IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Name>
<Date>

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 a map with folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

Summary of all results

- EDA Results
- Interactive Analysis
- Predictive analysis

Introduction

- Project background and context
- SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars, other providers cost of 162 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- Problems you want to find answers
- The project task is to predicting if the first stage of the SpaceX Falcon rocket will land successfully

Section 1

Methodology

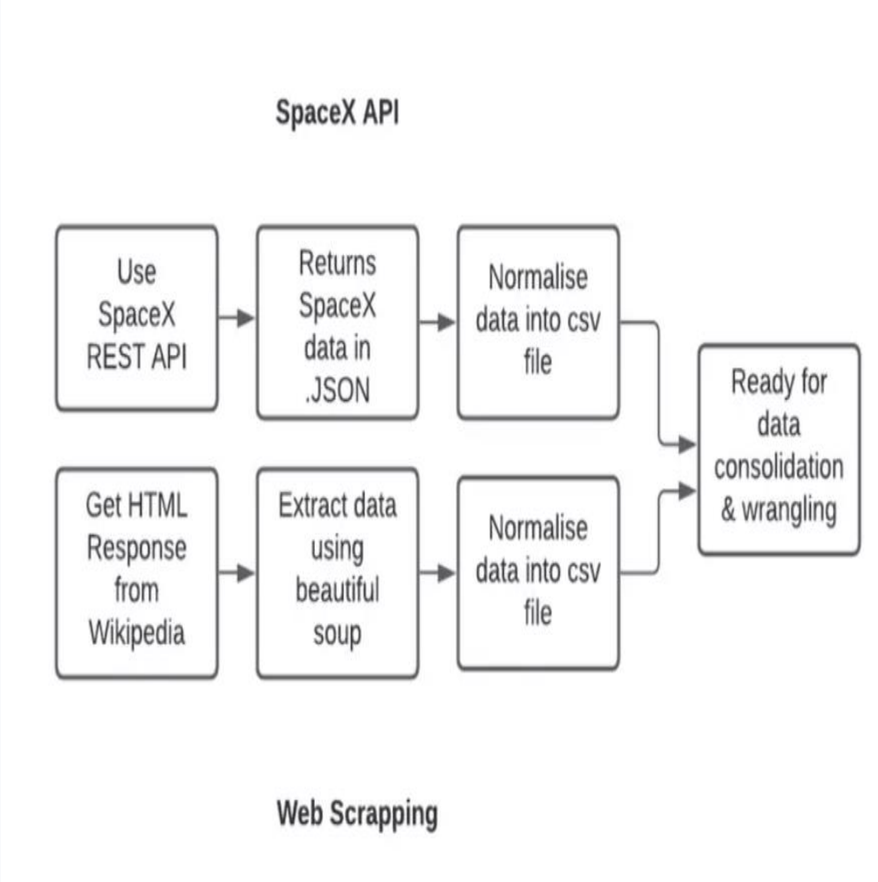
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One hot Encoding for Machine Learning and data Cleaning of Null values and irrelevant columns.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - K Nearest Neighbors , Logistic Regression Support Vector Machine, Decision Tree models have been built and evaluated for the best classifier

Data Collection

- Describe how data sets were collected.
  - From SpaceX REST API
  - Api.spacexdata.com/v4/
  - Web Scrapping From Wikipedia using BeautifulSoup



Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

```
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}
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class=[]
for i in range(0,len(df['Outcome'])):
    if df['Outcome'][i] in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

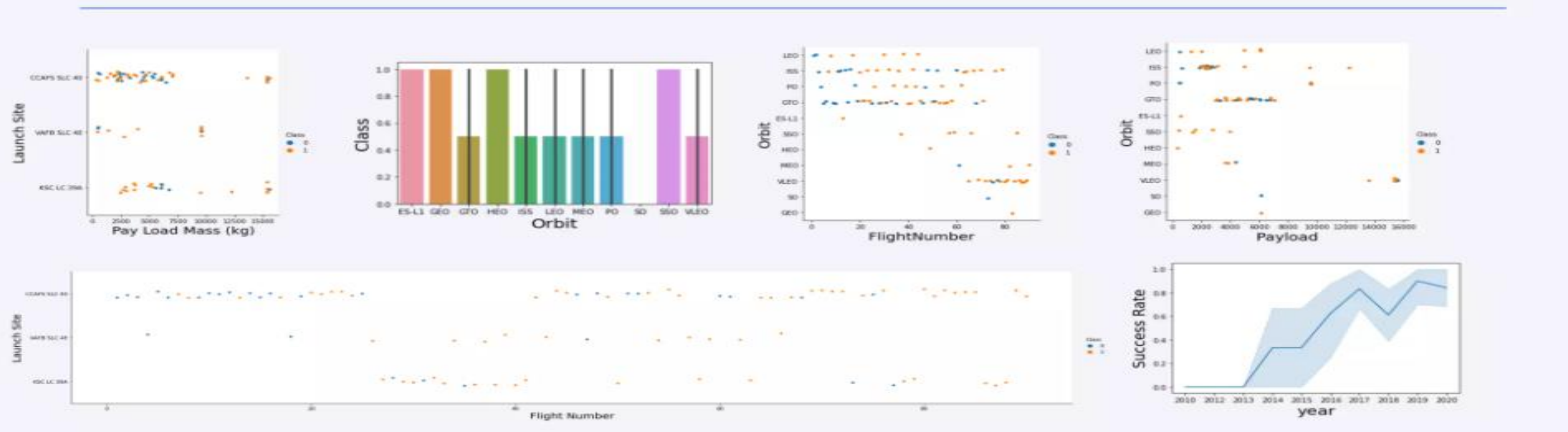
```
landing_outcomes=df['Outcome'].value_counts()
```

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 None ASDS
6 False Ocean
7 False RTLS
```

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

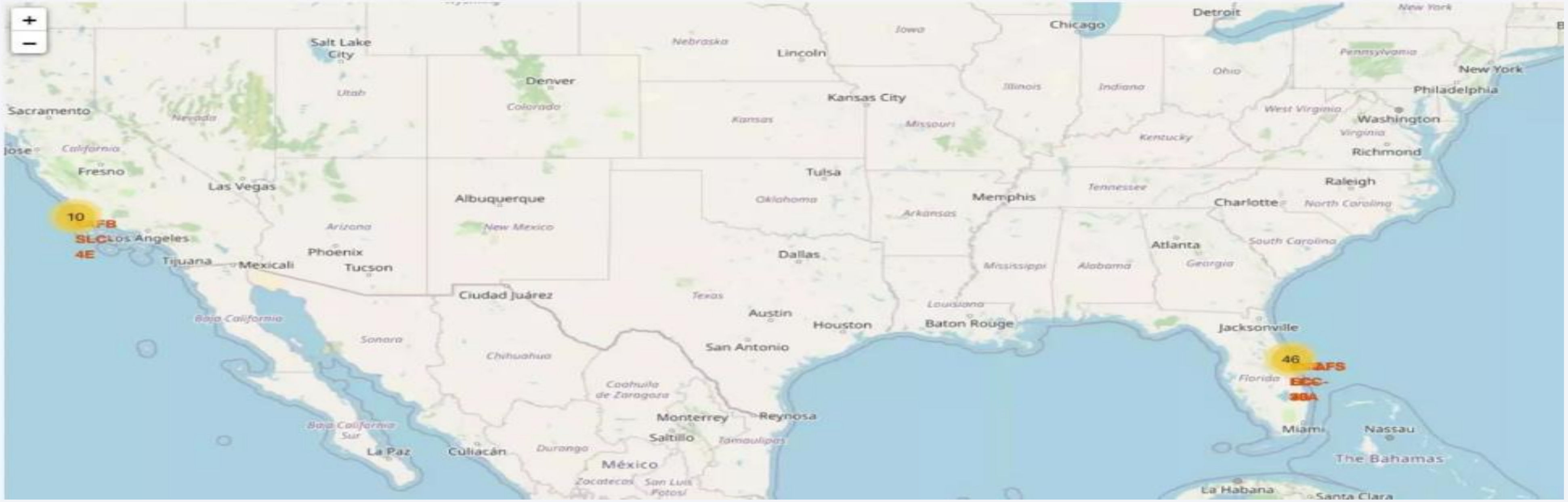


EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- 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
- •List the failed landing_outcomesin drone ship, their booster versions, and launch site names for in year 2015
- •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

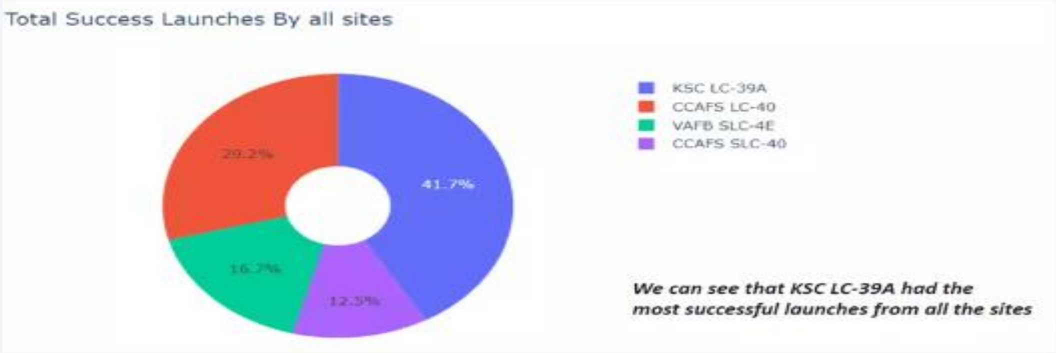
Build an Interactive Map with Folium

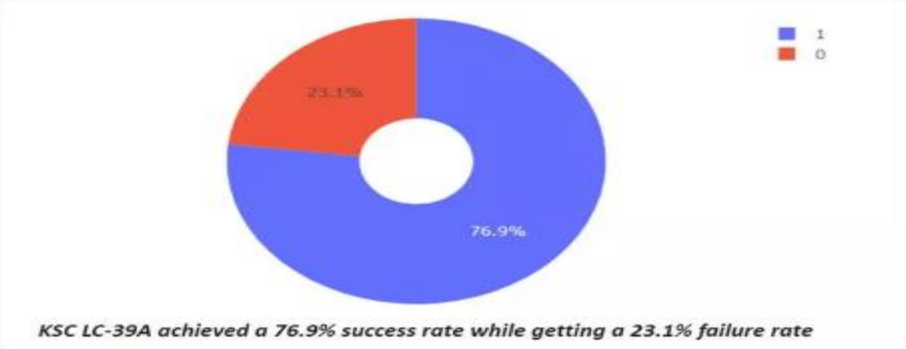
- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects

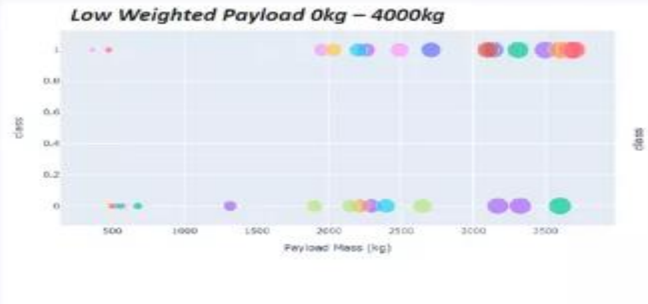


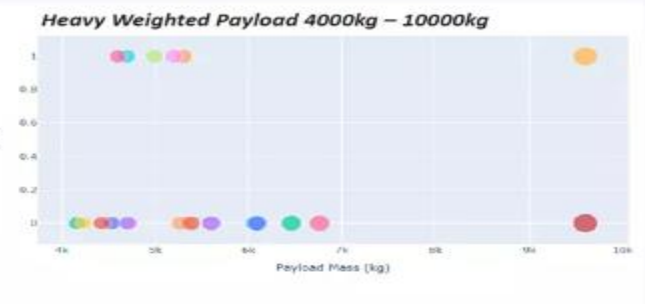
Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard









Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification modelBuild
- Building Model:
- •Transform data to Scale the columns
- •Split Data into Testing and Training Sets
- •Selected machine learning algorithms to use for classification (KNN, Decision Tree, SVM,Logistic Regression)
- •Use Grid Search and Cross Validation to find best tuning parameters for each model fittingon training sets

- Evaluate
- Evaluating Model:
- •Check accuracy of each model on training and testing sets
- •Plot Confusion Matrix

- Improve
- Improving Model:
- •Feature Engineering
- •Algorithm tuning

- Selectingthe best performing classifier:
- •Model with the best accuracy score on test set is the best model. If there is a tie, check accuracy on training set as well

Results

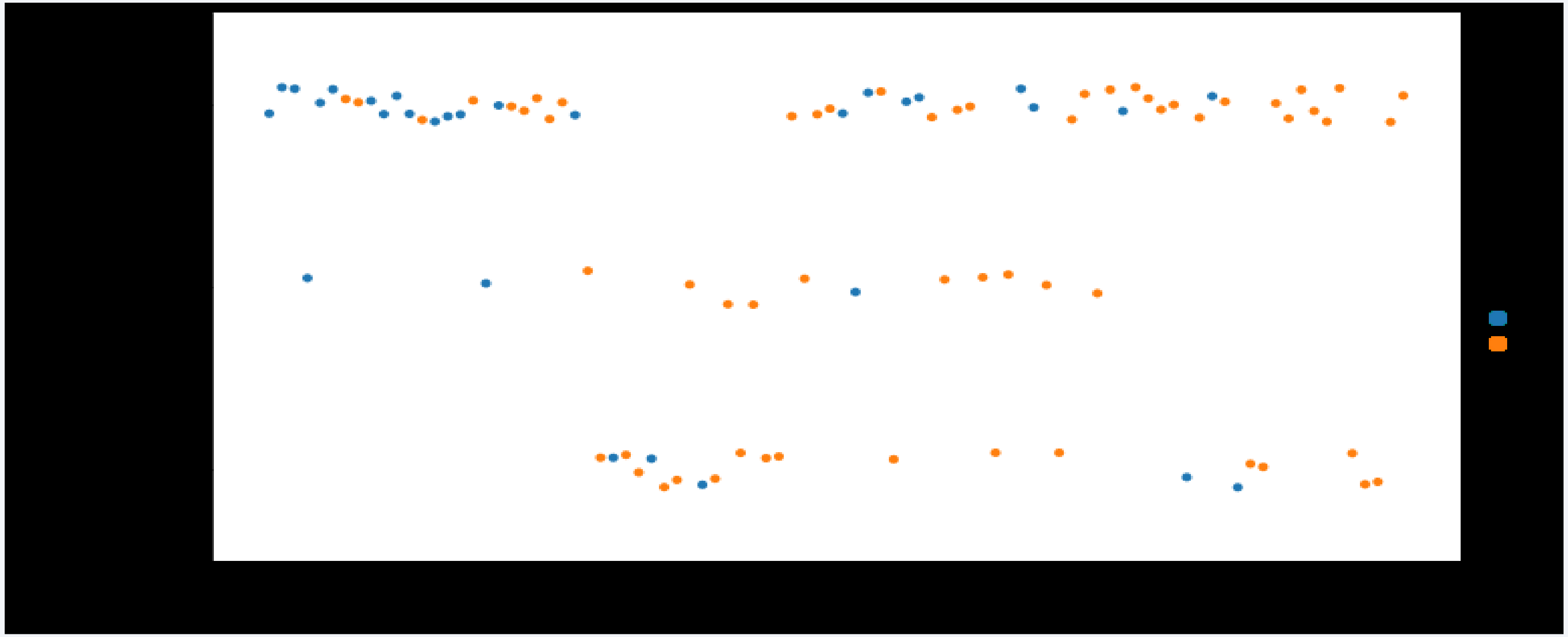
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

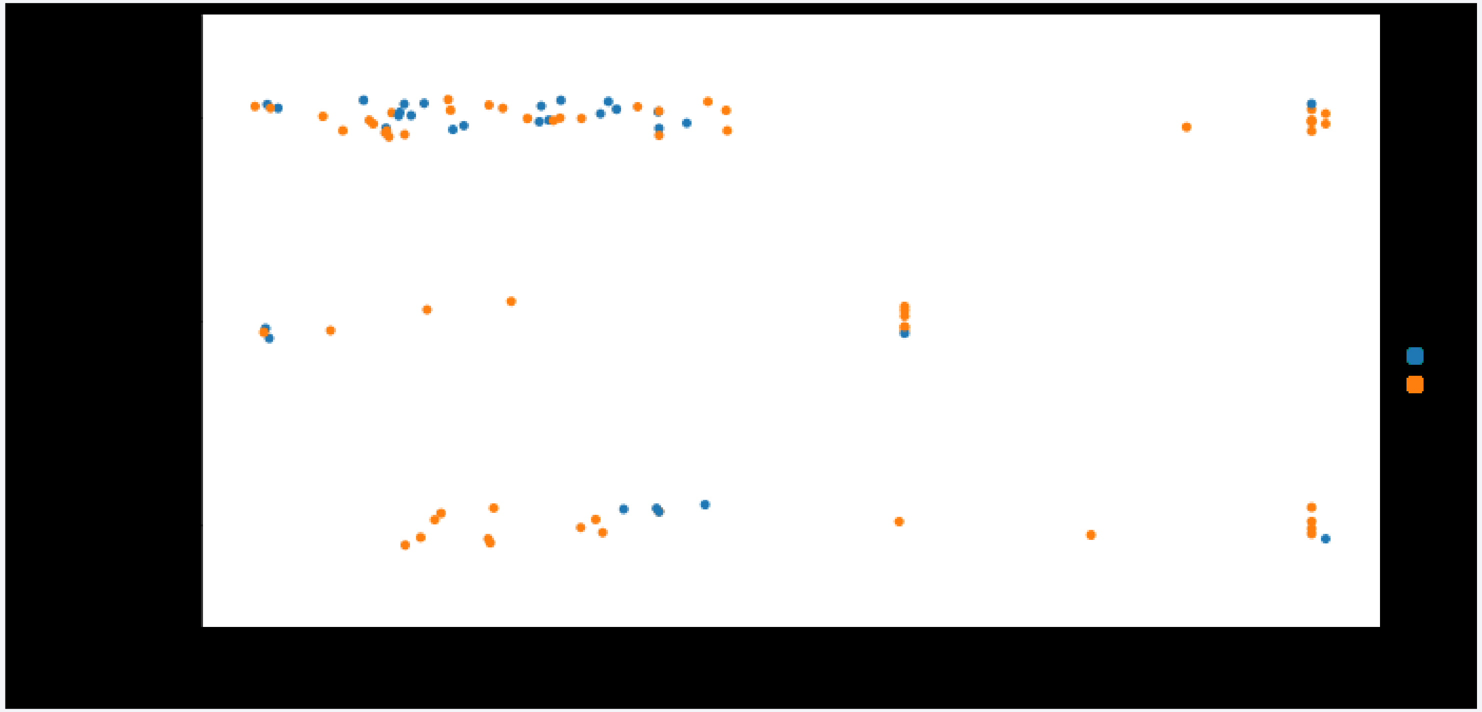
Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site



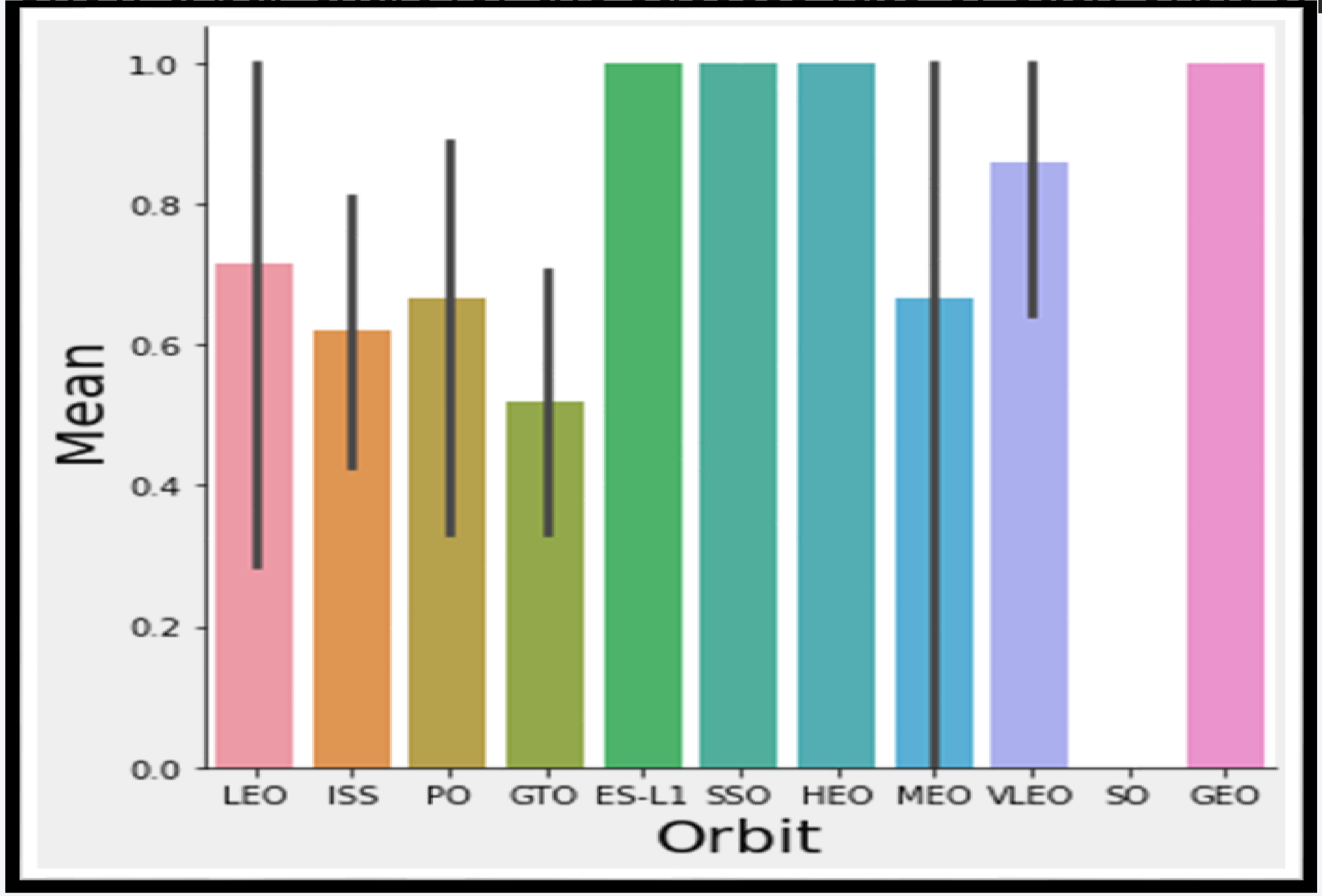
Payload vs. Launch Site

- Show a scatter plot
  of Payload vs. Launch Site



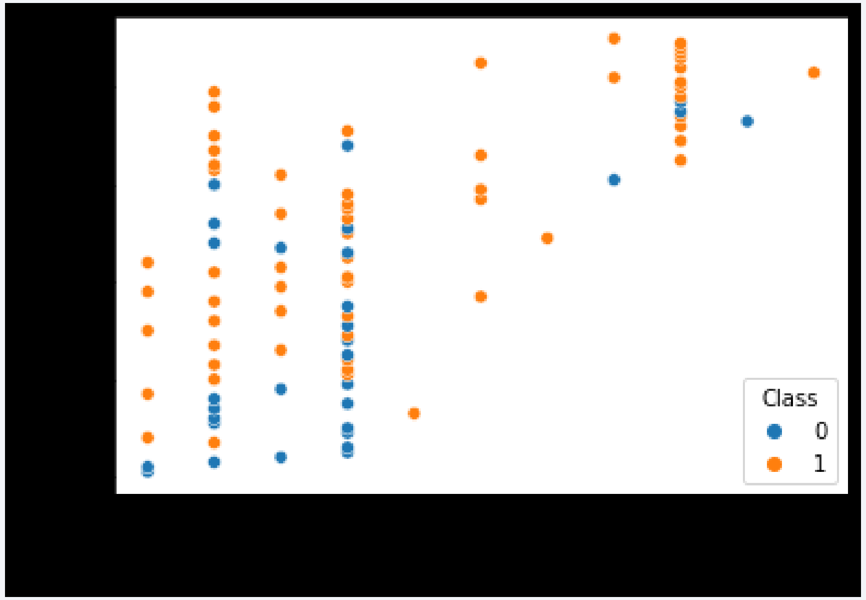
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type



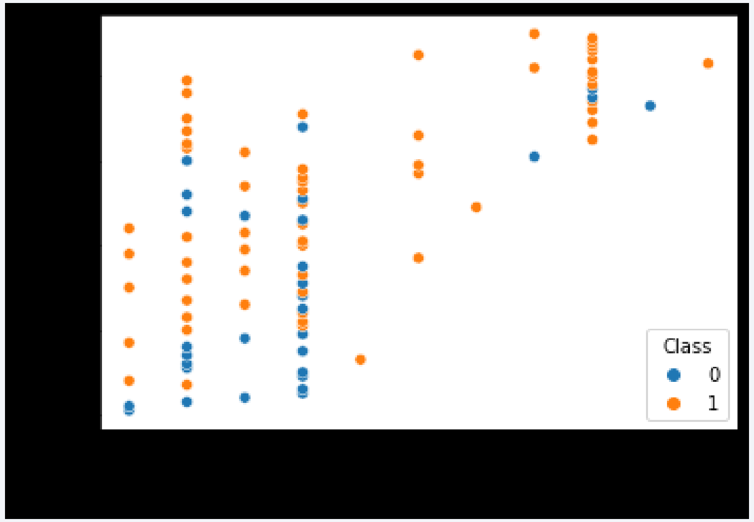
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type



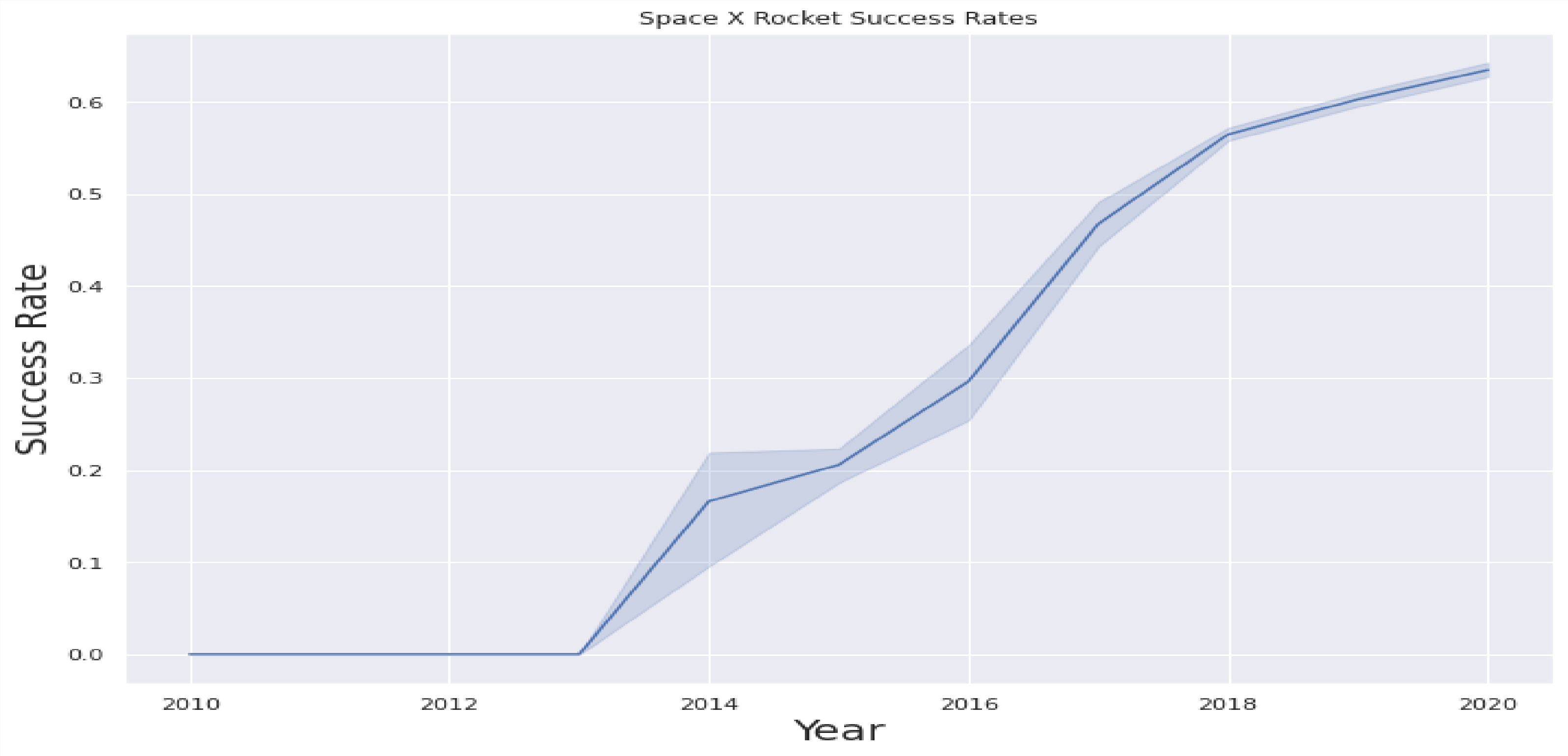
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type



Launch Success Yearly Trend

- Show a line chart of yearly average success rate



All Launch Site Names

- Find the names of the unique launch sites

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

- **Present your query result** with a short explanation here
- SELECT Unique Launch_Site from SpaceX

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

- `%%sqlSelect * from SpaceXwhere Launch_Sitelike 'CCA%'limit 5`

- Present your query result with a short explanation here

- **By selecting * we request all columns of the data frame**
- **The where clause requires results to have a Launch_Sitebeginning with "CCA"**
- **The limit clause restricts the database to only return the first 5 rows in the matching the request**

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

- ```
  %%sqlSelectcustomer, sum(payload_mass_kg) as"Total Payload Mass"
  from(Selectcustomer, payload_mass_kgfromSpaceXwherecustomer LIKE'NASA
  (CRS)')GROUPBYCUSTOMER
  ```

- Present your query result with a short explanation here

- **The sum callcaluclatesthe total**
- **The as clause renames the returned column to Total Payload mass**
- **We select from a subquerythat queries the list of customers and their payload masses but limits the search to rows where the customer is Nasa (CRS)**
- **The group by clause makes sure our sum combines the payloads of all NASA (CRS) rows**
  ```

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

- **%%sqlSelect**BOOSTER_VERSION, **AVG**(payload_mass_kg) **as**"AVERAGE Payload Mass" **from**(**Select**BOOSTER_VERSION, payload_mass_kg**from**SpaceX**where**BOOSTER_VERSION **LIKE**'F9 v1.1')**GROUPBY**BOOSTER_VERSION

- Present your query result with a short explanation here

- **The AVG callcalculates the average payload mass**
- **The as clause renames the returned column to AVERAGE Payload mass**
- **We select from a subquerythat queries the list of boster_versionsand their payload masses but limits the search to rows where the booster_versionis F9 v1.1**
- **The group by clause makes sure our average is calculated**

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

- **Using multi-line sqlmagic we can query the database as follows:**
- **%%sqlSelectMIN**(DATE) **as**"FIRST SUCCESS" **FROM**(**SELECT**DATE **FROM**SPACEX**WHERE**LANDING_OUTCOME

- Present your query result with a short explanation here

- **The MIN call selects the earliest date meeting the criteria**
- **The as clause renames the returned column to FIRST SUCCESS**
- **We select from a subquerythat queries the list of DATES from the SpaceX table where the LANDING_OUTCOME isSuccess (Ground Pad)**

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- **%%sqlSELECT**BOOSTER_VERSION, payload_mass_kg, Landing_Outcome**FROM**SPACEX**where**4000 **<**payload_mass_kg**and**payload_mass_kg**<**6000 **and**Landing_Outcome**=**'Success (drone ship)'

- Present your query result with a short explanation here

- **The where clause restricts results to those with a payload mass between 4000 and 6000 kgs and a Landing_Outcomeof Success (drone ship)**
- **The database returns the Booster Version, Payload mass,**

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

- **%%sqlSELECT**Mission_Outcome, **count**(Mission_Outcome) **as**"Total" **FROM**SPACEX**Groupby**Mission_Outcome

- Present your query result with a short explanation here

- **Select Mission Outcomes and count of each type of Mission outcome as the Total**
- **Group by clause ensures we count the success and failures as separate groups**

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

- `%%sqlSELECTUniqueBooster_version, payload_mass_kgFROMSPACEXwherepayload_mass_kg=(Selectmax(payload_mass_kg) fromSPACEX)`

- Present your query result with a short explanation here

- **Select Unique Booster Versions to ensure we get distinct results**
- **Returned Payload Mass to see what the Max Payload is**
- **Where clause checks that the payload mass equals the max payload which is found via a sub query**

2015 Launch Records

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

- **%%sqlSelect**landing_outcome, booster_version, launch_site, DATE **from**SPACEX**where**landing_outcome='Failure (drone ship)' **andYear**(DATE) =2015

- Present your query result with a short explanation here

- **Select Landing_Outcome, Booster Versions, Launch_Site, and Date as requested**
- **Where clause restricts results to have a landing outcome of Failure (Drone Ship) and a launch date in the year 2015**

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

- **%%sqlselect**landing_outcome, **count**(landing_outcome) **as**"Total" **from**Spacex**where**DATE **between**'2010-06-04' **and**'2017-03-20'**groupby**landing_outcome**orderby**"Total" **desc**

- Present your query result with a short explanation here

- **Select Landing_Outcomeand the count of each landing outcome listed as the Total**
- **Where clause restricts results to have a launch date between 2010-06-04 and 2017-03-20**
- **Group by clause ensures we count landing outcomes separately**
- **Order By desc orders the results in descending order as desired**

Section 3

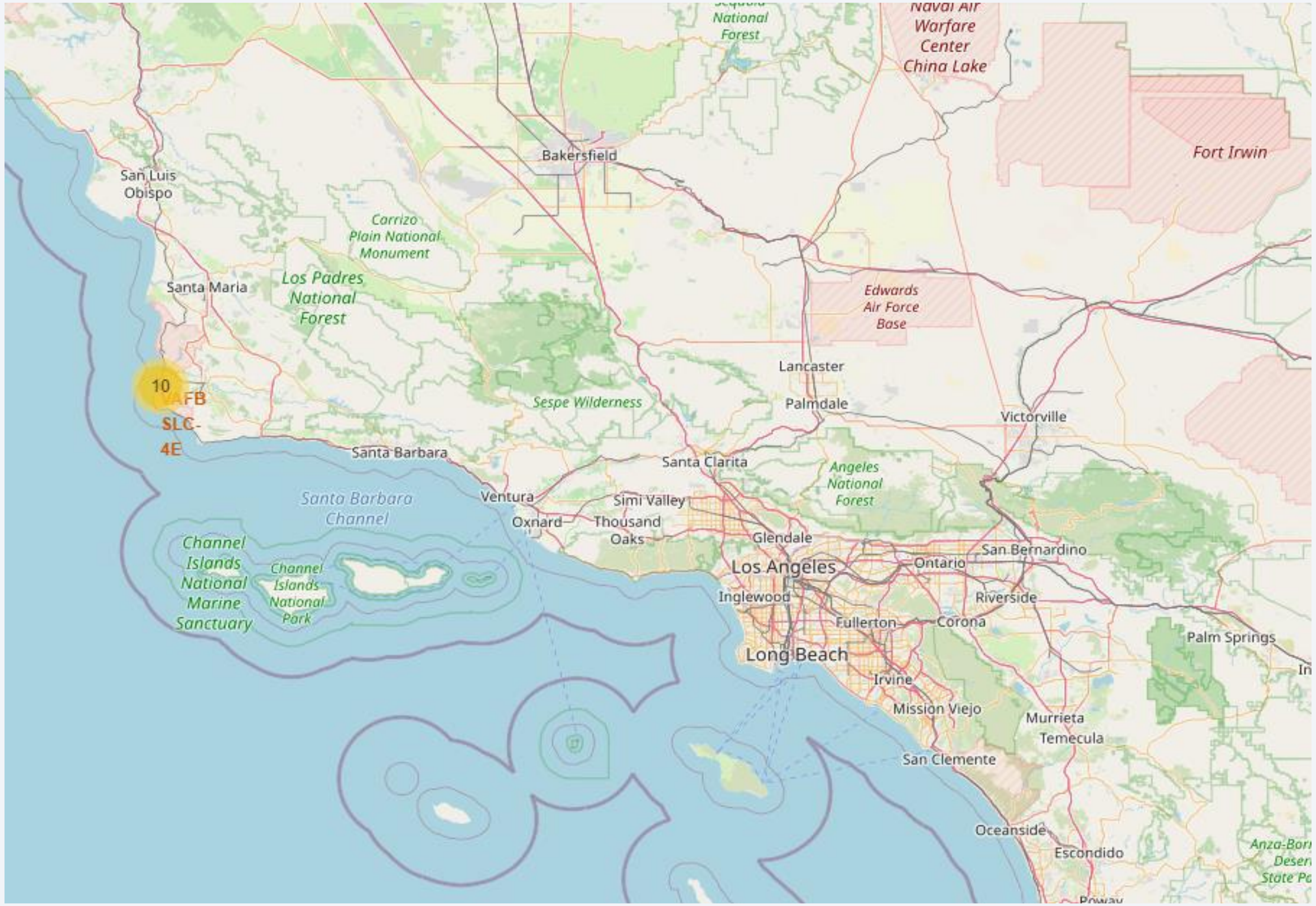
Launch Sites Proximities Analysis

Global Map of SpaceX Launch Sites

- Space X has Launch Sites Exclusively in the United States
- There are 4 total Launch Sites
- 1 is Located on the West Coast inCalifornia
- 3 are located on the East Coast in Florida

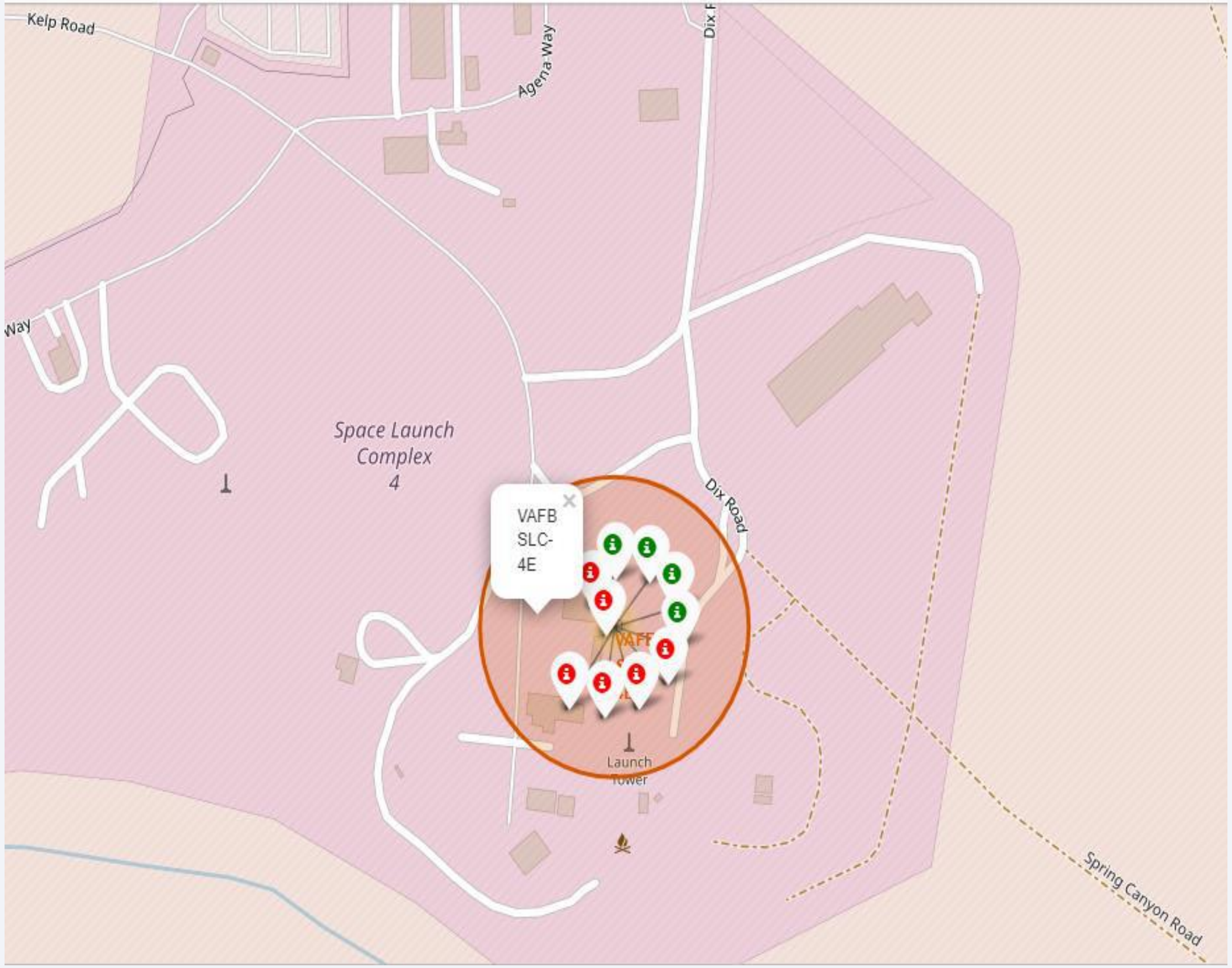
SpaceX VAFB SLC-4ELaunch Site

- •Space X has a single Launch Site Stationed off the Coast of Santa Mariain California
- •Our Cluster Marker indicates 10 Falcon 9 launches have taken place at this site



SpaceX VAFB SLC-4ELaunch Site

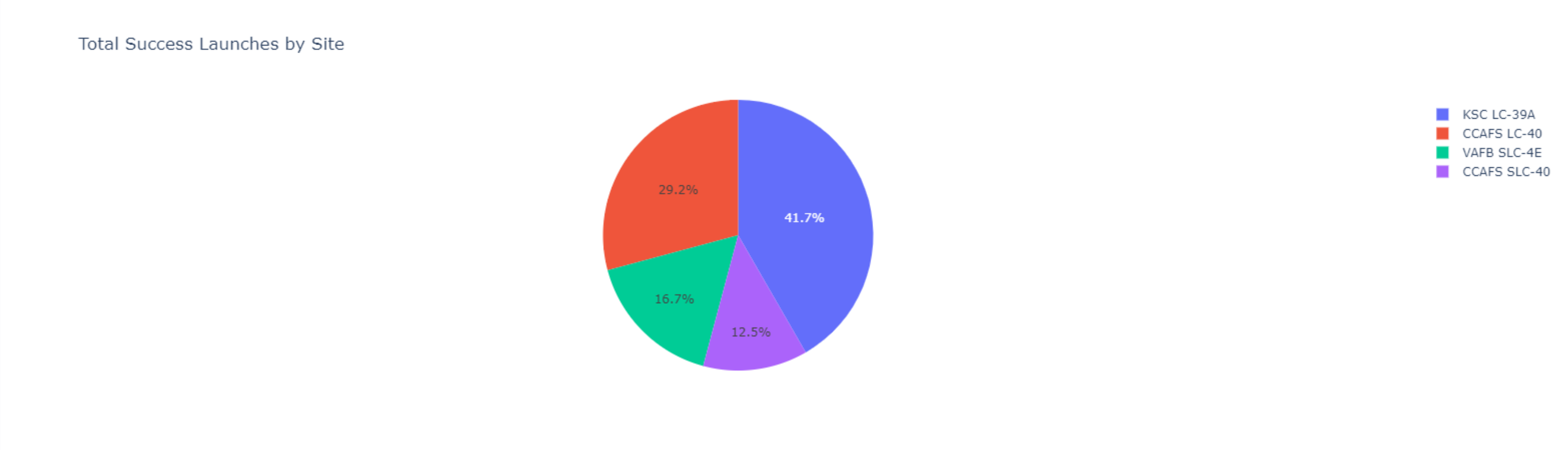
- •4 Rockets Landed
- •6 Rockets Failed to Land



Section 4

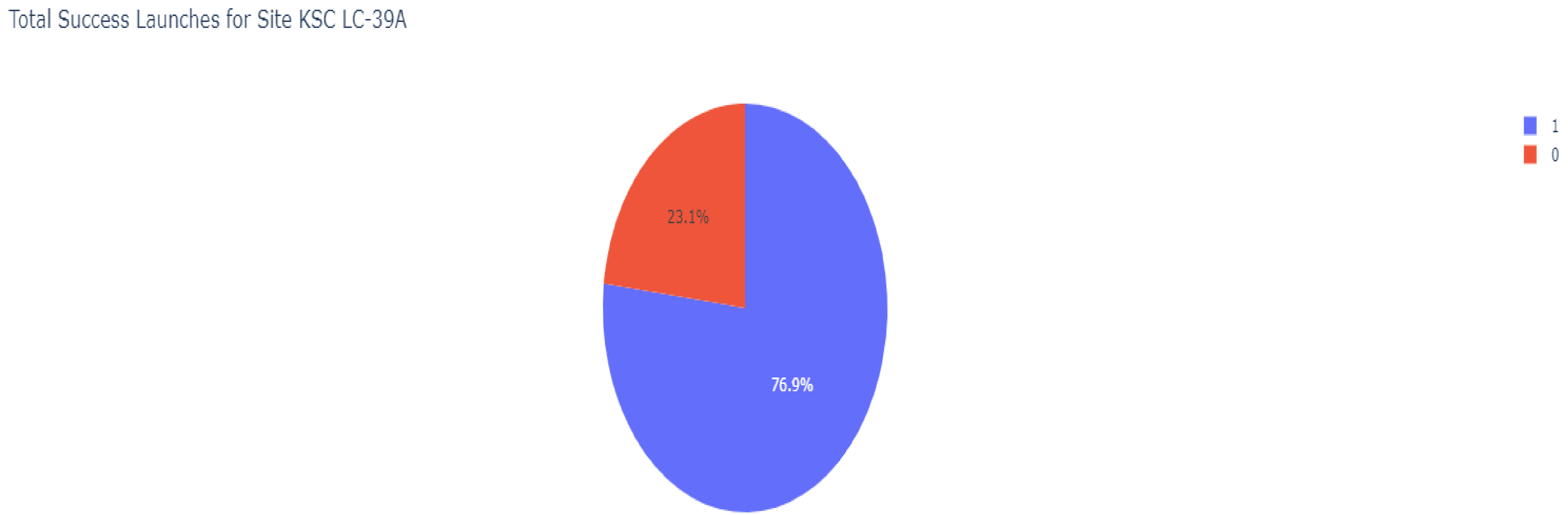
Build a Dashboard with Plotly Dash

Probability of launch site given Successful Landing



We see that the KSC LC-39A Launch site accounts for the largest percentage of the total number ofsuccessful landings at 41.7%

Launch Site with the Highest Probability of Success



- The KSC LC-39A Launch Site also has the highest probability of success per launch
- 76.9% of all launches at the KSC LC-39A Site Land Successfully
- 23.1%of all launches at the KSC LC-39A Site Fail to Land

Section 5

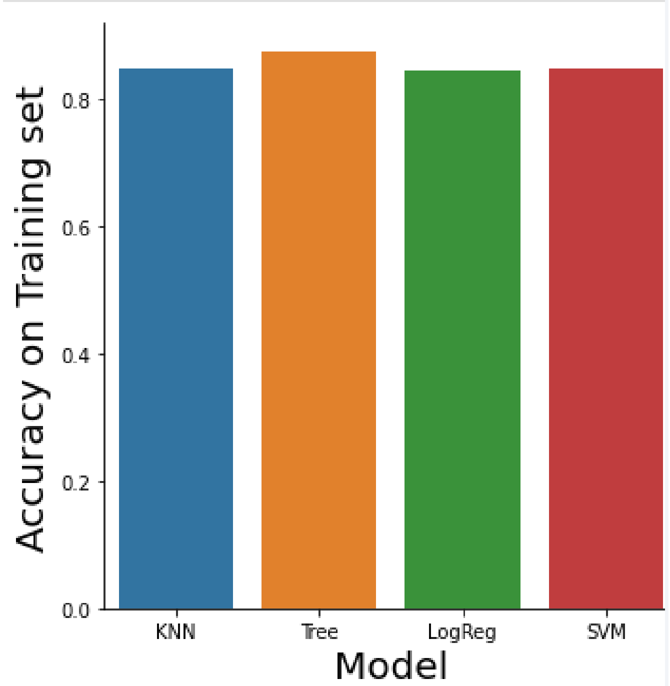
Predictive Analysis (Classification)

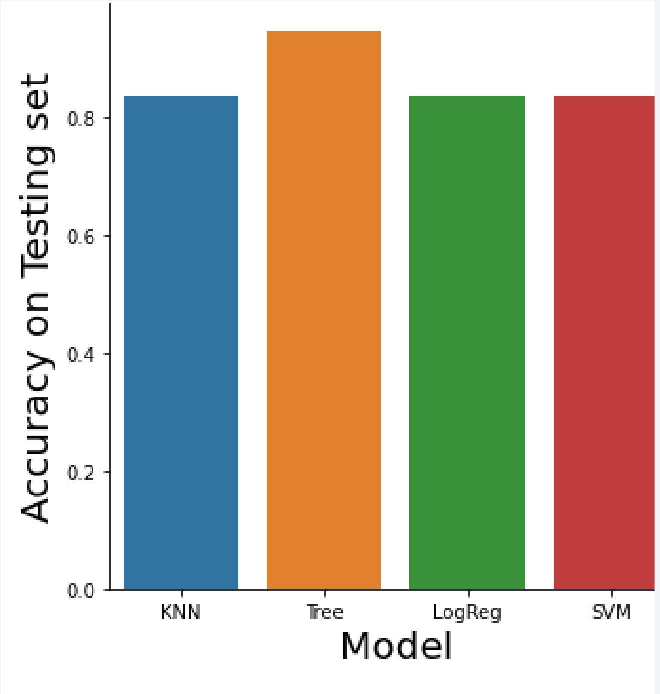
Classification Accuracy

•The data is split into a testing and training set for developing the model

- •Model Tuning is performed using Grid Search and 10 fold cross validation on training set
- •Bar Graphs Show Training and Testing accuracy of all 4 models
- •Decision Tree Model has the Highest accuracy on testing and training data
- •All models are resilient to new data and retain their accuracy we on the testing set
- •Decision Tree is clearly the best performing model

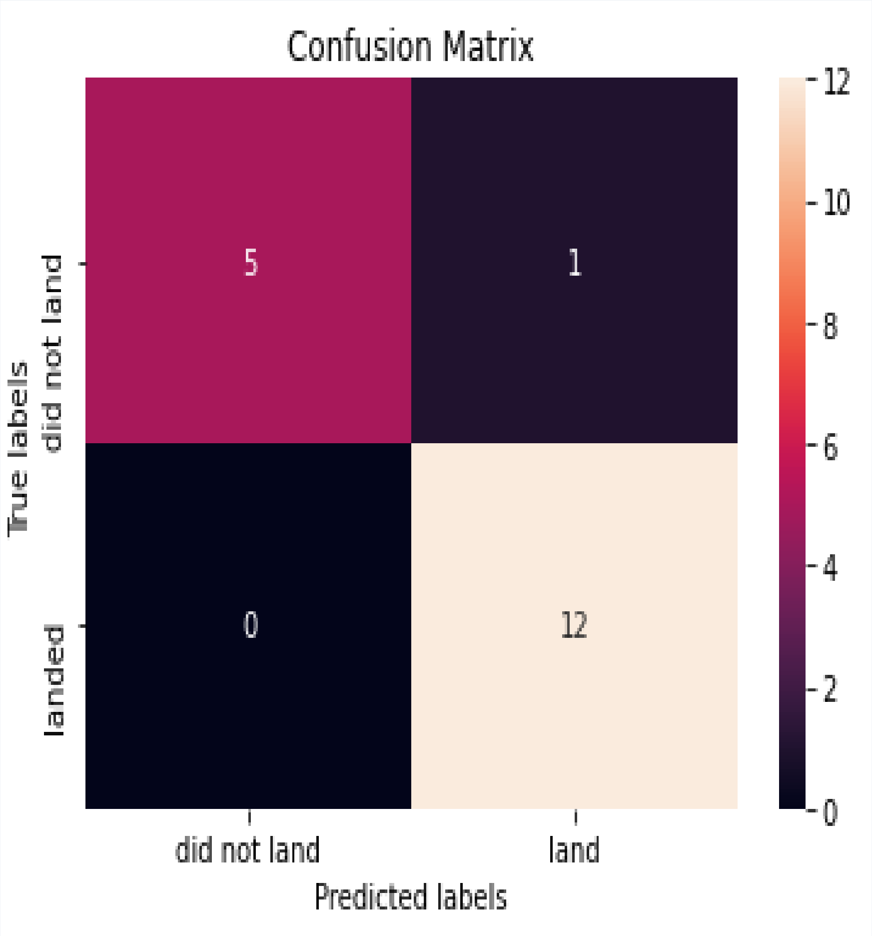
- Decision tree highest training accuracy=0.875
- Decision tree highest testing accuracy=0.944453





Confusion Matrix

- Decision Tree Confusion Matrix Testing Set
- •We see that our model is able to correctly predict that 5 of the 6 testing points that fail to land
- •Sensitivity = 5/6 = 83.33%
- •Our model is able to correctly predict that all 12 rockets that land will land
- •Specificity = 12/12 = 100%
- •Accuracy = (5+12)/(5+1+12+0) = 94.44%



Conclusions

- •We were able to build a decision tree model that can predict the probability of Falcon 9 Rocket Stage 1 Landing Successfully with an 94.44% accuracy on our out of sample data and 87.5% accuracy on in sample data
- •We found that low weight payloads are more likely to land successfully than heavy payloads
- •We found that SpaceX engineers have been improving the probability of success every year since 2013 but progress has begun to slow down reaching a maximum yearly success percentage of about 63%in 2020 meaning our model will have to be refined as time passes to keep up to date
- •The KSC LC-39A Launch Site also has the highest probability of success per launch
- •The type of orbit required for the launch has an impact on the landing success,theES-L1, SSO, HEO, and GEO orbits havethe highest rate of success for landing

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!