IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jawad Haider
17 July 2023

Outline

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

Executive Summary

- Problem Statement: Predicting the success of the first stage landing of the SpaceX Falcon 9 rocket, with cost implications for rocket launches and competitive bidding against SpaceX.
- Methodology Overview:
  - Data Collection: Obtained data from the SpaceX API and performed necessary data wrangling.
  - Data Wrangling: Conducted exploratory data analysis (EDA) to identify patterns and determine training labels.
  - SQL Notebook: Loaded the SpaceX dataset into a Db2 database and executed SQL queries for analysis.
  - Launch Site Analysis: Used Folium for interactive visual analytics to analyze launch site locations, success rates, and proximity calculations.
  - Machine Learning Prediction: Conducted EDA, created class labels, standardized data, and evaluated multiple machine learning methods.
- Results and Findings:
  - Machine Learning Models: Developed and evaluated various models, including SVM, Classification Trees, and Logistic Regression.
  - Performance Metrics: Achieved accuracy of 89% for predicting the success of first stage landings.
  - Key Insights: Identified significant factors influencing successful landings and provided actionable insights for alternate companies bidding against SpaceX.
- Conclusion and Future Steps:
  - Successfully addressed the problem of predicting first stage landing success for SpaceX Falcon 9 rockets.
  - Demonstrated the value of data science methodologies in making informed decisions regarding rocket launches and cost considerations.
  - Future steps involve further refining the models, considering additional features, and exploring new machine learning techniques

Introduction

- Project Background and Context:
  - This project serves as the final step in the IBM Data Science Professional Certificate and the Applied Data Science with Python Specialization.
  - You will assume the role of a Data Scientist working for a startup aiming to compete with SpaceX.
  - SpaceX's ability to reuse the first stage of the Falcon 9 rocket significantly reduces launch costs, making them an industry leader.

- Problems You Want to Find Answers:
  - The main problem is to predict the success of the first stage landing of the SpaceX Falcon 9 rocket.
  - Accurate predictions are crucial for determining the cost of a launch, which is essential for competitive bidding against SpaceX.
  - Alternate companies can use the predictions to make informed bids for rocket launches.

- Objective of the Project:
  - Develop a predictive model to determine the success of the first stage landing of the SpaceX Falcon 9 rocket.
  - Follow the complete data science methodology, including data collection, data wrangling, exploratory data analysis, data visualization, model development, model evaluation, and reporting findings.
  - By successfully completing this project, you will have a valuable addition to your data science and machine learning portfolio to showcase to potential employers.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected through the SpaceX API.
  - A GET request was made to the SpaceX API to retrieve the necessary data.
  - The SpaceX API provided access to information about launches, landing outcomes, and other relevant data points.
  - Data wrangling techniques were applied to clean and format the collected data for further analysis.
- Perform data wrangling
  - Data processing involved performing data wrangling and formatting to ensure its suitability for analysis.
  - Steps such as cleaning the data, handling missing values, removing duplicates, and transforming data types were applied.
  - Exploratory data analysis (EDA) techniques were utilized to gain insights, identify patterns, and understand the characteristics of the data.
  - Feature engineering techniques may have been applied to derive new features or transform existing ones, enhancing the predictive power of 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
  - How to build, tune, evaluate classification models

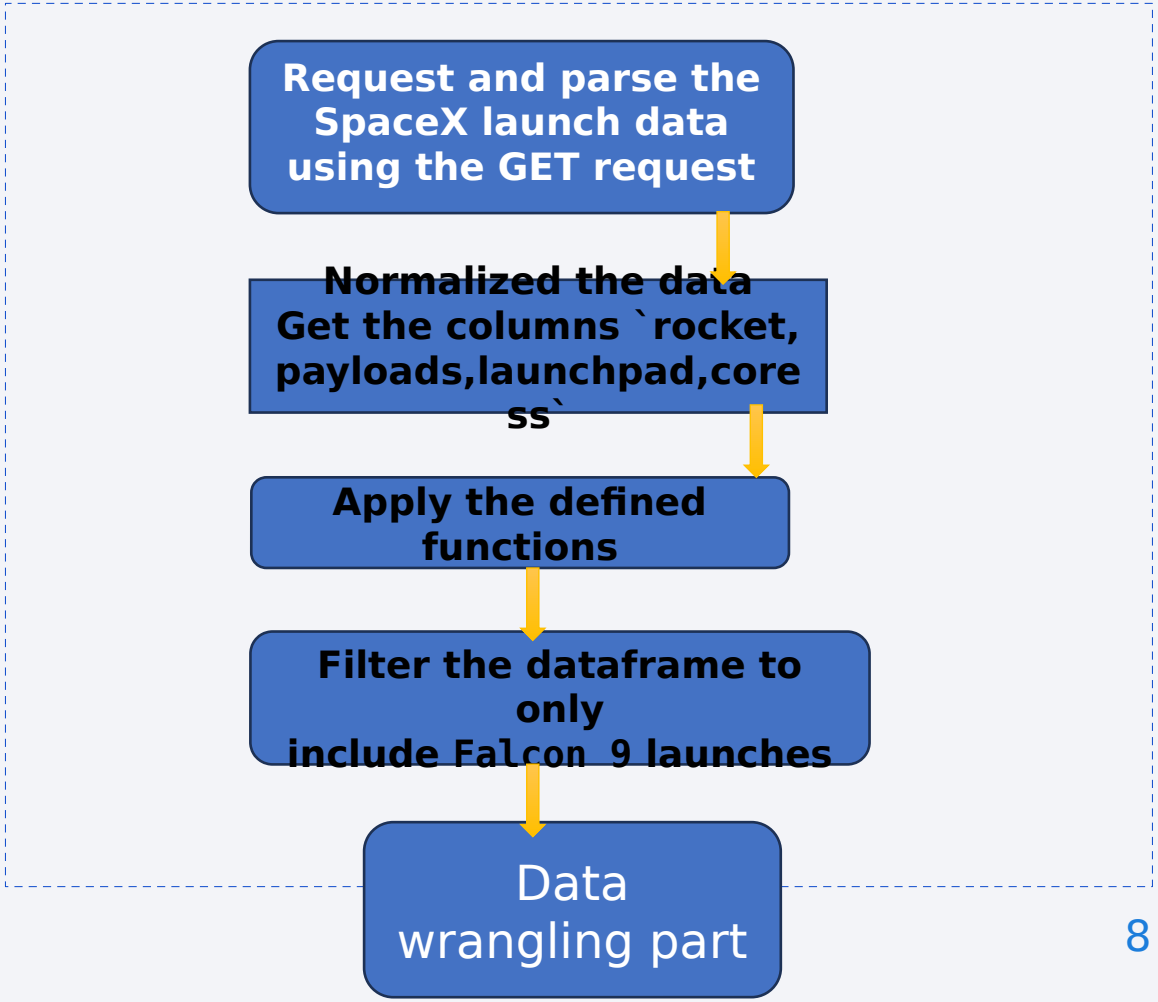
Data Collection

Identify Data Sources: The first step in the data collection process involved identifying the relevant data sources. These sources may include public databases, APIs, websites, or internal company databases.
Determine Data Requirements: Once the data sources were identified, the next step was to determine the specific data requirements for the project. This involved defining the variables, attributes, or metrics needed to address the research questions or solve the problem at hand.
Access Data Sources: After determining the data requirements, the data collection process involved accessing the identified data sources. This could be done through direct downloads, API calls, web scraping, or data acquisition from third-party providers.
Extract Data: The extracted data was then collected in its raw format. This raw data could be in various forms such as CSV files, JSON objects, or database tables. It was important to ensure that the data was collected accurately and completely from the selected sources.
Clean and Validate Data: Once the raw data was collected, the next step was to clean and validate it. This involved handling missing values, removing duplicates, and performing data quality checks. Data cleaning techniques, such as data imputation or outlier detection, were used to ensure the data's integrity and reliability.
Transform and Aggregate Data: In some cases, it was necessary to transform or aggregate the data to make it suitable for analysis. This step involved applying data preprocessing techniques such as feature scaling, normalization, or encoding categorical variables. Aggregation methods such as grouping, summarizing, or calculating derived variables were also applied as needed.
Perform Exploratory Data Analysis (EDA): Exploratory Data Analysis was conducted to gain insights into the collected data. This involved visualizing the data, identifying patterns, distributions, correlations, and potential outliers. EDA techniques such as histograms, scatter plots, or box plots were used to understand the data's characteristics.
Document Data Collection Process: Throughout the data collection process, it was essential to document the steps taken, data sources accessed, any transformations applied, and data quality checks performed. This documentation ensured transparency, reproducibility, and facilitated sharing the data collection process with stakeholders or collaborators.

Data Collection – SpaceX API

```
# Takes the dataset and uses the rocket column to call the API and append the data to the list

def getBoosterVersion(data):

# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):

# Takes the dataset and uses the payloads column to call the API and append the data to the lists

def getPayloadData(data):

# Takes the dataset and uses the cores column to call the API and append the data to the lists

def getCoreData(data):
```

- Github



Data Collection - Scraping

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'

# Lets take a subset of our dataframe# Use json_normalize meethod to convert the json result into a dataframe

data = pd.json_normalize(response.json()) keeping only the features we want and the flight number, and date_utc.

data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.

data = data[data['cores'].map(len)==1]

data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.

data['cores'] = data['cores'].map(lambda x : x[0])

data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time

data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches

data = data[data['date'] <= datetime.date(2020, 11, 13)]`
```

Request and parse the SpaceX launch data using the GET request

Normalize the json

Getting the info about launches using IDS

Scrapping each column from each part

https://github.com/qalmaqihir/Applied-Data-Science-Capstone/blob/master/jupyter-labs-spacex-data-collection-api.ipynb

Data Wrangling

Data Collection: The first step in the data processing pipeline involved collecting the data from relevant sources, such as APIs, databases, or external files. This step ensured that the necessary data was obtained for further processing and analysis.
Data Inspection: Once the data was collected, it was inspected to gain a preliminary understanding of its structure, content, and quality. This involved examining the data's dimensions, variable types, and identifying any missing or inconsistent values.
Data Cleaning: The data cleaning process aimed to address missing values, outliers, and inconsistencies in the dataset. Techniques such as imputation, removal of duplicates, and handling outliers were employed to ensure data quality and integrity.
Data Transformation: Data transformation involved converting variables into appropriate formats, scaling numerical features, and encoding categorical variables. This step ensured that the data was in a suitable format for analysis and modeling.
Feature Engineering: Feature engineering was performed to create new features or derive additional meaningful insights from the existing data. This step included techniques such as creating interaction terms, polynomial features, or applying domain-specific transformations.
Data Integration: In some cases, data integration was required to merge or combine multiple datasets into a single unified dataset. This step involved matching and aligning data based on common variables or keys.
Data Reshaping: Data reshaping was performed to restructure the dataset, particularly when dealing with data in wide or long formats. Techniques such as pivoting, melting, or stacking were applied to reshape the data to better suit the analysis requirements.
Data Splitting: The dataset was split into training and testing subsets to facilitate model training

https://github.com/qalmaqihir/Applied-Data-Science-Capstone/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb

https://github.com/qalmaqihir/Applied-Data-Science-Capstone/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_1_E3_labs-jupyter-spacex-data_wra

EDA with Data Visualization

Histograms: Histograms were plotted to visualize the distribution of numerical variables. They provide insights into the central tendency, spread, and shape of the data. Histograms help identify skewed distributions, outliers, and potential data issues.
Box Plots: Box plots were used to display the distribution of numerical variables and detect potential outliers. They provide information about the median, quartiles, and the presence of extreme values. Box plots are particularly useful for comparing variables across different groups or categories.
Scatter Plots: Scatter plots were employed to explore relationships between two numerical variables. They allow visual examination of the correlation or association between variables and help identify any patterns or trends. Scatter plots are useful for identifying potential linear or non-linear relationships.
Bar Charts: Bar charts were used to represent categorical variables or to compare frequencies or proportions across different categories. They provide a visual representation of the distribution of categorical data and are effective for highlighting differences or similarities between groups.
Heatmaps: Heatmaps were used to visualize the correlation matrix between variables. They provide a color-coded representation of the strength and direction of the relationships between variables. Heatmaps help identify highly correlated variables and can guide feature selection or dimensionality reduction.
Pie Charts: Pie charts were utilized to represent proportions or percentages of different categories within a variable. They provide a visual summary of categorical data and help understand the relative distribution or composition of the data.
Line Charts: Line charts were employed to plot trends or patterns over time or a continuous variable. They enable the visualization of temporal or sequential changes and help identify patterns, seasonality, or trends in the data.

https://github.com/qalmaqihir/Applied-Data-Science-Capstone/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL

SELECT statement: Used to retrieve specific columns or variables from a table.
WHERE clause: Used to filter data based on specified conditions.
GROUP BY clause: Used to group data by one or more columns for aggregation purposes.
COUNT() function: Used to count the number of rows or occurrences.
SUM() function: Used to calculate the sum of a numerical column.
AVG() function: Used to calculate the average value of a numerical column.
MAX() and MIN() functions: Used to find the maximum and minimum values of a column, respectively.
ORDER BY clause: Used to sort the result set in ascending or descending order based on specified columns.
LIMIT clause: Used to restrict the number of rows returned in the result set.

https://github.com/qalmaqihir/Applied-Data-Science-Capstone/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Markers: Markers were added to the Folium map to indicate specific locations, such as launch sites or points of interest. Each marker represents a geographical point with a unique latitude and longitude coordinate. Markers are useful for visualizing the spatial distribution of data or highlighting important locations on the map.
Circles: Circles were used to represent areas of interest or influence around a specific location. They are defined by a center point (latitude and longitude) and a radius. Circles can help visualize the extent or coverage of certain events or phenomena, such as the range of a rocket's landing zone or the proximity of certain locations to a launch site.
Lines: Lines were utilized to illustrate connections or paths between different locations. They are created by connecting two or more latitude-longitude points. Lines can be used to show trajectories, flight paths, or routes, providing a visual representation of movement or spatial relationships between locations.
Polygons: Polygons were added to outline specific areas or regions of interest. They are defined by a series of latitude-longitude points that form a closed shape. Polygons can be used to highlight boundaries, geographical regions, or areas of significance on the map.

https://github.com/qalmaqihir/Applied-Data-Science-Capstone/blob/master/lab_jupyter_launch_site_location%20(1).ipynb
https://github.com/qalmaqihir/Applied-Data-Science-Capstone/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_locatio
lite.ipynb

Build a Dashboard with Plotly Dash

Bar Charts: Bar charts were used to visualize categorical data or compare frequencies/proportions across different categories. They provide a clear visual representation of data distribution and are effective for highlighting differences or similarities between groups. Bar charts help users quickly identify patterns or trends in the data.
Line Charts: Line charts were employed to plot trends or patterns over time or a continuous variable. They allow users to visualize temporal or sequential changes and help identify patterns, seasonality, or trends in the data. Line charts enable users to understand the evolution of data and make informed decisions based on historical trends.
Scatter Plots: Scatter plots were utilized to explore relationships between two numerical variables. They enable users to visually examine the correlation or association between variables and help identify any patterns or trends. Scatter plots are particularly useful for identifying potential linear or non-linear relationships and outliers.
Heatmaps: Heatmaps were added to visualize the correlation matrix between variables. They provide a color-coded representation of the strength and direction of the relationships between variables. Heatmaps help users identify highly correlated variables and guide feature selection or dimensionality reduction decisions.
Dropdown Menus: Dropdown menus were included to provide users with interactive selection options. Users can choose specific variables or categories to view data subsets or compare different scenarios. Dropdown menus enhance the dashboard's interactivity and allow users to customize their data exploration based on their specific interests or requirements.
Slider Controls: Slider controls were added to enable users to dynamically adjust specific parameters or filter data based on a range of values. Sliders enhance the dashboard's flexibility and allow users to explore different aspects of the data by adjusting thresholds or ranges.
Data Table: A data table was incorporated to display the underlying data in a tabular format. It provides users with detailed information and enables them to access specific data points or examine individual records. The data table enhances the dashboard's transparency and allows users to explore the data at a granular level.

https://github.com/qalmaqihir/Applied-Data-Science-Capstone/blob/master/SpaceX_dash_app.py

Predictive Analysis (Classification)

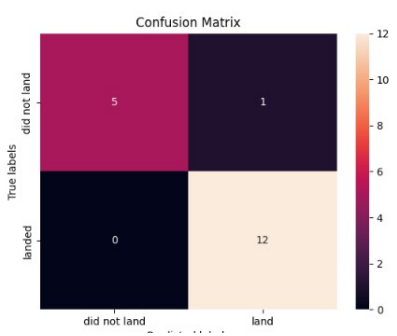
Model Selection: Choose a set of classification algorithms suitable for your problem. This may include algorithms like logistic regression, decision trees, random forests, support vector machines (SVM), or neural networks.
Initial Model Building: Build an initial model using one of the selected algorithms. Train the model on the training dataset and evaluate its performance on the testing dataset using appropriate evaluation metrics such as accuracy, precision, recall, or F1-score.
Model Evaluation: Assess the performance of the initial model and analyze the evaluation metrics. Identify any areas of improvement or potential issues, such as overfitting or underfitting.
Model Improvement: To improve the model, consider the following steps:
Feature Engineering: Analyze the impact of different features and consider adding, removing, or transforming features to enhance model performance.
Hyperparameter Tuning: Adjust the model's hyperparameters to find the optimal configuration. This can be done using techniques like grid search or randomized search, evaluating different combinations of hyperparameters.
Cross-Validation: Apply cross-validation techniques, such as k-fold cross-validation, to obtain a more robust estimate of the model's performance.
Iterative Model Building: Iterate the model development process by fine-tuning the model, evaluating its performance, and making necessary adjustments based on the evaluation results.
Model Selection: Compare the performance of different models based on evaluation metrics. Select the best performing model as the final model for your classification task.
Model Deployment: Once you have the best performing classification model, deploy it in a production environment for practical use. Ensure it is integrated into the target system and meets any performance or security requirements.

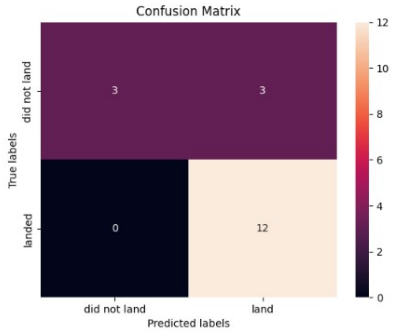
https://github.com/qalmaqihir/Applied-Data-Science-Capstone/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Predictio
https://github.com/qalmaqihir/Applied-Data-Science-Capstone/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

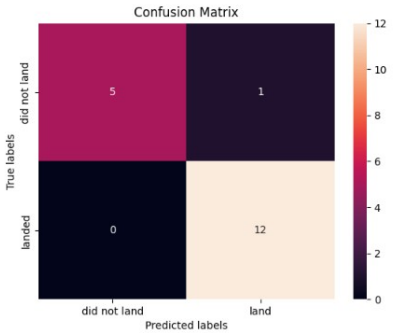
Results

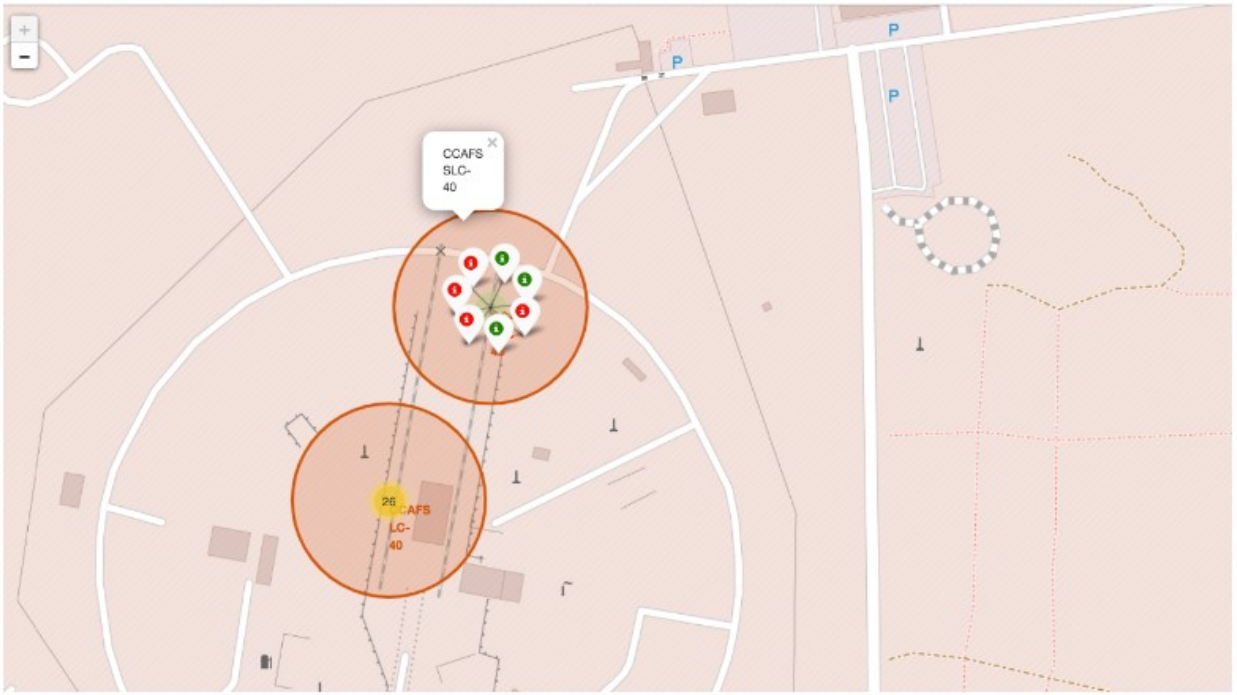
- Exploratory data analysis results
- Interactive analytics demo in screenshots

- Predictive analysis results









Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

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

- Show the screenshot of the scatter plot with explanations

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations

Success Rate vs. Orbit Type

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

- Show the screenshot of the scatter plot with explanations

Flight Number vs. Orbit Type

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

- Show the screenshot of the scatter plot with explanations

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations

Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations

All Launch Site Names

Find the names of the unique launch sites

Present your query result with a short explanation here

Launch Site Names Begin with 'CCA'

Find 5 records where launch sites begin with `CCA`

Present your query result with a short explanation here

Total Payload Mass

Calculate the total payload carried by boosters from NASA

Present your query result with a short explanation here

Average Payload Mass by F9 v1.1

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

Present your query result with a short explanation here

First Successful Ground Landing Date

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

Present your query result with a short explanation here

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

Present your query result with a short explanation here

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes

Present your query result with a short explanation here

Boosters Carried Maximum Payload

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

Present your query result with a short explanation here

2015 Launch Records

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

Present your query result with a short explanation here

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

Present your query result with a short explanation here

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

Replace <Folium map screenshot 1> title with an appropriate title

Explore the generated folium map and make a proper screenshot to include all launch sites’ location markers on a global map

Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

Replace <Folium map screenshot 2> title with an appropriate title

Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

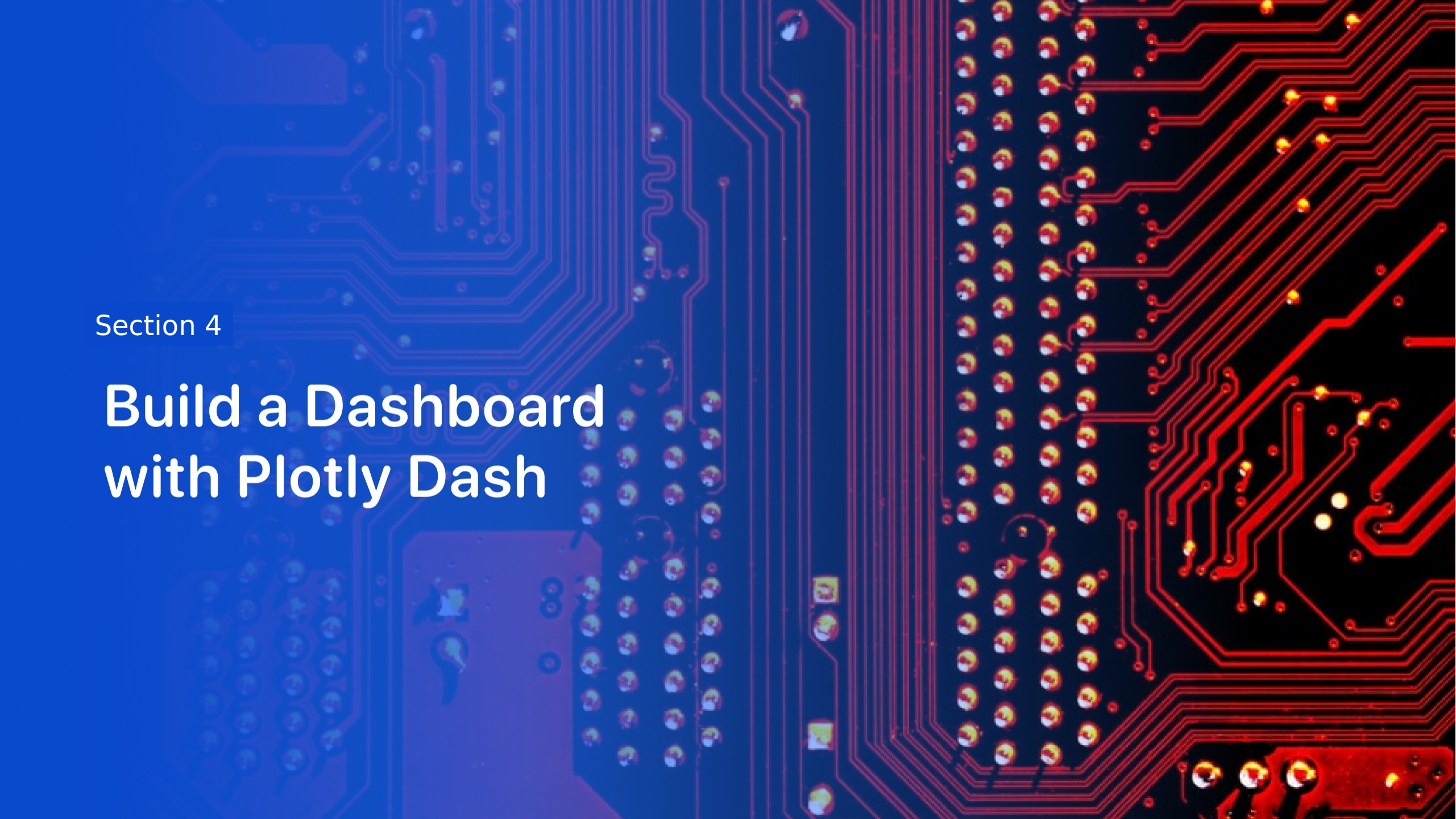
Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

Replace <Folium map screenshot 3> title with an appropriate title

Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

Explain the important elements and findings on the screenshot


Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

Replace <Dashboard screenshot 1> title with an appropriate title

Show the screenshot of launch success count for all sites, in a piechart

Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

Replace <Dashboard screenshot 2> title with an appropriate title

Show the screenshot of the piechart for the launch site with highest launch success ratio

Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

Replace <Dashboard screenshot 3> title with an appropriate title

Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

Conclusions

Point 1

Point 2

Point 3

Point 4

...

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!