IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mantas Zilevicius

12 Mar 2023

Outline

- EXECUTIVE SUMMARY
- INTRODUCTION
- METHODOLOGY
- RESULTS
- CONCLUSION
- APPENDIX

Executive Summary

Summary of Methodologies

- Data was collected via API and Web Scraping
- Applied EDA using Data Wrangling, SQL queries, Data visualization including Plotly Dash and Folium
- Used different ML models for predictions

Summary of all results

- KSC LC 39A is the best launch site and the best performing booster version is FT
- Machine Leaning models can help to predict landing outcome with high accuracy rate

Introduction

Project background and context

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch

Problems you want to find answers

- What factors has the most influence for Falcon 9 first stage to land successfully?
- Does launch site location has any influence on outcome?
- How can SpaceX increase successful landing rate for Falcon 9 first stage?

Section 1

Methodology

Methodology

- Executive Summary
- Data collection methodology:
  - Data was collected from 2 data sources (from SpaceX API and Wikipedia using Web Scraping technique)
- Perform data wrangling
  - Data was cleaned and prepared for further analysis and Machine Learning
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The collected data has been normalized, divided into training and test sets, and assessed by four different classification models using varying parameter combinations to determine accuracy.

Data Collection

Data sets were collected From SpaceX API and Wikipedia using Web Scraping

- Data was collected via API and Web Scraping methods
- Collected data was added to Pandas dataframe
- Data was cleaned for missing values and filtered
- Finally, data sets was saved to CSV files

Data Collection – SpaceX API

Requested data via Space X API

Results returned as JSON file

Data from JSON file loaded to Pandas dataframe

Data was filtered, cleaned, modified to include just Falcon 9 launches

Finally, data was stored to CSV file

GitHub URL

https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/2492c865bc1f8afee35714dc638870ea8d94b155/Week%201/API/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Data was collected from Wikipadia and loaded to BeautifulSoup object

Extracted relevant data and stored to dictionary (tables, headers columns and rows)

Dictionary loaded to Pandas dataframe

Finally, data stored to CSV file

GitHub URL

https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/2492c865bc1f8afee35714dc638870ea8d94b155/Week%201/Webscraping/jupyter-labs-webscraping.ipynb

Data Wrangling

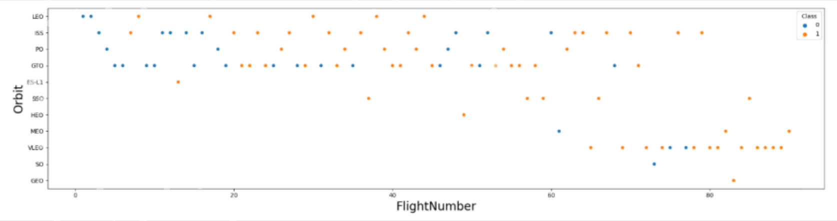
- Calculated the number of launches on each site
- Calculated the number and occurrence of each orbit
- Calculated the number and occurrence of mission outcome per orbit type
- Created a landing outcome label from Outcome column
- Data stored to CSV file

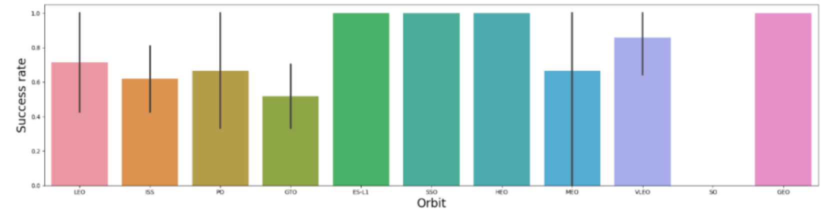
GitHub URL https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/2492c865bc1f8afee35714dc638870ea8d94b155/Week%201/Wragling/data_wrangling.ipynb

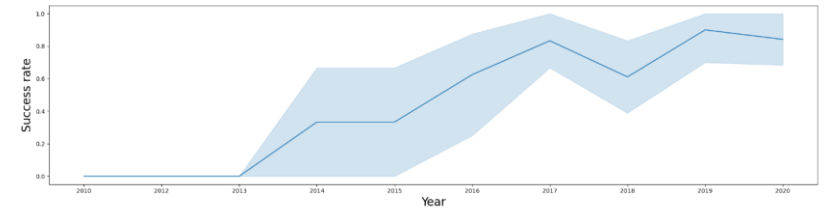
EDA with Data Visualization

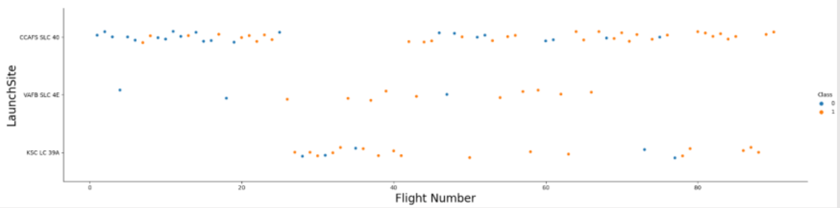
- Plotted Catpot or Scater point charts on relationships between different parameters to compare success or failed outcomes
- Plotted Bar chart to compare success rate for each Orbit
- Plotted Line chart to compare progression of success rate by different years

GitHub URL https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/2492c865bc1f8afee35714dc638870ea8d94b155/Week%202/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb









EDA with SQL

- **Displayed the names of the unique launch sites in the space mission**
- **Displayed 5 records where launch sites begin with the string 'CCA’**
- **Displayed the total payload mass carried by boosters launched by NASA (CRS)**
- **Displayed average payload mass carried by booster version F9 v1.1**
- **Found first successful landing outcome in ground pad was achieved**
- **Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000**
- **Listed the total number of successful and failure mission outcomes**
- **Listed the names of the booster versions which have carried the maximum payload mass by using a subquery**
- **Listed the records which displayed the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015**
- **Ranked the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.**

GitHub URL https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/2492c865bc1f8afee35714dc638870ea8d94b155/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Markers to indicated location for launch sites on the map
- Circle markers to highlight the launch site on the map
- Icon markers to create icons on the map to represent outcomes
- Marker clusters to group map objects on the map with possibility to split them by zooming in
- Lines to show distance between launch site and other objects

GitHub URL

https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/c1c1d68edc2221a1327b28f50c72d0418f508e62/Week%203/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Added interactive Pie chart to compare success rates between different Launch site or Success / Failed outcomes rate between launch site
- Added interactive Shatter point chart to compare outcomes in different payload mass ranges

GitHub URL https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/c1c1d68edc2221a1327b28f50c72d0418f508e62/Week%203/spacex_dash_app.py

Predictive Analysis (Classification)

- Data was loaded to Pandas dataframe, transformed and split to train and test data
- Were built different ML models (Logistic Regresion, SVM, Decision Tree and KNN)
- Models was tuned by hyperparameters using GridSearchCV
- Evaluated models by checking accuracy of each models and plotted confusion matrixes for each
- Results summarized by plotted bar chart and found bet performed model

Results

Exploratory data analysis results

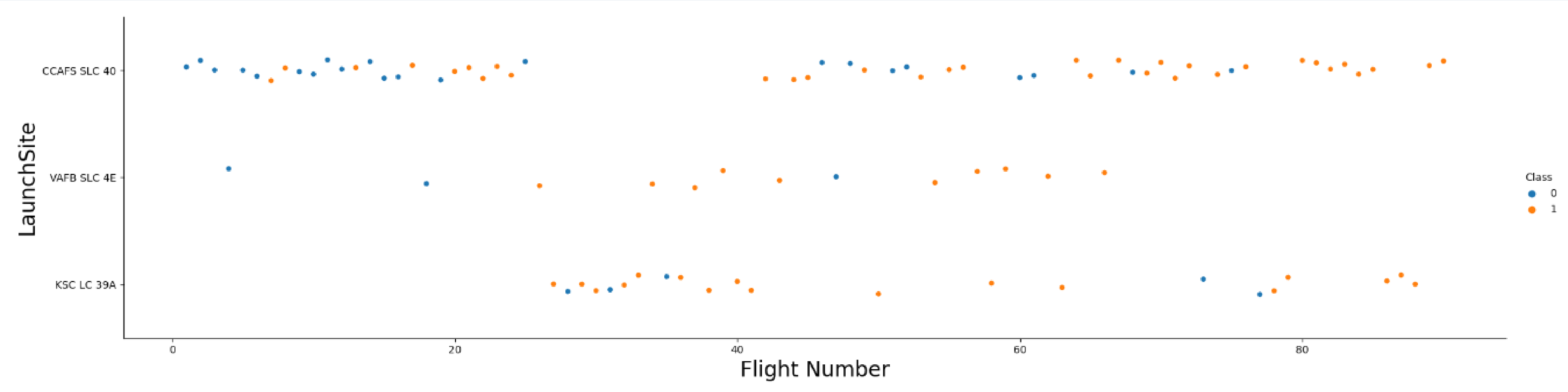
Interactive analytics demo in screenshots

Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



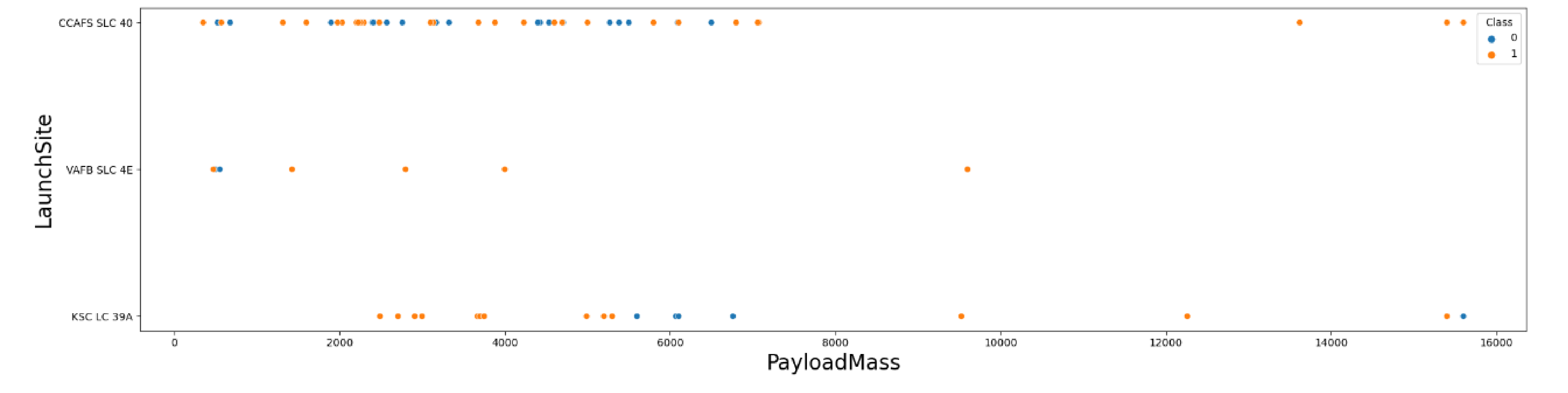
Git Hub [URL: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb](https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb)

As you can see in screenshot above site CCADS SLC 40 has the highest number of flights and the lowest.

The most successful site is KSC LC 39A and VARFB SLC 4E with rounded 77% success rate. The lowest success rate is for CCADS SLC 40 about 60%.

Also, you can see that success rate in all launch sites increasing over time. At the fist 20 launches was more failed landings then in the last 20 launches.

Payload vs. Launch Site



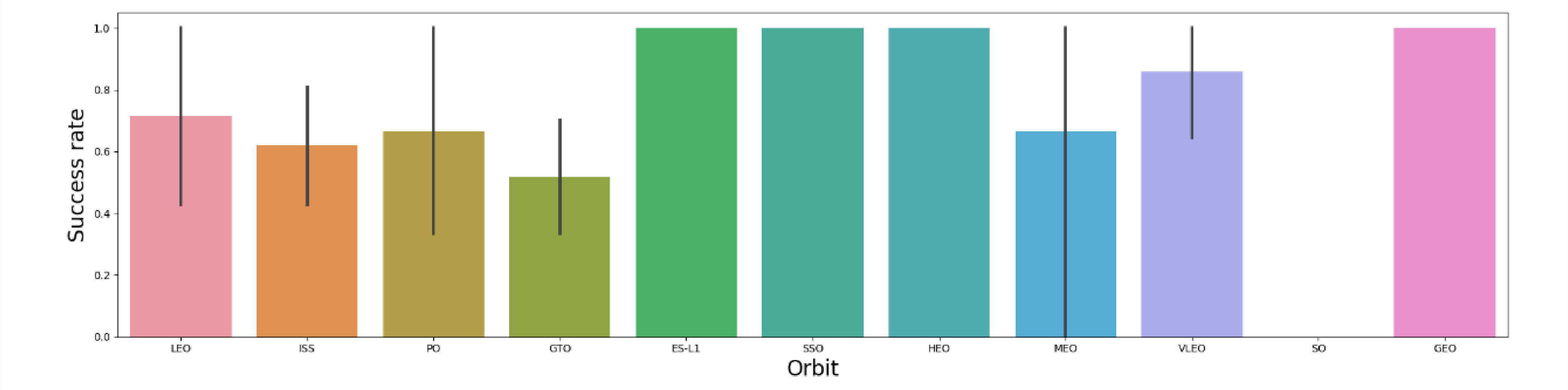
Git Hub [URL: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb](https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb)

You can see that the higher payload mass then 7000 kg have the higher success rate

Also, VARB SLC 4E has 100% success rate with higher mass then 1000 kg

Moreover, KSC LC 39A has high success just in Payload mass of ~6000 kg has most of failed landings.

Success Rate vs. Orbit Type

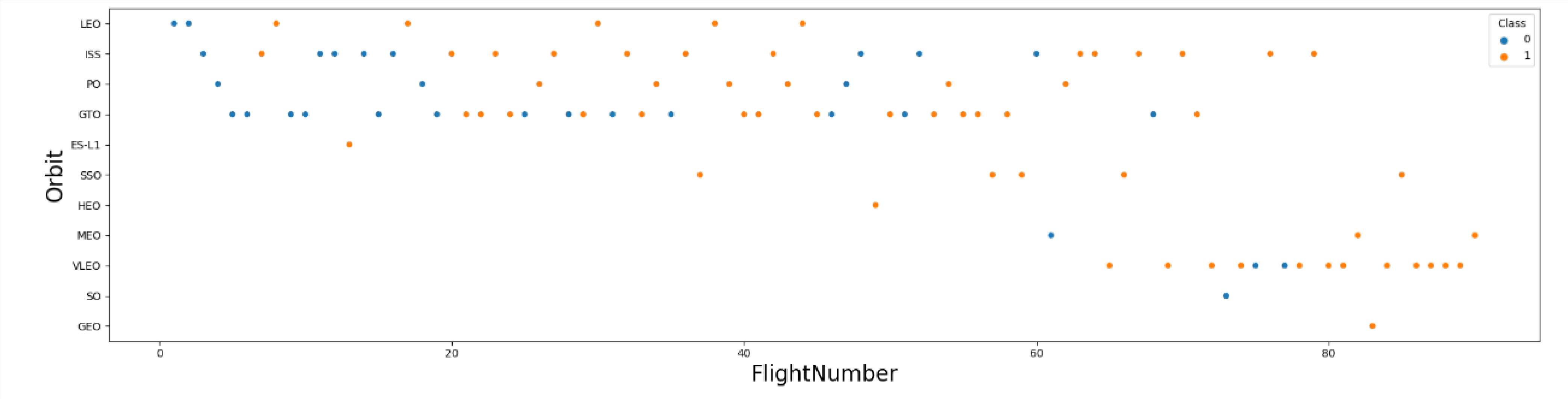


Git Hub URL: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

The highest success rate (100%) is for ES-L1, SSO, HEO and GEO orbits

Orbit SO has 0% success rate

Flight Number vs. Orbit Type

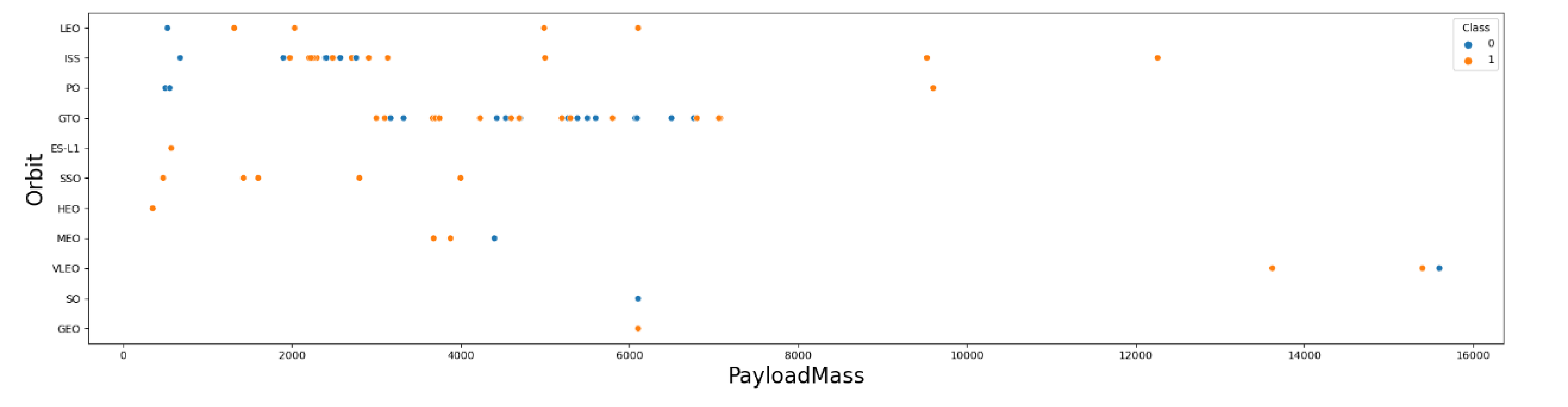


Git Hub URL: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

Now you can see more insights from the data. While ES-L1, SSO, HEO and GEO orbits have 100% success rate they have only few flights launched each. Also, most of the flights launched in higher launch number. As you remember from previous slides success rate increased by higher flight number. Moreover, SSO has only 1 launched flight.

On the other hand, success rate increased for in total for all sites over time.

Payload vs. Orbit Type

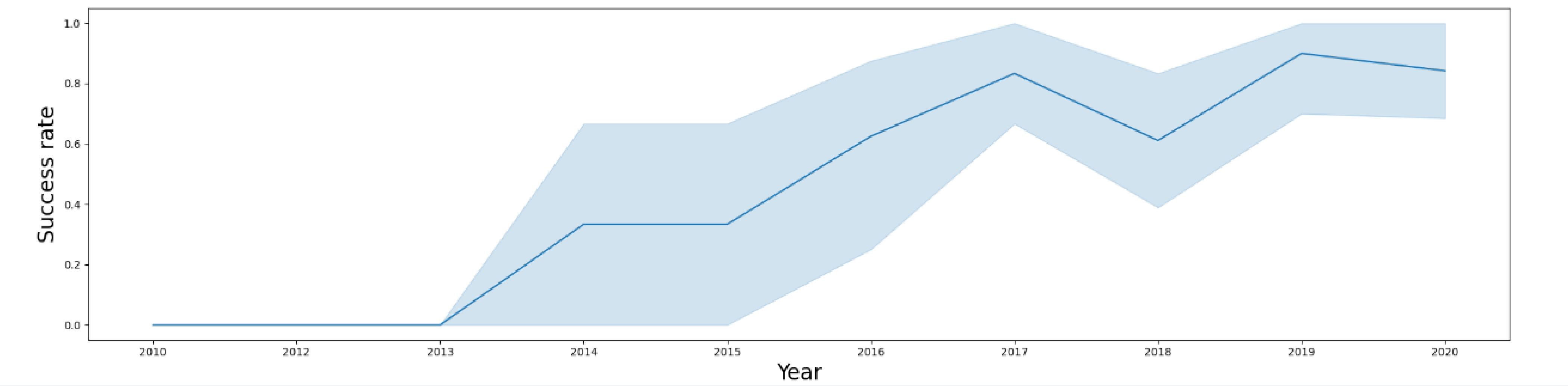


Git Hub [URL: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb](https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb)

As we remember from previous slide the higher payload mass has positive effect for success rate. Just 1 failed flight in VLEO orbit. Could be just outlier.

Also, the majority payload ~6000 kg mass is related to GTO orbit. As you remember from previous slide that has low success rate.

Launch Success Yearly Trend



Git Hub URL: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

The success rate is increasing overtime. Most likely because the technologies get more advanced.

From 2010 to 2013 was 0 successful landings.

All Launch Site Names

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

Git Hub: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Selected Launch site column and grouped by Launch_site. This represent unique site names

Launch Site Names Begin with 'CCA'

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Git Hub: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Were selected all columns from the table where in launch site name begins with latter combination 'CCA'

Total Payload Mass

Total_payload_mass
45596

Git Hub: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Calculated total payload mass where customer is NASA (CRS)

Average Payload Mass by F9 v1.1

Average_payload_mass
2928.4

Git Hub: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Calculated average payload mass for booster version F9 v1.1

First Successful Ground Landing Date

First_succesful_landing
22-12-2015

Git Hub: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Selected first date from the Landing_outcome column where outcome was Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

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

Git Hub: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Selected all booster versions where mass is between 4000 kg and 6000 kg. Also, second filter applied where Landing outcome is Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

Mission_Outcome	Total
Failure (in flight)	1
Success	100

Git Hub: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Calculated total mission outcome where status of mission outcome is successful or failure. The outcome grouped where letters combination are like %Success% or the rest left as actual (failed)

Boosters Carried Maximum Payload

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

Git Hub: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Selected all boosters from booster_version column where Payload mass is the biggest. Used secondary select statement in where filter

2015 Launch Records

month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Git Hub: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

- Selected 2015 all boosters' records where landing outcome “Failure (drone ship)”

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Landing _Outcome	total
Success	20
Success (drone ship)	8
Success (ground pad)	6

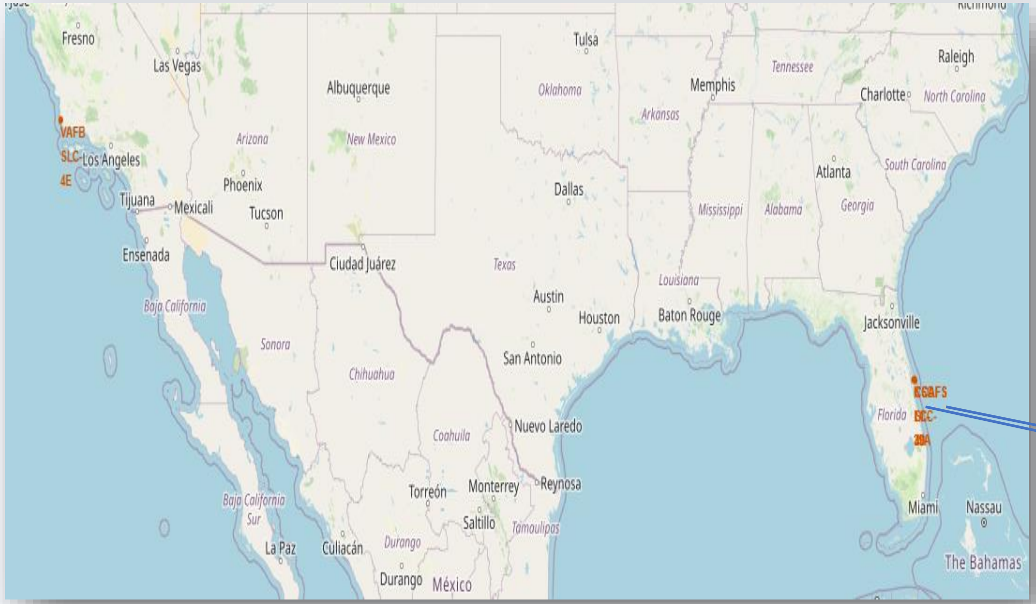
Git Hub: https://github.com/Oziris5/IBM-Data-Science-Capstone-Project/blob/main/Week%202/jupyter-labs-eda-sql-coursera_sqllite.ipynb

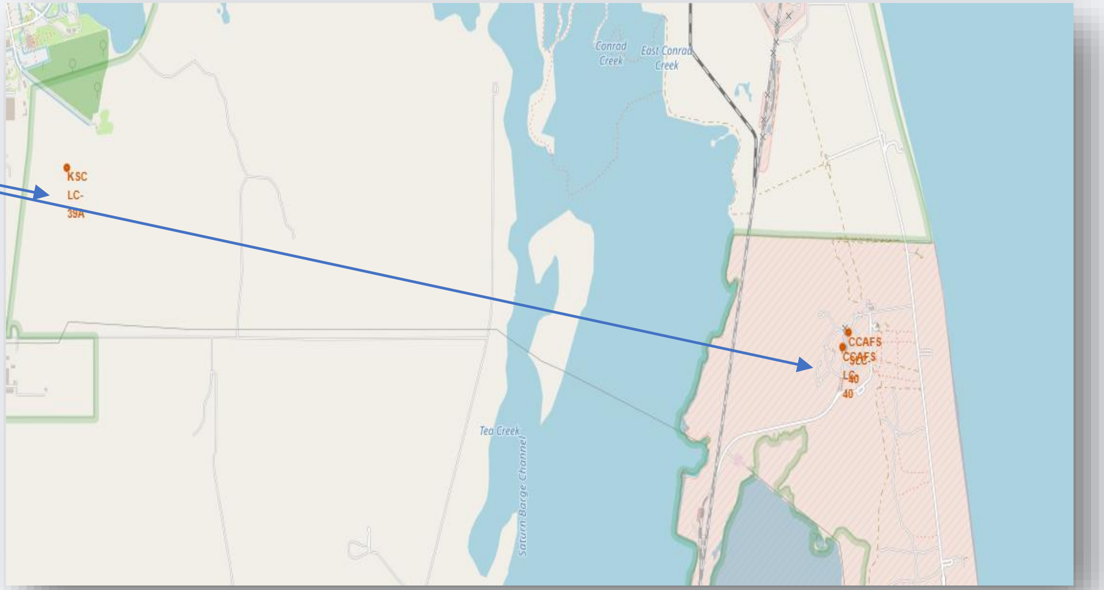
- Selected count of all types of successful landing outcomes between 2010-06-04 and 2017-03-20. Then result sorted by descending order.

Section 3

Launch Sites Proximities Analysis

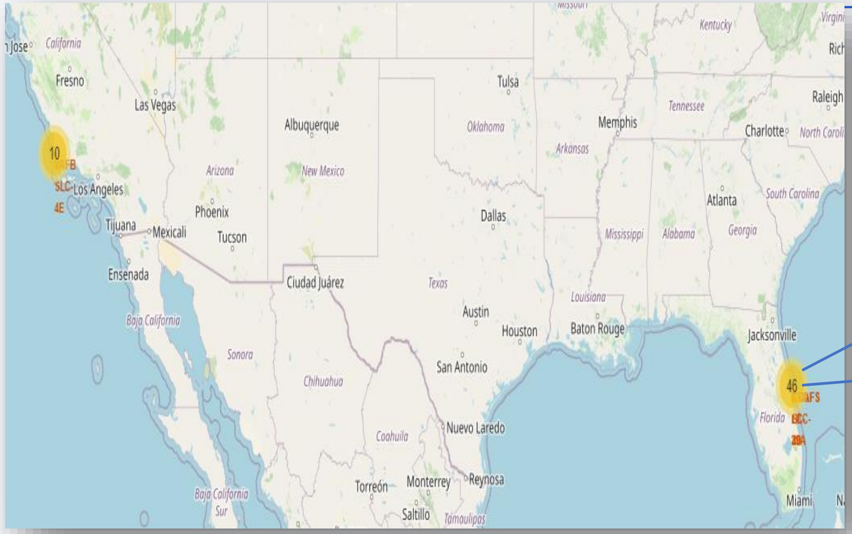
All Launch Sites

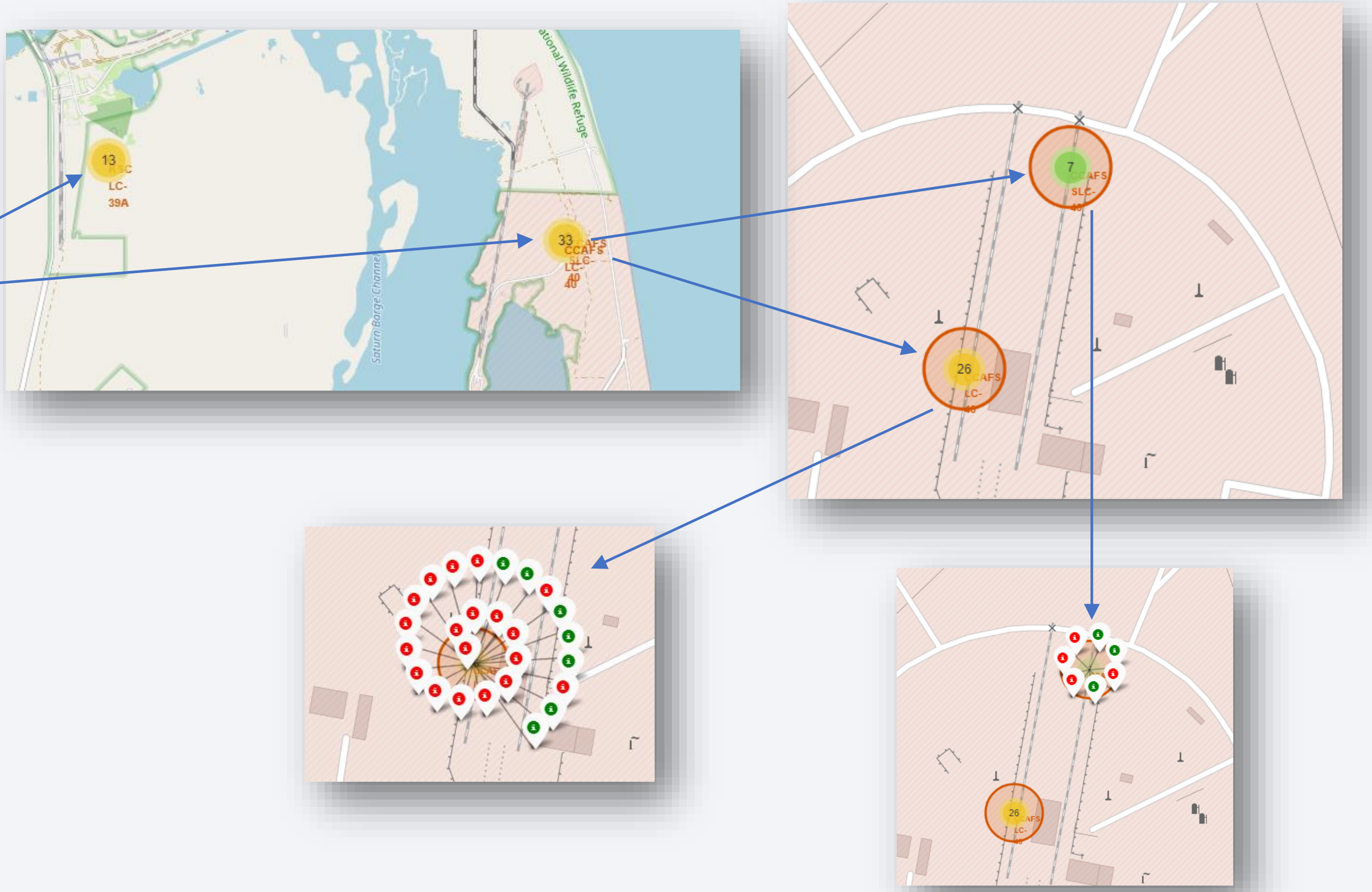




All Space X launch sites are in USA and all near cost and other logistic lines

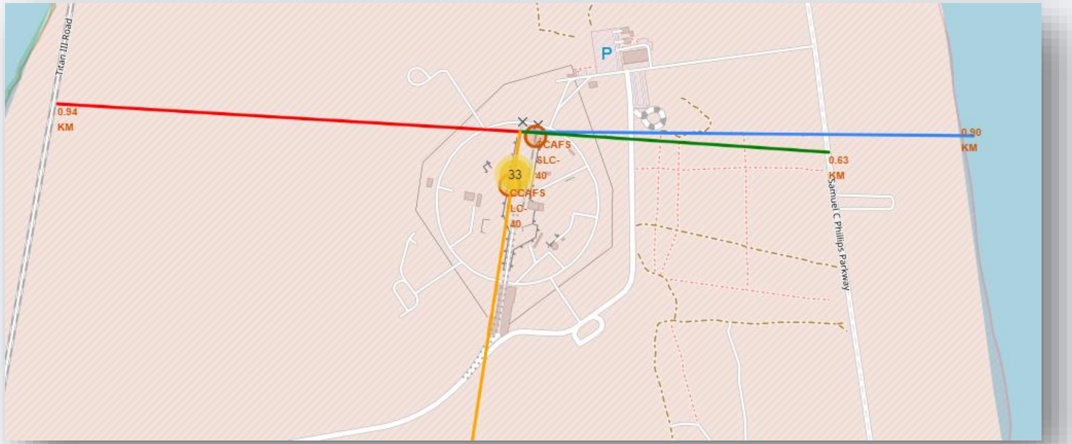
Launch results by site

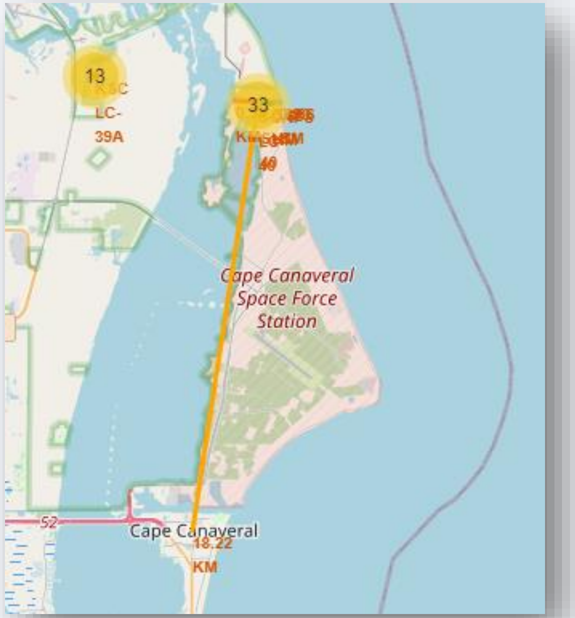




- Green marker indicates successful launches and red indicates failures
- Zooming in and out summaries or split sites launches if they are close enough

Logistics and the nearest objects





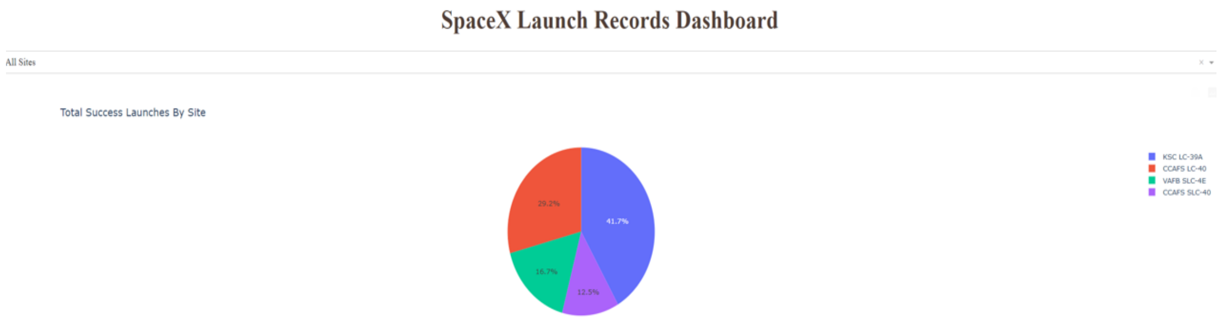
- Was calculated distances from different objects
- Red line indicates distances between launch site and the nearest railway
- Blue line indicates distances between launch site and the nearest cost
- Green line indicates distances between launch site and the nearest highway
- Orange line indicates distances between launch site and the nearest city

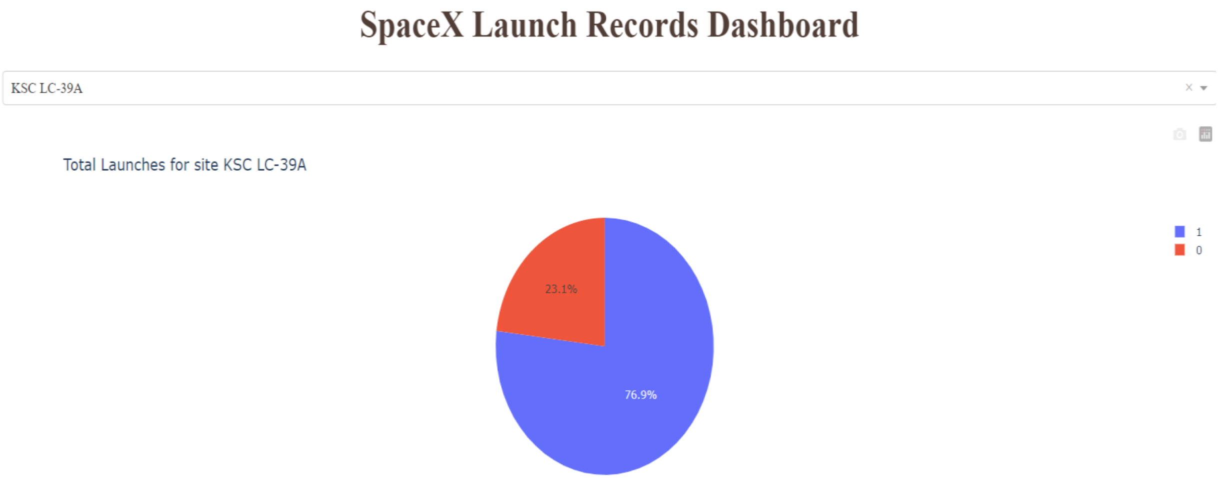
Section 4

Build a Dashboard with Plotly Dash

Successful Launches by Site

- We can see that KSC LC-39A has highest success rate and VAFB SLC-4E has the lowest comparing with other launch sites.



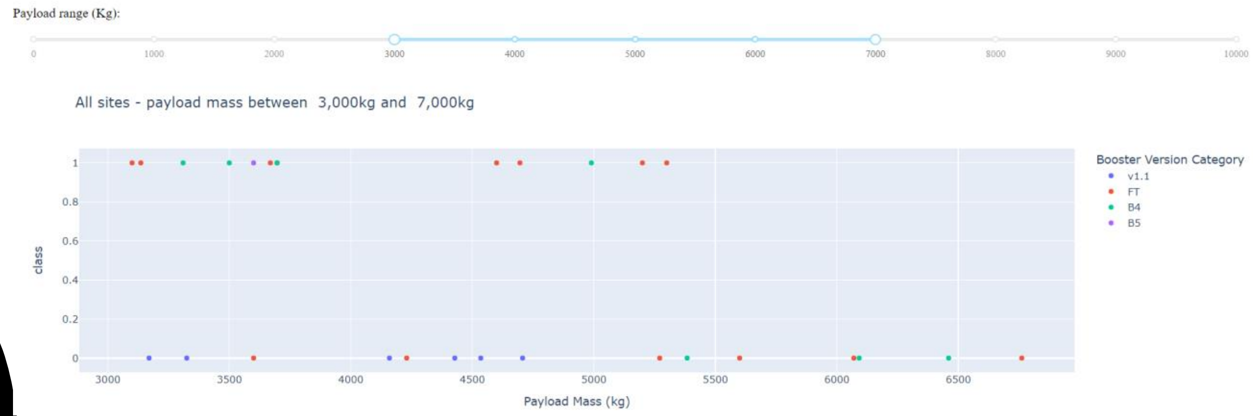


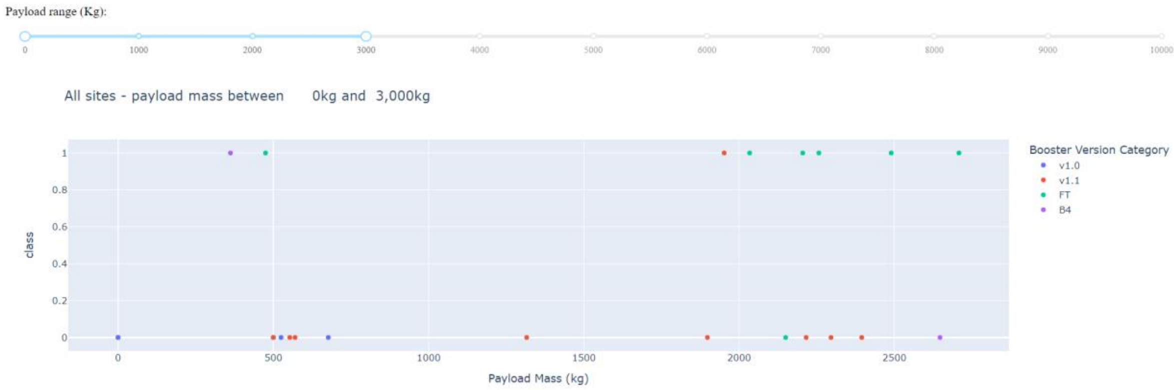
Site with Highest Success Rate

- KSC LC-39A site with the highest success rate has 76.9% successful launches and 23,1% failed.

Most Successful Booster Version

- In payload mass between 0 kg and 3 000 kg FT booster version has highest success rate . Also, in payload mass between 3 000 kg and 7 000 kg FT launches was most successful too.



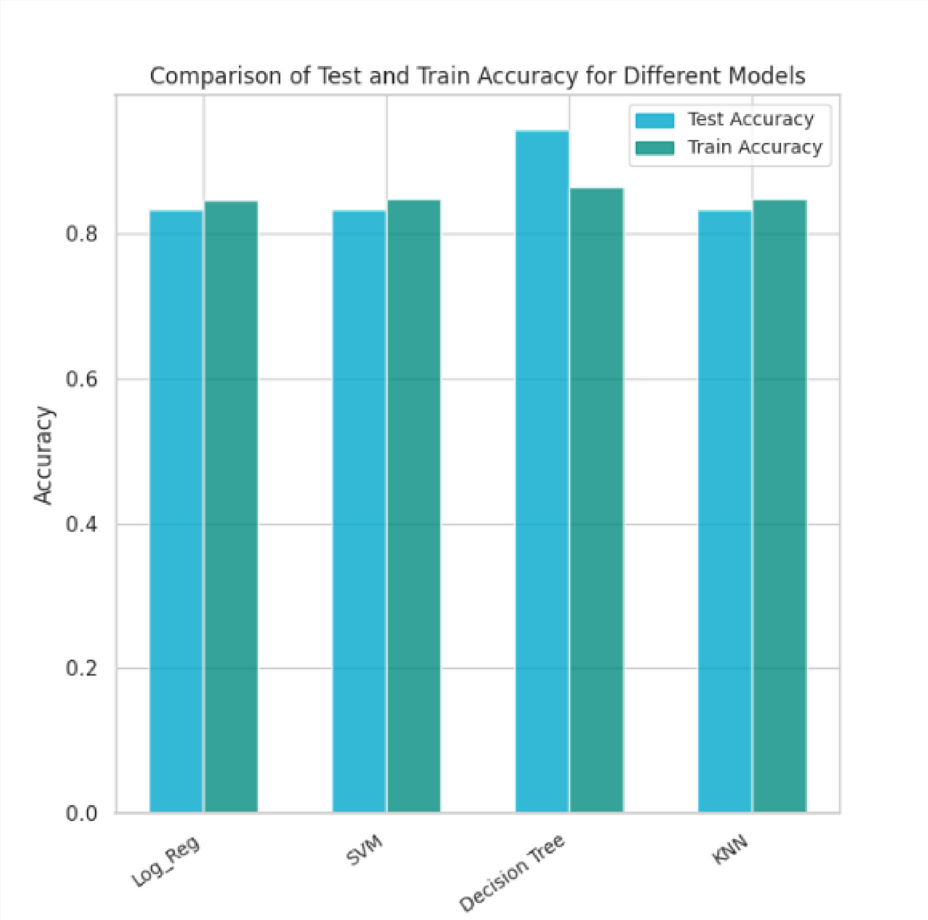


Section 5

Predictive Analysis (Classification)

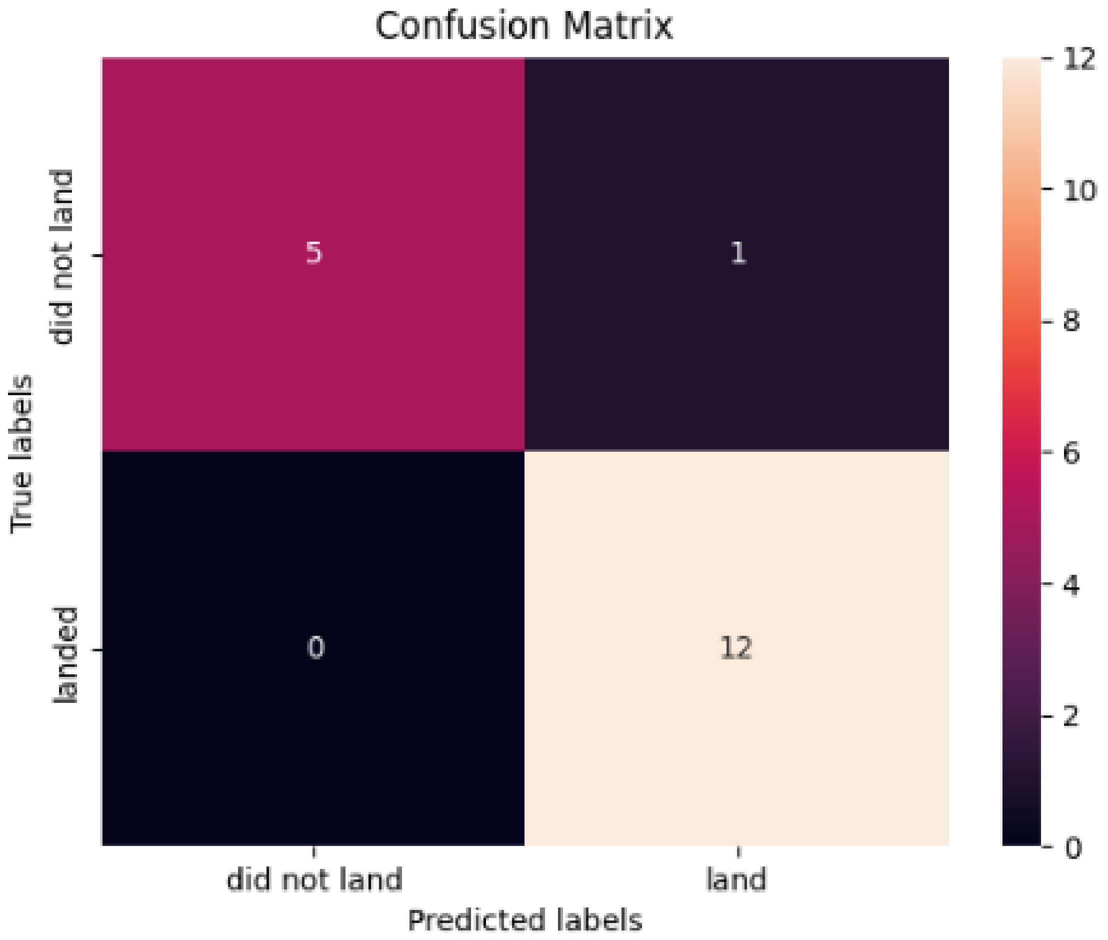
Classification Accuracy

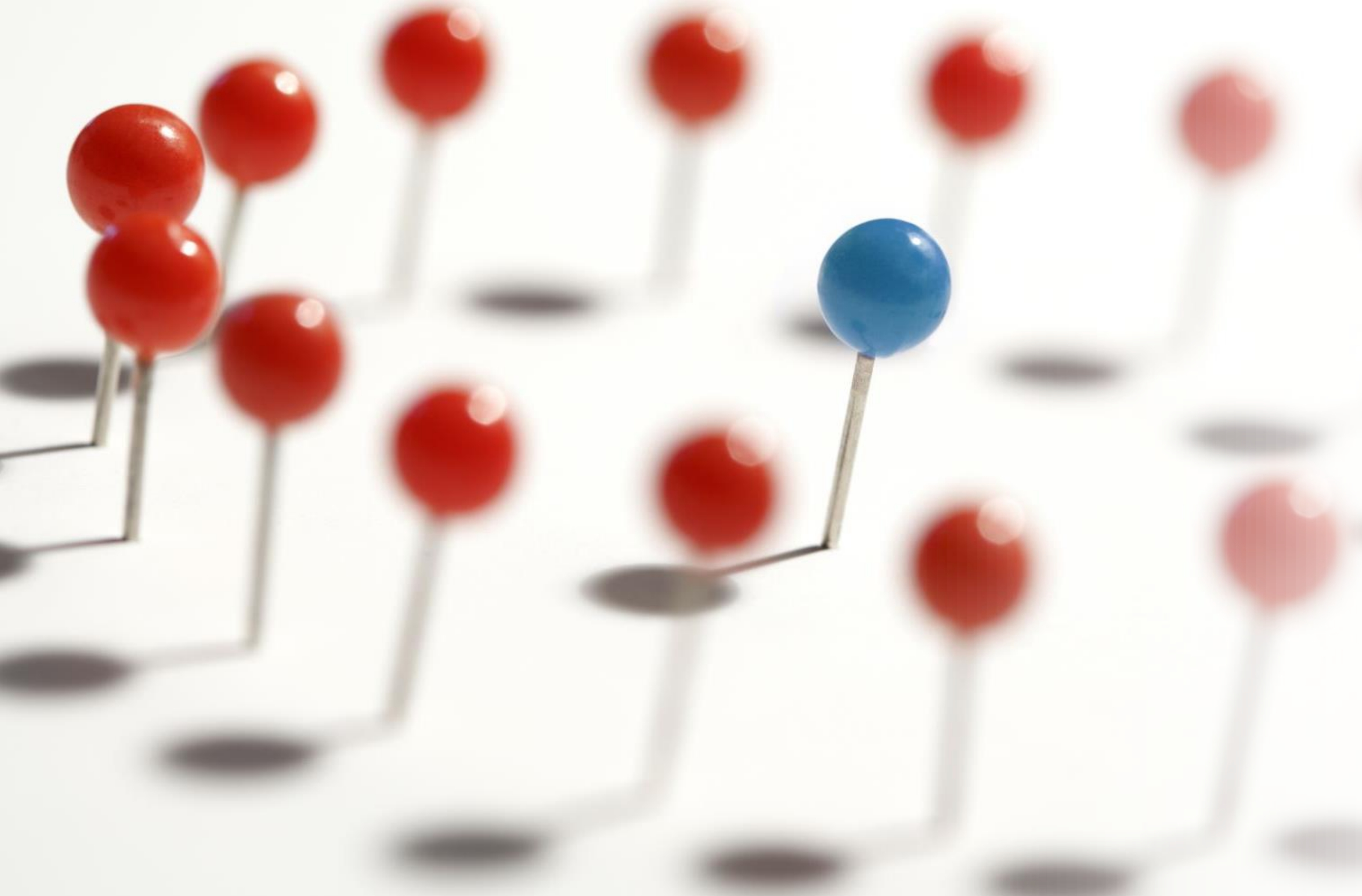
- The decision tree performed the best comparing to other models on test and train accuracy



Confusion Matrix for Decision Tree

- Even decision tree true positive was same as other models comparing confusing matrix. The True negative was higher by 2 comparing to other models but False positive was lower by 2.





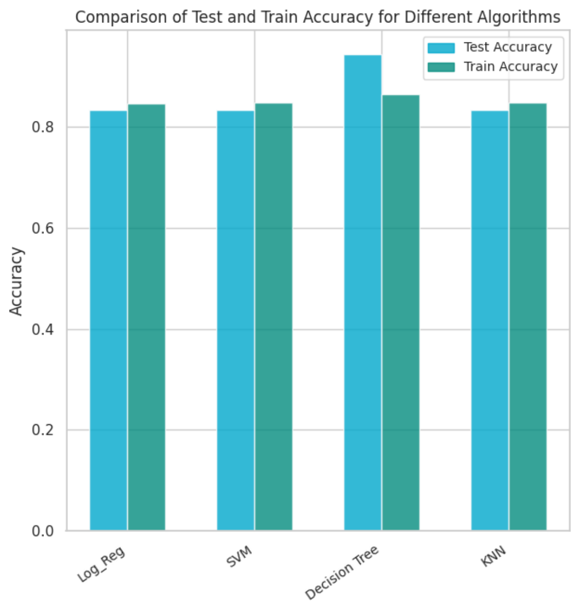
Conclusions

- KSC LC-39A is the most successful launch site
- Ft booster version is the most successful.
- Success rate increased in total for all sites over time
- ML Decision Tree model is best performing model and should used for predicting landing outcome

Appendix

```python
performance = pd.DataFrame({'Algorithm' : ['LogisticRegression', 'SVM', 'Decision Tree','KNN']})

lr_score_test = logreg_cv.score(X_test, Y_test)
svm_score_test = svm_cv.score(X_test, Y_test)
tree_score_test = tree_cv.score(X_test, Y_test)
knn_score_test = knn_cv.score(X_test, Y_test)

lr_score_train = logreg_cv.best_score_
svm_score_train = svm_cv.best_score_
tree_score_train = tree_cv.best_score_
knn_score_train = knn_cv.best_score_

performance['Test Accuracy'] = [lr_score_train, svm_score_train, tree_score_train, knn_score_train]
performance['Train Accuracy'] = [lr_score_test, svm_score_test, tree_score_test, knn_score_test]

print(performance)

data = {'Log_Reg': {'Test Accuracy': lr_score_test, 'Train Accuracy': lr_score_train},
        'SVM': {'Test Accuracy': svm_score_test, 'Train Accuracy': svm_score_train},
        'Decision Tree': {'Test Accuracy': tree_score_test, 'Train Accuracy': tree_score_train},
        'KNN': {'Test Accuracy': knn_score_test, 'Train Accuracy': knn_score_train}}


x = list(data.keys())
test_acc = [d['Test Accuracy'] for d in data.values()]
train_acc = [d['Train Accuracy'] for d in data.values()]
bar_width = 0.3
colors = ['#00a8cc', '#038c7f']
fig, ax = plt.subplots(figsize=(7, 7))
ax.bar(x, test_acc, width=bar_width, color=colors[0], alpha=0.8)
ax.bar([i + bar_width for i in range(len(x))], train_acc, width=bar_width, color=colors[1], alpha=0.8)

# Add labels and titles
ax.set_xlabel('Algorithm', fontsize=12)
ax.set_ylabel('Accuracy', fontsize=12)
ax.set_title('Comparison of Test and Train Accuracy for Different Algorithms', fontsize=12)

# Add custom legend
legend_elements = [plt.Rectangle((0, 0), 1, 1, color=colors[i], alpha=0.8) for i in range(2)]
ax.legend(legend_elements, ['Test Accuracy', 'Train Accuracy'], fontsize=10)

# Adjust the position of the x-axis labels
ax.set_xticks([i + bar_width / 2  for i in range(len(x))])
ax.set_xticklabels(x, fontsize=10, rotation=35, ha='right')

# Show the plot
plt.show()
```

	Algorithm	Test Accuracy	Train Accuracy
0	LogisticRegression	0.846429	0.833333
1	SVM	0.848214	0.833333
2	Decision Tree	0.864286	0.944444
3	KNN	0.848214	0.833333



Thank you!