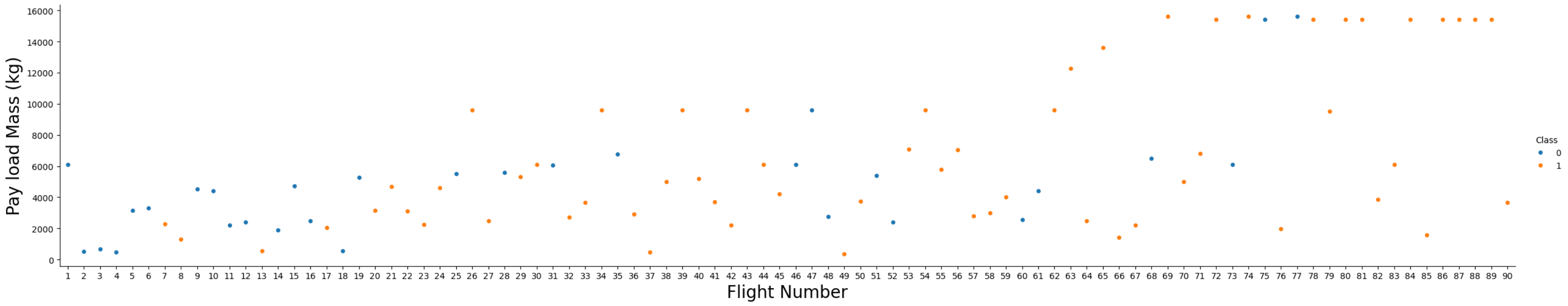
- Data Wrangling process is as illustrated at below.

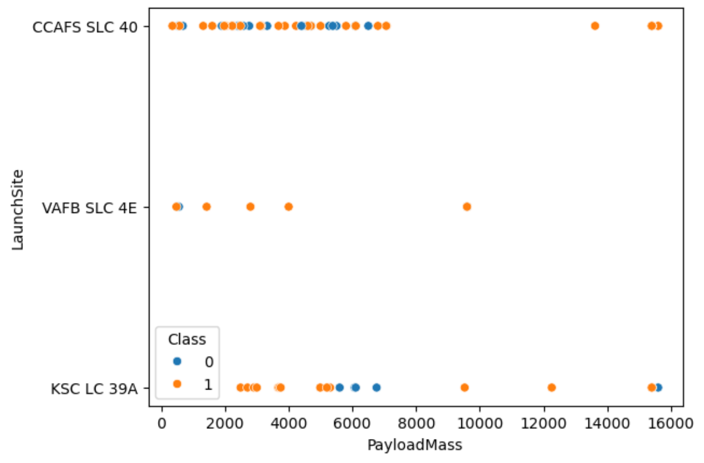


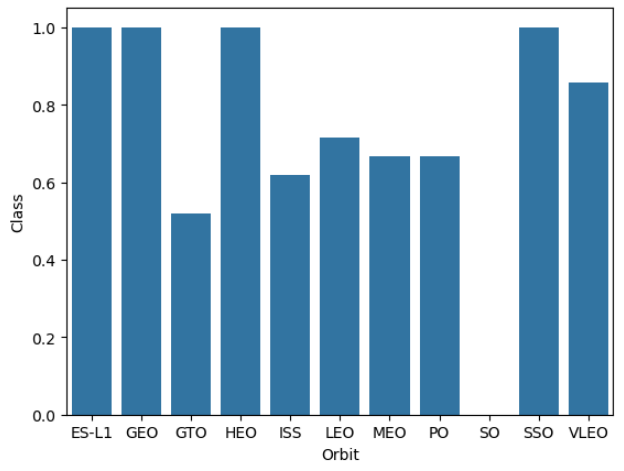
- https://github.com/cuneytharp/testrepo/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

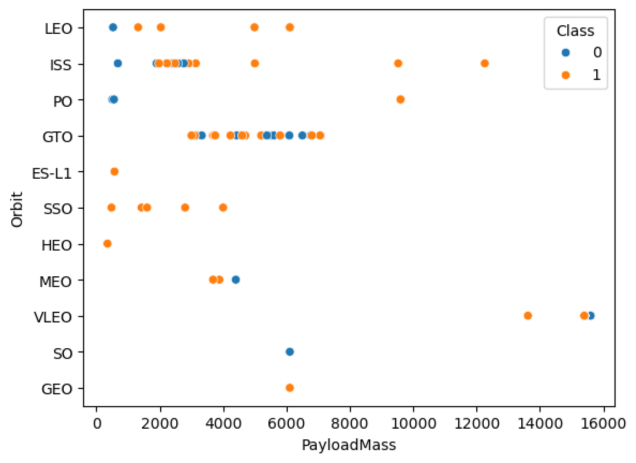
EDA with Data Visualization

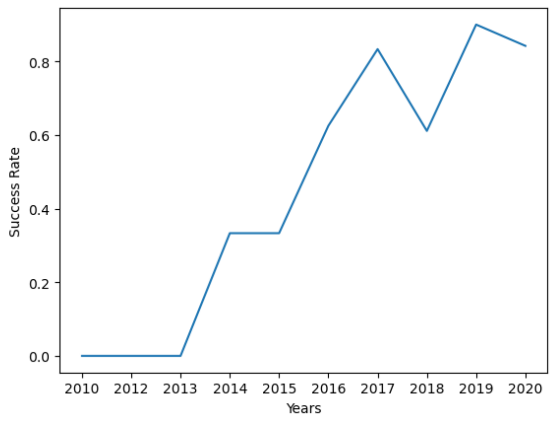
- Catplot was used to present flightnumber/payload and payloadmass/launchSite relation.Barchart was used to show orbit based success rates(1=100%).Scatterplot was used to show payloadmass and orbit relation.Lineplot was used to show yearly success rate











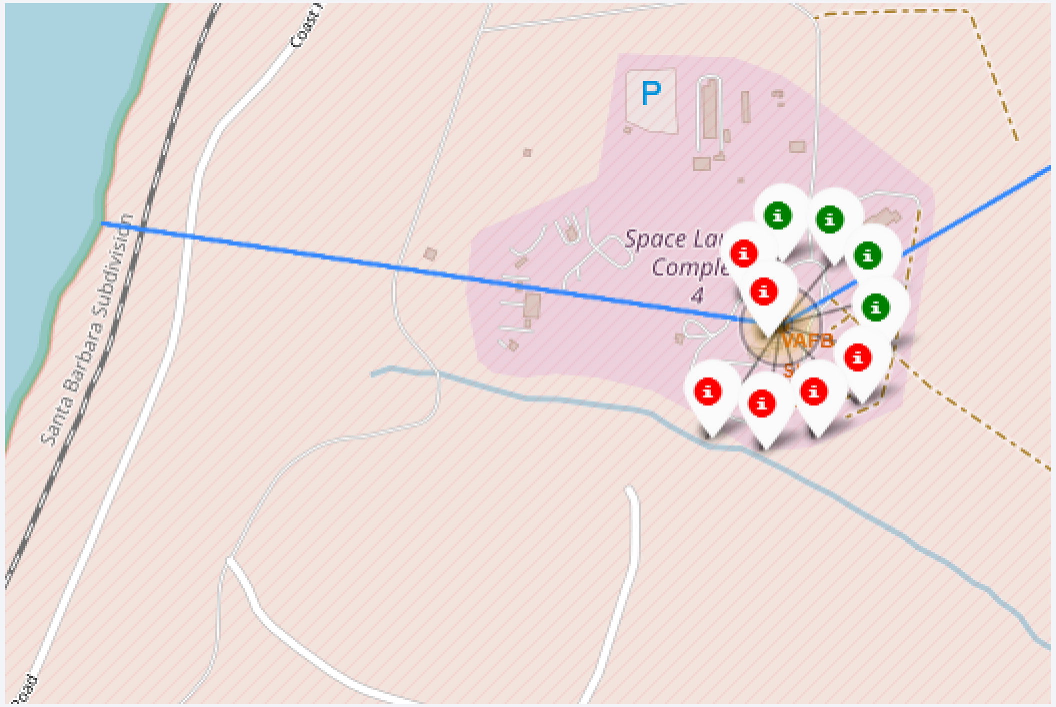
- https://github.com/cuneytharp/testrepo/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- **Display the names of the unique launch sites in the space mission**
- %sql select DISTINCT launch_site from SPACEXTABLE
- **Display 5 records where launch sites begin with the string 'KSC'**
- %sql select * from SPACEXTABLE WHERE launch_site LIKE 'KSC%' LIMIT 5
- **Display the total payload mass carried by boosters launched by NASA (CRS)**
- %sql SELECT sum(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE Customer LIKE "NASA (CRS)"
- **Display average payload mass carried by booster version F9 v1.1**
- %sql SELECT AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE Booster_Version = "F9 v1.1"
- **List the date where the succesful landing outcome in drone ship was acheived.**
- %sql SELECT Date FROM SPACEXTABLE WHERE Landing_Outcome ="Success (drone ship)"
- **List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000**
- %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome ="Success (ground pad)" AND PAYLOAD_MASS__KG_>4000 AND PAYLOAD_MASS__KG_<6000
- **List the total number of successful and failure mission outcomes**
- %sql SELECT COUNT(Mission_Outcome) as "Successful Mission" FROM SPACEXTABLE WHERE Mission_Outcome ="Success"
- %sql SELECT COUNT(Mission_Outcome) as "Failed Mission" FROM SPACEXTABLE WHERE Mission_Outcome !="Success"
- https://github.com/cuneytharp/testrepo/blob/main/jupyter-labs-eda-sql-edx_sqllite.ipynb

Build an Interactive Map with Folium

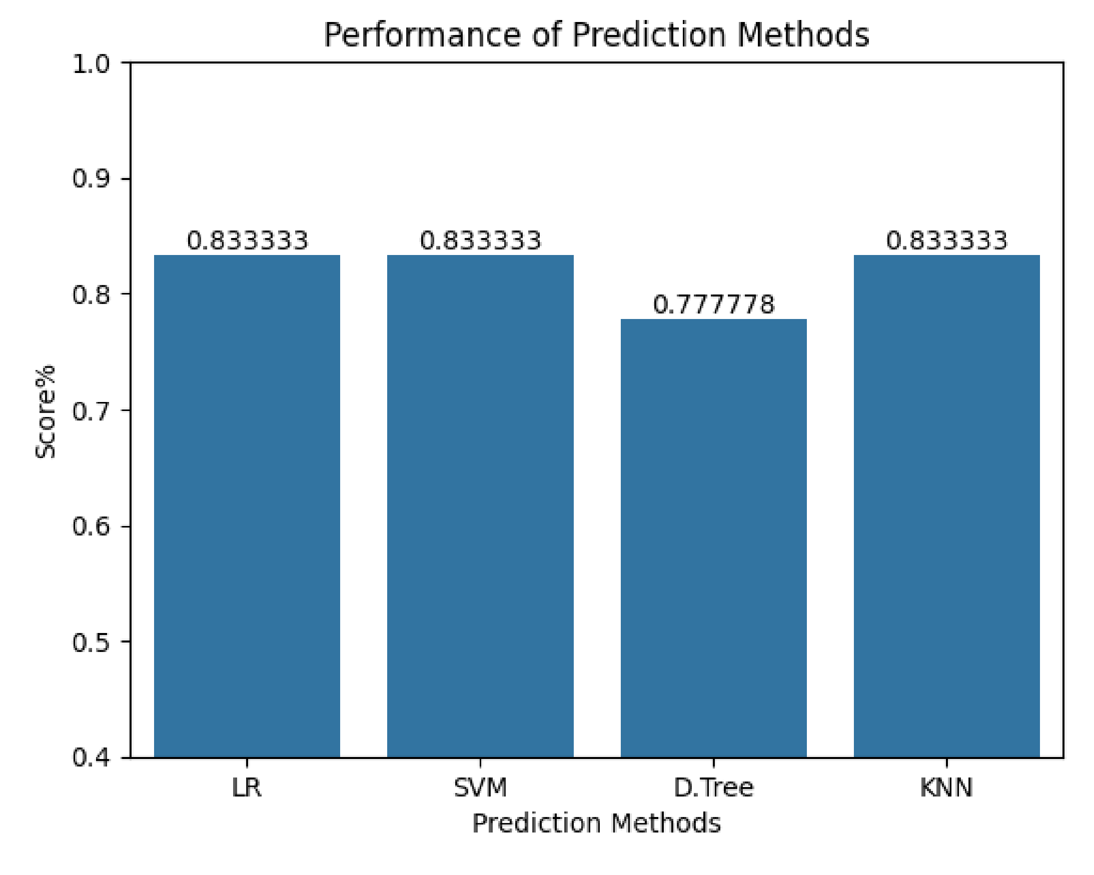
- Markers, circles, cluster were used to visualize data. Mouseposition was used to get coordinates where mouse hovers. Polyline was used to show distance between coastline and launch site.Coloring data was gathered by a written function.

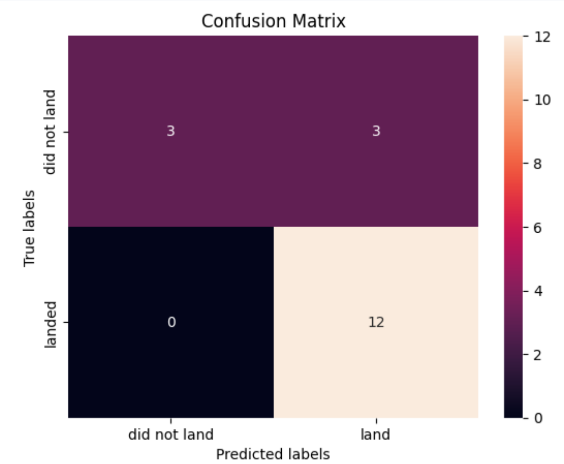


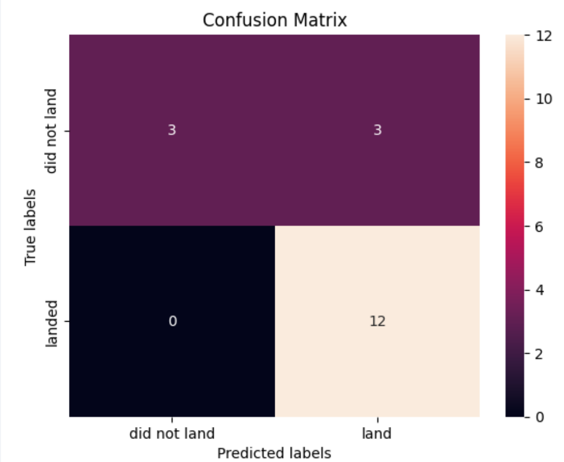
- https://github.com/cuneytharp/testrepo/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb

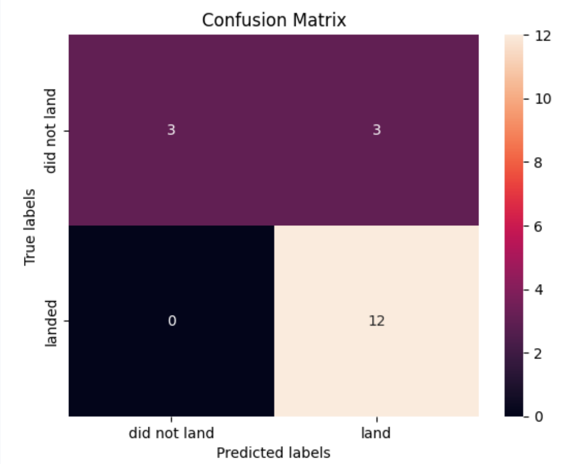
Predictive Analysis (Classification)

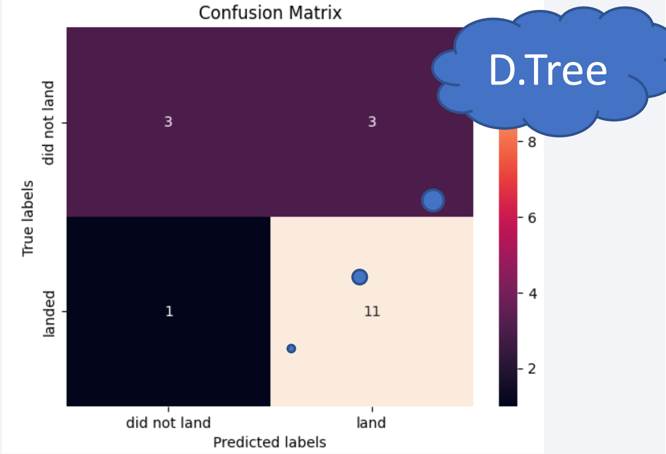
- Logistic regression, decision tree, KNN and SVM methods were performed and %83.3 accuracy could be achieved .
- GridsearchCV was used for hyperparameter tuning with unique specified parameter lists.











Results

- SVM,KNN and Logistic Regression models are best performing models for this data set(%83.3)
- As seen, success rate of SpaceX is increasing continuously which may mean we can expect better success rates in future.
- GEO,HEO,SSO, and ES L1 orbits have the best success rate.
- All launch sites are near to ocean

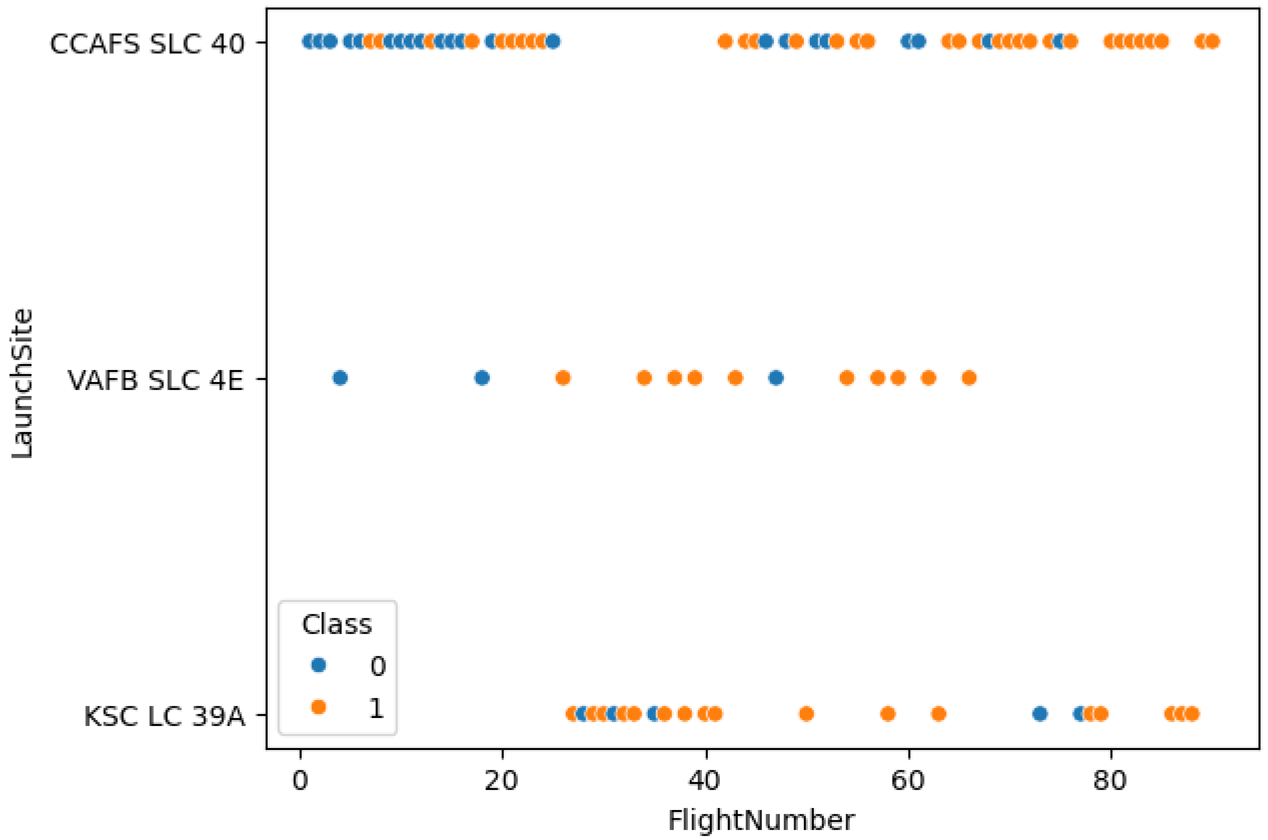
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

```
In [7]: # 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
        sns.scatterplot(data=df, x="FlightNumber", y="LaunchSite",hue="Class")
```

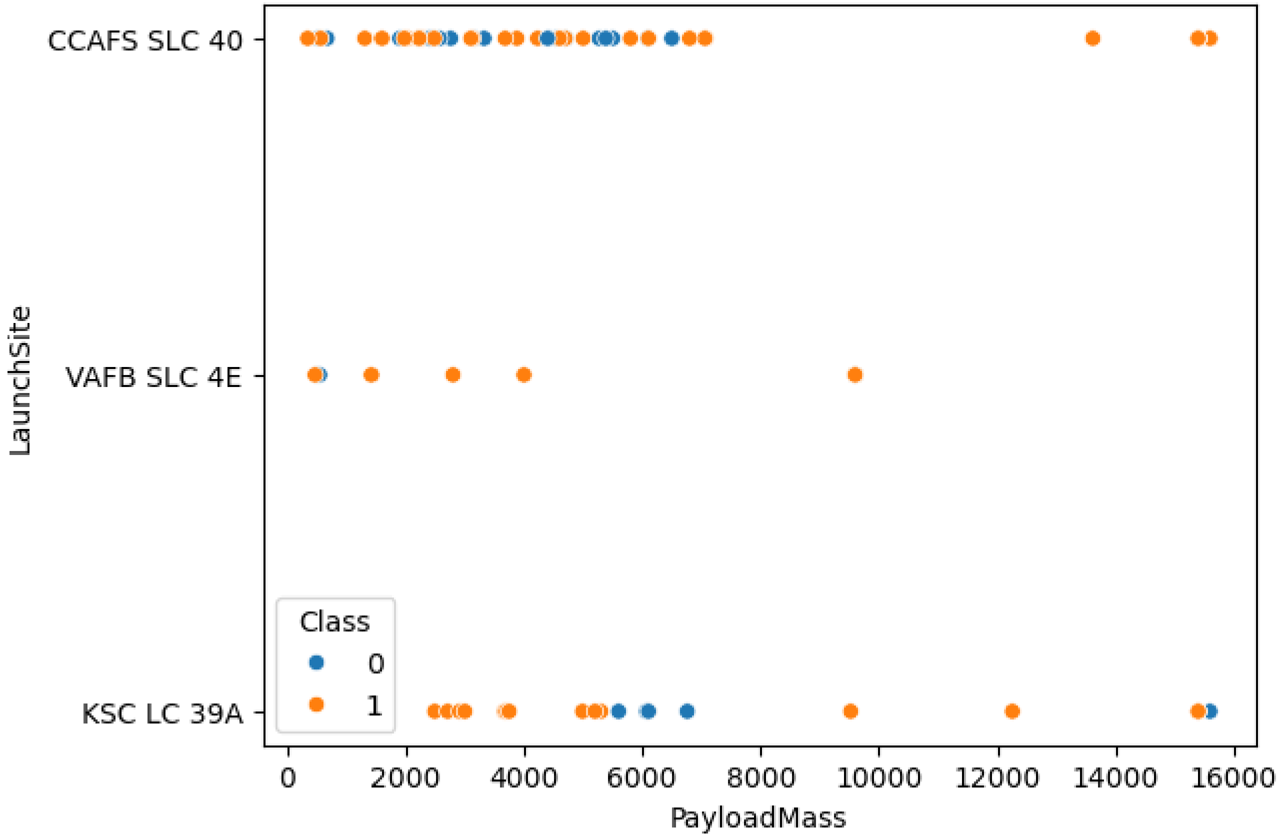
```
Out[7]: <AxesSubplot:xlabel='FlightNumber', ylabel='LaunchSite'>
```



Payload vs. Launch Site

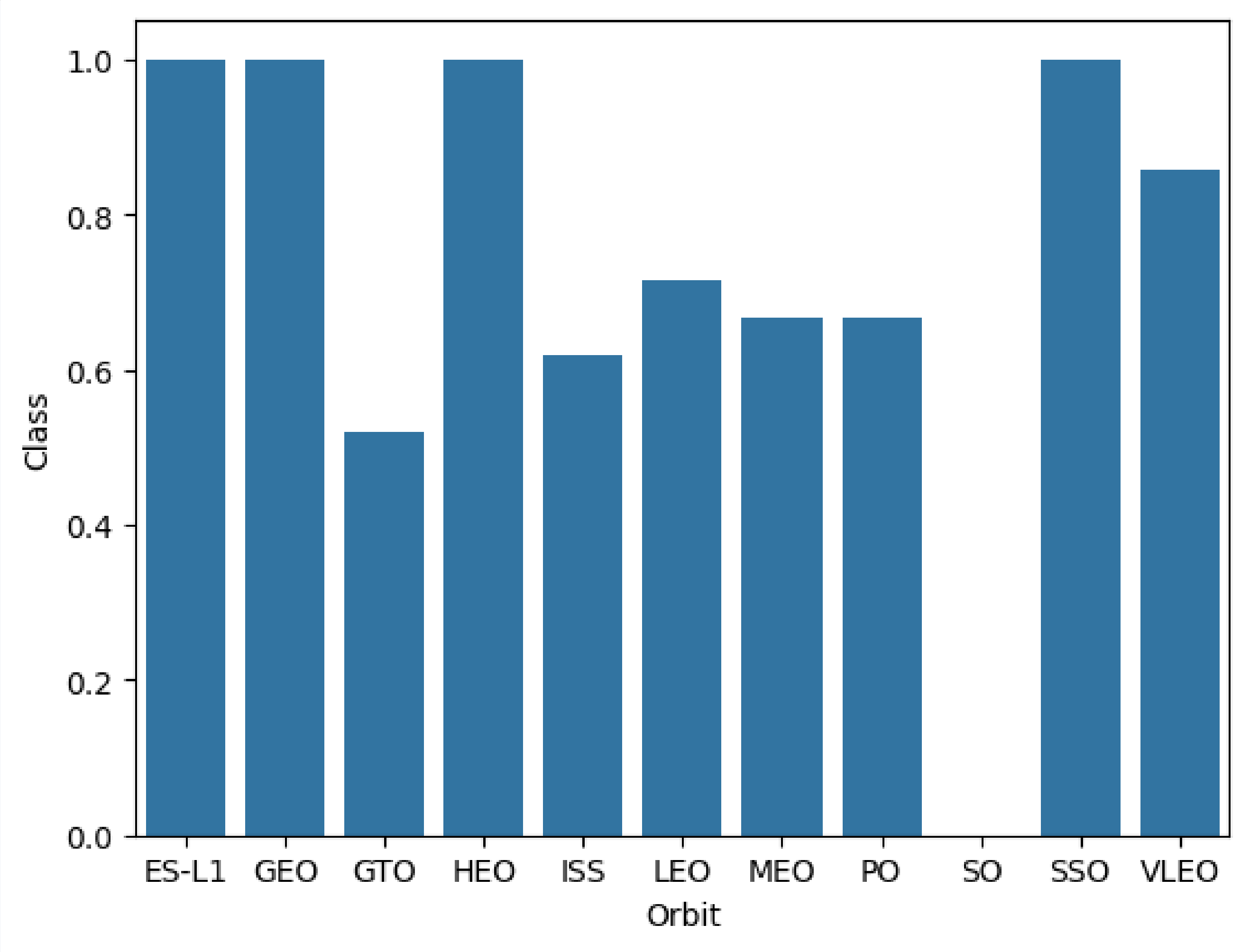
```
In [9]: # Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
        sns.scatterplot(data=df,x="PayloadMass",y="LaunchSite",hue="Class")

Out[9]: <AxesSubplot:xlabel='PayloadMass', ylabel='LaunchSite'>
```



Success Rate vs. Orbit Type(Class, 1=100%)

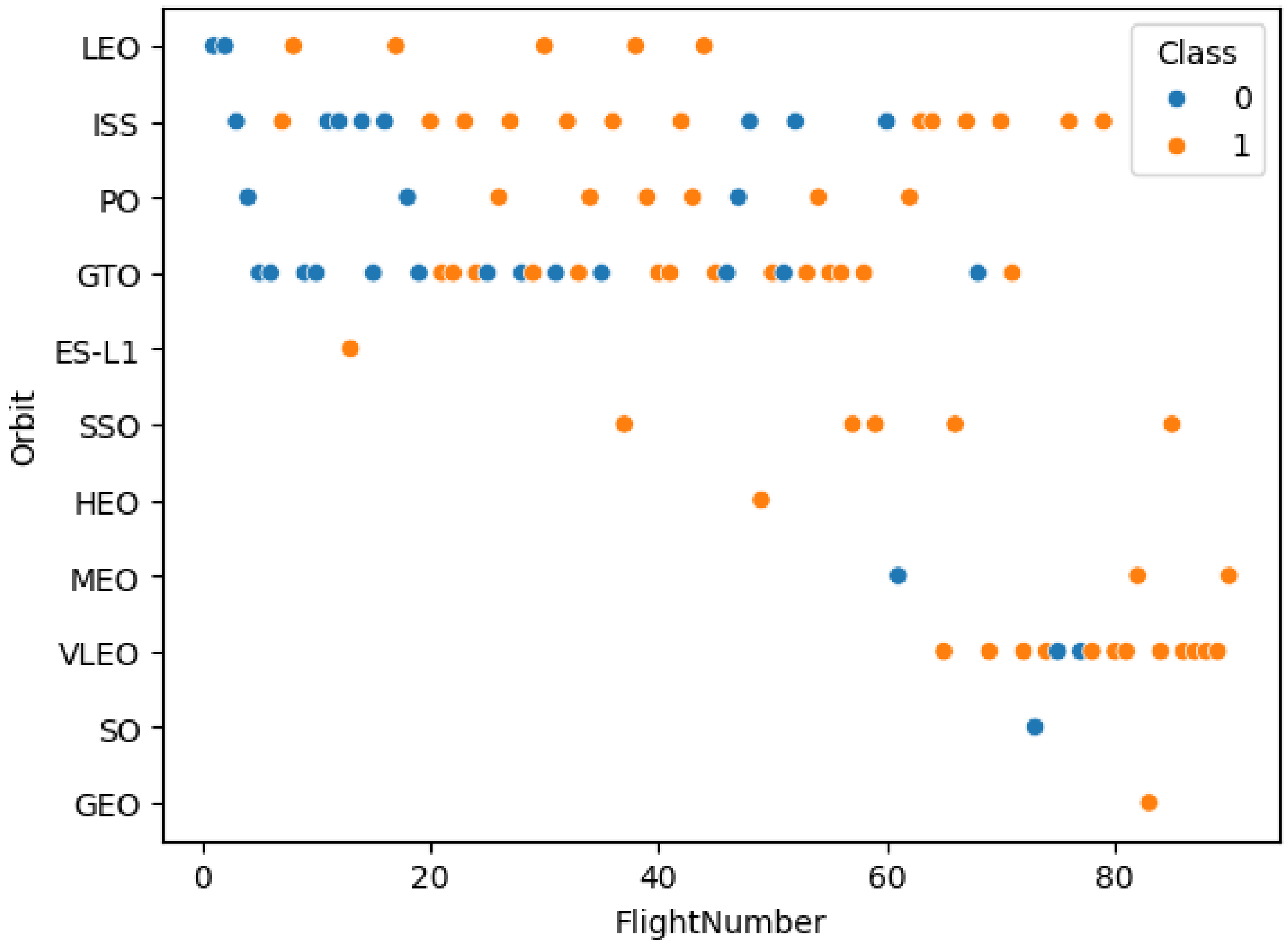
```
new_df= df.groupby("Orbit").mean()
new_df.reset_index(inplace=True)
sns.barplot(data=new_df, x= "Orbit",y="Class")
```



Flight Number vs. Orbit Type

```
sns.scatterplot(data=df,x="FlightNumber",y="Orbit",hue="Class")
```

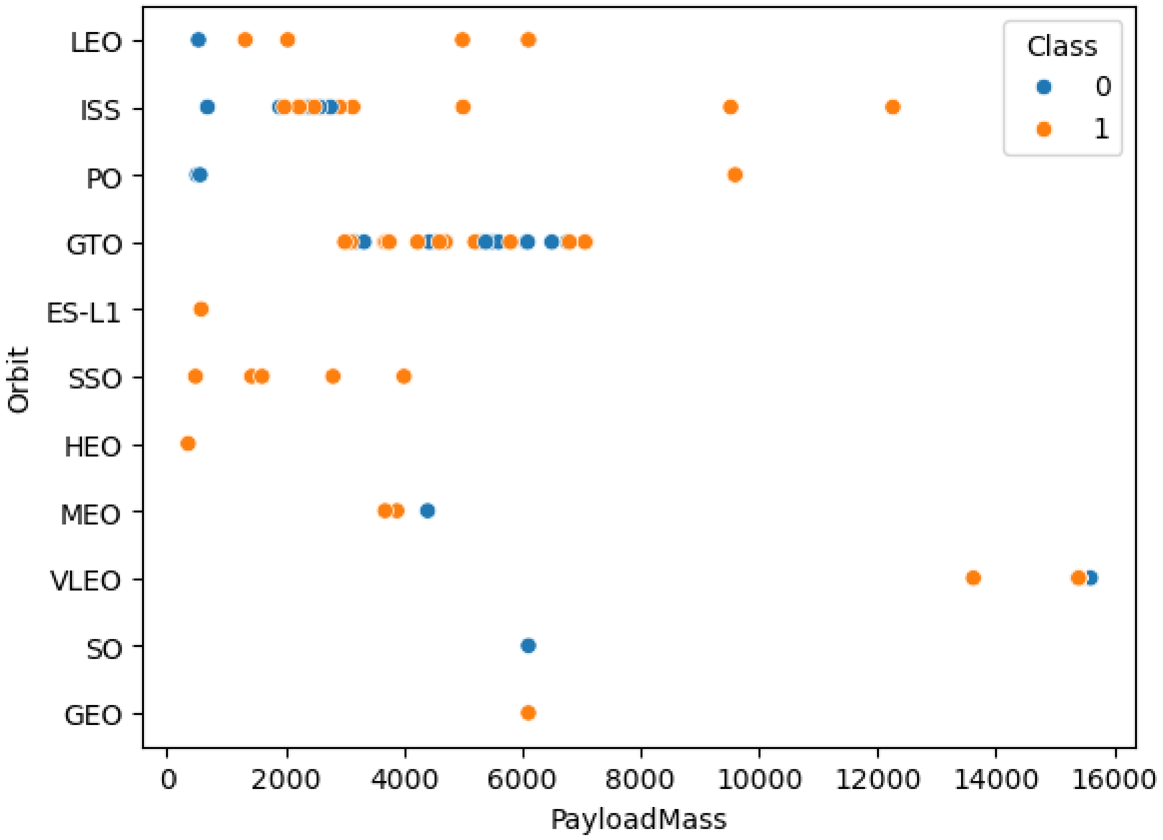
```
Out[13]: <AxesSubplot:xlabel='FlightNumber', ylabel='Orbit'>
```



Payload vs. Orbit Type

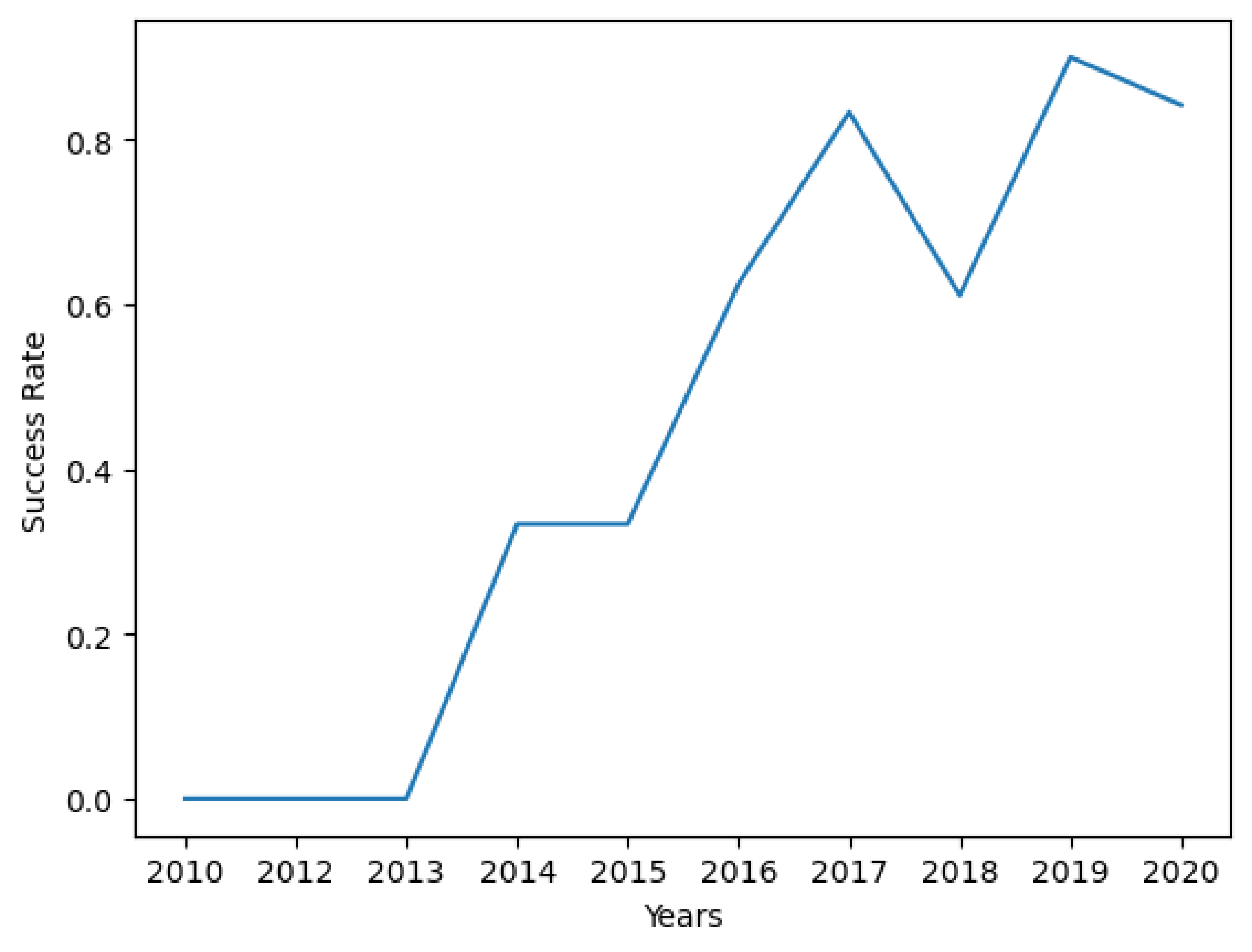
```
sns.scatterplot(data=df, x="PayloadMass",y="Orbit",hue="Class")
```

```
Out[15]: <AxesSubplot:xlabel='PayloadMass', ylabel='Orbit'>
```



Success increases with payload over 10000kg

Launch Success Yearly Trend



After 2013, there is significant improvement in success rate

All Launch Site Names

```
In [7]: %sql select DISTINCT launch_site  from SPACEXTABLE

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

Out[7]:

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

- Query with «DISTINCT» parameter gets unique values of the column like df[x].unique()

Launch Site Names Begin with 'KSC'

```
In [8]: %sql select * from SPACEXTABLE WHERE launch_site LIKE 'KSC%' LIMIT 5

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

Out[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

- LIMIT describes how many results will be returned from query.
- LIMIT 5 is similiar to df.head()

Total Payload Mass

In [9]:
```
%sql SELECT sum(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE Customer LIKE "NASA (CRS)"
```

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

Out[9]:

sum(PAYLOAD_MASS__KG_)
45596

- Sum function within query sums the returned query for PAYLOAD_MASS_KG

Average Payload Mass by F9 v1.1

```
In [10]: %sql SELECT AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE Booster_Version = "F9 v1.1"

 * sqlite:///my_data1.db
Done.

Out[10]:
```

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

```
[13]: %sql SELECT min(Date) FROM SPACEXTABLE WHERE Landing_Outcome ="Success (drone ship)"

 * sqlite:///my_data1.db
Done.

[13]:
```

min(Date)
2016-04-08

- Min function returns the minimum value of the specified field. We also filter landing outcome by «Success (drone ship)»
- With min, we return the earliest success.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
[14]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome ="Success (drone ship)" AND PAYLOAD_MASS__KG_>4000 AND PAYLOAD_MASS__KG_<6000

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

[14]:

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

- After WHERE Clause we can specify our filters and join them by AND operation for filtering.

Total Number of Successful and Failure Mission Outcomes

```
[22]: %sql SELECT COUNT(Mission_Outcome) as "Total Mission" FROM SPACEXTABLE

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

[22]:

Total Mission
101

```
[23]: %sql SELECT COUNT(Mission_Outcome) as "Successful Mission" FROM SPACEXTABLE WHERE Mission_Outcome ="Success"

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

[23]:

Successful Mission
98

```
[24]: %sql SELECT COUNT(Mission_Outcome) as "Failed Mission" FROM SPACEXTABLE WHERE Mission_Outcome !="Success"

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

[24]:

Failed Mission
3

- With «as», we can specify column name. Count gathers the count number of the specified column. We can distinguish them by filtering according to the success condition of the outcome.

Boosters Carried Maximum Payload

```
[28]: %sql select Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ =(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)

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

[28]:

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

2015 Launch Records

```
[32]: %sql select substr(Date,6,2) as month,substr(Date,0,5) as year, Booster_Version,Launch_Site from SPACEXTABLE where Landing_Outcome = "Success (ground pad)" and (substr(Date,0,5)='2017')
```

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

[32]:

month	year	Booster_Version	Launch_Site
02	2017	F9 FT B1031.1	KSC LC-39A
05	2017	F9 FT B1032.1	KSC LC-39A
06	2017	F9 FT B1035.1	KSC LC-39A
08	2017	F9 B4 B1039.1	KSC LC-39A
09	2017	F9 B4 B1040.1	KSC LC-39A
12	2017	F9 FT B1035.2	CCAFS SLC-40

- As seen , KSC LC-39A seems a very good launch site matching these conditions

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

[34]:
```
%sql select * from SPACEXTABLE where Date between "2010-06-04" and "2017-03-20" group by Landing_Outcome order by date desc
```

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

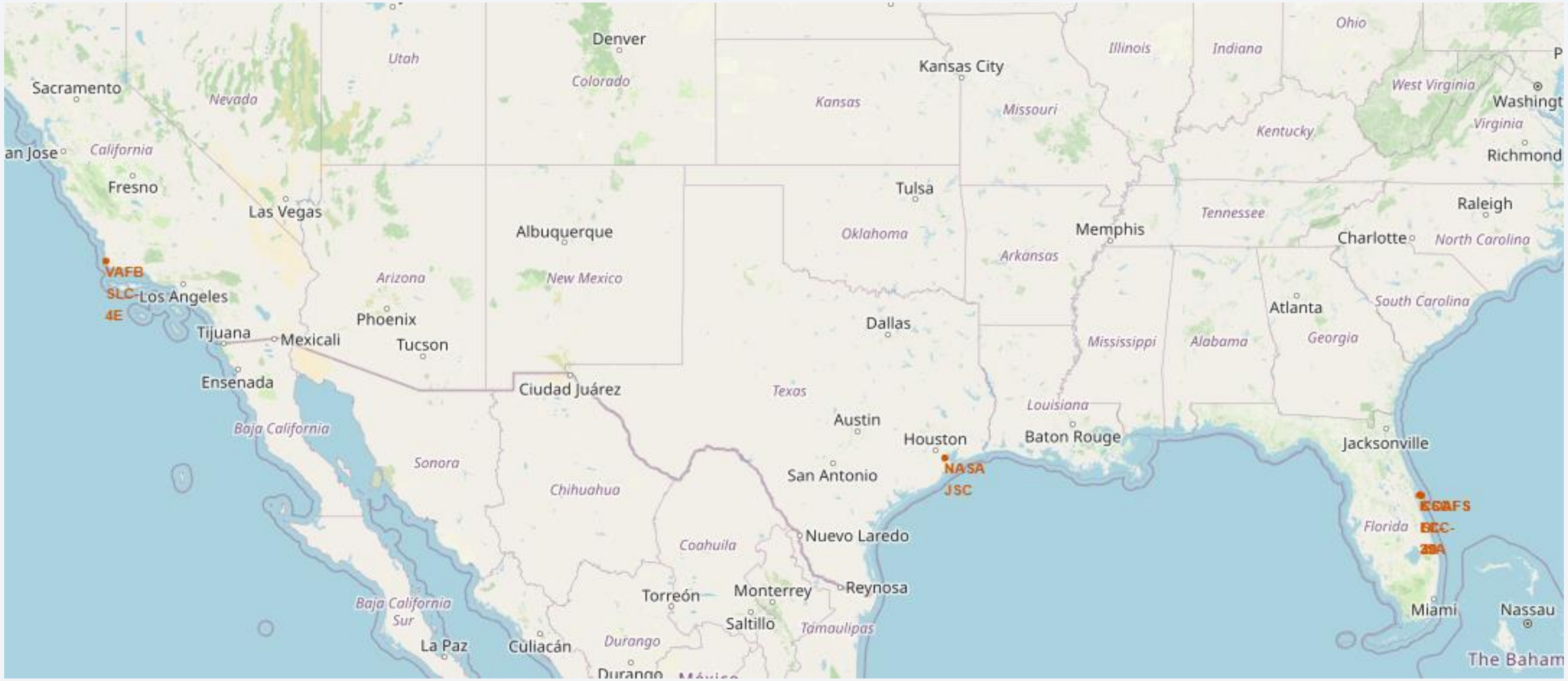
[34]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)
2015-06-28	14:21:00	F9 v1.1 B1018	CCAFS LC-40	SpaceX CRS-7	1952	LEO (ISS)	NASA (CRS)	Failure (in flight)	Precluded (drone ship)
2015-01-10	9:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2014-04-18	19:25:00	F9 v1.1	CCAFS LC-40	SpaceX CRS-3	2296	LEO (ISS)	NASA (CRS)	Success	Controlled (ocean)
2013-09-29	16:00:00	F9 v1.1 B1003	VAFB SLC-4E	CASSIOPE	500	Polar LEO	MDA	Success	Uncontrolled (ocean)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)

Section 3

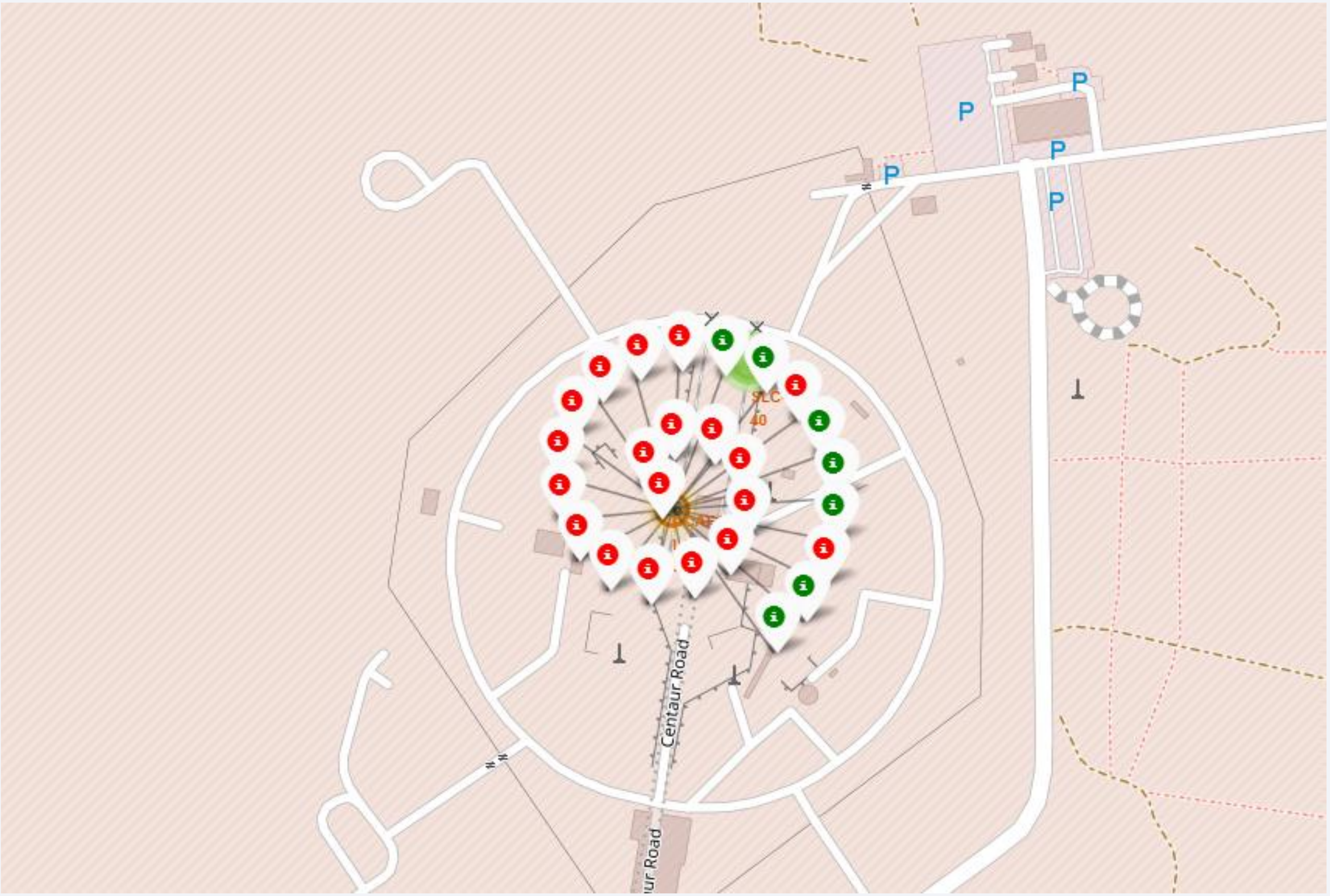
Launch Sites Proximities Analysis

Launch sites of SpaceX



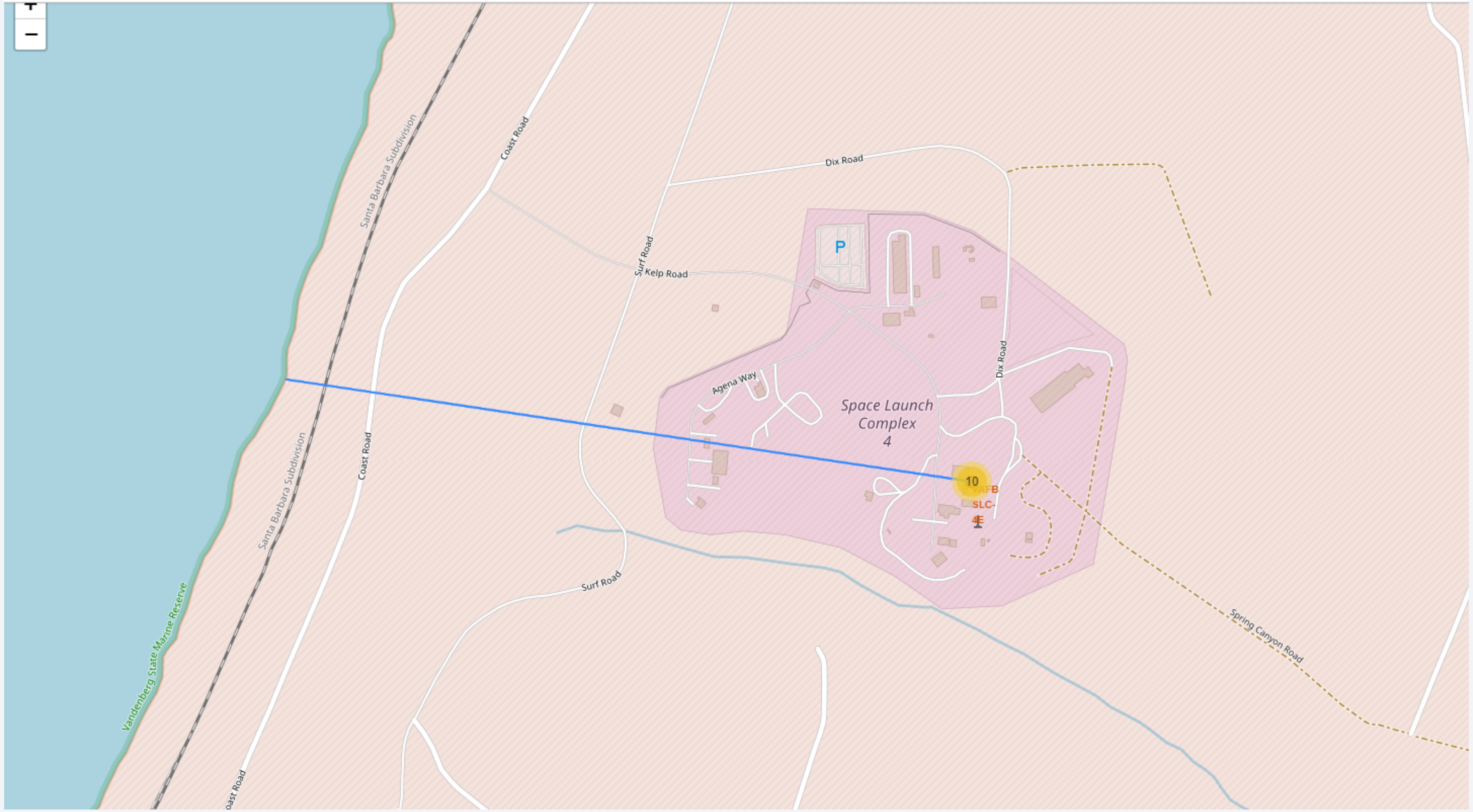
- As seen, all launch sites are near to ocean for safety reasons. If landing is unsuccessful, it will not harm any people located on residential areas.

Outcomes from SLC-40



- Red shows unsuccessful and green shows successful results

Distance from coastal line

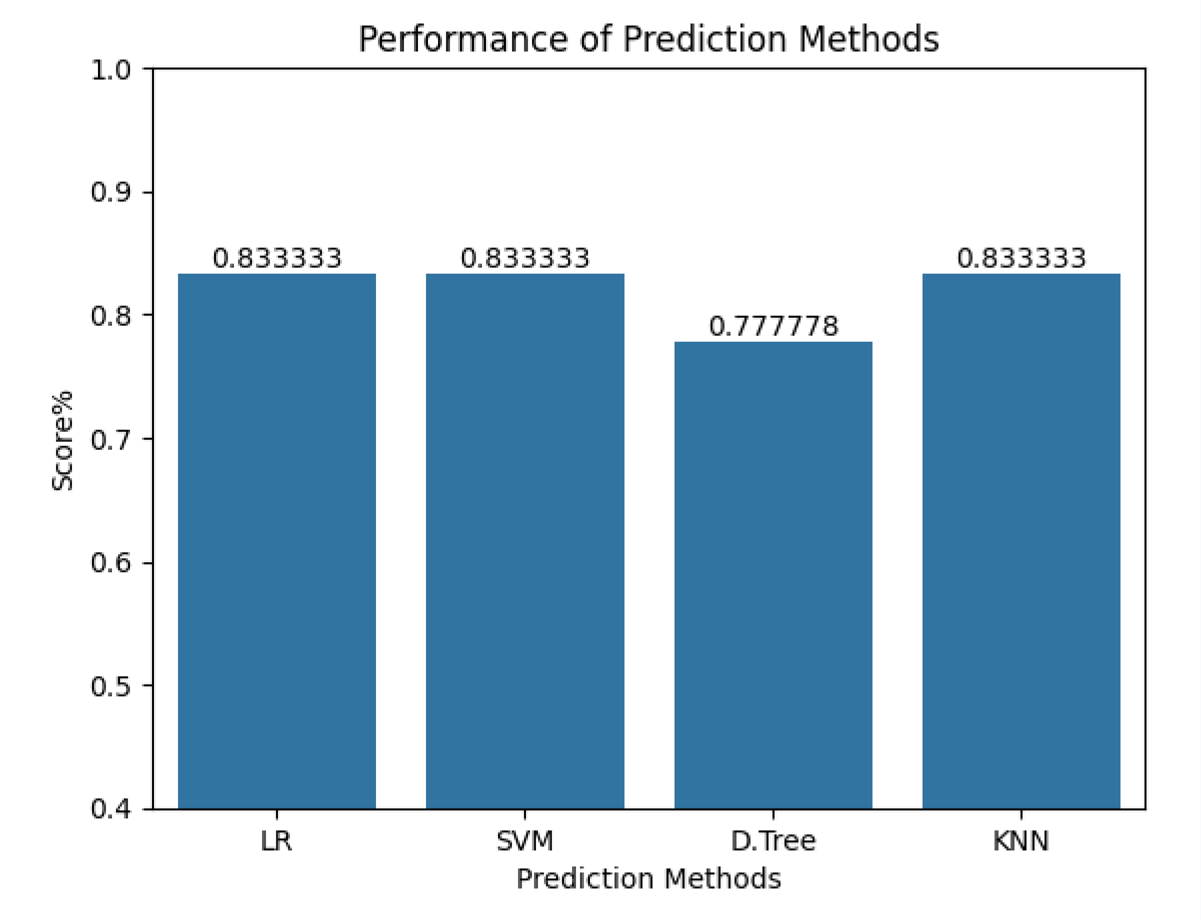


- Launch site is 1.34km away from coastal side.

Section 5

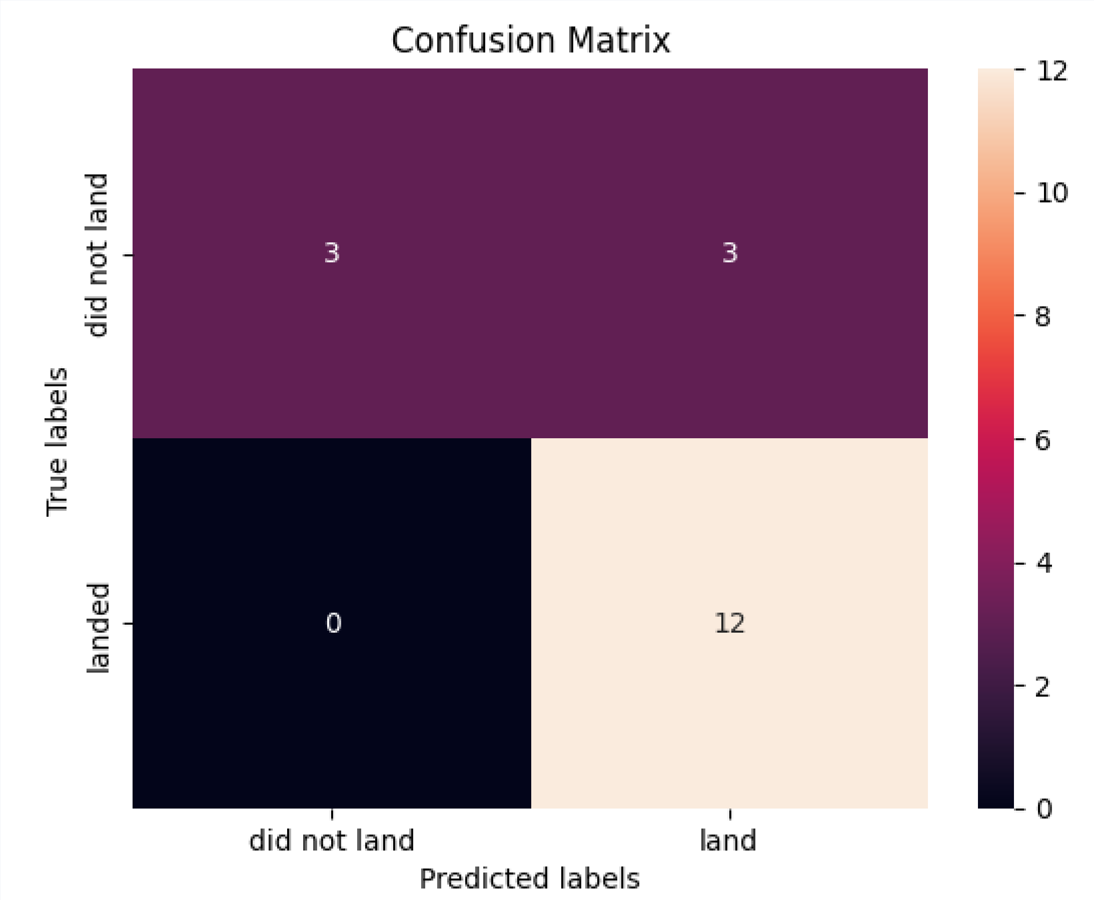
Predictive Analysis (Classification)

Classification Accuracy



- As seen, logistic regression,support vector machines and KNN both perform the same.

Confusion Matrix



- Prediction is very accurate on landed outcomes.

Conclusions

- SVM,KNN and Logistic Regression models are best performing models for this data set
- As seen, success rate of SpaceX is increasing continuously which may mean we can expect better success rates in future.
- GEO,HEO,SSO, and ES L1 orbits have the best success rate.
- All launch sites are near to ocean.

Thank you!