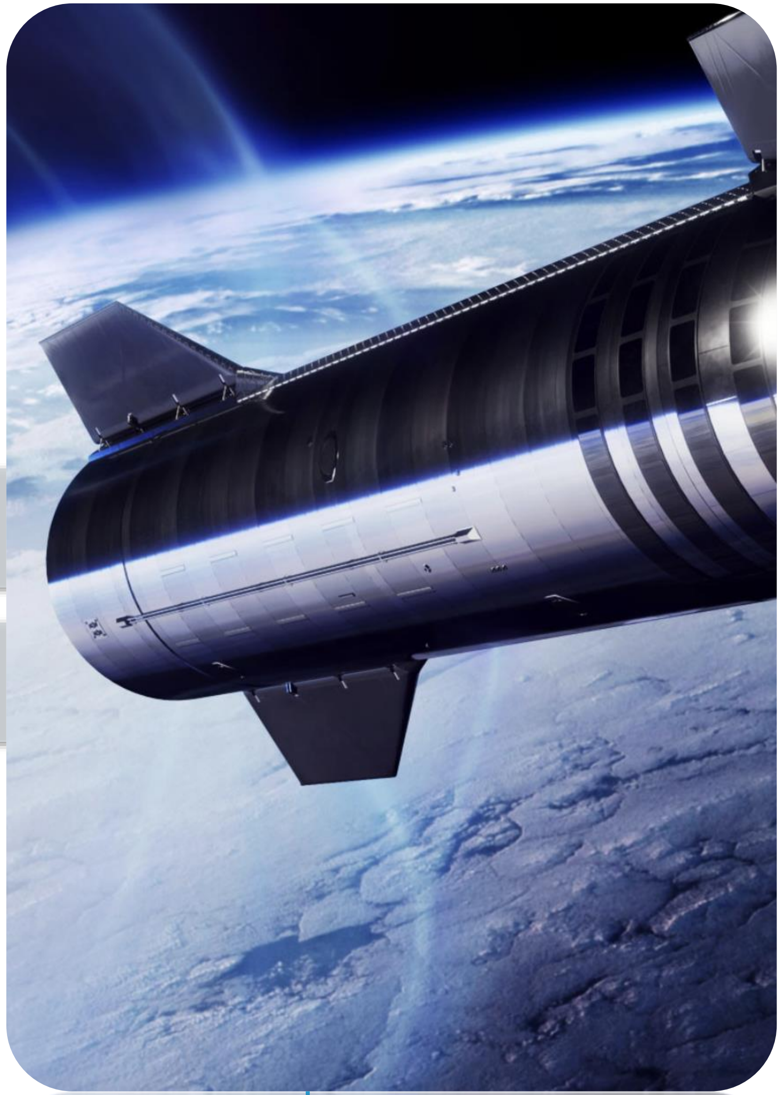
Data Science & SPACEX

By: Michal Uhrinek
Date: 15.11.2022

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



EXECUTIVE SUMMARY

Summary of Methodology

- Data Collection using API technology
- Data Collection with web scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualisation
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

INTRODUCTION

In this capstone I will take the role of a data scientist working for a new rocket company

determine the price of each launch of SPACE X

- Do this by gathering information about Space X and creating dashboard for the team.
- And also determine if Space X will reuse the first stage. (use same rockets, same materials..)
- Instead of using rocket science to determine if the first stage will land successfully, you will train a machine learning model and use public information to predict if Space X will reuse the first stage.

SECTION 1.

METHODOLOGY

METHODOLOGY

EXECUTIVE SUMMARY:

Data collection - methodology:

- Get request to the Space X API and web scraping from Wikipedia
  - Perform data wrangling
    - Clean the data

- Perform exploratory data analysis (EDA), using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Creating machine learning model.

Data collection

The Data Sets are collected by:

- Space X API request
- Web Scrapping

Phase 1	• Enter the URL of the page you want to analyze for this project
Phase 2	• Request and parse the Space X launch data using GET request
Phase 3	• Decode the response content as a JSON file and turn it into a Pandas Dataframe
Phase 4	• Now use the API again to get information about launches using IDs given for each launch
Phase 5	• Filter the data frame to only include Falcon 9 launches and replace null values and get required output (clean data in dataframe)

Data collection – SPACEX API

- Data collection by web-scrapping process is give a flow chart as you can see. For completed Notebook link given below.
- Git Hub URL link:
- https://github.com/mikediego/Mike/blob/62cc4fab04f9b3e8d15eae6e86de15b016000dae/SPACEX-DATACOLLECTION-api.ipynb

1. Falcon9 Launch wiki page from URL
2. Create a Beautiful Soup object from the HMTL
3. Extract all column and names from the HTML table
4. Create an empty dictionary with keys from the column names.
5. Fill up dictionary with launch records extracted from the table rows.
6. Convert dictionary into a CSV dataset.

Data wrangling

- Data wrangling process is give a flow chart as you can see. For completed Notebook link given below.
- Git Hub URL link:
- https://github.com/mikediego/Mike/blob/62cc4fab04f9b3e8d15eae6e86de15b016000dae/Data%20wrangling%20-%20Complete%20the%20EDA.ipynb

Calculate the number of launches on each site.

Calculate the number of occurrences on each orbit.

Calculate the number and occurrence of mission outcome per orbit type.

Create a landing outcome label from Outcome column

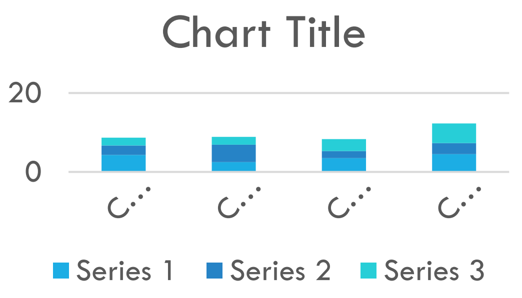
EDA with data visualization

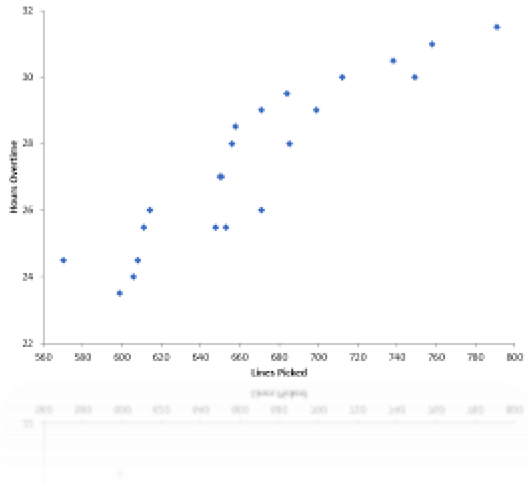
Types of charts used:

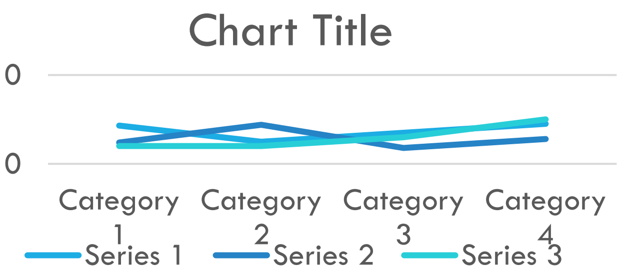
- Scatter plot – Flight Number vs Payload Mass , Flight Number vs Launch Site, Payload and Launch Sites, Flight Number and Orbit Type, Payload and Orbit type
- Bar chart – Success rate of each orbit
- Line plot – success rate and date

EDA with data visualization complete notebook link is below:

GITHUBLINK: https://github.com/mikediego/Mike/blob/62cc4fab04f9b3e8d15eae6e86de15b016000dae/Exploratory%20Data%20Analysis%20using%20AIDA.ipynb







EDA WITH SQL

Summary of SQL queries that were used:

- Display the names of the unique launch sites in the space mission
- Display 5records where launch sites begin with string: ‘cca’
- Display average 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 the ground pas was achieved
- List the names of 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 version which have carried the maximum payload mass. Use a subquery
- List the failed landing outcomes in drone ship, their booster version, and launch site names for in year 2015
- Rank the count of a landing outcomes (such as failure (drone ship) or success (ground pad)) between the date 2010-06-04
- and 2017-03-20, in description order:
- GITHUB LINK:

BUILD AN INTERACTIVE MAP WITH FOLIUM

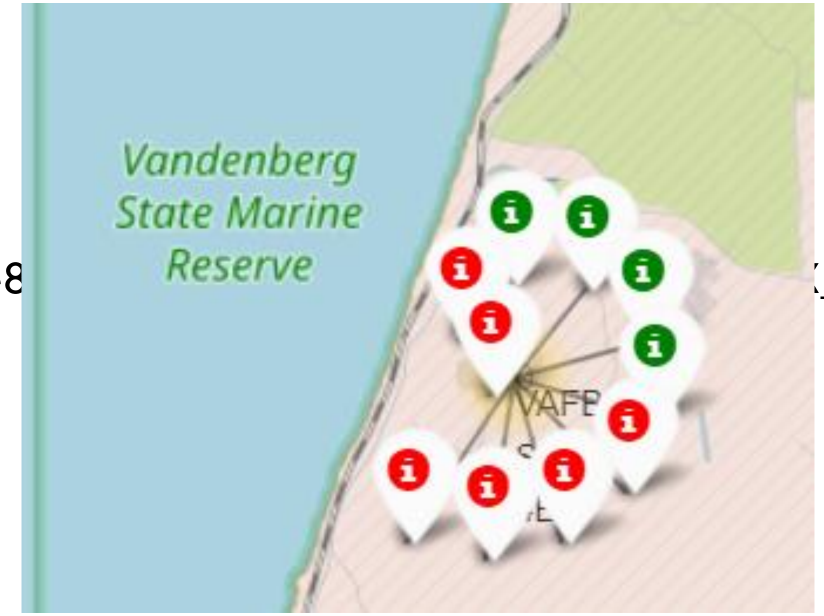
Folium markers were used to show SPACE X launch sites and their nearest important landmark lines like: railway, highway, cities and coastline.

RED = represent rocket launch failures

GREEN = successful rocket launch

GITHUB LINK:

https://github.com/mikediego/Mike/blob/62cc4fab04f9b3e8 _
SQL_LOADDATA.ipynb



Build a dashboard with plotly dash

Pie chart and scatter charts were used to visualize the launch records of SPACE X

These charts displayed the rocket launch success rate per launch site. We are able to get an understanding of the factors that may have been influencing the success rate at each site. Such as payload mass, booster version.

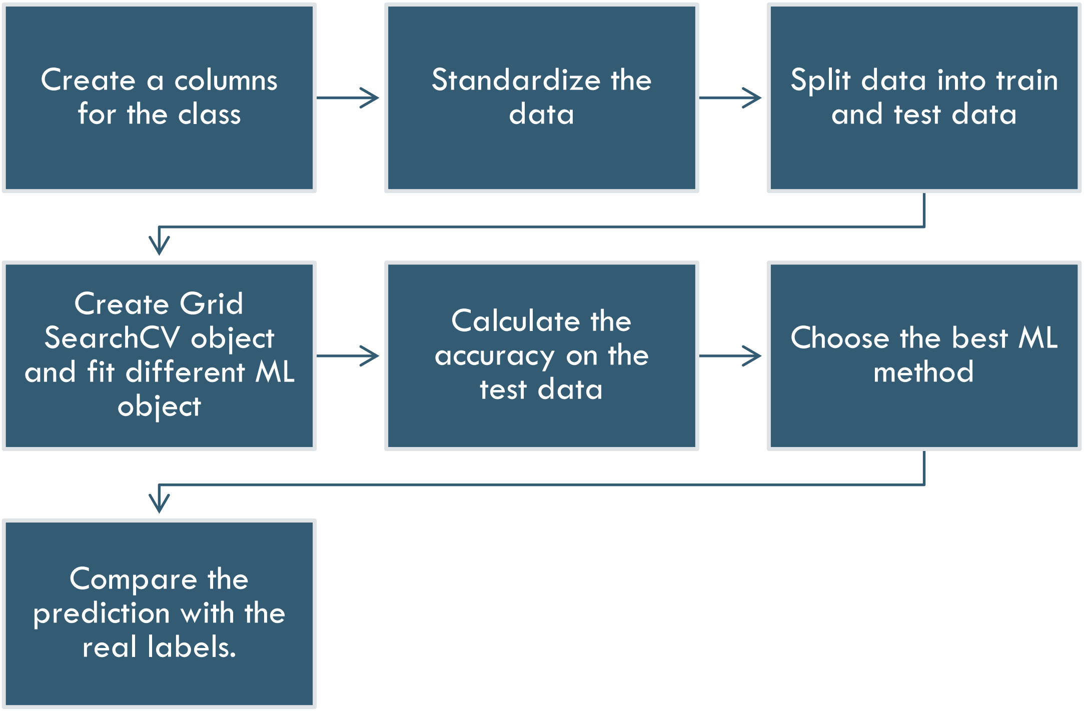
Succesful launches were represented by 1 while failures were represented by 0.

PREDICTIVE ANALYSIS

Scikit-learn is a Machine-learning library that was used for predictive analysis. The following took place:

- Create a machine model pipeline to predict if the first stage will land given the data.

GITHUB LINK:
https://github.com/mikediego/Mike/blob/62cc4fab04f9b3e8d15eae6e86de15b016000dae/SpaceX_Dashboard.py



Results

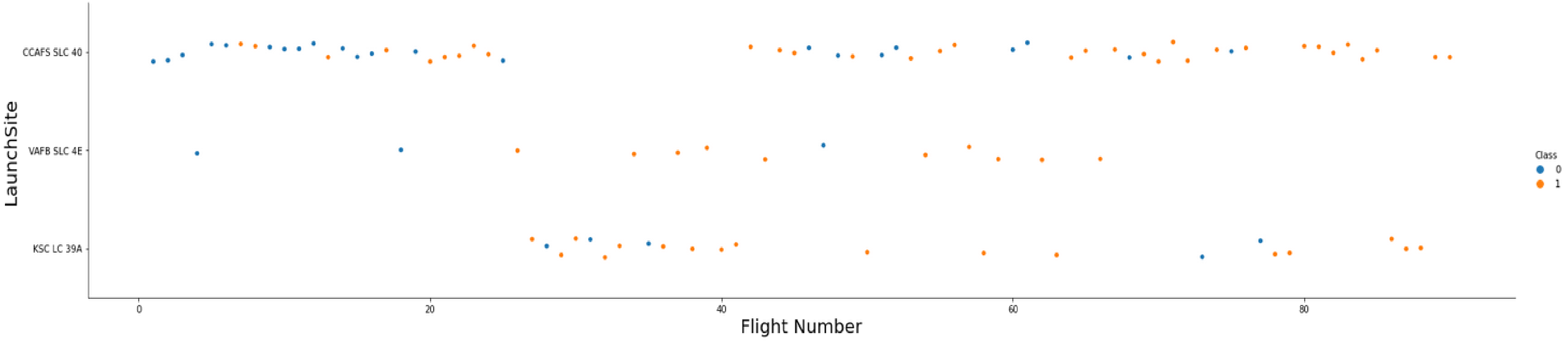
- The exploratory data analysis has shown us that successful landing outcomes are somewhat correlated with flight number. It was also apparent that successful landing outcomes have had a significant increase since the year 2015.
- All launch sites are located near the coast line. Maybe this make it easier to launch it near the water.
- Sites are also located near highways and railways. This may facilitate, transportation of equipment and research materials.
- The machine learning were able to predict landing success of rockets with an accuracy score of 83,33% check the score

SECTION 2

INSIGHTS DRAWN FROM EDA

FLIGHT NUMBER VS. LAUNCH SITE

```
# 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.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



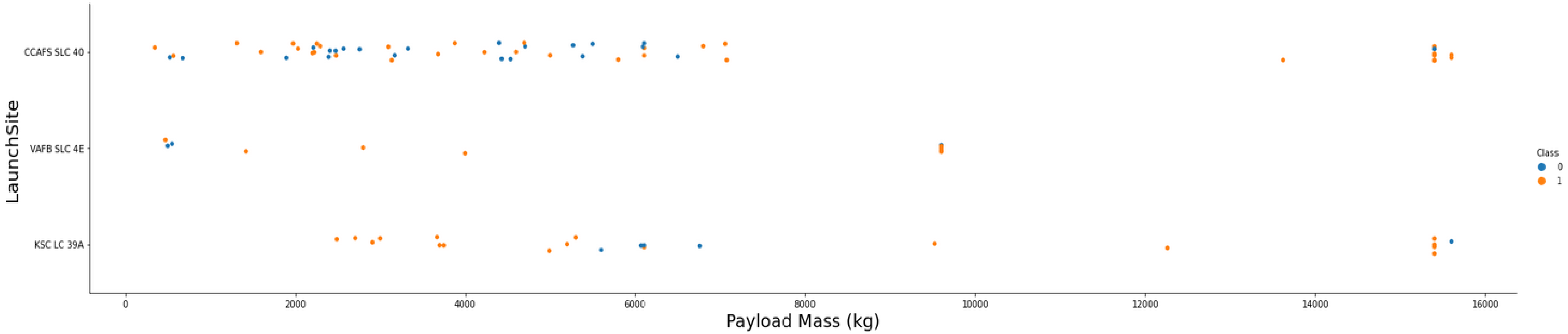
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

It shows that there were more successful landings as the flight numbers increased. Launch site CCAFS SLC40 had the most number of landings.

Payload mass vs. launch site

We also want to observe if there is any relationship between launch sites and their payload mass.

```
# 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.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass (kg)",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



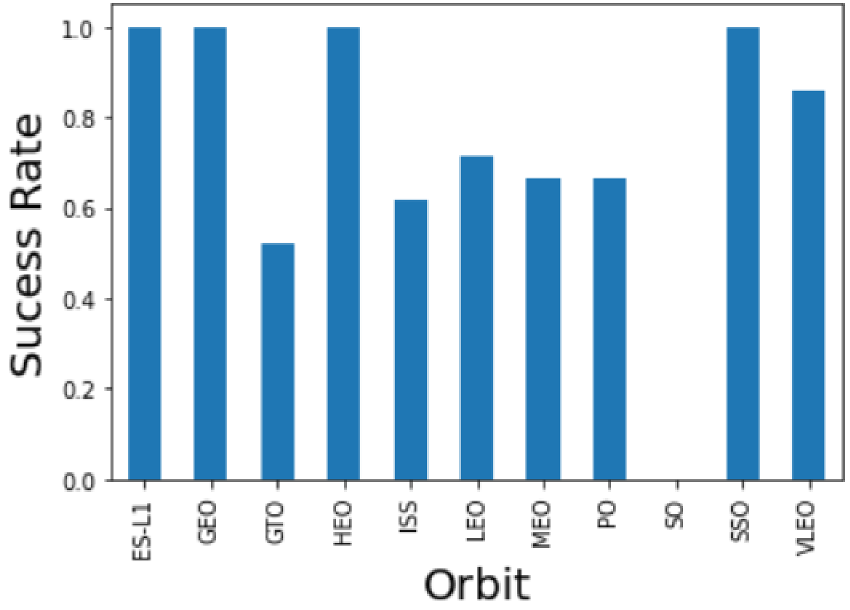
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success rate vs. orbit type

The highest success rate orbits are:

- **ES-L1**
- **GEO**
- **SSO**
- **HEO**

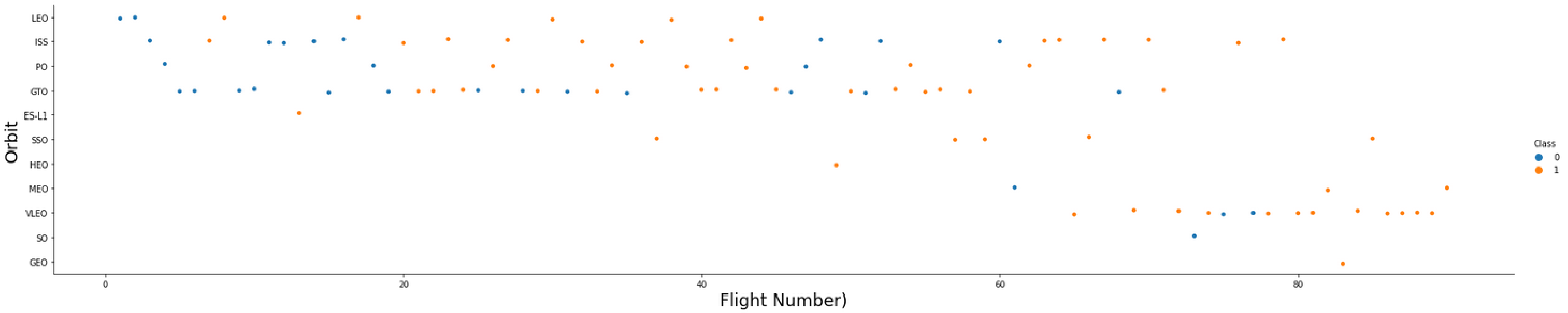
```python
# HINT use groupby method on Orbit column and get the mean of Class column
df.groupby(['Orbit']).mean()['Class'].plot(kind='bar')
plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Sucess Rate",fontsize=20)
plt.show()
```



Analyze the ploted bar chart try to find which orbits have high sucess rate.

Flight number vs. orbit type

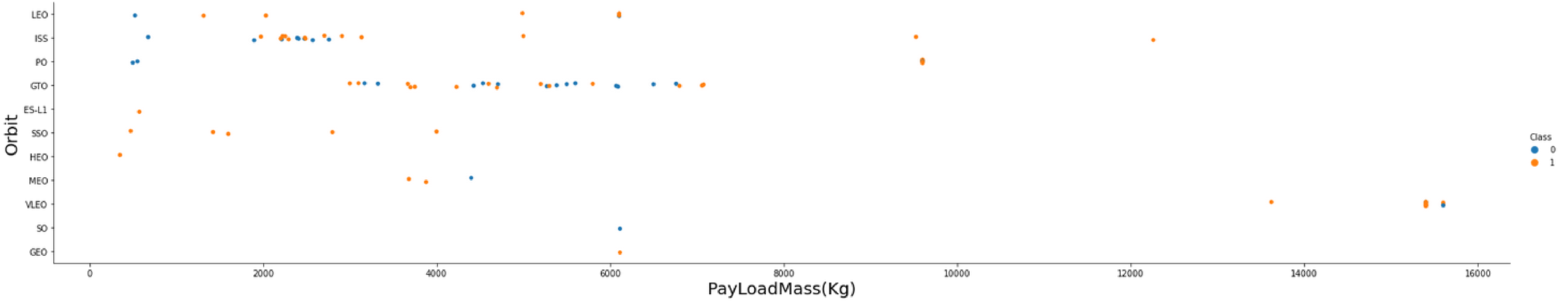
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload mass vs. orbit type

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayLoadMass(Kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

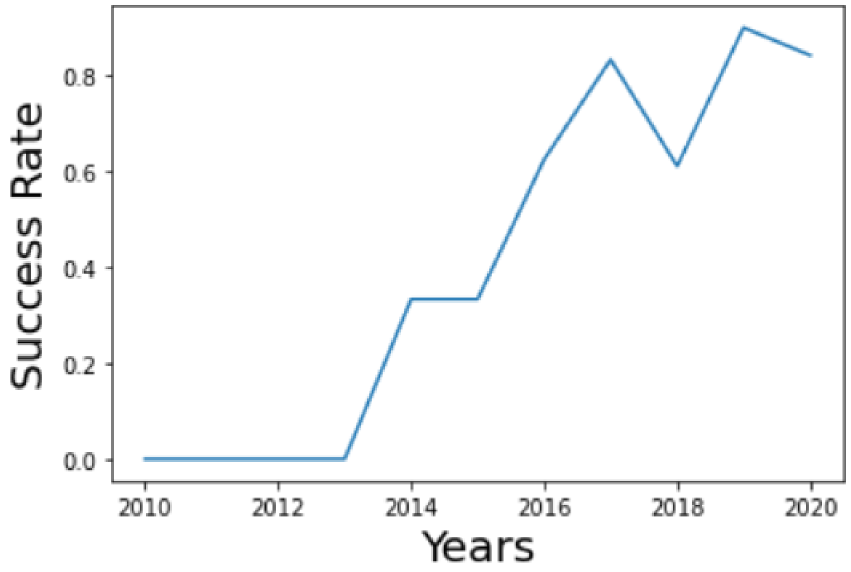


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

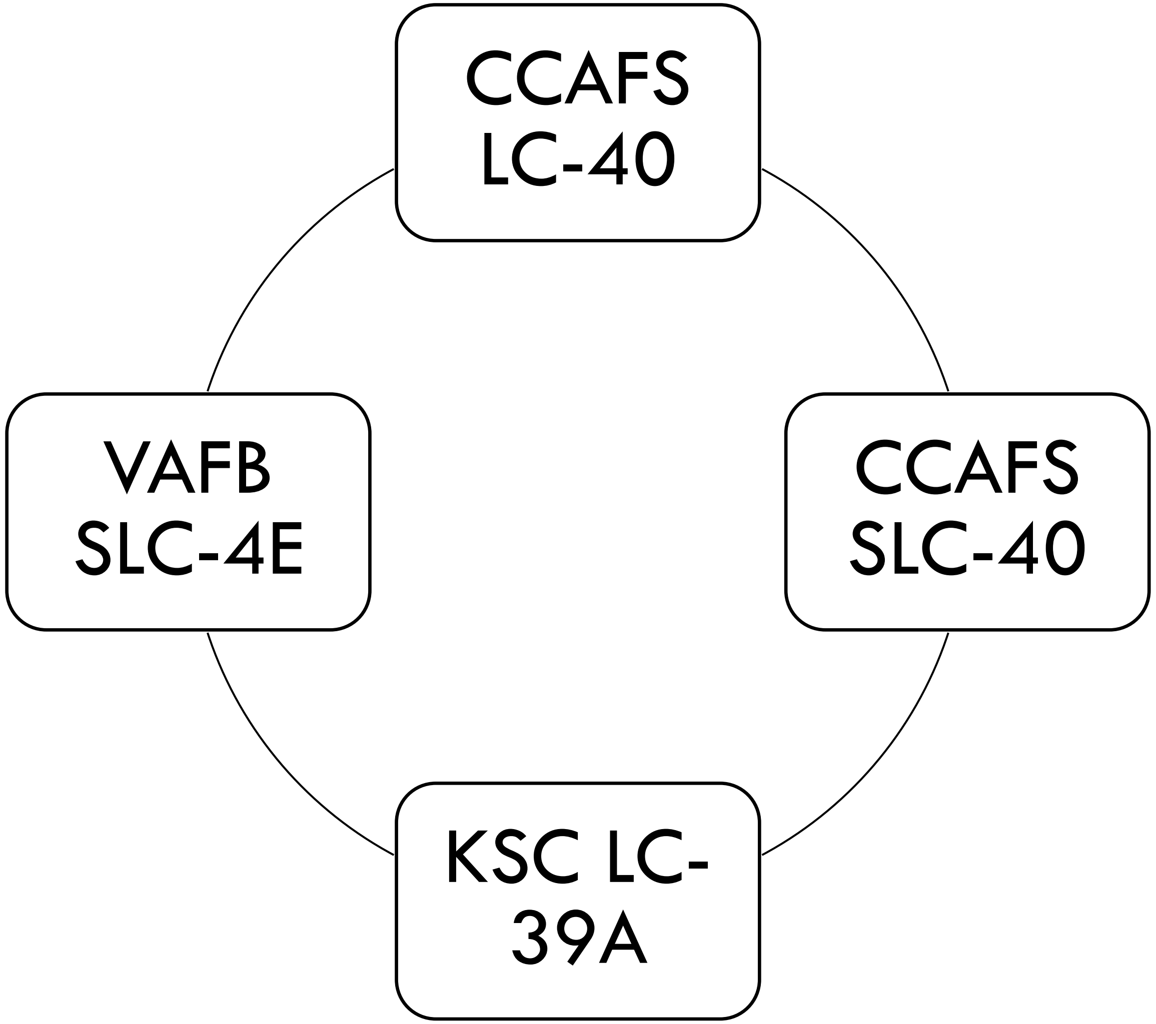
Launch success early trend

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
df['Year'] = pd.DataFrame(Extract_year(df['Date'])).astype('int')
sns.lineplot(x = df['Year'].unique() , y = df.groupby(['Year'])['Class'].mean())
plt.xlabel("Years",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



you can observe that the sucess rate since 2013 kept increasing till 2020

All launch site names



Launch Site Names Beginning with 'CCA'

```
In [18]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
 * ibm_db_sa://gfd86828:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[18]:

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

These are 5 records where launch sites begin with the letters 'CCA'. As we can see, there are other organizations besides Space X that were testing their rockets.

Total Payload Mass

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

In [23]:

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA(CRS)';
```

```
 * ibm_db_sa://gfd86828:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[23]:

1

- The information in the picture displays the total payload mass carried by boosters launched by NASA

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://gfd86828:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

1
2928

The average payload mass carried by F9 v1.1 was 2928 kg

First Successful Ground Landing Date

- First Successful Ground Landing Date was in Dec 22, 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

It appears that there are only 4 Boosters with a payload mass between 4000 and 6000 which are:

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
]: %sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'
```

```
 * ibm_db_sa://gfd86828:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

]:

1
100

The Above picture show the total number of successful and failure mission outcomes

Boosters That Carried the Maximum Payload Mass

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

In [34]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT max(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * ibm_db_sa://gfd86828:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[34]:

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

From the above picture it shows that 12 boosters have carried the maximum payload mass of 15600 kg.

2015 Launch Records - Failed Landing Outcomes

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

In [18]:

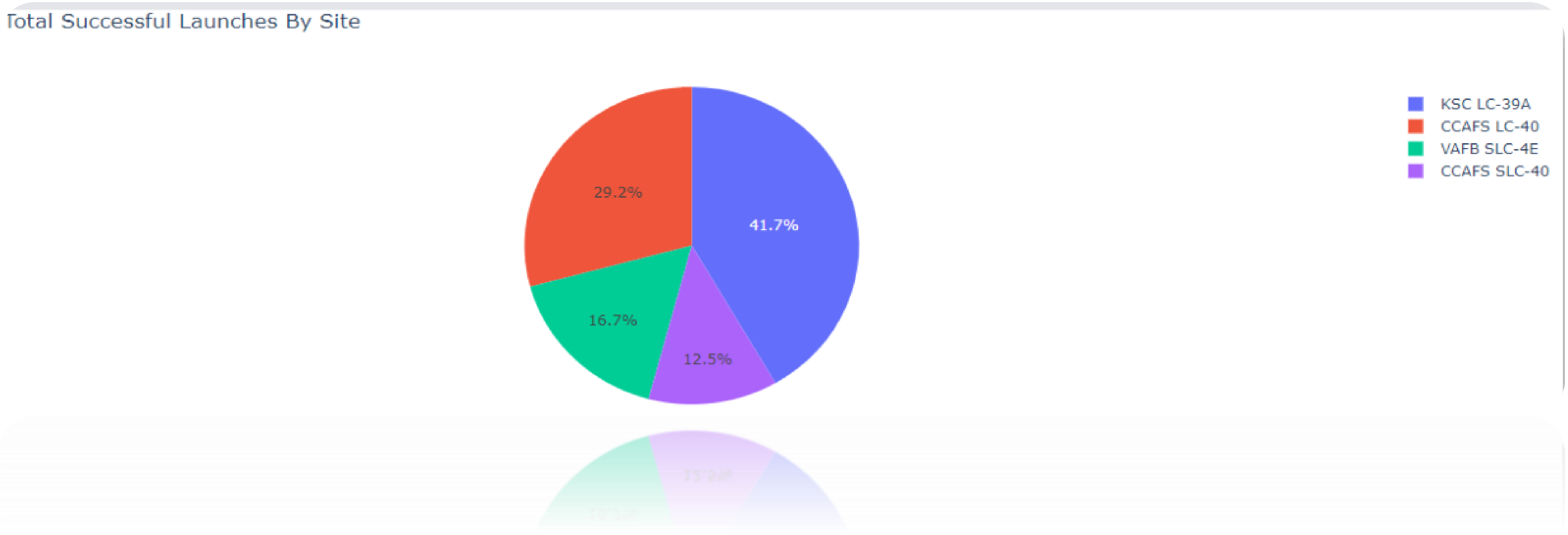
```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

- 2 boosters F9 v1.1B1012_CCAFS LC-40 and F9v1.1B1015 CCAFS LC-40 failed to land at 2015

TOTAL SUCCESSFUL LAUNCHES BY SITE



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

In [42]:

```
%sql select * from SPACEXTBL where Landing__Outcome = 'Success (ground pad)' or and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```

```
 * ibm_db_sa://gfd86828:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb
Done.
```

Out[42]:

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)
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

- The number of successful landings have increased since 2015.

Section 3

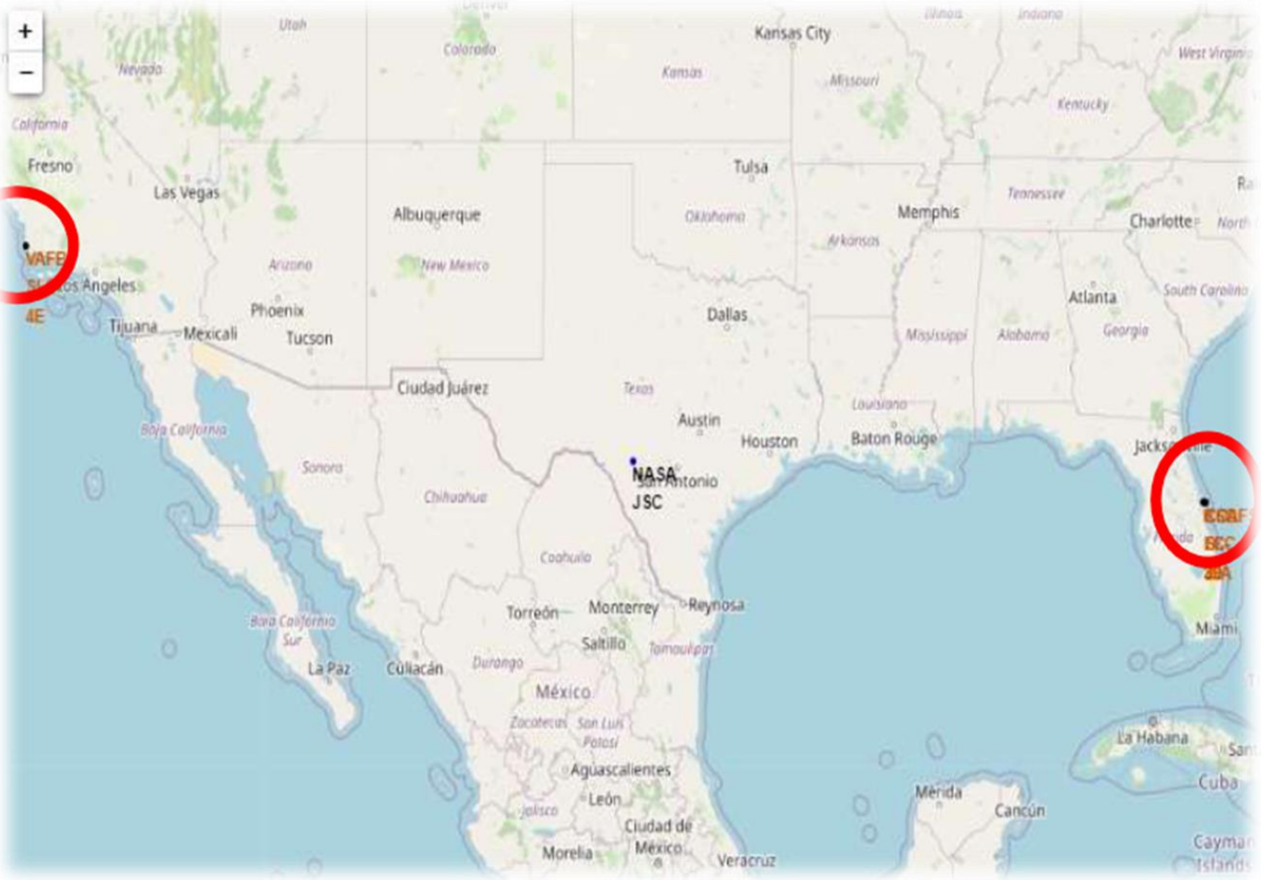
BUILD AN INTERACTIVE MAP WITH FOLIUM

SECTION 3

BUILD AN INTERACTIVE MAP WITH FOLIUM

Launch Site Locations

All launch sites are in very close proximity to the coast
and they are also a couple
thousand kilometers away from the equator line.

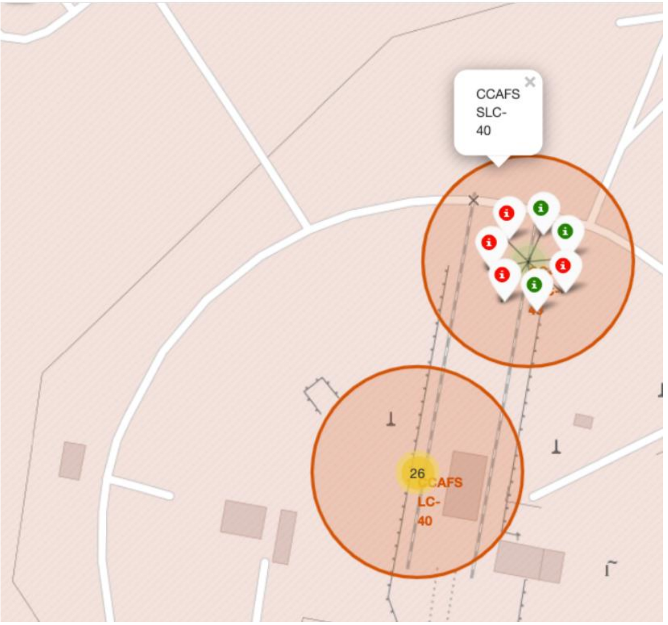


GITHUB LINK:

https://github.com/mikediego/Mike/blob/62cc4fab04f9b3e8d15eae6e86de15b016000dae/SpaceX_visualisation_folium.ipynb

Success Rate of Rocket Launches

- The successful launches are represented by a green marker while the red marker represents failed rocket launches.




```
# Function to assign color to launch outcome
def assign_marker_color(launch_outcome):
    if launch_outcome == 1:
        return 'green'
    else:
        return 'red'

spacex_df['marker_color'] = spacex_df['class'].apply(assign_marker_color)
spacex_df.tail(10)
```

	Launch Site	Lat	Long	class	marker_color
46	KSC LC-39A	28.573255	-80.646895	1	green
47	KSC LC-39A	28.573255	-80.646895	1	green
48	KSC LC-39A	28.573255	-80.646895	1	green
49	CCAFS SLC-40	28.563197	-80.576820	1	green
50	CCAFS SLC-40	28.563197	-80.576820	1	green
51	CCAFS SLC-40	28.563197	-80.576820	0	red
52	CCAFS SLC-40	28.563197	-80.576820	0	red
53	CCAFS SLC-40	28.563197	-80.576820	0	red
54	CCAFS SLC-40	28.563197	-80.576820	1	green
55	CCAFS SLC-40	28.563197	-80.576820	0	red

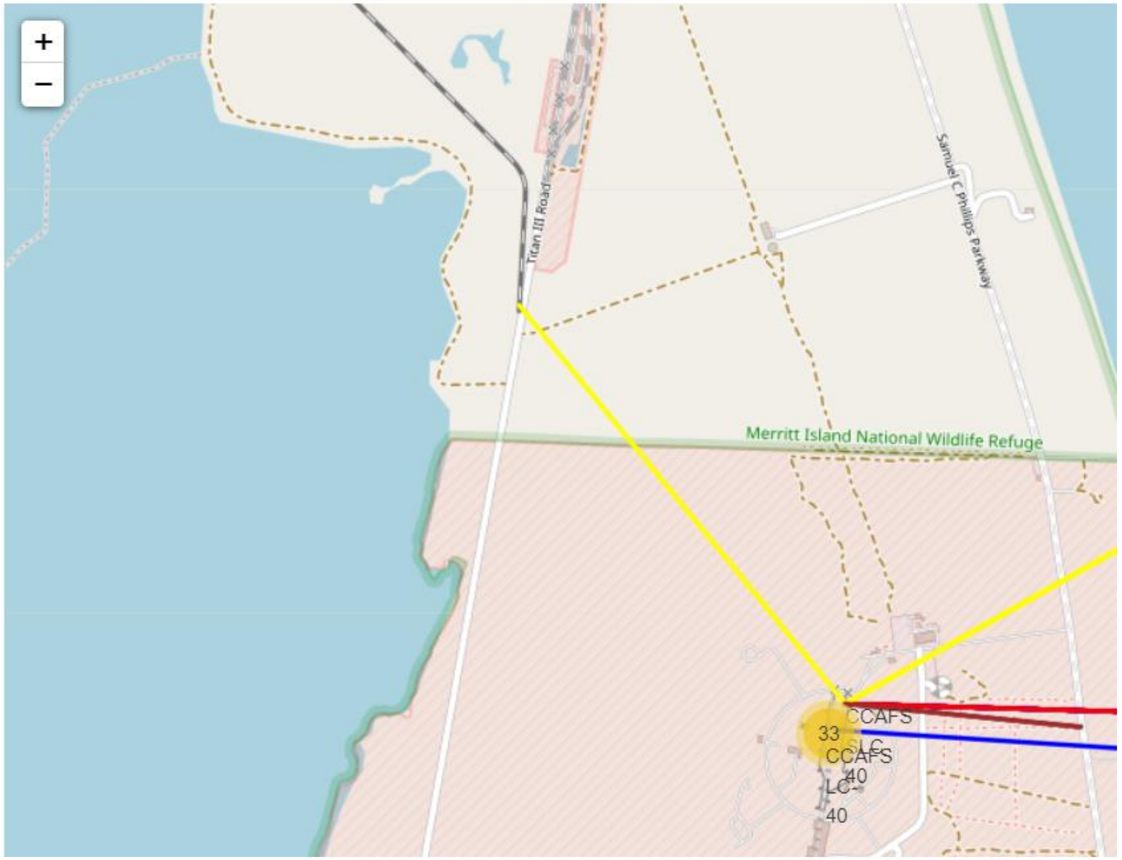
Surrounding Landmarks

It appears that launch sites are usually set up at least 18 km away from cities. This may be because of the desire to prevent any crashes near populated areas.

It is also apparent that launch sites are in very close proximity to railways and highways. Perhaps, due to the necessary transportation requirements for rocket parts.

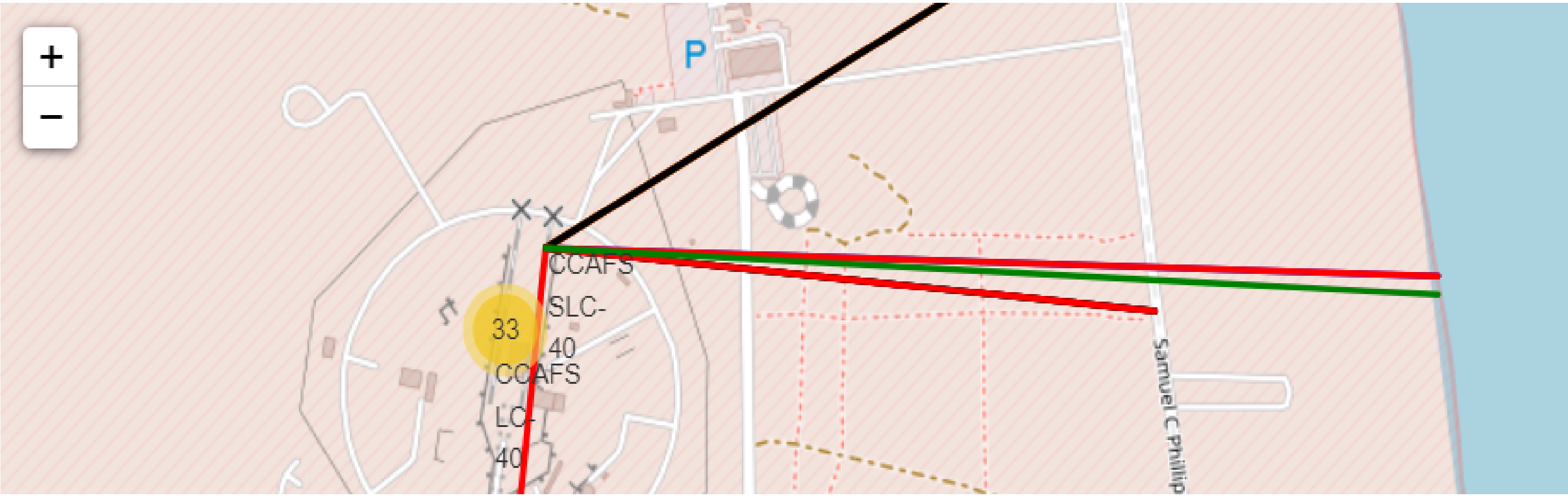
The sites are close the coast line. This is evident with the many rocket landing tests on water bodies like the ocean.

```
# highway
points = [[28.56316,-80.57684], [28.56264,-80.57071]]
folium.PolyLine(points, color='brown').add_to(site_map)
lines=folium.PolyLine(locations=[(28.56316,-80.57684),(28.56264,-80.57071)], weight=1)
site_map
```



```
# Create a `folium.PolyLine` object using the coastline coordinates and launch site coordinate
# lines=folium.PolyLine(locations=coordinates, weight=1)
points = [[28.56316,-80.57684], [28.56278,-80.56785]]
folium.PolyLine(points, color='green').add_to(site_map)
lines=folium.PolyLine(locations=[(28.56316,-80.57684),(28.56278,-80.56785)], weight=1)

site_map
```

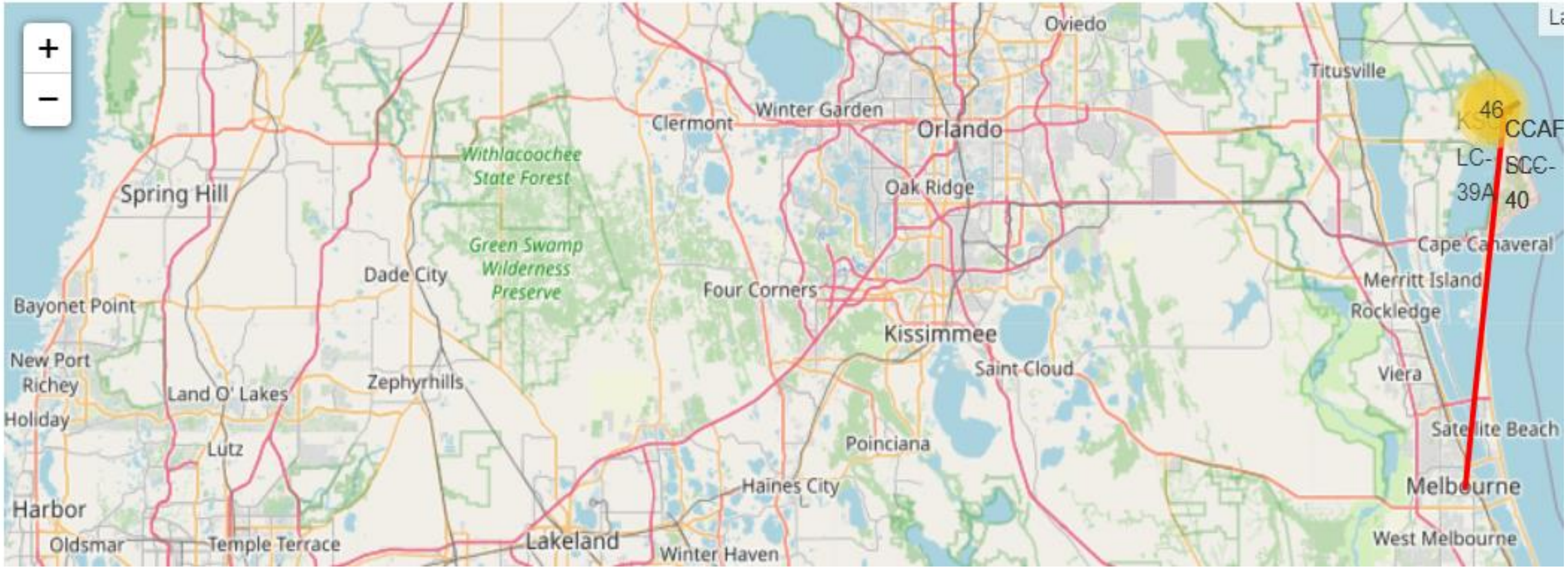


COASTLINE

```
# highway
points = [[28.56316,-80.57684], [28.56264,-80.57071]]
folium.PolyLine(points, color='brown').add_to(site_map)
lines=folium.PolyLine(locations=[(28.56316,-80.57684),(28.56264,-80.57071)], weight=1)
site_map
```

HIGHWAY

```python
# city Melbourne
folium.PolyLine(points, color='red').add_to(site_map)
lines=folium.PolyLine(locations=[(28.56316,-80.57684),(28.10238,-80.63416)], weight=2)
site_map
```



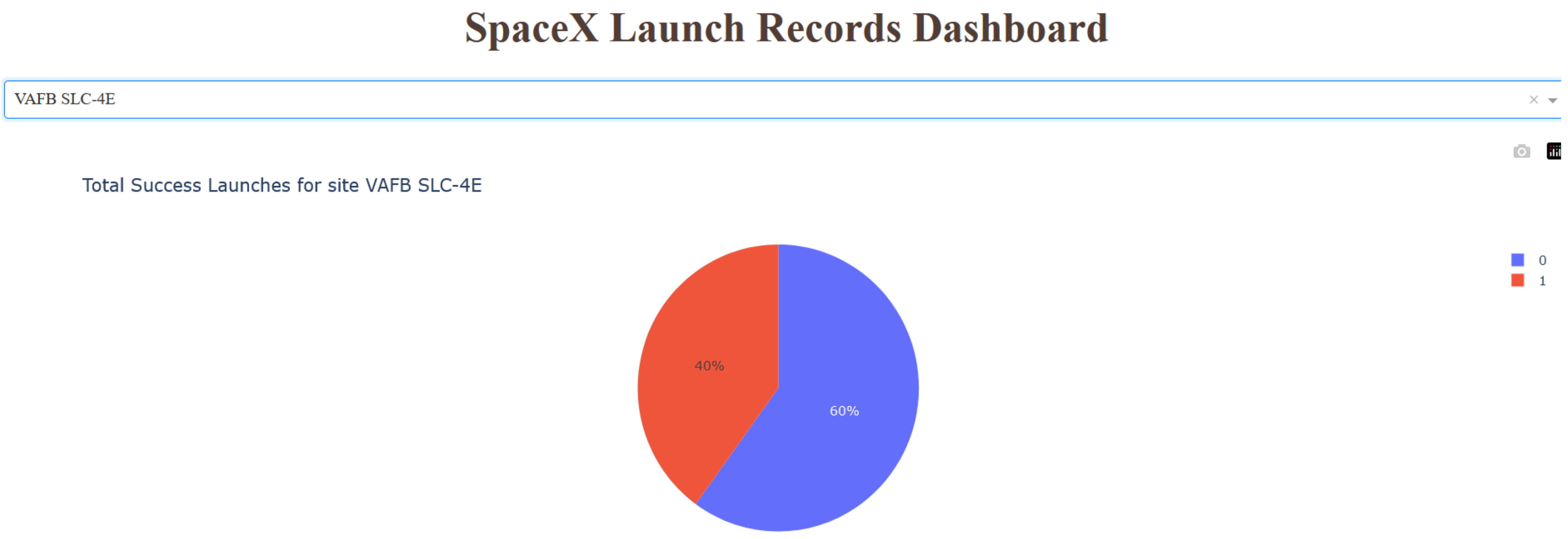
MELBOURNE

SECTION 4

BUILD A DASHBOARD WITH PLOTLY

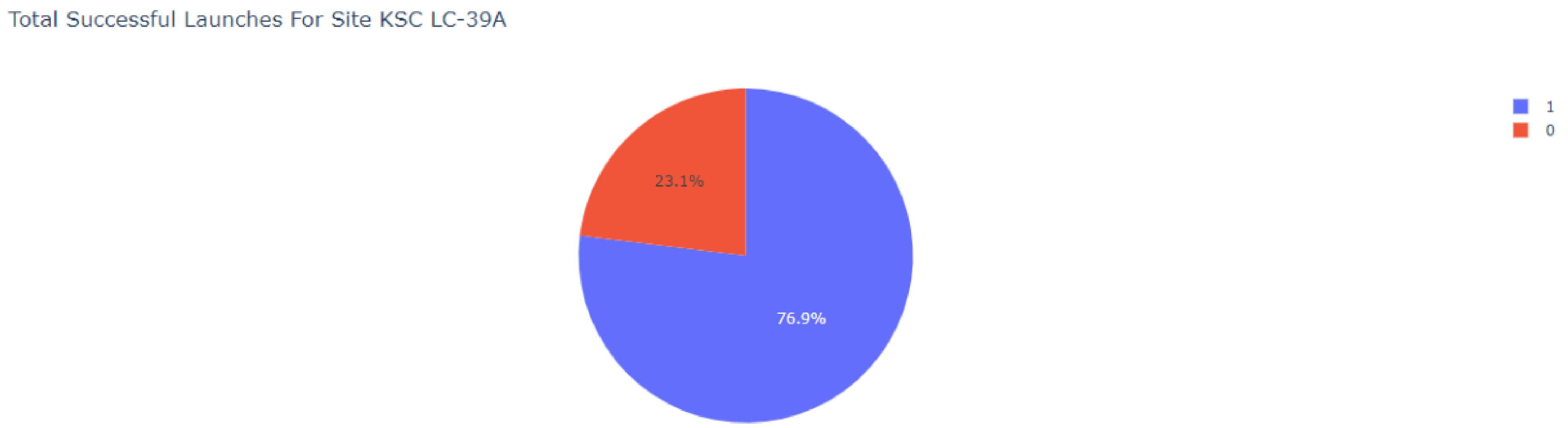
BUILD A DASHBOARD WITH PLOTLY

Successful Launches by Site



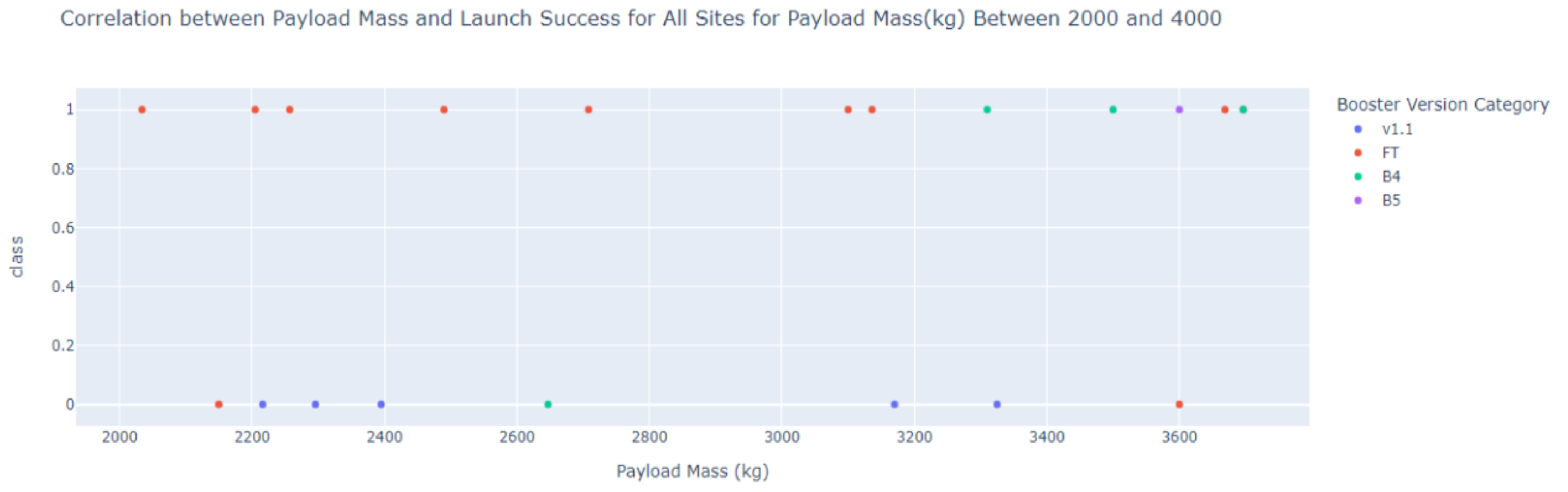
You can see from the plot that Site VAFB SLC-4E has only 1 successful launch.

Total Successful Launches for Site KSC LC-39A



You can see that 76.9% of the total launches at site KSC LC-39A were successful. This is a the highest success rate of all the different launch sites.

Payload Mass vs. Launch Success for All Sites



It appears that the payload range between 2000 kg and 4000 kg has the highest success rate.

Section 5

PREDICTIVE ANALYSIS(CLASSIFICATION)

Predictive analysis (classification)

Classification Accuracy

Find the method performs best:

```
accuracy = [svm_cv_score, logreg_score, knn_cv_score, tree_cv_score]
accuracy = [i * 100 for i in accuracy]

method = ['Support Vector Machine', 'Logistic Regression', 'K Nearest Neighbour', 'Decision Tree']
models = {'ML Method':method, 'Accuracy Score (%)':accuracy}

ML_df = pd.DataFrame(models)
ML_df
```

	ML Method	Accuracy Score (%)
0	Support Vector Machine	83.333333
1	Logistic Regression	83.333333
2	K Nearest Neighbour	83.333333
3	Decision Tree	83.333333

You can see that All the methods have an identical accuracy score of 83.33%, so we decided to use Logistic Regression for the classification.

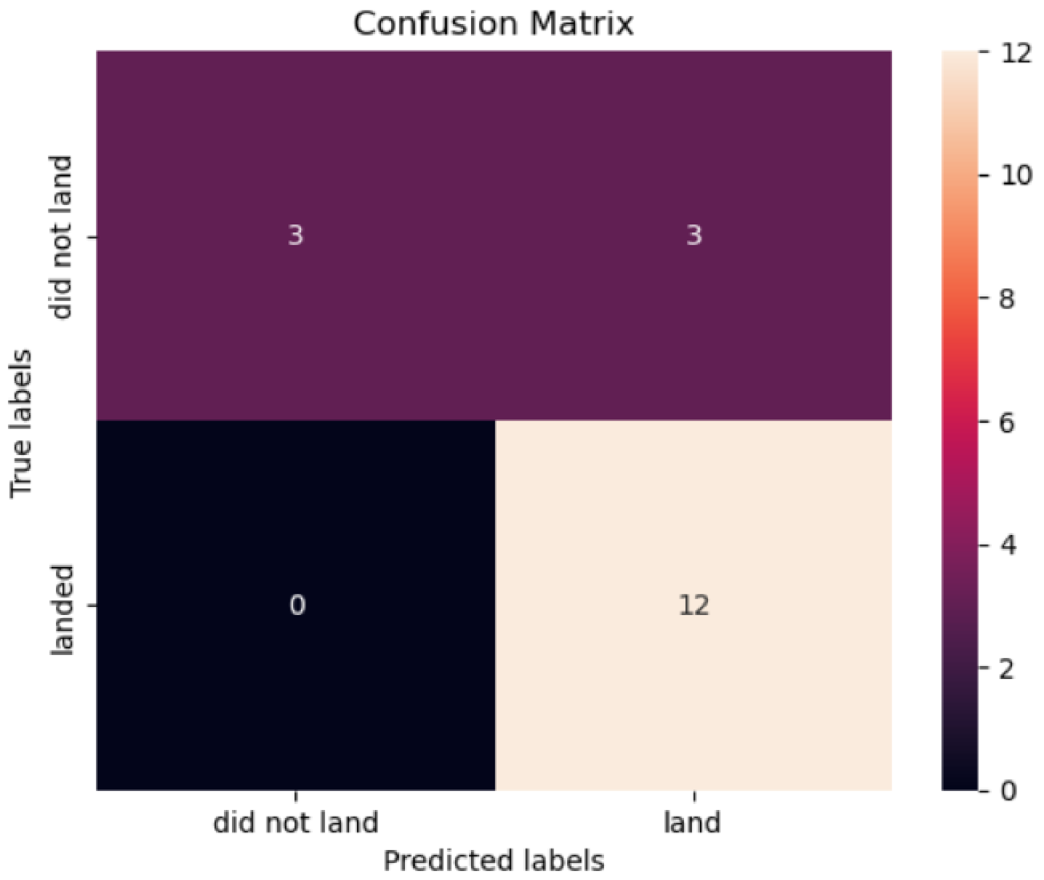
Confusion Matrix

- The chart shows the confusion matrix of the Logistic Regression model that was chosen.
- The model only failed to accurately predict
  3 labels.

GITHUB LINK:

https://github.com/mikediego/Mike/blob/62cc4fab04f9b3e8d15eae6e86de15b016000dae/SpaceX_MachineLearning%202022.ipynb

```
#confusion matrix
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



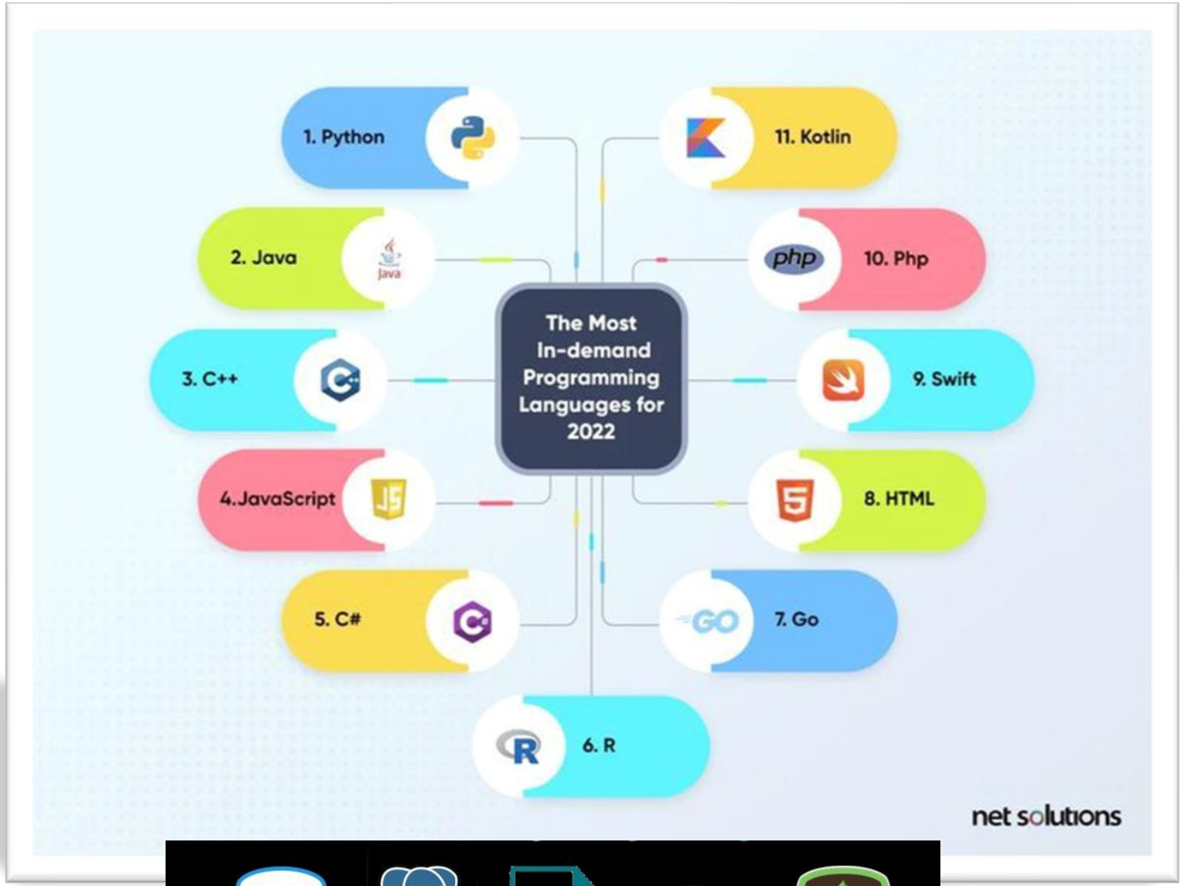
Conclusions

In order to compete with SpaceX Through this process, a general picture of their success methods are:

- All their launch sites are located near the coast, away from nearby cities. This enabled to them to test their rocket landings without much interference.
- Site KSC LC-39A had the highest launch success rate out of all the launch sites.
- From 2015 onwards, the success rate of rocket landings significantly increased. It was also apparent that landing success increased with flight number

All this data was used to train a machine learning model that is able to predict the landing outcome of rocket launches with 83.33% accuracy.

DISCUSSION about new and nowadays trend in programming and databases



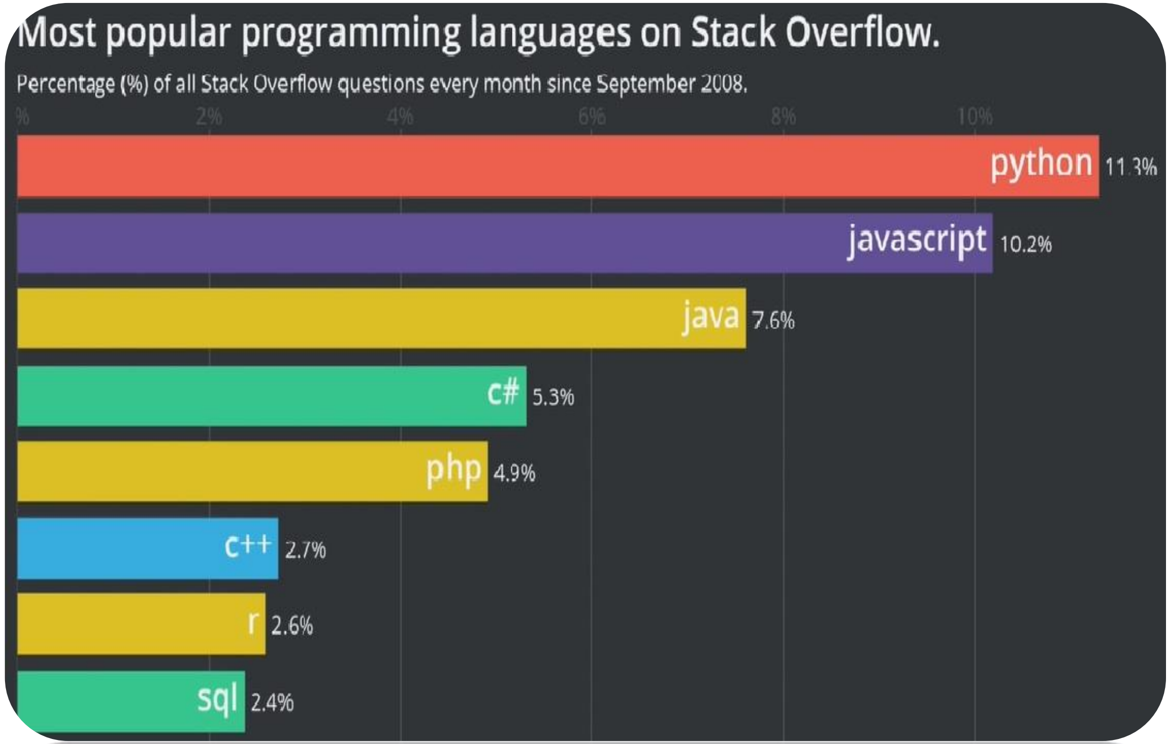
What to use?

Where?

When?

PROGRAMMING LANGUAGE TRENDS

Current Year



Next Year

1. **Kotlin**
2. **TypeScript**
3. **Swift**
4. **R**
5. **Scala**

For more information why those langues will play future roll and become more popupar next year clic here: *https://www.javaassignmenthelp.com/blog/future-programming-languages/*

PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

Findings

Finding 1. Languages like Python, Java, R, JavaScript, SQL will still be used.

Finding 2. Those languages are evolving so there is no problem with them being 'old'.

Finding 3. NOSQL and SQL are popular database languages, but sooner or later we switch towards NOSQL because we don't need more than 1table, 1file, 1document. It is no relational database so it doesn't need 4tables to get you one result which saves time, money and storage.

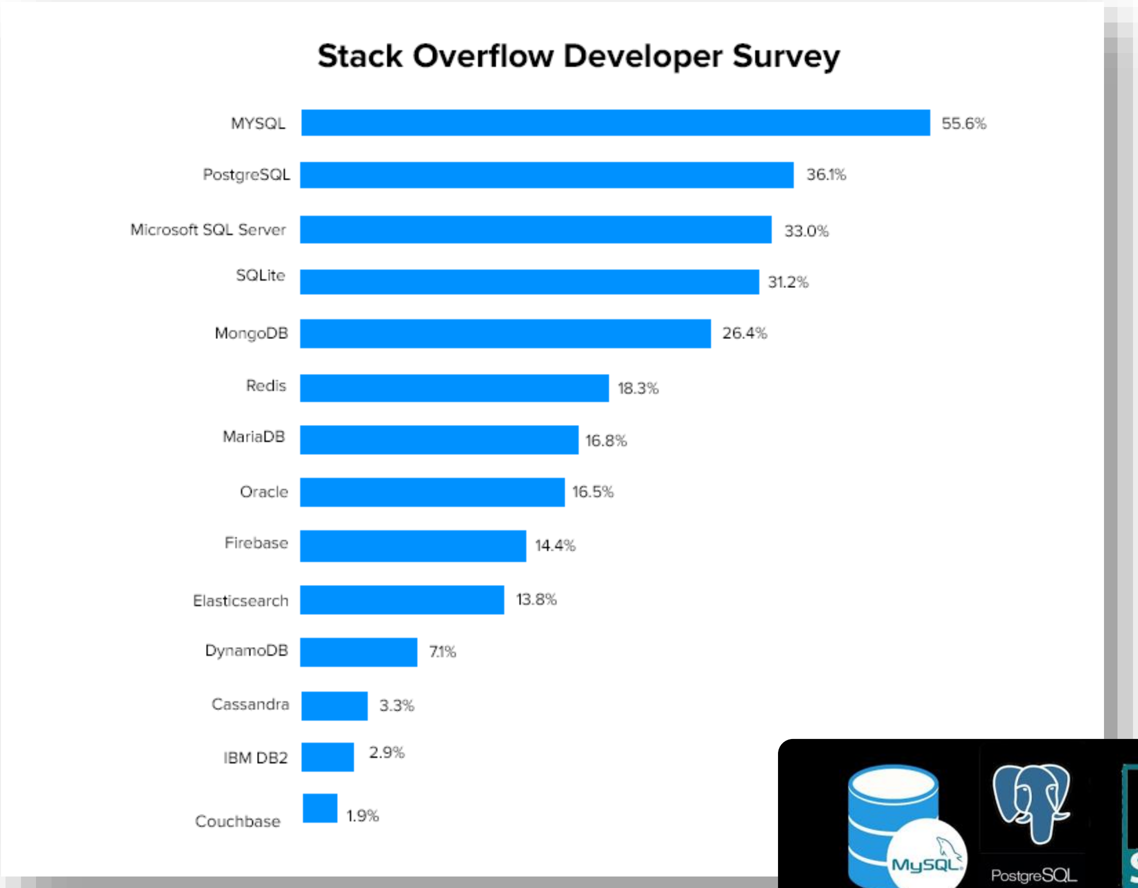
Finding 4. R has a huge usage in AI and next years its gonna be used for machine learning and AI even more.

Implications

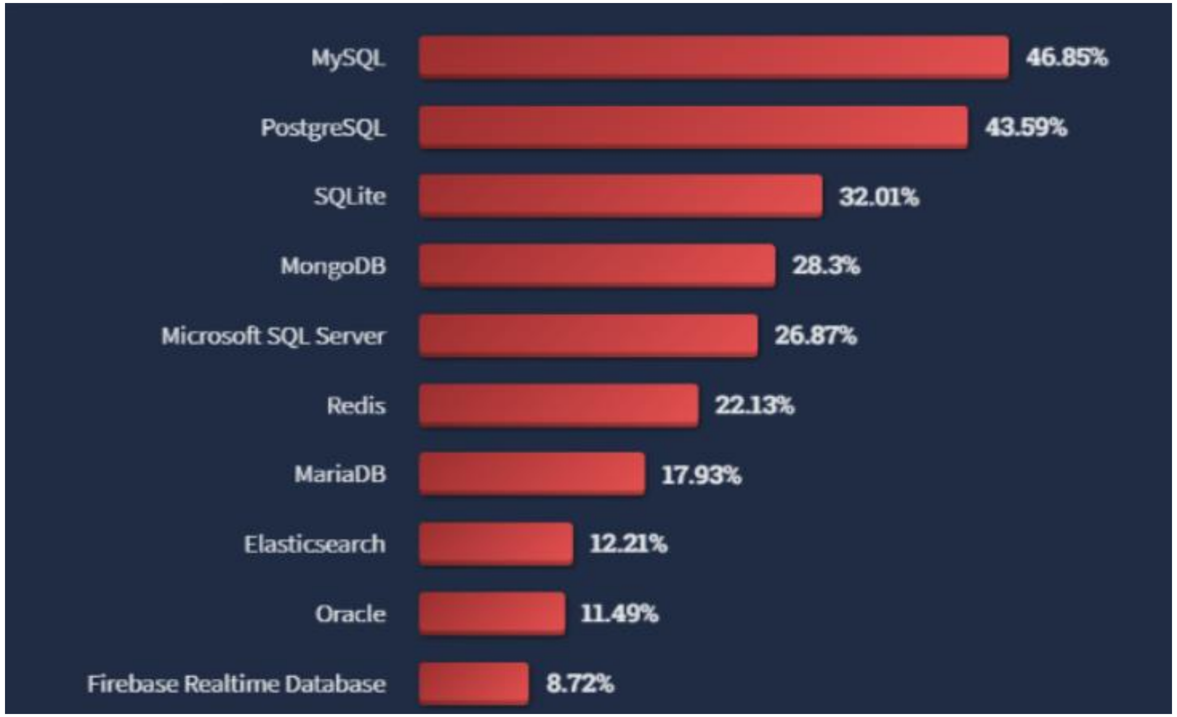
- Implication 1. Make sure you stay up to date with those language and always learn something new. Python was here 30years and its still evolving so if you stop, in 5years you will need to start study again.
- Implication 2. Take a look at new languages because JavaScript will be soon with Java replaced by TypeScreept, and Kotlin. Because its efficiency and language its easier to learn and write.

DATABASE TRENDS

Current Year



Next Year



DATABASE TRENDS - FINDINGS & IMPLICATIONS

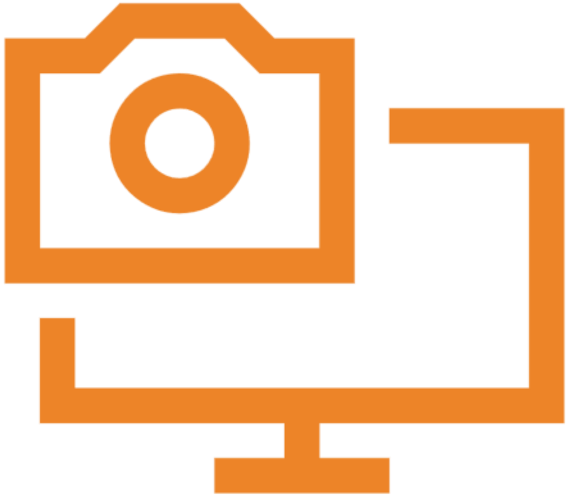
Findings

- Finding 1. Technologies like artificial intelligence, machine learning, data science, and data engineering all rely on some data source somehow. Countless surveys and blogs have data-related jobs at the top with all the emerging technologies today. Since you need SQL to work with databases, learning SQL is definitely among the first steps towards a career that involves data crunching or use in any way.
- Finding 2. MySQL is not the best if you want advanced data protection features like throttling and masking. It is also not the best with semi-structured data like JSON. SQL. FACEBOOK, YOUTUBE, UBER use it in their applications. YouTube uses MySQL to store all the metadata for the videos.
- PostgreSQL handles semi-structured data such as JSON and has great support for distributed. WHATSUP, TWITCH, NASA, APPLE, SPOTIFY, INSTAGRAM, REDEDIT.
- Finding 3 ORACLE - In fact, it is one of the most mature and stable databases today. It is used by major fortune 500 companies around the world for their transactions. Its not free and for 100 employees it can cost 1000s of $. NETFLIX, LINKED IN, EBAY
- **NoSQL** databases (aka "not only SQL") are non-tabular databases and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads. Storing financial data and healthcare records). Misconception - NoSQL databases like MongoDB do, in fact, support ACID transactions.
- **MongoDB -** Works with semi-structured data such as JSON or XML…
  - MongoDB is an object-oriented document-based database that stores data inside a collection of documents rather than in rows and columns like in the other databases
  - Naturally, the lack of pure structure may come at the cost of some ACID transaction inconsistency. If you primarily deal with structured data in your transactions, it is better to use one of the other options.
- **Microsoft SQL Server – PAIND. BUT CHEAPER THAN ORACLE.**
  - Microsoft's release of cloud-based Azure SQL database. Azure is a multi-tenant database in which different customers can access a single instance of the database and is a platform-as-a-service (PaaS) offering. Due to the similarities, you can learn SQL Server then switch to Azure as needed. SQL from A to Z in SQL Server is a great track to get you started if you want to start learning SQL for a Microsoft product.

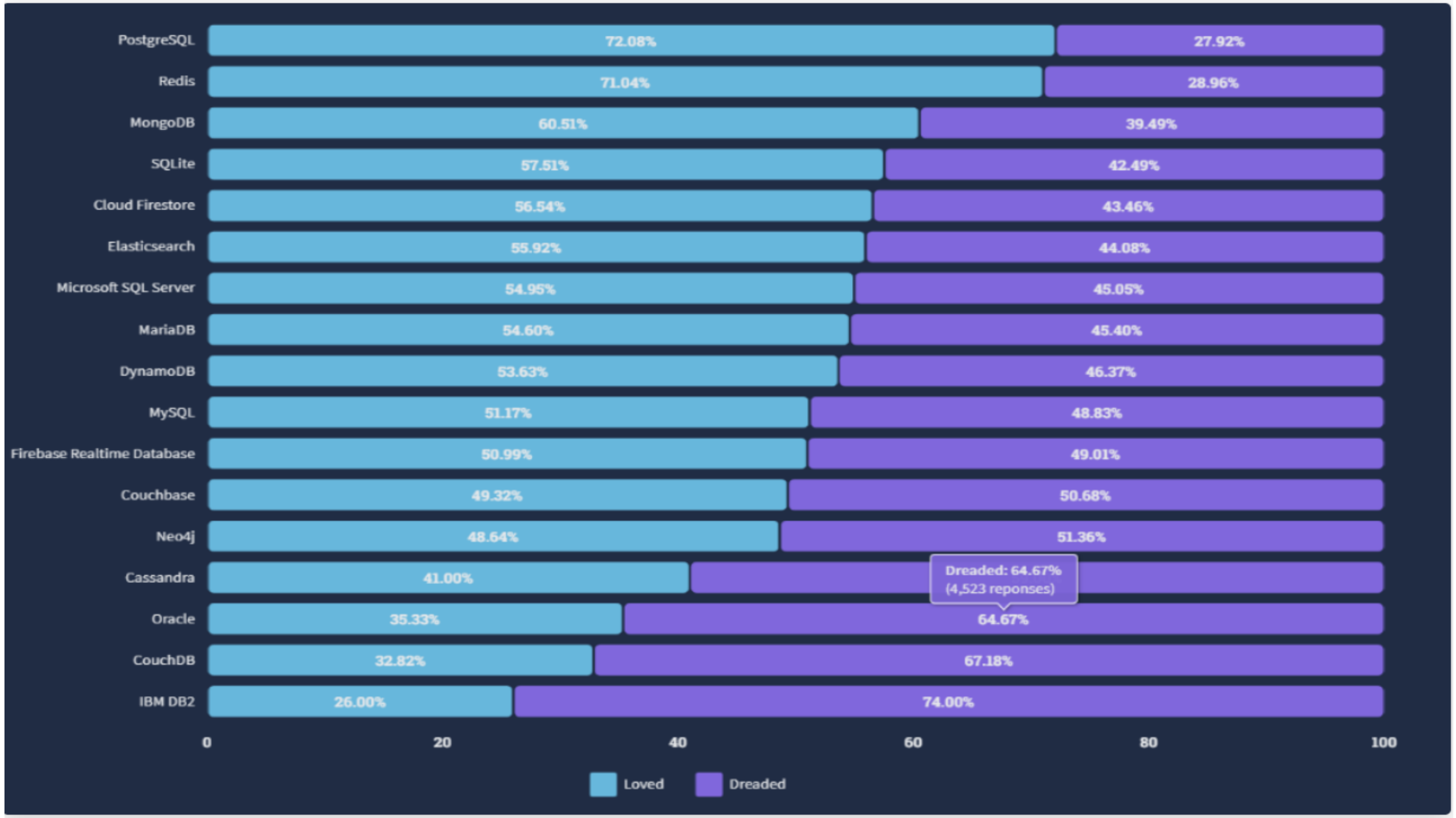
Implications

- Implication 1: If you want to work with unstructured data then you can easily go for NoSQL database.
- Implication 2: If you will work strictly with Windows and Miscrosoft products than I recommend to go for Miscrosoft SQL Server. You can work with unstructured data, its paid and you can use SQL on Azure.
- Implication 3: Generally you shoud take a closer look at MySQL, ORACLE, PostgreSQL for more wider usage.

APPENDIX



With nowadays data and questioning developers and users which database systems they would like to use and that they regret to learn we can see the outcome here:



THANK YOU

CREATED BY: MICHAL UHRINEK