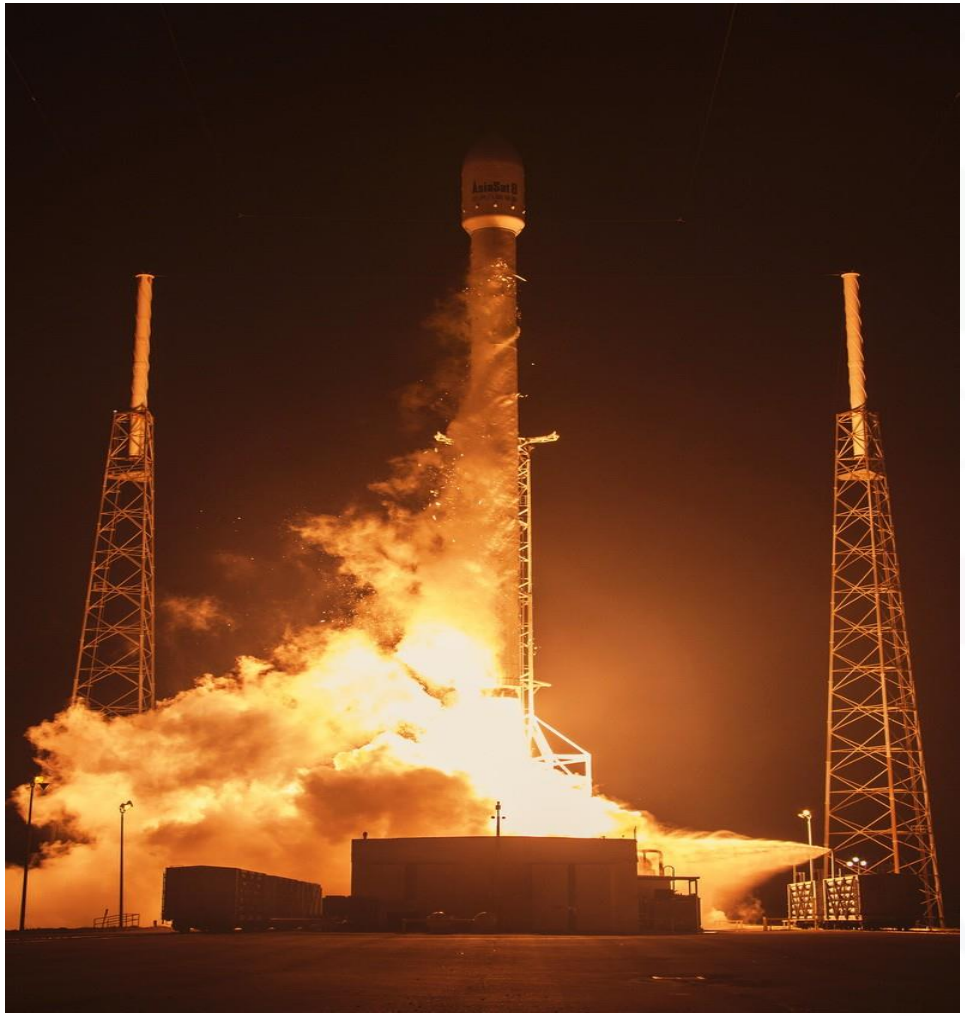


SPACEX FALCO-9 CAPSTONE PROJECT

Yasser Shabbir

15th January 2024

OUTLINE

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

EXECUTIVE SUMMARY

- Data Collection
- Data Wrangling – done with
  - Pandas
  - Numpy
- Exploratory Data Analysis.

  . SQL
- Visualization.
- Predictive Analysis with Machine Learning.

INTRODUCTION

- **Space-X Goal**
  - Sending spacecraft to the international space station.
  - Providing satellite internet to the whole world with Starlink technology.
  - Taking people and cargo into space and contributing to space exploration.

FALCON 9 ROCKETS

- Falcon 9 is a two-stage reusable rocket developed and manufactured by SpaceX, a private aerospace company founded by Elon Musk. Here are some key features and missions of Falcon 9 rockets

  - Reusability: One of the notable features of Falcon 9 is its reusability. The first stage of the rocket is designed to return to Earth after launch, landing vertically either on land (at SpaceX's landing zones) or on an autonomous drone ship in the ocean. This reusability significantly reduces the cost of space launches.

  - Payload Capacity: Falcon 9 is capable of delivering a variety of payloads to orbit, including satellites, cargo resupply missions to the International Space Station (ISS), and even crewed missions. It has a payload capacity of up to 22,800 kilograms (50,300 pounds) to low Earth orbit (LEO) and up to 8,300 kilograms (18,300 pounds) to geostationary transfer orbit (GTO).

- Starlink: Falcon 9 plays a crucial role in SpaceX's ambitious Starlink project, which aims to provide global broadband internet coverage from a network of thousands of small satellites in low Earth orbit. Falcon 9 launches numerous batches of Starlink satellites to gradually build up the constellation.

- Satellite Deployment: Falcon 9 is frequently used to deploy satellites for commercial customers, government agencies, and scientific research. It offers the flexibility to deliver satellites into different orbits, such as LEO, GTO, and sun-synchronous orbit (SSO).

METHODOLOGY

- Data Collection through Web-Scrapping from Spacx API
- Performed Exploratory Data Analysis (EDA) and Visualization with Structure Queried Language (SQL), Folium, Plotly.
- Performed Predictive Analysis with Machine Learning using
  - Logistic Regression
  - K- Nearest
  - Decision Tree
  - Support Vector Machine

RESULTS

- Data Collection

Api:
spacex_url="https://api.spacexdata.com/v4/launches/past"

Web page:

"https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"

RESULTS

Python

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitud
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.56185
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.56185
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.56185
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.63209
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.56185
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	12	B1060	-80.603956	28.60805
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	13	B1058	-80.603956	28.60805
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	12	B1051	-80.603956	28.60805
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	12	B1060	-80.577366	28.56185
93	90	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	8	B1062	-80.577366	28.56185

90 rows × 17 columns

Data Collection - SpaceX API

- Request and parse the SpaceX launch data using the GET request Filter the dataframe to only include Falcon 9 launches
- **Web Scrapping - Wikipedia**

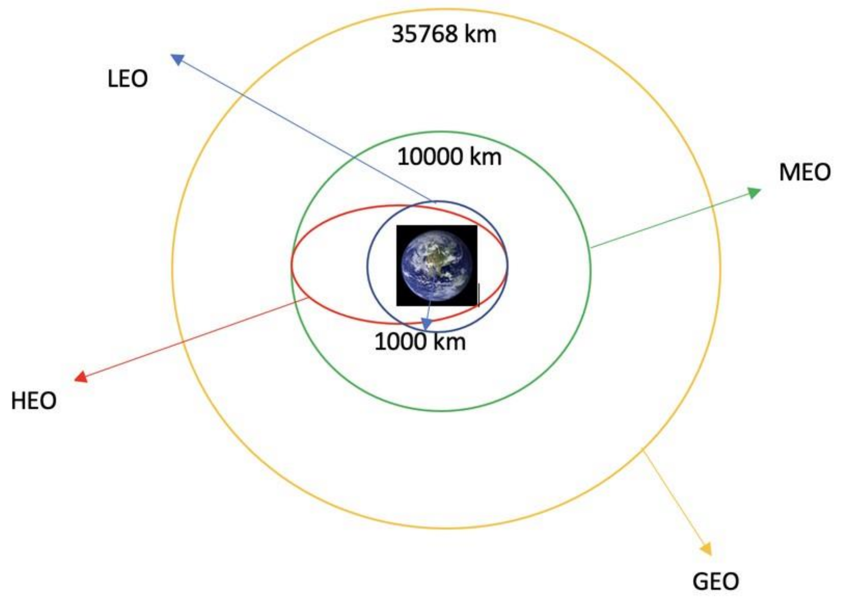
  Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia

  - Request the Falcon9 Launch Wiki page from its URL
  - Extract all column/variable names from the HTML table header Create a data frame by parsing the launch HTML tables

RESULTS

Data Wrangling

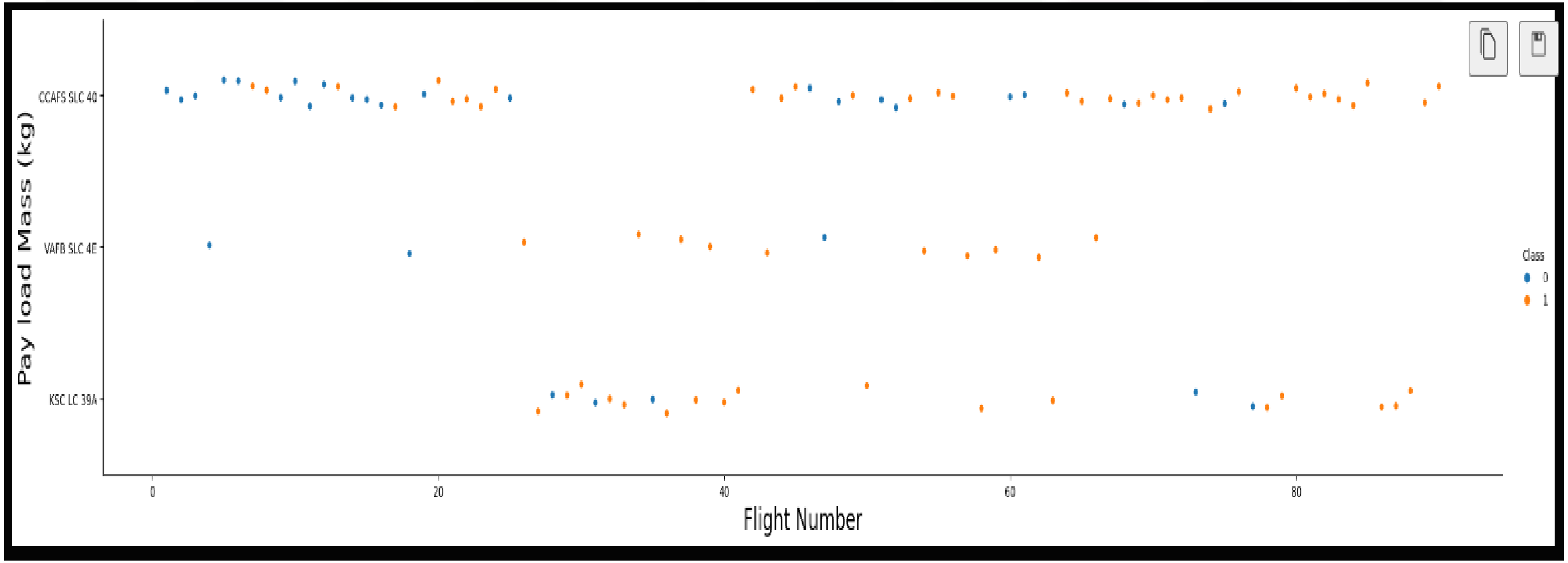
- Calculate the number of launches on each site Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcome per orbit type Create a landing outcome label from Outcome column



RESULTS

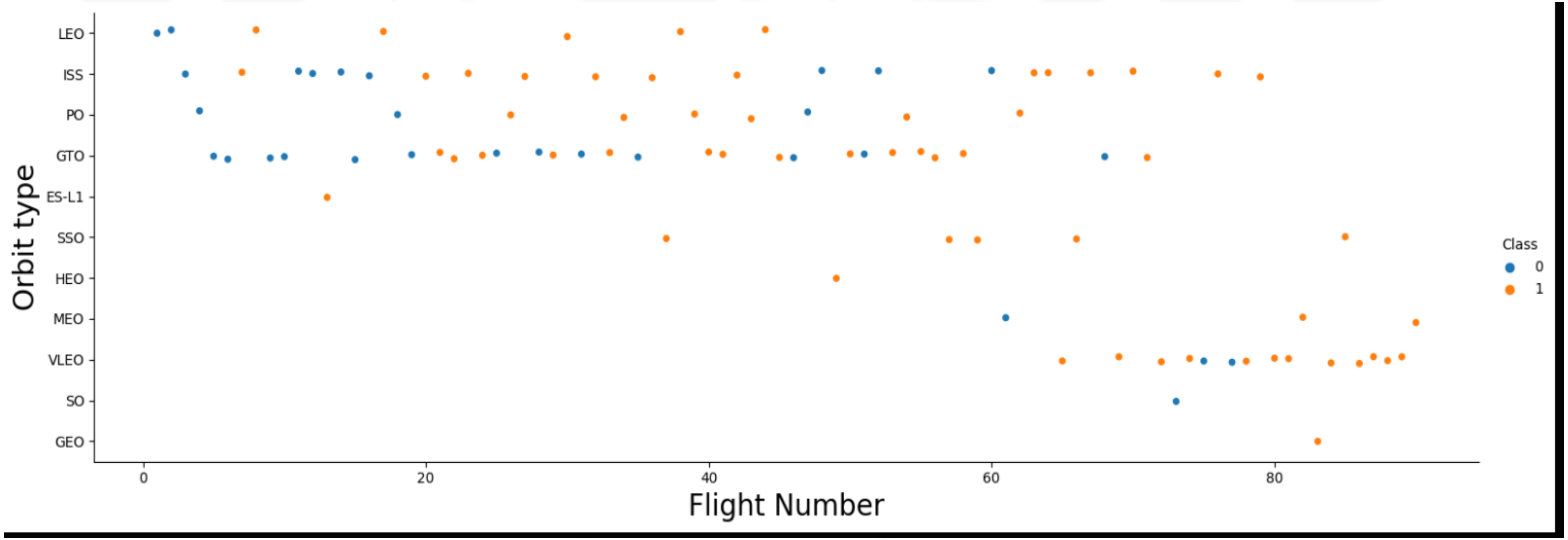
EDA done with Data Visualization

- Scatter Plot showing the Flight Number and Payload Massa



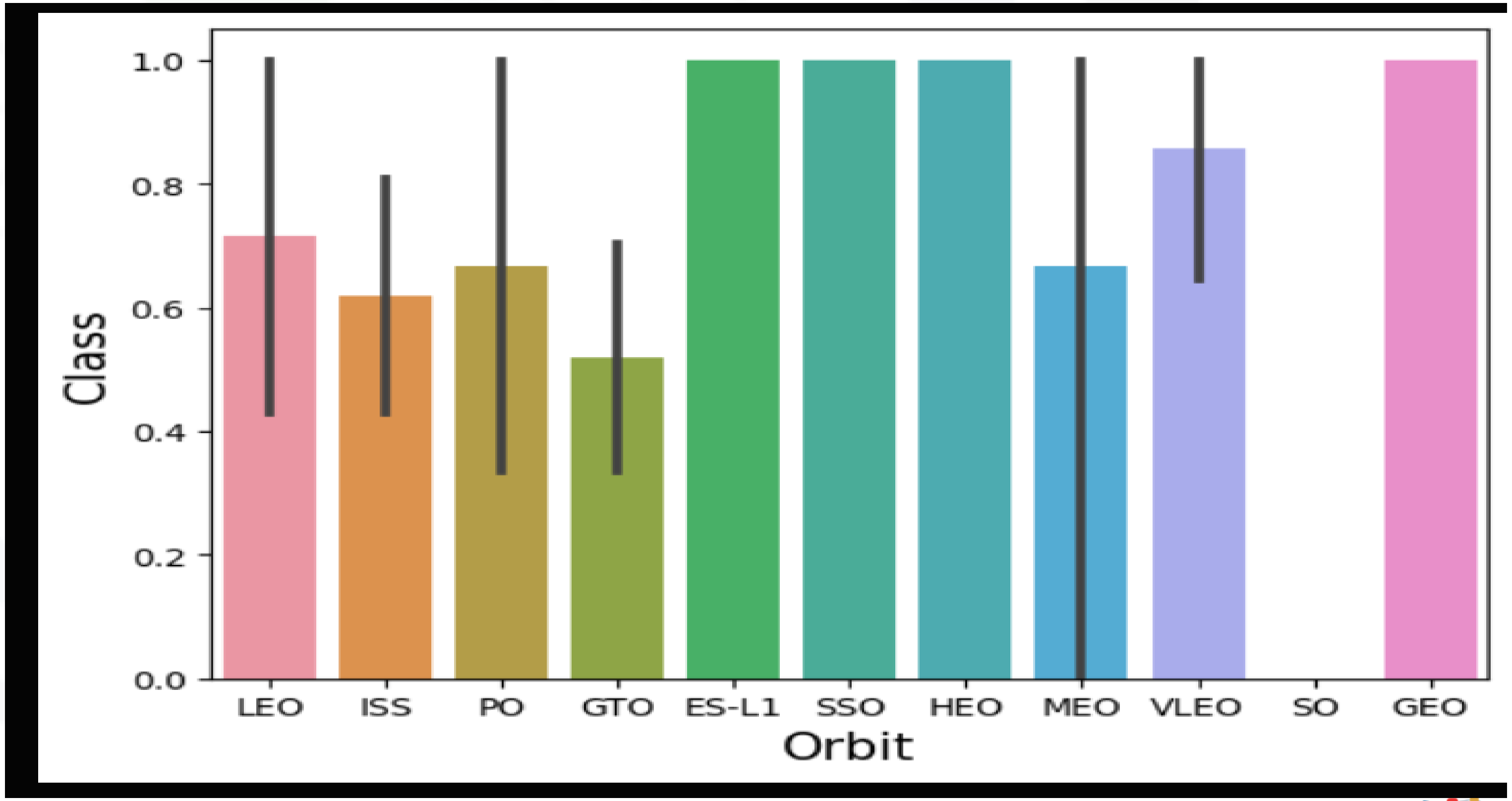
RESULTS

. Scatter Plot showing the relationship between Flight Number and Orbit Type



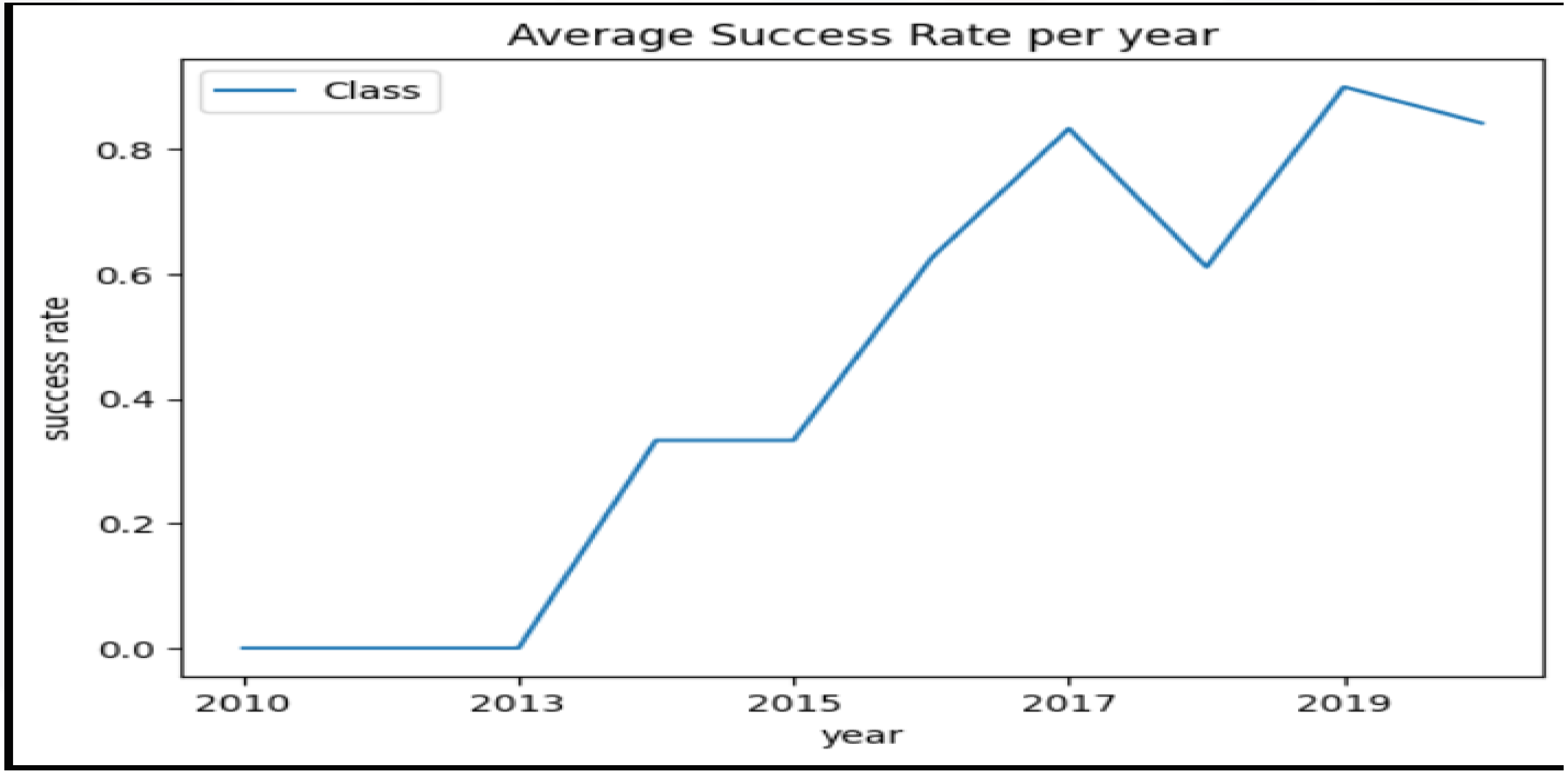
RESULTS

- Bar Chart showing the Classes of Each Orbit.



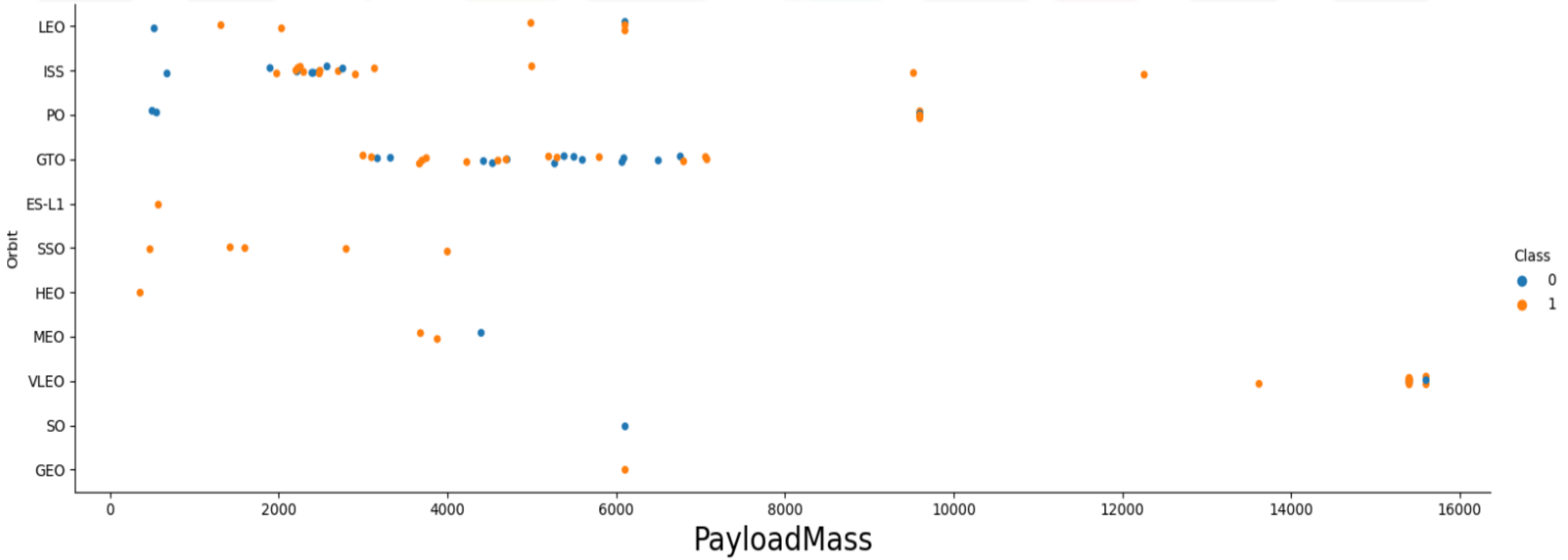
RESULTS

- A Line Plot showing the Average Success rate per year



RESULTS

- Scatter Plot showing the different Payload Masses of each Orbit



RESULTS

EDA with SQL

- 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 achieved.
    - List the names of the boosters which have success in drone ship and
    - have payload mass greater than 4000 but less than 6000
    - List the total number of successful and failure mission outcomes

- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- 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
- Rank the count of successful landing_outcomes between the date 04- 06-2010 and 20-03-2017 in descending order.

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

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

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

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

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

Customer	Total_NASA_CRS_mass
NASA (CRS)	45596

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

[24]

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

avg_Booster_versionF9_v1_1
2928.4

List the date when the first successful landing outcome in ground pad was achieved.

* sqlite:///my_data1.db
Done.

Mission_Outcome	FIRST SUCCESSFUL LANDING DATE
Success	2015-12-22
Success (payload status unclear)	2018-01-08

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

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

List the total number of successful and failure mission outcomes

Mission_Outcome	Total (Success or failure)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

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

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

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.

* sqlite:///my_data1.db
Done.

Date	Booster_Version	Launch_Site	Landing_Outcome
2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

[33]

Python

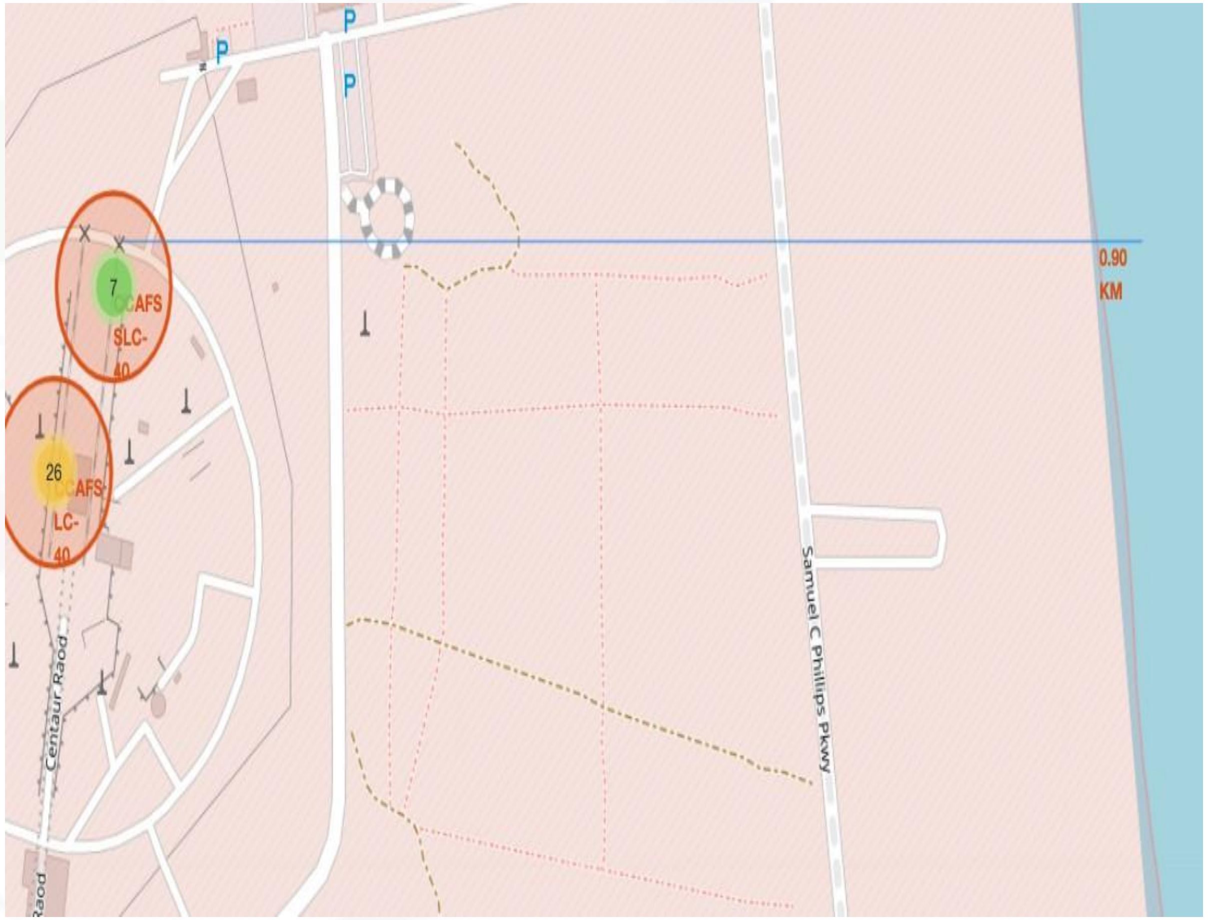
* sqlite:///my_data1.db
Done.

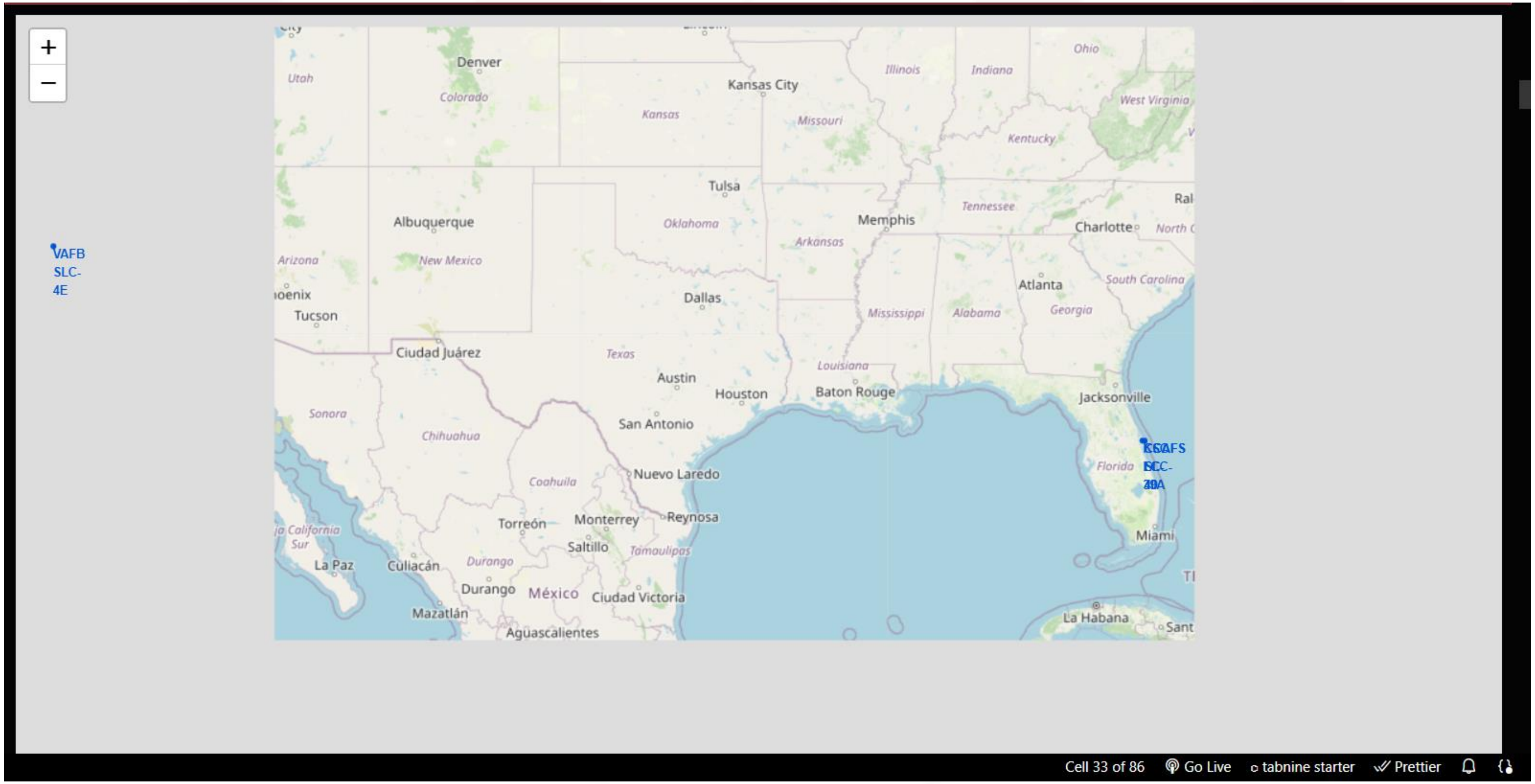
Landing_Outcome	Total Count
Success (ground pad)	3
Failure (drone ship)	5

Cell 1 of 37 Go Live tabnine starter

Interactive Visual Analytics with Folium

- Mark all launch sites on a map
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximities




+
−
VAFB
SLC-
4E
Denver
Utah
Colorado
Kansas City
Kansas
Missouri
Illinois
Indiana
Ohio
West Virginia
Kentucky
Tulsa
Oklahoma
Memphis
Tennessee
Arkansas
Charlotte
Albuquerque
New Mexico
Arizona
Tucson
Dallas
Atlanta
South Carolina
Georgia
Mississippi
Alabama
Ciudad Juárez
Texas
Louisiana
Austin
Houston
Baton Rouge
Jacksonville
San Antonio
Sonora
Chihuahua
Florida
Nuevo Laredo
Coahuila
Torreón
Monterrey
Reynosa
Saltillo
Tamaulipas
Miami
La Paz
Culiacán
Durango
México
Ciudad Victoria
Mazatlán
Aguascalientes
La Habana
Cell 33 of 86
Go Live
tabnine starter
Prettier


Montana
Minnesota
Québec
Portland
Idaho
Saint Paul
Wisconsin
Michigan
Ottawa
Montréal
Maine
Oregon
Boise
South Dakota
Toronto
Milwaukee
Wyoming
Detroit
Hamilton
New York
Chicago
Boston
Salt Lake City
Nebraska
Iowa
Lincoln
Pennsylvania
Denver
Utah
Colorado
Illinois
Indiana
Ohio
New York
Philadelphia
Kansas City
Sacramento
Nevada
Kansas
Missouri
West Virginia
Washington
Virginia
Kentucky
San Jose
California
Richmond
Fresno
Tulsa
Las Vegas
Albuquerque
Oklahoma
Memphis
Tennessee
Raleigh
Charlotte
North Carolina
Arkansas
Arizona
New Mexico
Los Angeles
Phoenix
Dallas
Atlanta
South Carolina
Tijuana
Mexicali
Tucson
Mississippi
Alabama
Georgia
Ciudad Juárez
Texas
Louisiana
Austin
Houston
Baton Rouge
Jacksonville
Baja California
Sonora
San Antonio
Chihuahua
Florida
Nuevo Laredo
Coahuila
Torreón
Monterrey
Reynosa
Miami
Nassau
Saltillo
Tamaulipas
Baja California Sur
La Paz
Culiacán
Durango
The Bahamas
Durango
México
Ciudad Victoria
Mazatlán
La Habana
Santa Clara
Aguascalientes
Cuba
Turks and Caicos Islands
Mérida
Camaguey
Tepic
Jalisco
León
Cancún
Ciudad de México
Campeche
Cayman Islands
Santiago de Cuba
República Dominicana
Morelia
Xalapa
Ayiti
Puebla
Villahermosa
Jamaica
Acapulco
Oaxaca
Belize
Tuxtla Gutiérrez
Cell 33 of 86
Go Live
tabnine starter
Prettier

MACHINE LEARNING PREDICTIVE ANALYSIS

- Create a NumPy array from the column Class in data, by applying the method to_numpy() then assign it to the variable Y,make sure the output is a Pandas series (only one bracket df['name of column’]).
- Standardize the data in X then reassign it to the variable X using the transform provided below.
- Use the function train_test_split to split the data X and Y into training and test data. Set the parameter test_size to 0.2 and random_state to
- 2. The training data and test data should be assigned to the following labels.
- Create a logistic regression object then create a GridSearchCV
- object logreg_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters.

- We output the GridSearchCV object for logistic regression. We display the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best_score_.

We output the GridSearchCV object for logistic regression. We display the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best_score_.

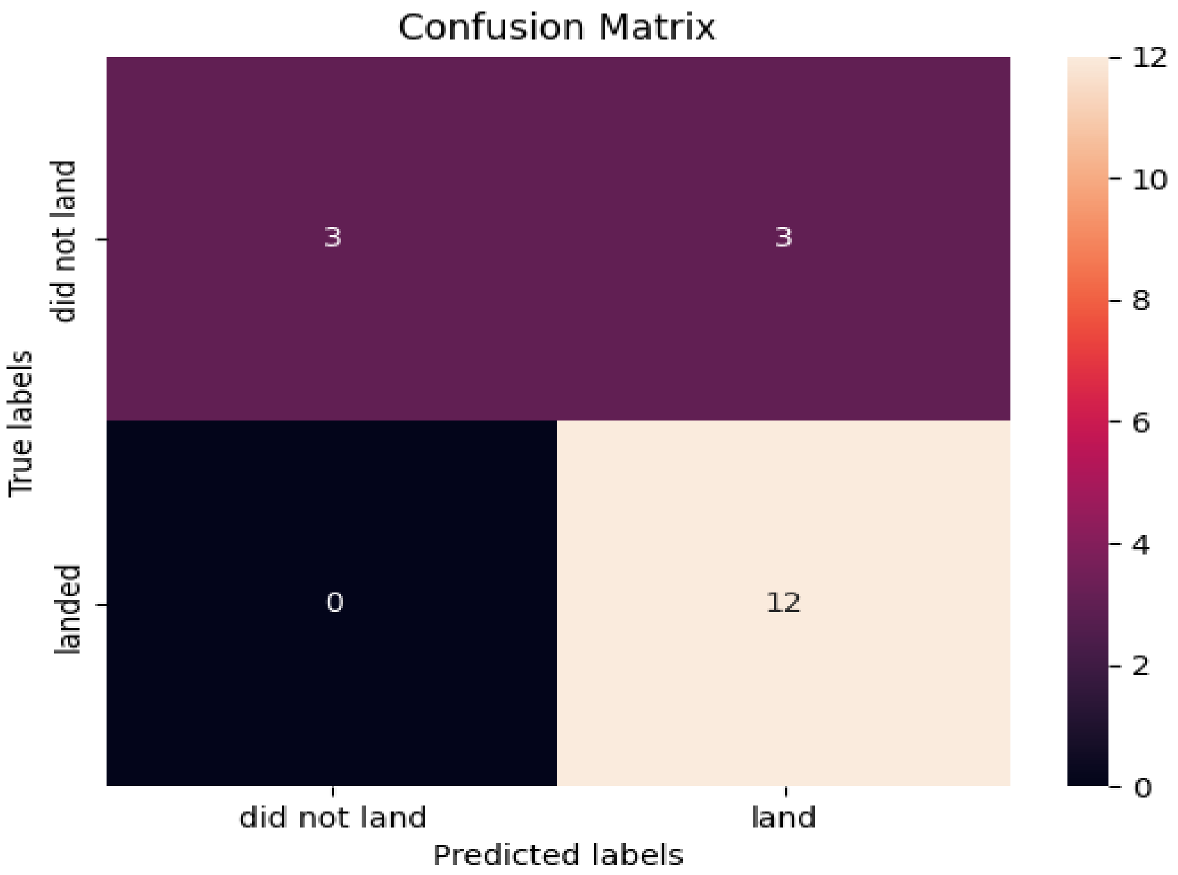
```python
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

✓ 0.0s Python

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

- Calculate the accuracy on the test data using the method score: **0.8333333333333334**

CONFUSION-MATRIX



- Create a support vector machine object then create a GridSearchCV object svm_cv with cv - 10. Fit the object to find the best parameters from the dictionary parameters.
- **Accuracy: 0.8482142857142856**
- Calculate the accuracy on the test data using the method score:
- **Accuracy: 0.8333333333333334**

- Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters
- **Accuracy: 0.8714285714285713**
- Calculate the accuracy of tree_cv on the test data using the method score:
- **Accuracy: 0.8333333333333334**

- Create a k nearest neighbors object then create a GridSearchCV object knn_cv with cv = 10. Fit the object to find the best parameters from the dictionary parameters

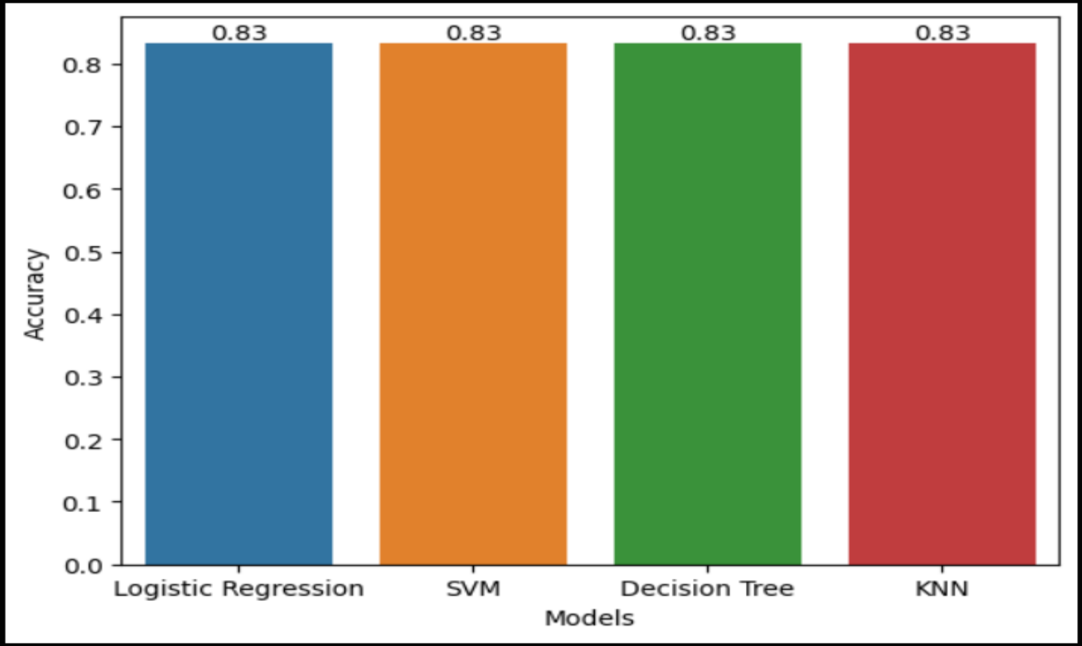
- **Accuracy: 0.8482142857142858**

- Calculate the accuracy on the test data using the method score:
- **Accuracy: 0.8333333333333334**

Bar-Plot Showing the Accuracy of The Algorithms

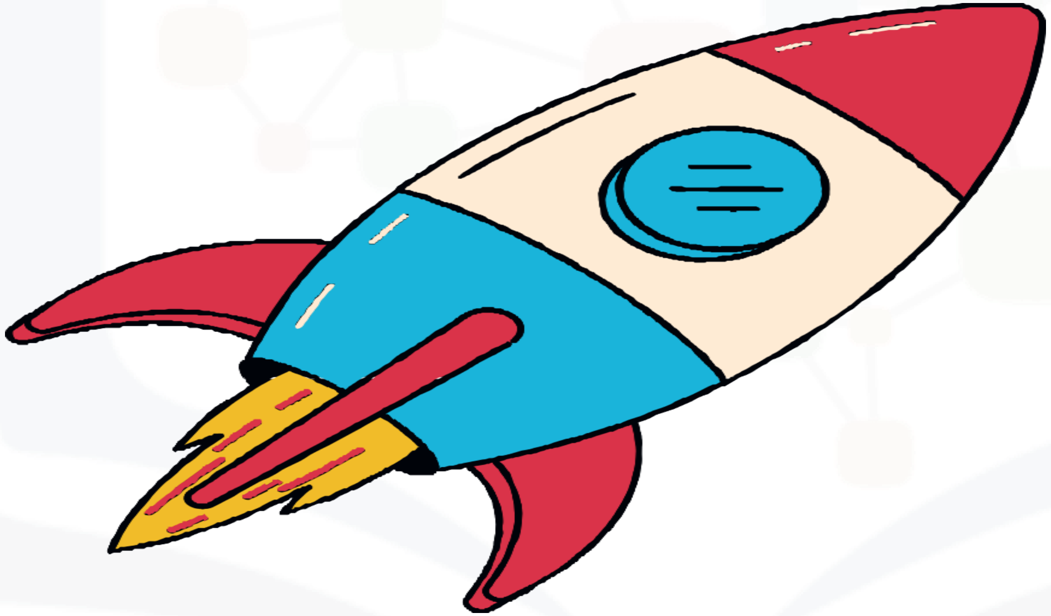
```
# plot the barchart
ax = sns.barplot(x = 'Models', y = 'Accuracy', data = df)
ax.bar_label(ax.containers[0], fmt='%.2f')
plt.show()
```

✓ 0.3s

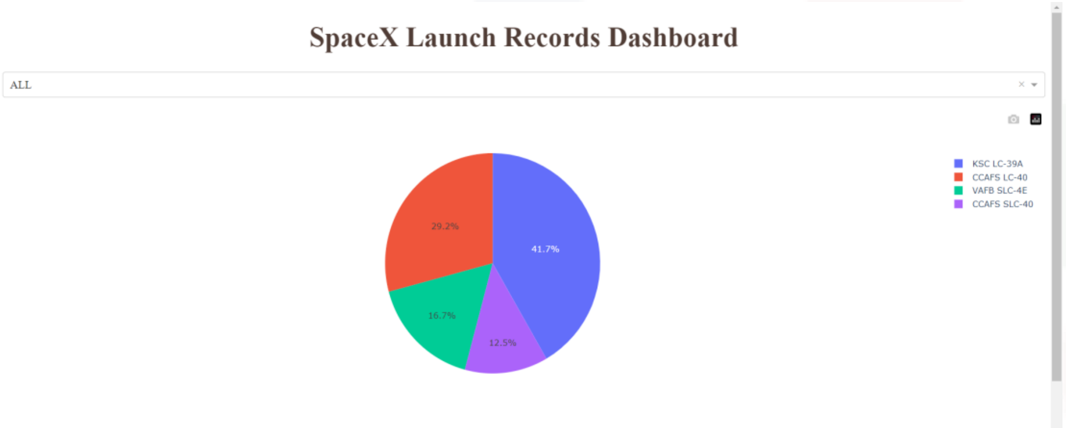


Conclusion Form the Machine-Learning Algorithm

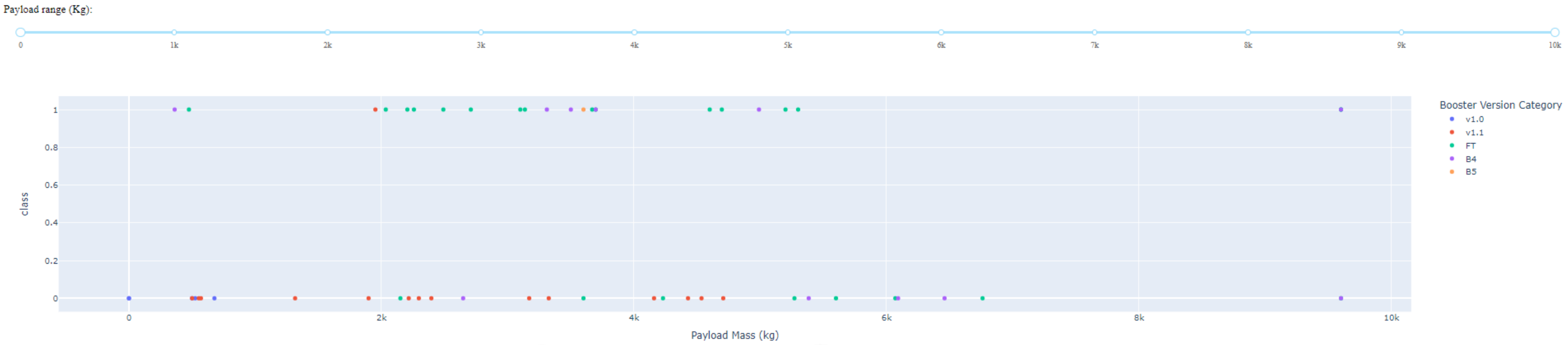
- **"Practically all these algorithms give the same result"**



DASHBOARD



SpaceX Launch Records Dashboard
ALL
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40
41.7%
29.2%
16.7%
12.5%



Payload range (Kg):
0
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k
class
Payload Mass (kg)
Booster Version Category
v1.0
v1.1
FT
B4
B5

CONCLUSION

- There is a correlation between launch site and success rate Payload mass is also associated with the success rate.: the more massive the payload, the less likely the first stage will return

- For orbit type, SO has the least success rate while ES-L1, GEO, HEO and SSO have the highest success rate According to the yearly trend

- There has been an increase in the success rate since 201 3 kept increasing till 2020 • With best parameter provided, decision tree classifier used in prediction yielded the highest accuracy of 84% .

THE END

Done By
Yasser Shabbir

SPACEX