IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

KARPAGA SELVI

4 April 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection from SpaceXAPI and Wiki site using REST API and Web Scraping methods
  - Data wrangling convert the collected data with target values
  - Data Visualization with scatter, bar and line plots gives the insight of relationship between attributes and trends
  - EDA with SQL helps to get more understanding of the collected data
  - Interactive visualization techniques like folium shows geographical relationship of launch sites
  - Dash board provides live data visualization techniques
  - Predictive analysis : various candidate algorithms are trained with different parameters and validated for accuracy
  - Best model is selected by testing accuracy (high)
- Summary of all results
  - Decision Tree model have high Training Accuracy
  - All model are equally perform in testing data with same accuracy and confusion matrix
  - Decision tree can be considered as suitable or best among available because of it less computational Complexity
  - This Model is easy to interpret

Introduction

- Project background and context
  - Space Rockets Stage 1 is very expensive but reusable if landed properly during the previous launch
  - If stage 1 is properly landed and captured that will reduce the cost of Space Travel by approximately more than 50%.
  - This study is to develop a predictive model to predict Whether the stage 1 will be landed properly based on previous historical data of launching
- Problems you want to find answers
  - Given Details like rocket used, launching site, date , payload types, payload masses, Orbit, reused etc, we need to predict the possibility of proper landing or not

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from REST API and FROM WIKI site.
  - By using the API , endpoints data is collected
  - Web Scraping techniques are employed to collect data from wiki sites
- Perform data wrangling
  - Distribution of Launching Sites and Orbits are calculated
  - A new attribute ‘Class’ is added to dataset to specify Success full landing od stage 1 as 1 and 0 for failure

Methodology

- Perform exploratory data analysis (EDA) using visualization and SQL
  - Relationship between various attributes to target class is studied by scatter plots
  - Success rate with respect to time is studied
  - Categorical variables or nominal values are converted to numeric data by applying one hot encoding techniques
- Perform interactive visual analytics using Folium and Plotly Dash
  - Launching Sites, number of successful and unsuccessful launching in the sites are marked over a geographical map using Folium
  - Proximity of the lancing site to coastal lines are identified from the geographical interactive map

Methodology

- Perform predictive analysis using classification models
  - Data is split as training and testing data
  - Model are trained using training data – KNN, Logistic Regression, svm and Decision tree methods are explored
- How to build, tune, evaluate classification models
  - While training the model parameters are fine tuned by GridSearchCV method and 10 fold cross validation is employed
  - Model Accuracy is calculated from test data and confusion matrices are drawn

Data Collection

- Data Collection
  - SpaceX API - Using API
  - Wiki Pedia - Web Scraping using Beautiful Soup method

Data Collection – SpaceX API

- Present your data collection with SpaceX REST flowcharts

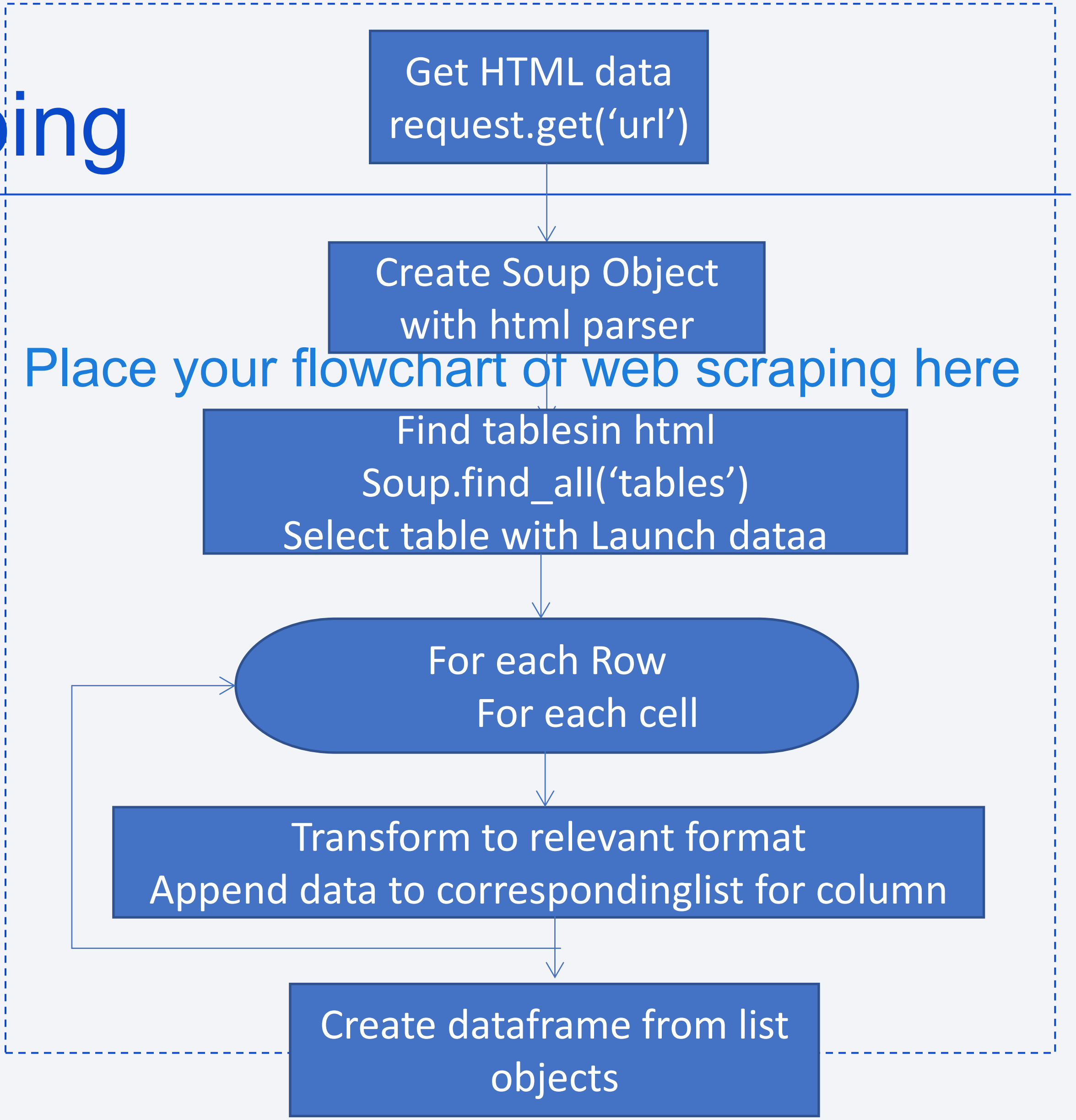
- GitHub URL: https://github.com/drkarpagaselvi/SpaceX-predictive-Analysis.git

- **File Name :** datacollectionfromAPI.ipynb.ipynb

Place y calls
here

Data Collection - Scraping

- Web Scraping FlowChart
- GitHub URL: https://github.com/drkarpagaselvi/SpaceX-predictive-Analysis.git
- File Name : datacollectiowebsscraping.ipynb.ipyn



Data Wrangling

- The frequency distribution of launch sites, orbits and landing outcomes are calculated
- From the landing outcome target class for the data set is added to data set such that 1 for Successful Landin and 0 for Failure
- GitHub URL: https://github.com/drkarpagaselvi/SpaceX-predictive-Analysis.git
- File Name : wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts

Chart Type	Reasons
Scatter Plots	To determine the relationship of various attributes among them themselves an with target class
Bar chart	Relationship between categorical attributes and target class
Line chart	Trends to show the how an attribute changes with respect to another (Eg Time)

- GitHub URL: https://github.com/drkarpagaselvi/SpaceX-predictive-Analysis.git

- File Name : datavisualization.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - select distinct(Launch_site) from spacexdataset
  - select Launch_site from spacexdataset where launch_site like 'CCA%' limit 5
  - select sum(payload_mass__kg_) from spacexdataset where customer='NASA (CRS)‘
  - select avg(payload_mass__kg_) from spacexdataset where Booster_version = 'F9 v1.1‘
  - select date from spacexdataset where landing__outcome ='Success (ground pad)' order by date limit 1
  - select booster_version from spacexdataset where landing__outcome = 'Success (drone ship)' and (payload_mass__kg_ >4000 and payload_mass__kg_ <6000)
  - select mission_outcome, count(mission_outcome) from spacexdataset group by mission_outcome
  - select Booster_version from spacexdataset where payload_mass__kg_ = (select max(payload_mass__kg_) from spacexdataset)
  - select date, booster_version, launch_site , landing__outcome from spacexdataset where landing__outcome ='Failure (drone ship)' and year(date) = 2015
  - select landing__outcome , count(landing__outcome) as freq1 from spacexdataset where date > Date('2010-06-04') and date < Date('2017-03-20') group by landing__outcome order by freq1 DESC
- GitHub URL: https://github.com/drkarpagaselvi/SpaceX-predictive-Analysis.git
- File Name : SQL EDA.ipynb

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
- GitHub URL: https://github.com/drkarpagaselvi/SpaceX-predictive-Analysis.git
- File Name : VisualAnalytics-dashboards.ipynb

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- GitHub URL: https://github.com/drkarpagaselvi/SpaceX-predictive-Analysis.git
- File Name : 7 spacex_dash_app.py

Predictive Analysis (Classification)

Developing best performing classification model Summary

Model Name	Parameters tested	Chosen Parameter	Training Accuracy	Testing Accuracy
Logistic Regression	"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']	'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'	0.846	0.833
Decision Tree	parameters = {'criterion': ['gini', 'entropy'], 'splitter': ['best', 'random'], 'max_depth': [2*n **for** n **in** range(1,10)], 'max_features': ['auto', 'sqrt'], 'min_samples_leaf': [1, 2, 4], 'min_samples_split': [2, 5, 10]}	'criterion': 'entropy', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'	0.873	0.833
SVM	'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'), ‘ C': np.logspace(-3, 3, 5), 'gamma':np.logspace(-3, 3, 5)	'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'	0.848	0.833
KNN	parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'], 'p': [1,2]}	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}	0.848	0.833

Predictive Analysis (Classification)

- Model development process Algorithm

  1. Preprocessing data
  2. Split test and train set
  3. Choose candidate Models
  4. For each Models in candidate Models

     1.Choose the possible values for fine tuning parameters

     2.Train the model with All possible set of tuning parameters and model with best parametes using GridSearchCV (Model Validation by 10 fold cross validation)

     3.Evaluate All models with test data

     4. Select the model with highest test Accuracy

- GitHub URL: https://github.com/drkarpagaselvi/SpaceX-predictive-Analysis.git
- File Name : Prediction.ipynb

Results

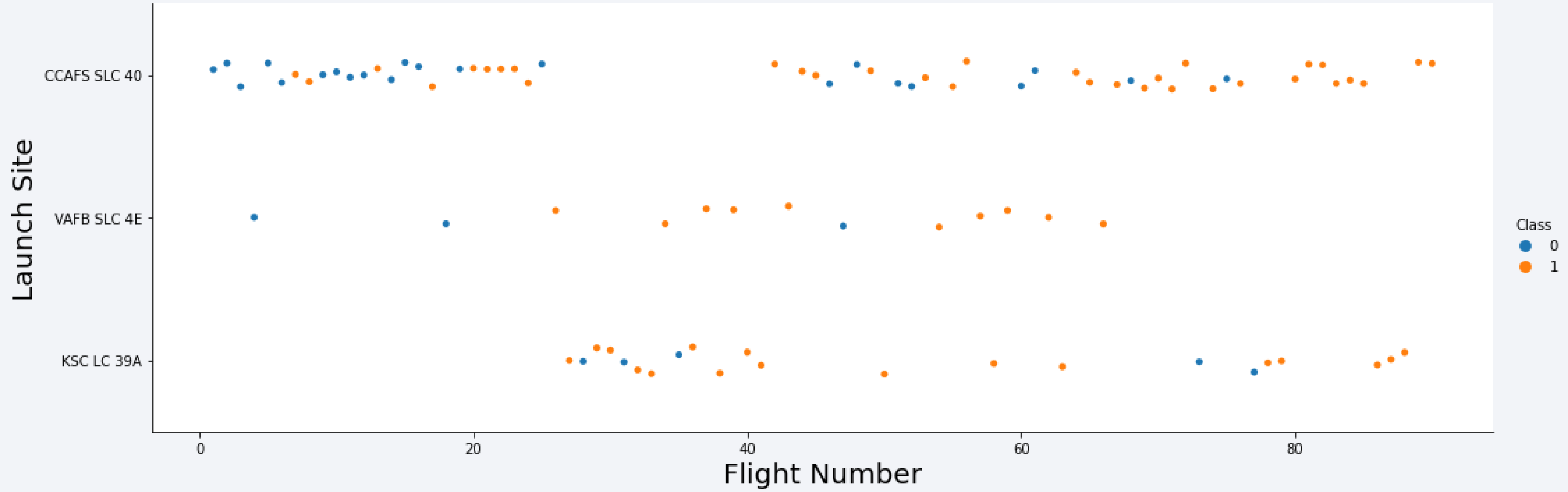
- Exploratory Data analysis results
  - Orbits ES-L, GEO,HEO, SSO – High Success rates
  - Launch Site VAFB SLC 4E – No Rockets Launched for Heavy payloads
  - Success rate is increasing with respect to Time
  - Various Queries are Executed to understand the data and results are presented in slides 27 to 36
- Interactive analytics demo in screenshots
- Predictive analysis results
  - Decision Tree is having higher training Accuracy Decision Tree is having higher training Accuracy
  - All Models are having same testing Accuracy 83.333..
  - Decision tree – choosen – simple to interpret and less computational complexity

Section 2

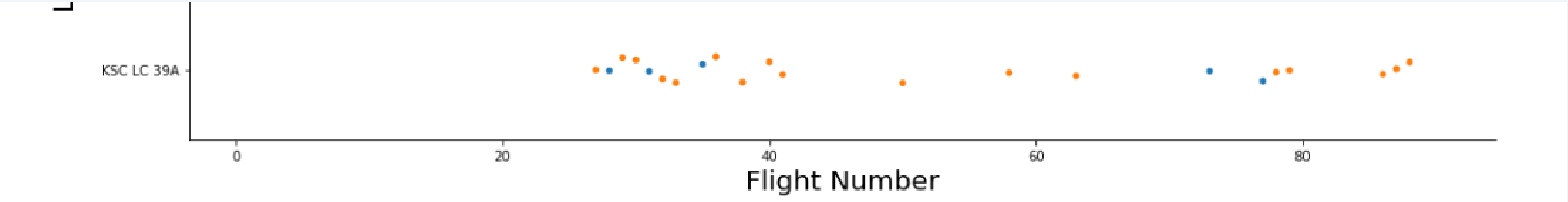
Insights drawn from EDA

Flight Number vs. Launch Site

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



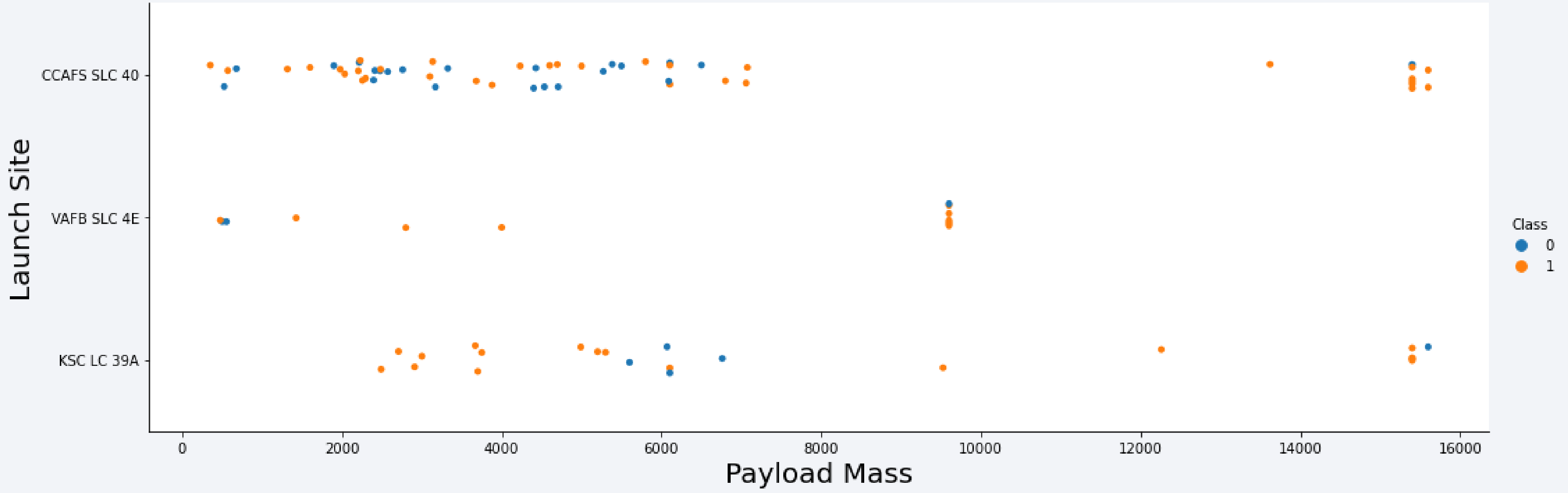
- Show the screenshot of the scatter plot with explanations



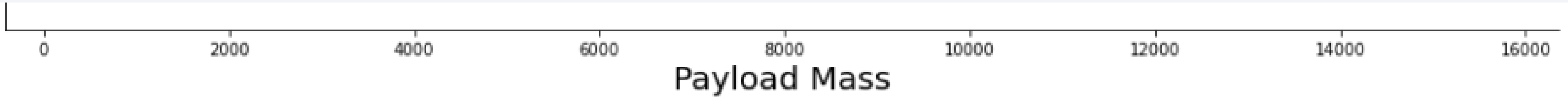
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots. higher the flight number , failure is remarkably less Launch site CCSFA SLC 40 Success and failures are mixed Launch sites VAFB SLC 4e and KCS LC 39 A are having more success rate but compartively less number of Launches

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site



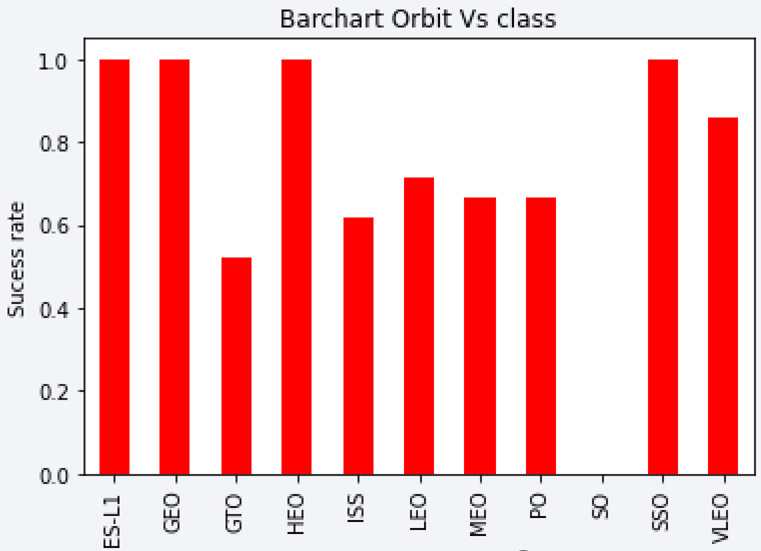
- Show the screenshot of the scatter plot with explanations



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

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

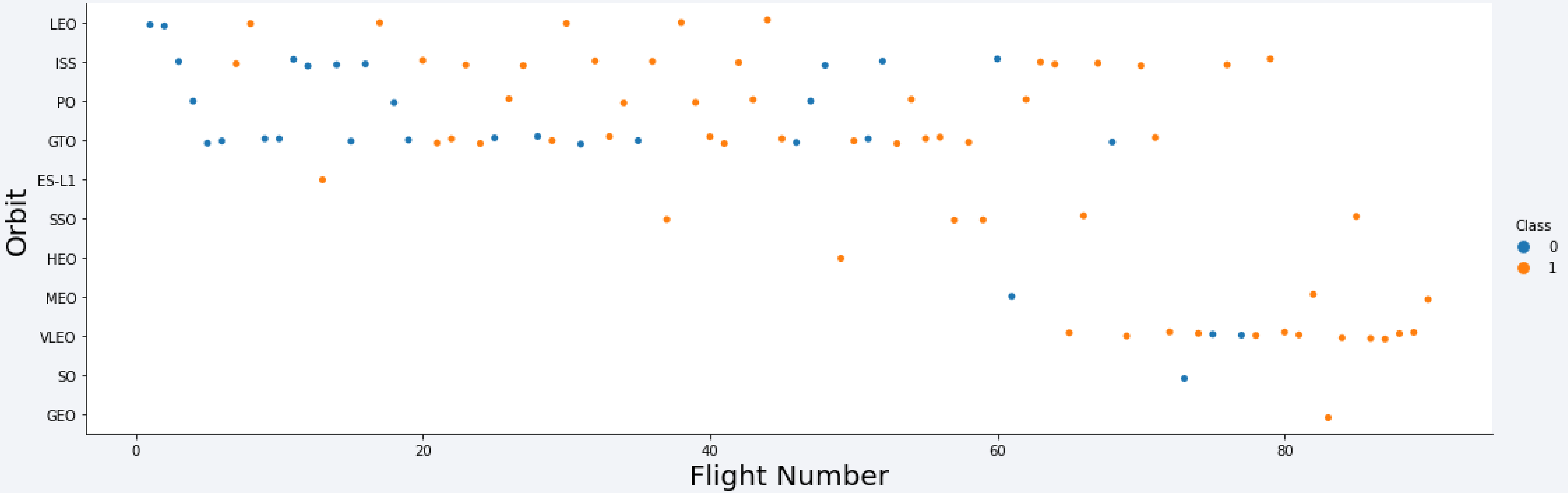


- Show the screenshot of the bar plot with explanations

Analyze the ploted bar chart try to find which orbits have high sucess rate. ES-L,GEO,, HEO, SSO Orbits are having high Sucess rate SO sucess rate is zero All other orbits are having more than 50 percent sucess rate

Flight Number vs. Orbit Type

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

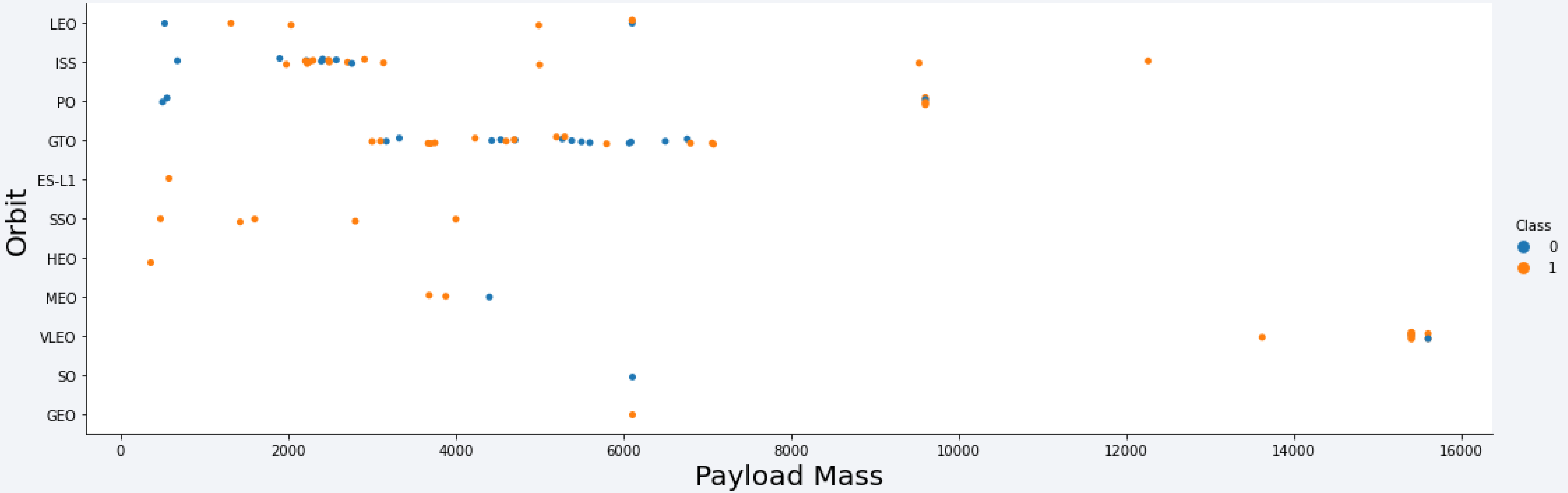


- Show the screenshot of the scatter plot with explanations

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 vs. Orbit Type

- Show a scatter point of payload vs. orbit type



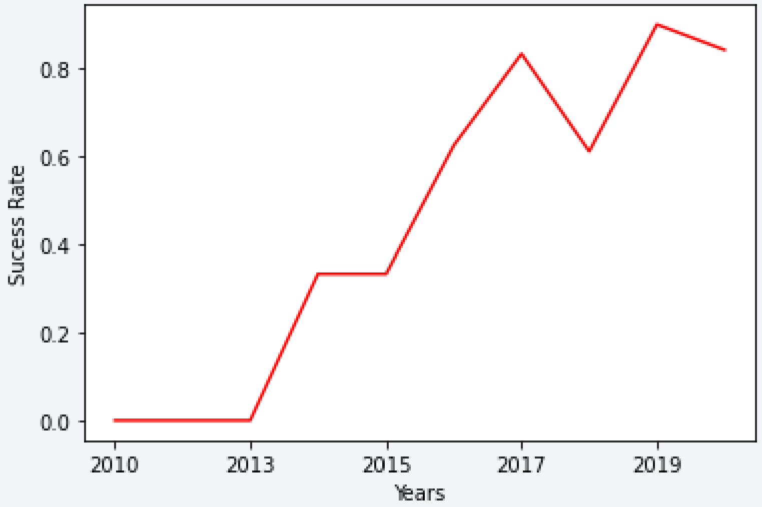
- Show the screenshot of the scatter plot with explanations

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 Yearly Trend

- Show a line chart of yearly average success rate



- Show the screenshot of the scatter plot with explanations

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

All Launch Site Names

- Find the names of the unique launch sites
  - **%sql** select distinct(Launch_site) from spacexdataset
- Present your query result with a short explanation here
  - Distinct will return the unique values of the column

Done.

Out[6]:

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

Launch Site Names Begin with 'CCA'

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

  Query : %**sql** select Launch_site from spacexdataset where launch_site like 'CCA%' limit 5

- Present your query result with a short explanation here
  - % - Wild card caracter
  - Like class compares the similarity
  - Limit clause limits the retrived recordset

Out[10]:

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

  Query:%**sql** select sum(payload_mass__kg_) from spacexdataset where customer='NASA (CRS)'

- Present your query result with a short explanation here
  - Sum Aggregate Function sums a column in recordset
  - Where clause is used to for condition

Out[14]:

1
45596

Average Payload Mass by F9 v1.1

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

Query; **%sql** select avg(payload_mass__kg_) from spacexdataset where Booster_version = 'F9 v1.1'

- Present your query result with a short explanation here
  - Avg- Aggregate Function sums a column in recordset
  - Where clause is used to for condition

1
2928

First Successful Ground Landing Date

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

Query: **%sql** select date from spacexdataset where landing__outcome ='Success (ground pad)' order by date limit 1

Out[29]:

DATE
2015-12-22

- Present your query result with a short explanation here
    - Order by clause sorts the record set by the column specified.
    - Default assending order
    - Limit 1 means first date will be retrived
    - Where clause used to check condition of Success (groundpad)

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

Query: select booster_version from spacexdataset where landing__outcome = 'Success (drone ship)' and (payload_mass__kg_ >4000 and payload_mass__kg_ <6000)

- Present your query result with a short explanation here

  ‘and’ logical operator checks for more than one condition in

  where clause

Done.

Out[31]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Query :%**sql** select mission_outcome, count(mission_outcome) from spacexdataset group by mission_outcome

- Present your query result with a short explanation here
- Group by clause used to accumulate data on specific value
- Column and count(column) with group by gives frequency distribution of the column values

Out[32]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

Query: **%sql** select Booster_version from spacexdataset where payload_mass__kg_ = (select max(payload_mass__kg_) from spacexdataset)

- Present your query result with a short explanation here
- Sub Query is used to get the Max pay load mass values
- The result of sub Query is value for Condition in main Query

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Query :%**sql** select date, booster_version, launch_site , landing__outcome from spacexdataset where landing__outcome ='Failure (drone ship)' and year(date) = 2015
- Present your query result with a short explanation here

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 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
- Query **%sql** select landing__outcome , count(landing__outcome) as freq1 from spacexdataset where date > Date('2010-06-04') and date < Date('2017-03-20') group by landing__outcome order by freq1 DESC
- Present your query result with a short explanation here

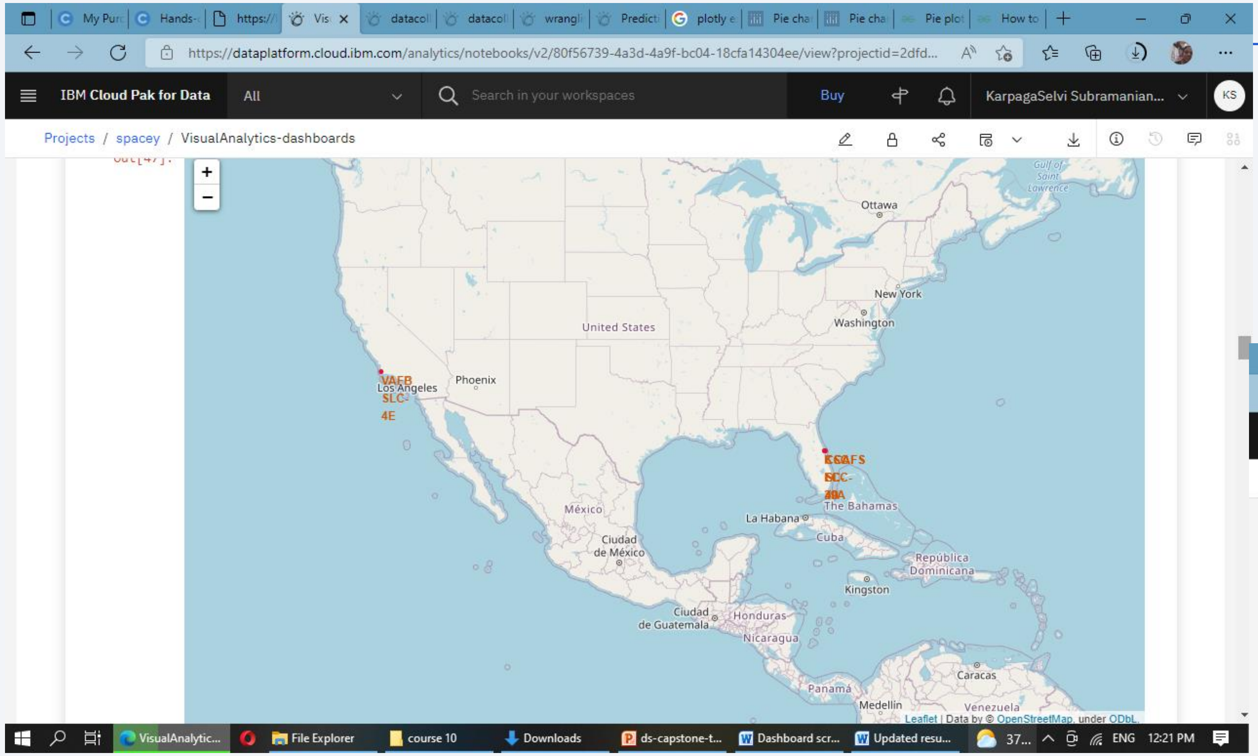
Out[51]:

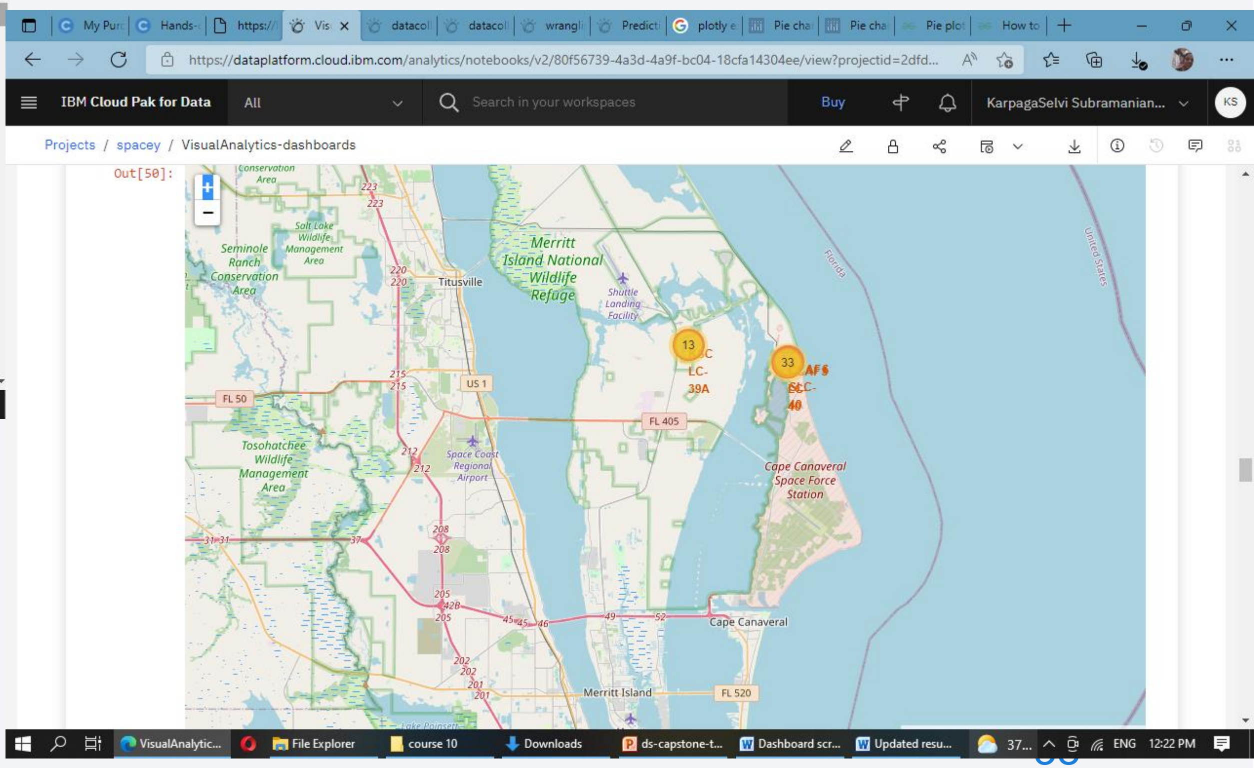
landing__outcome	freq1
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Uncontrolled (ocean)	2
Failure (parachute)	1
Precluded (drone ship)	1

Section 3

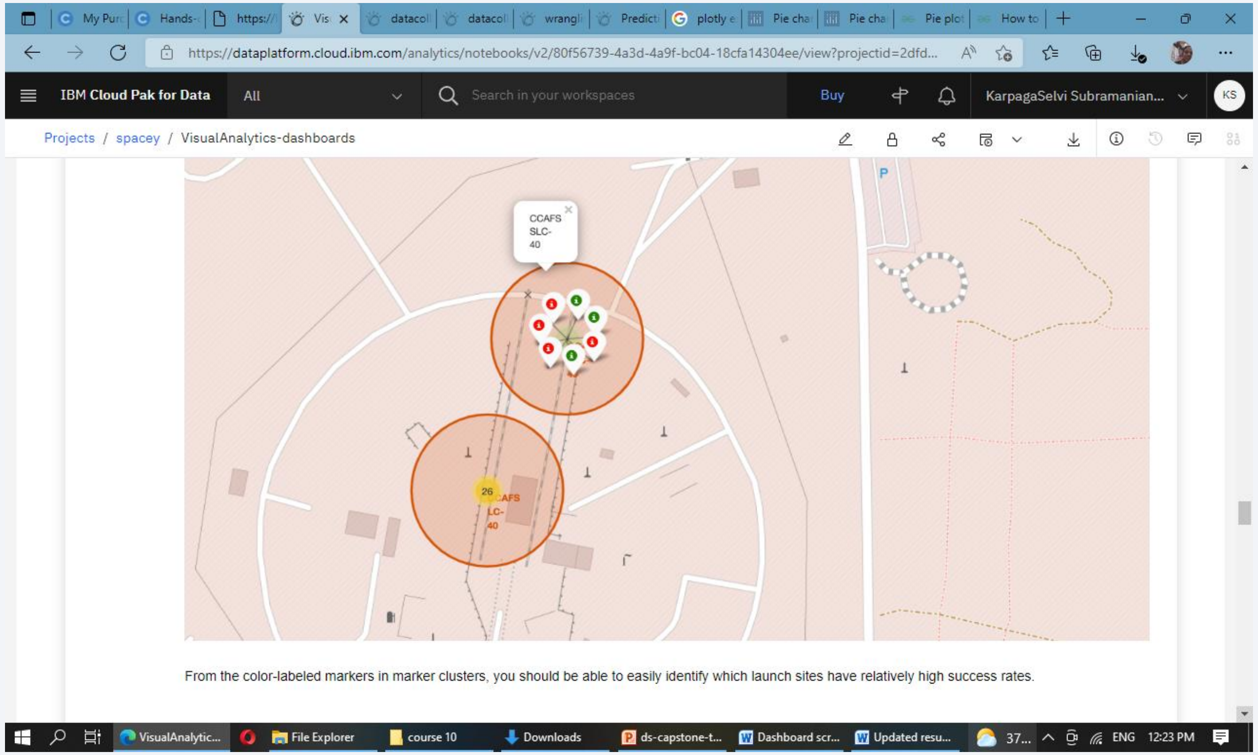
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>





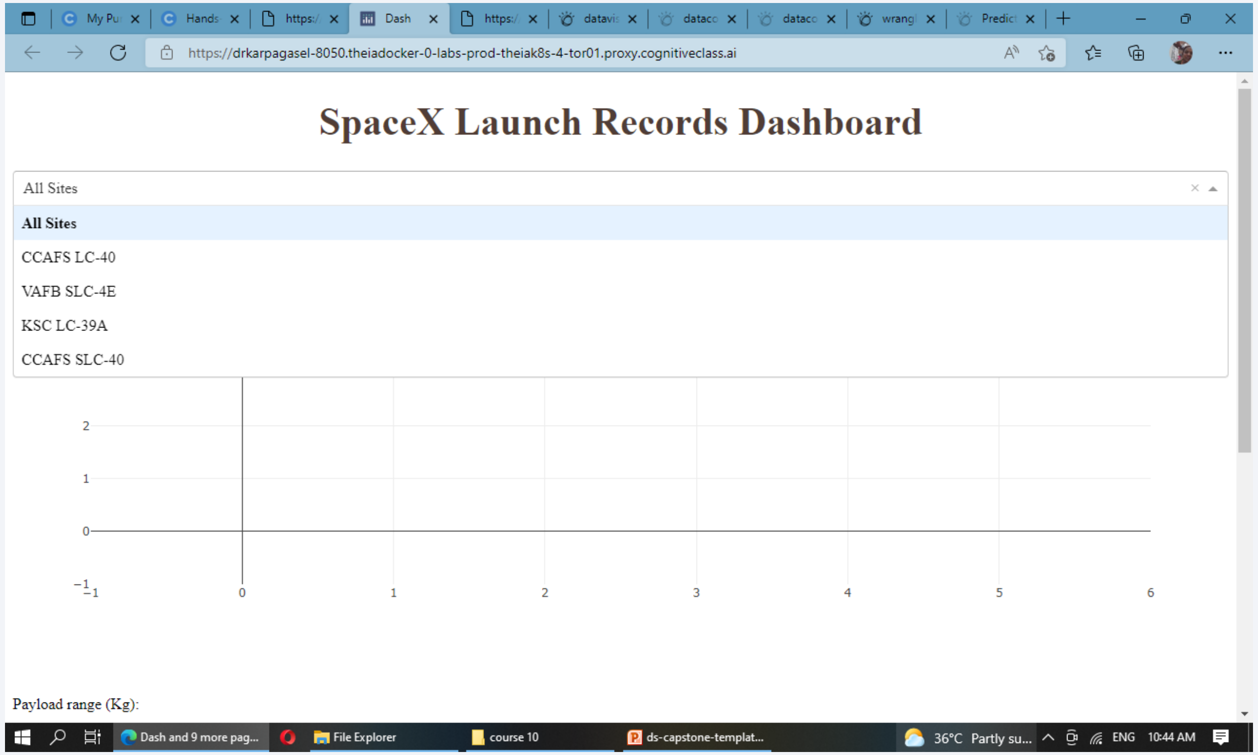
<Folium Map Screenshot 2>



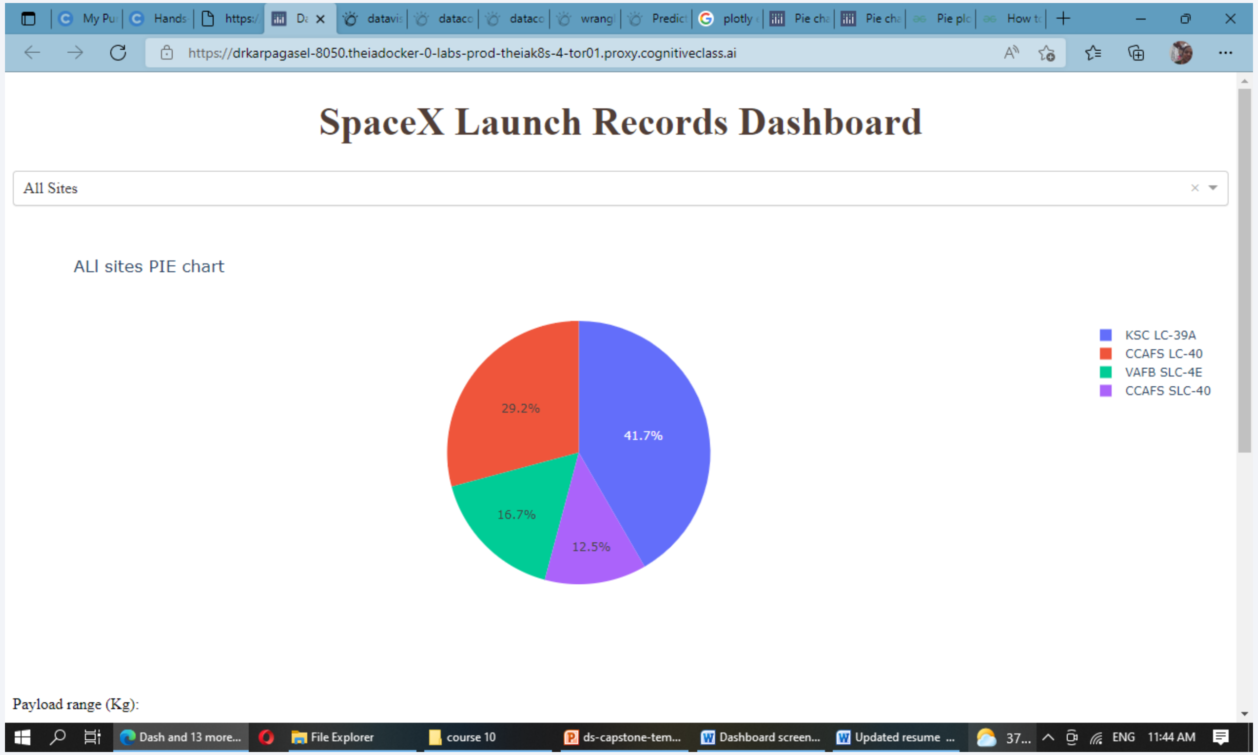
Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



<Dashboard Screenshot 2>

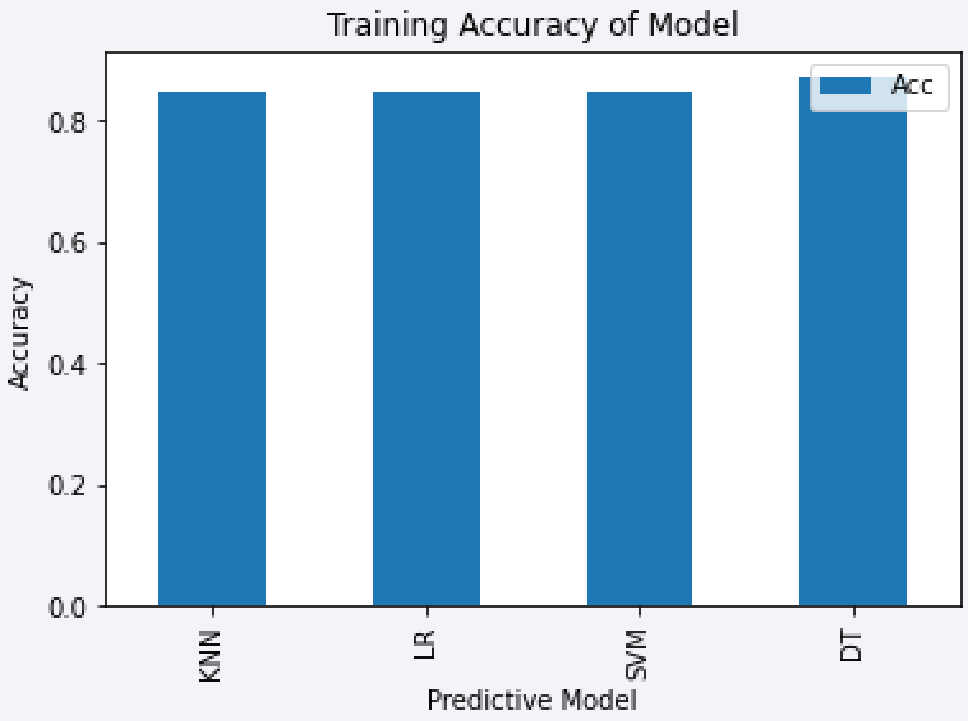


Section 5

Predictive Analysis (Classification)

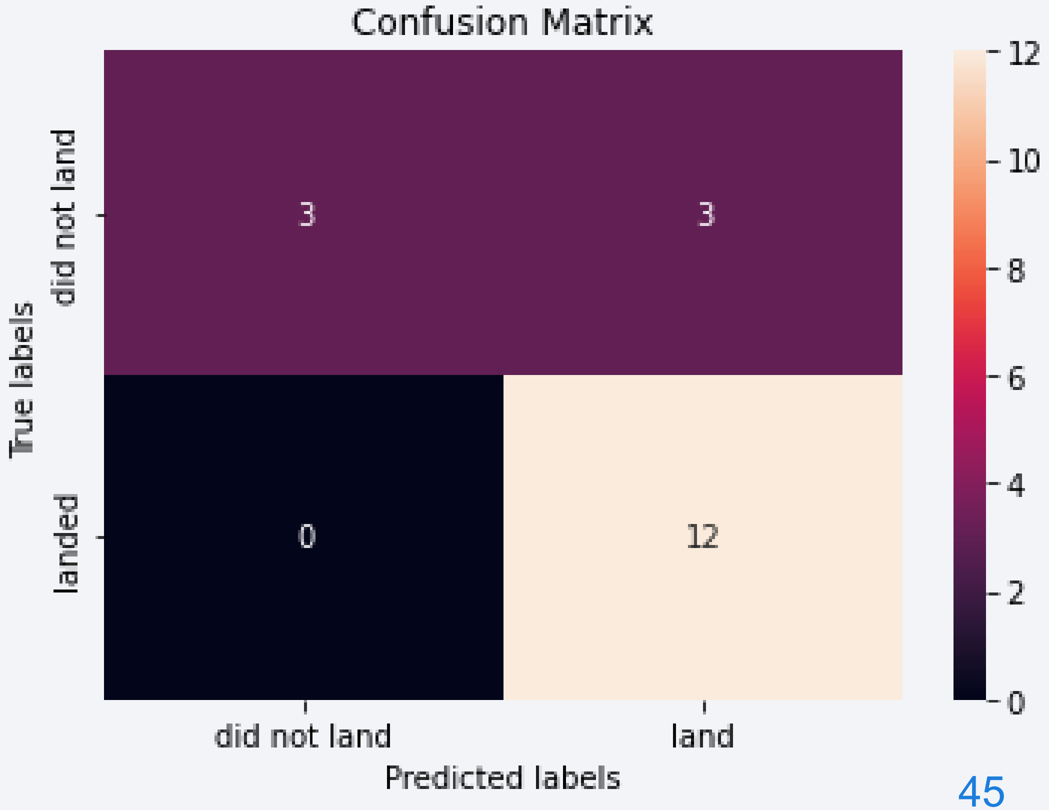
Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart
  - Training Accuracy for all 4 model are shown . Decision Tree is having higher training Accuracy
- Find which model has the highest classification accuracy
  - All Models Testing Accuracy is 83.33..
  - All Model Confusion Matrix having 3 False positive for test data



Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation
- All Models are having Same Confusion matrix with 3 False Positive.



Conclusions

- Decision Tree model have high Training Accuracy
- All model are equally perform in testing data with same accuracy and confusion matrix
- Decision tree can be considered as suitable or best among available because of it less computational Complexity
- This Model is easy to interpret

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!