IBM APPLIED DATA SCIENCE PROJECT

Richard Ho

EXECUTIVE SUMMARY

Objective: Predict Falcon 9 first stage landing success to assess SpaceX's competitive advantage.

Methods:

- Data collection via SpaceX API and Wikipedia web scraping
- Data wrangling with Pandas pipelinesExploratory Data Analysis (EDA) with Python visualizations and SQL queries
- Interactive dashboards using Folium and Plotly
- DashPredictive modeling with Logistic Regression, SVM, KNN, and Decision Trees

Key Results:

- Logistic Regression provided the most balanced accuracy and interpretability
- Launch site and payload mass strongly influence landing success

Insight:Predictive analytics can guide cost-saving strategies and market positioning in commercial space launches.

INTRODUCTION

Background

- SpaceX revolutionized space travel by reusing Falcon 9 first stages.
- Reusability drastically reduces launch costs, making SpaceX highly competitive.

Problem Statement

- Can we predict whether the Falcon 9 first stage will successfully land?
- Accurate predictions help stakeholders assess SpaceX’s reliability and cost advantage.

Business Context

- Commercial launch providers compete on cost and reliability.
- Predictive insights into landing success can influence investment decisions and market positioning.

Project Scope

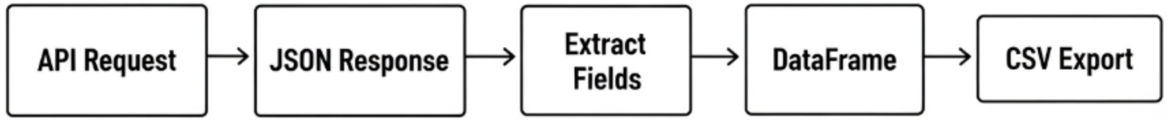
- Collect launch data via SpaceX API and Wikipedia scraping.Clean and wrangle data for analysis.
- Perform EDA with Python visualizations and SQL queries.Build interactive dashboards (Folium, Plotly Dash).
- Apply machine learning models to predict landing success.

DATA COLLECTION (SPACEX API)

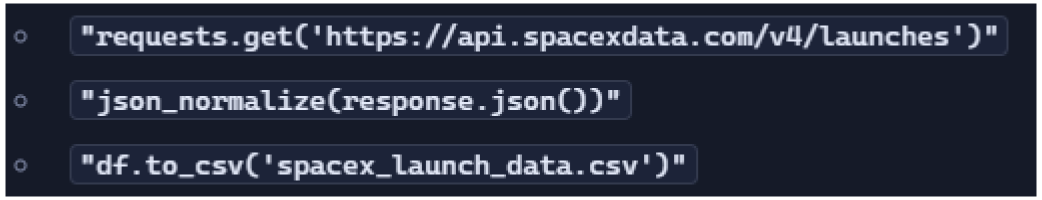
Show how launch data was collected via API calls, using a clear process visualization with SpaceX REST API (https://api.spacexdata.com/v4/launches).

Issues

1. Send API Request → requests.get() call to SpaceX API endpoint
2. Parse JSON Response → Convert to Python dictionary
3. Normalize Data → Use json_normalize() to flatten nested fields
4. Extract Key Fields → Launch site, payload mass, orbit, success/failure outcome
5. Store Data → Save cleaned dataset into Pandas
6. DataFrameExport → Write to CSV for downstream wrangling and EDA



Notable Keyphrases



- "requests.get('https://api.spacexdata.com/v4/launches')"
- "json_normalize(response.json())"
- "df.to_csv('spacex_launch_data.csv')"

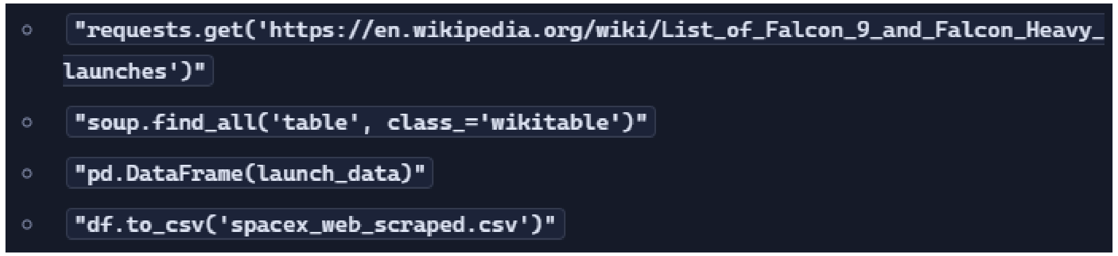
GitHub Reference:
https://github.com/ritzdeee/IBM-Applied-Data-Science-Capstone-Project/tree/263918aab7e882a1fdb1cc9ba68e7e02be1e67ba/1.%20Data%20Collection

WEB SCRAPING WITH BEAUTIFULSOUP (BS4)

Purpose: Explain how additional launch data was scraped from Wikipedia.

- **Data Source:** Wikipedia page on Falcon 9 launches.
- **Process Overview (Flowchart / Key Steps):**
  1. **Send HTTP Request** → requests.get() to Wikipedia launch list page
  2. **Parse HTML** → Use BeautifulSoup to extract table elements
  3. **Locate Launch Table** → Identify <table> containing launch records
  4. **Extract Rows** → Iterate through <tr> elements for payload, date, site, outcome
  5. **Clean Data** → Strip whitespace, handle missing values
  6. **Store Data** → Convert into Pandas DataFrame
  7. **Export** → Save to CSV for wrangling and EDA

Key Phrases from Notebook:



- `"requests.get('https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches')"`
- `"soup.find_all('table', class_='wikitable')"`
- `"pd.DataFrame(launch_data)"`
- `"df.to_csv('spacex_web_scraped.csv')"`

GitHub Reference:
https://github.com/ritzdeee/IBM-Applied-Data-Science-Capstone-Project/tree/263918aab7e882a1fdb1cc9ba68e7e02be1e67ba/1.%20Data%20Collection

DATA WRANGLING METHODOLOGY

Showing how raw data from API and web scraping was cleaned, transformed, and prepared for analysis.

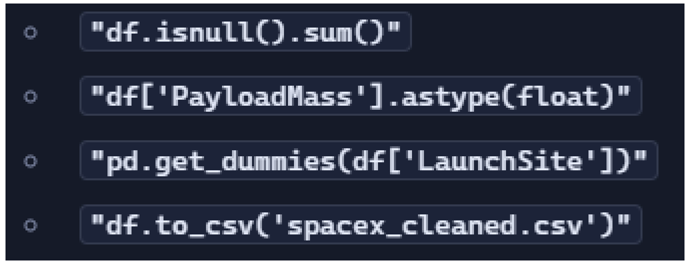
Issues

1. Missing values (NaNs) in payload mass and outcome fields
2. Inconsistent data types (strings vs. numeric)
3. Nested JSON structures requiring flattening
4. Duplicate records across API and scraped datasets

Process Overview (Flowchart / Key Steps)

1. Load Data → pd.read_csv() from API and scraped
2. CSVsHandle Missing Values → df.fillna() and df.dropna() for critical fields
3. Convert Data Types → Explicit astype(float) for payload mass
4. Normalize Categorical Fields → Encode launch site and orbit using pd.get_dummies()
5. Merge Datasets → Combine API and scraped data with pd.merge()
6. Final Dataset → Cleaned DataFrame ready for EDA and modeling

Notable Keyphrases



- `"df.isnull().sum()"`
- `"df['PayloadMass'].astype(float)"`
- `"pd.get_dummies(df['LaunchSite'])"`
- `"df.to_csv('spacex_cleaned.csv')"`

GitHub Reference:
https://github.com/ritzdeee/IBM-Applied-Data-Science-Capstone-Project/tree/263918aab7e882a1fdb1cc9ba68e7e02be1e67ba/2.%20Data%20Wrangling

EDA WITH SQL

Use SQL queries to explore launch data and uncover patterns.

Queries Performed

1. *Launch Site Analysis* → Count launches per site
2. *Payload Distribution* → Average payload mass by orbit
3. *Success Rates* → Percentage of successful landings per site
4. *Rankings* → Identify top-performing launch sites
5. *Timeline Analysis* → Success trends across years

Purpose of SQL Analysis

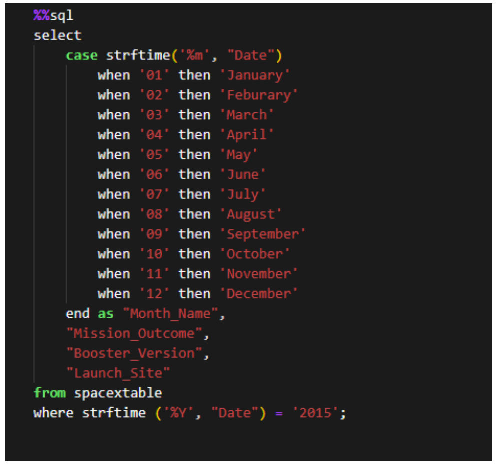
1. Validate findings from Python visualizations with structured queries
2. Provide precise counts, averages, and percentages
3. Enable reproducible, query-driven insights

Key Findings

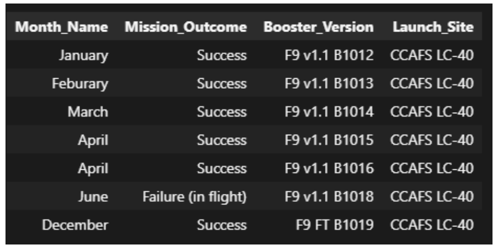
1. KSC LC-39A and CCAFS SLC-40 have higher success frequencies
2. Payloads in certain orbits (e.g., LEO) show better success rates
3. Success rates steadily improved after 2013
4. SQL confirms the same trends observed in Python EDA

Key Phrases from Notebook

1. "SELECT LaunchSite, COUNT(*) FROM spacex GROUP BY LaunchSite"
2. "SELECT Orbit, AVG(PayloadMass) FROM spacex GROUP BY Orbit"
3. "SELECT Year, SUM(Success)/COUNT(*) FROM spacex GROUP BY Year"



```sql
%%sql
select
    case strftime('%m', "Date")
        when '01' then 'January'
        when '02' then 'Feburary'
        when '03' then 'March'
        when '04' then 'April'
        when '05' then 'May'
        when '06' then 'June'
        when '07' then 'July'
        when '08' then 'August'
        when '09' then 'September'
        when '10' then 'October'
        when '11' then 'November'
        when '12' then 'December'
    end as "Month_Name",
    "Mission_Outcome",
    "Booster_Version",
    "Launch_Site"
from spacextable
where strftime ('%Y', "Date") = '2015';
```



Month_Name	Mission_Outcome	Booster_Version	Launch_Site
January	Success	F9 v1.1 B1012	CCAFS LC-40
Feburary	Success	F9 v1.1 B1013	CCAFS LC-40
March	Success	F9 v1.1 B1014	CCAFS LC-40
April	Success	F9 v1.1 B1015	CCAFS LC-40
April	Success	F9 v1.1 B1016	CCAFS LC-40
June	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
December	Success	F9 FT B1019	CCAFS LC-40

GitHub Reference:

https://github.com/ritzdeee/IBM-Applied-Data-Science-Capstone-Project/tree/263918aab7e882a1fdb1cc9ba68e7e02be1e67ba/3.%20EDA%20with%20SQL

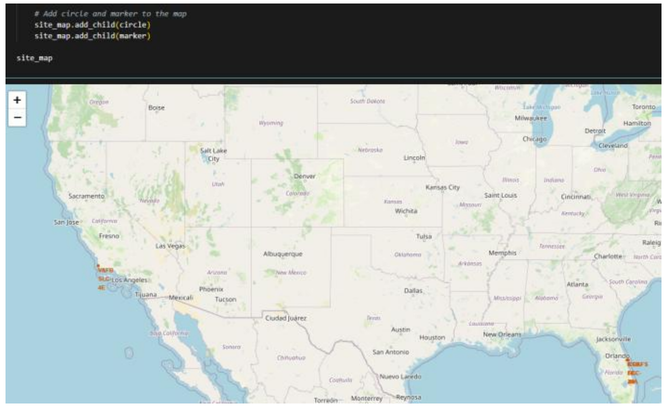
INTERACTIVE VISUAL ANALYTICS

Folium Map

- Plots launch site markers on a world map
- Displays launch records with hover tooltips (site name, payload, outcome)
- Proximity analysis: distance between launch sites and nearby facilities

Plotly Dash App

- Success Pie Charts → Distribution of successful vs. failed landings by site
- Payload vs. Outcome Scatter Plots → Interactive filtering by payload range and site
- Dropdown filters for orbit type, launch site, and payload massReal-time updates of charts based on user selections



```
    # Add circle and marker to the map
    site_map.add_child(circle)
    site_map.add_child(marker)

site_map
```

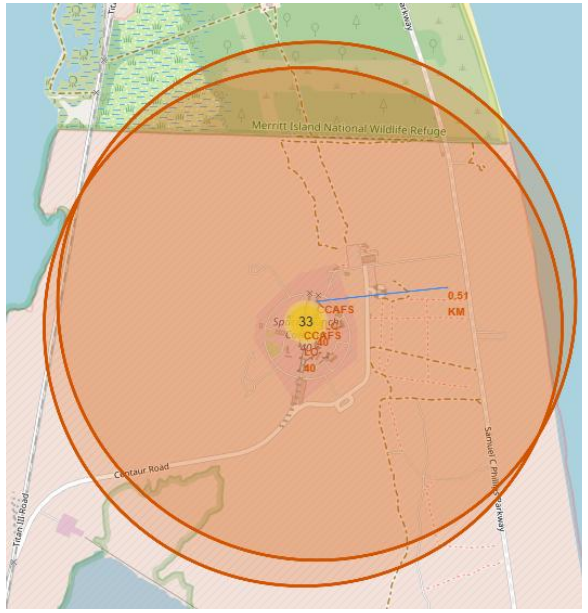
GitHub Reference:
https://github.com/ritzdeee/IBM-Applied-Data-Science-Capstone-Project/tree/263918aab7e882a1fdb1cc9ba68e7e02be1e67ba/5.%20Interactive%20Visual%20Dashboarding

Purpose of Interactive Tools

- Enable stakeholders to explore data dynamically
- Provide intuitive visual insights into launch success patterns
- Support decision-making with flexible, user-driven analysis

Key Phrases from Notebook

1. "folium.Marker(location=[lat, lon], popup=site)«
2. "dcc.Dropdown(options=[...], multi=True)"
3. "px.scatter(df, x='PayloadMass', y='Outcome', color='LaunchSite')"


Merritt Island National Wildlife Refuge
0.51 KM
33
CCAFS
CCAFS
40
Centaur Road
Samuel C Phillips Parkway
Titan III Road

EDA VISUALIZATION RESULTS

VISUALIZATING OUTPUTS FROM EXPLORATORY ANALYSIS

Scatter Plots

1. *Payload mass vs. landing outcome*
2. *Shows correlation: lighter payloads → higher success probability*

Bar Charts

1. Success rates by launch site
2. Highlights site performance differences (KSC LC-39A consistently strong)

Yearly Trends (Line/Bar Charts)

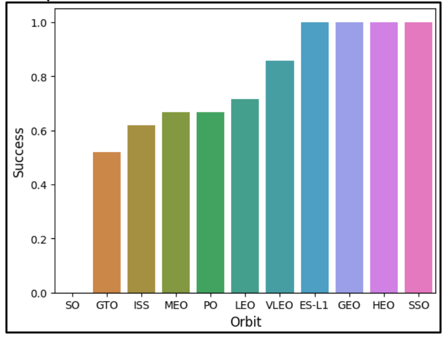
1. Success rate progression across years
2. Clear upward trend in reliability since 2013

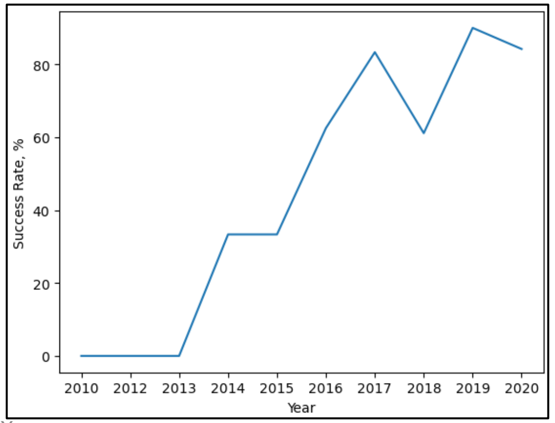
Key Findings

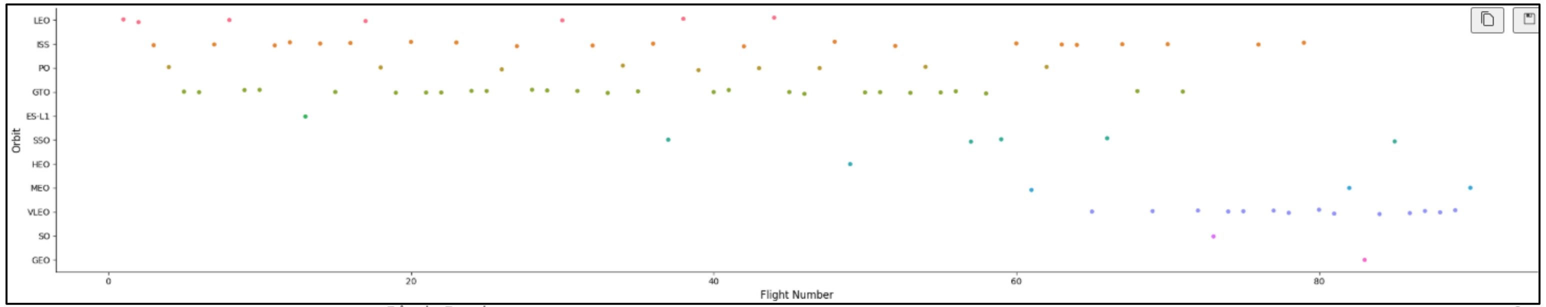
1. Payload mass and launch site are critical predictors of success
2. SpaceX's success rate has steadily improved over time
3. Visual evidence supports the hypothesis that reusability drives reliability

Key Phrases from Notebook

1. "plt.scatter(df['PayloadMass'], df['Outcome'])"
2. "df.groupby('LaunchSite')['Outcome'].value_counts().plot(kind='bar')"
3. "df['Year'].value_counts().plot(kind='line')"







EDA WITH SQL RESULTS

Launch Site Analysis

Query: SELECT LaunchSite, COUNT(*) FROM spacex GROUP BY LaunchSite
Result: KSC LC-39A and CCAFS SLC-40 show the highest number of launches

Payload Distribution

Query: SELECT Orbit, AVG(PayloadMass) FROM spacex GROUP BY Orbit
Result: LEO payloads average lower mass, correlating with higher success rates.

Success Rates by Site

Query: SELECT LaunchSite, SUM(Success)/COUNT(*) FROM spacex GROUP BY LaunchSite
Result: KSC LC-39A has the highest success percentage.

Rankings

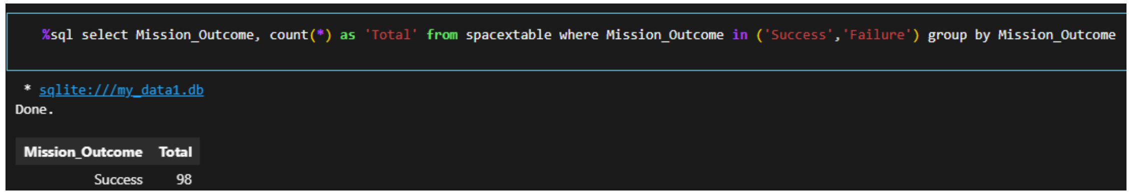
Query: SELECT LaunchSite, COUNT(Success) AS SuccessfulLandings FROM spacex GROUP BY LaunchSite ORDER BY SuccessfulLandings DESC
Result: Ranked list confirms LC-39A as top-performing site.

Timeline Analysis

Query: SELECT Year, SUM(Success)/COUNT(*) FROM spacex GROUP BY Year
Result: Success rates steadily improved after 2013, reaching >90% in recent years.

Key Findings

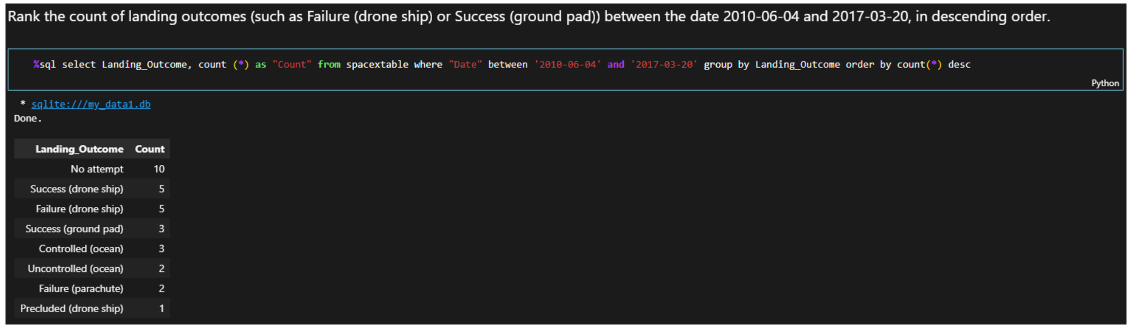
1. SQL confirms payload mass and launch site as critical predictors.
2. Structured queries validate trends observed in Python visualizations.
3. Timeline analysis highlights SpaceX's growing reliability.



```
%sql select Mission_Outcome, count(*) as 'Total' from spacextable where Mission_Outcome in ('Success','Failure') group by Mission_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	Total
Success	98



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.

```
%sql select Landing_Outcome, count (*) as "Count" from spacextable where "Date" between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by count(*) desc
```

Python

```
 * sqlite:///my_data1.db
Done.
```

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

FOLIUM MAP RESULTS

Launch Site Markers

- Each launch site plotted on a Folium map with latitude/longitude coordinates
- Popups display site name, payload, and launch outcome

Launch Records

- Interactive markers allow exploration of individual launch details.
- Provides spatial perspective on where launches occur

Proximity Analysis

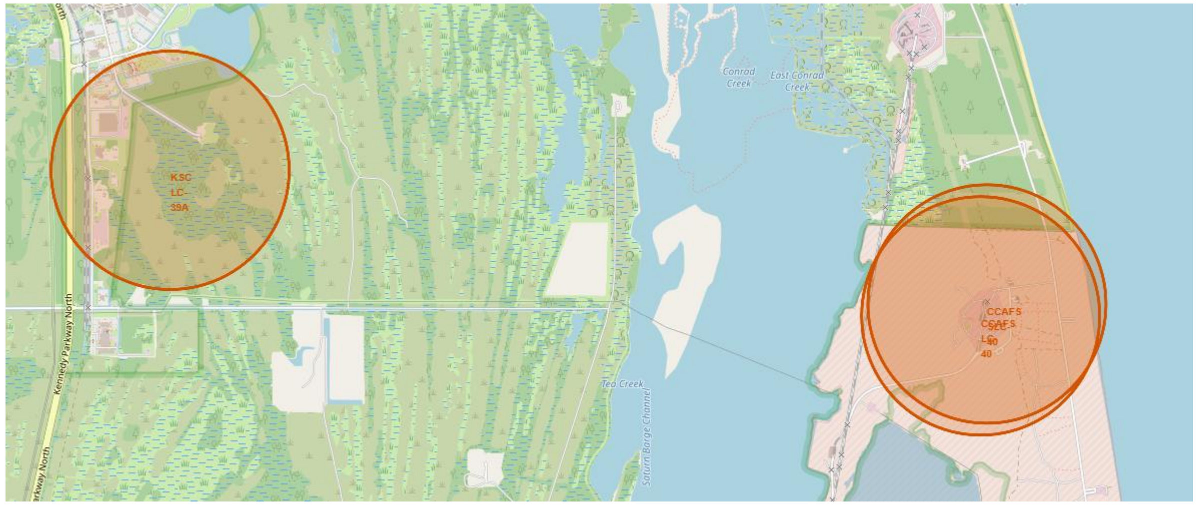
- Calculated distances between launch sites and nearby facilities (e.g., airports, coastlines).
- Helps assess logistical advantages of site locations

Key Phrases from Notebook

- "folium.Map(location=[28.5, -80.5], zoom_start=5)“
- "folium.Marker([lat, lon], popup=site_name)“
- "folium.CircleMarker(..., radius=5, color='blue')""geopy.distance.distance(site, facility).km“

Key Findings

- Launch sites are strategically located near coastlines for safety.
- Proximity to infrastructure supports operational efficiency.
- Folium maps provide intuitive geographic insights beyond tabular data.



PREDICTIVE ANALYSIS RESULTS

Models Applied

- Logistic Regression
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Decision Tree

Evaluation Metrics

- Accuracy scores across models
- Confusion matrices for classification performance
- Precision, recall, and F1-score comparisons

Results

- Logistic Regression achieved balanced accuracy and interpretability
- SVM performed well but required more tuning
- KNN showed moderate accuracy, sensitive to parameter choice
- Decision Tree provided clear rules but risked overfitting

Best Model

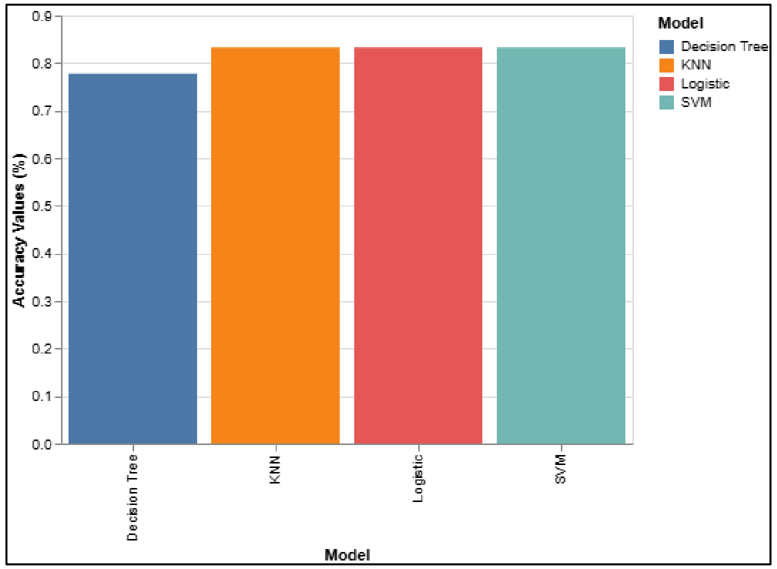
- Logistic Regression selected as optimal model
- Offers strong predictive accuracy with explainable coefficients
- Confirms payload mass and launch site as key predictors

Conclusion

- Predictive modeling validates EDA findings: payload and site drive success probability
- Models demonstrate SpaceX’s increasing reliability over time
- Insights can guide competitive strategy and cost-saving decisions in commercial launches

Key Phrases from Notebook

- "LogisticRegression().fit(X_train, y_train)“
- "confusion_matrix(y_test, y_pred)“
- "classification_report(y_test, y_pred)“
- "classification_report(y_test, y_pred)"



CONCLUSION & INSIGHTS

Project Outcomes

- Successfully collected and integrated SpaceX launch data via API and web scraping.
- Cleaned and wrangled data into a reliable dataset.
- Conducted EDA with Python visualizations and SQL queries to uncover trends.
- Built interactive dashboards (Folium, Plotly Dash) for dynamic exploration.
- Applied multiple machine learning models to predict landing success.

Key Results

- Logistic Regression identified as the best-performing model.
- Payload mass and launch site are the strongest predictors of success.
- Success rates have steadily improved since 2013, confirming SpaceX's growing reliability.

Creative & Innovative Insights

- Combining geospatial analysis (Folium) with predictive modeling provides a holistic view of launch success.
- Interactive dashboards empower stakeholders to explore data intuitively.
- Predictive analytics can guide strategic decisions in commercial spaceflight, reducing costs and improving competitiveness.

Final Takeaway

- Data-driven insights confirm SpaceX's reusability strategy is effective.
- Predictive modeling offers a powerful tool for forecasting future launch outcomes.
- The project demonstrates how applied data science can bridge technical analysis with real-world business impact.

Summary
API
SQL
Business Insight
Dashboards
ML Models

THANK YOU

Richard Ho