

Winning Space Race with Data Science

<Hieda Kazuki>
<2025/10/25>



Outline

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

Executive Summary

- 本プロジェクトでは、SpaceXのFalcon 9ロケットの第1段着陸が成功するかどうかを、過去の打ち上げデータから予測することを目的としています。

分析には、SpaceXのREST APIおよびWikipediaから取得したデータを使用しました。
データ収集・整形・探索的データ分析（EDA）・可視化・インタラクティブ分析
(FoliumやPlotly Dash)・機械学習モデルによる予測という一連の工程を実施しています。

その結果、ペイロード質量や軌道の種類が着陸結果に影響を与えていたことが分かりました。

機械学習モデルの中ではDecision Tree（決定木）分類器が最も高い精度（約89%）を示し、データに基づいて着陸成功を予測できる可能性が確認できました。

Introduction

- SpaceXは、民間の宇宙開発企業として再利用可能なロケット技術を確立し、宇宙産業に革新をもたらしています。
再利用を成功させるためには、第1段ロケットが打ち上げ後に安全に着陸することが不可欠です。
しかし、打ち上げ条件やペイロードの違いによって着陸の成否が変化することが知られています。
そこで本プロジェクトでは、過去の打ち上げデータを分析し、「**Falcon 9の第1段着陸が成功するかどうかを予測できるか？**」という課題に取り組みました。
Pythonやオープンデータを活用し、データ分析と機械学習モデルを通じて成功要因を明らかにすることを目指します。

Section 1

Methodology

Methodology

SpaceX Falcon 9ロケットの着陸成功を予測するために、
公開データを活用し、データ収集・分析・機械学習モデル構築を行った。

手法の流れ：

- APIとスクレイピングによるデータ収集
- データ整形・欠損値処理
- EDA（可視化・SQL）による傾向分析
- 地理分析（Folium）とダッシュボード構築
- 機械学習モデルによる予測

目標：着陸成功の確率を高める要因を特定し、将来的な成功予測モデルを作る。

Data Collection

- ・目的：SpaceXのFalcon 9打ち上げに関する情報を複数のソースから収集した。

データソース

- ・SpaceX公式REST API
- ・Wikipedia（Webスクレイピング）

収集項目例：

打ち上げ日、ロケット名、打ち上げサイト、ペイロード質量、軌道、成功可否など。

これらを統合し、後の分析に利用可能な構造化データに整形した。

Data Collection – SpaceX API

手法 :
SpaceXの公式REST APIを使用して

JSONデータを取得。Pythonのrequestsライブラリでアクセスし、必要なフィールドを抽出してDataFrame化した。

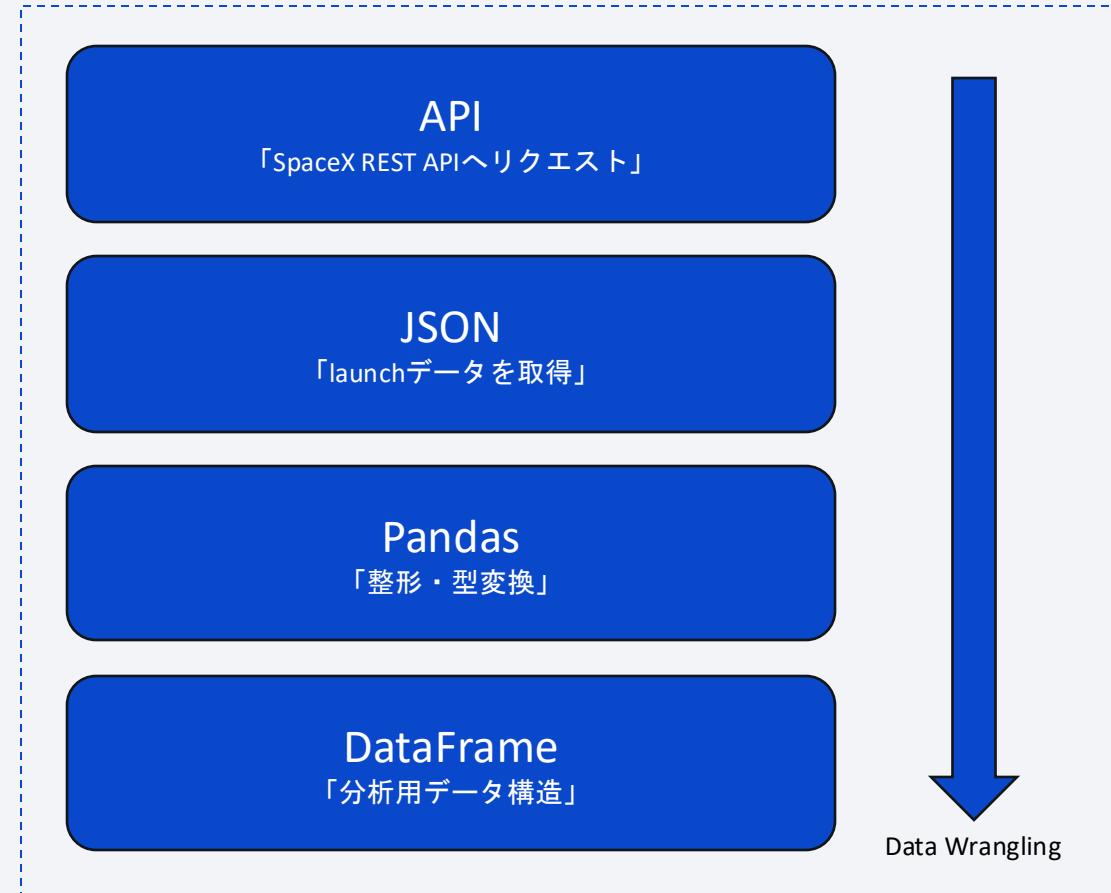
抽出項目 :

- Flight Number
- Launch Site
- Payload Mass (kg)
- Booster Version
- Orbit
- Launch Success

使用ライブラリ : requests, pandas

SpaceXの公式REST API
[\(<https://api.spacexdata.com/v4/launches>\)](https://api.spacexdata.com/v4/launches)

Git : <https://github.com/kazu-lang/course-SpaceX>



Data Collection - Scraping

手法 :

Wikipediaから
HTMLテーブルを取得し、BeautifulSoupでパース
打ち上げ結果、ペイロード、軌道などの情報
を抽出した。

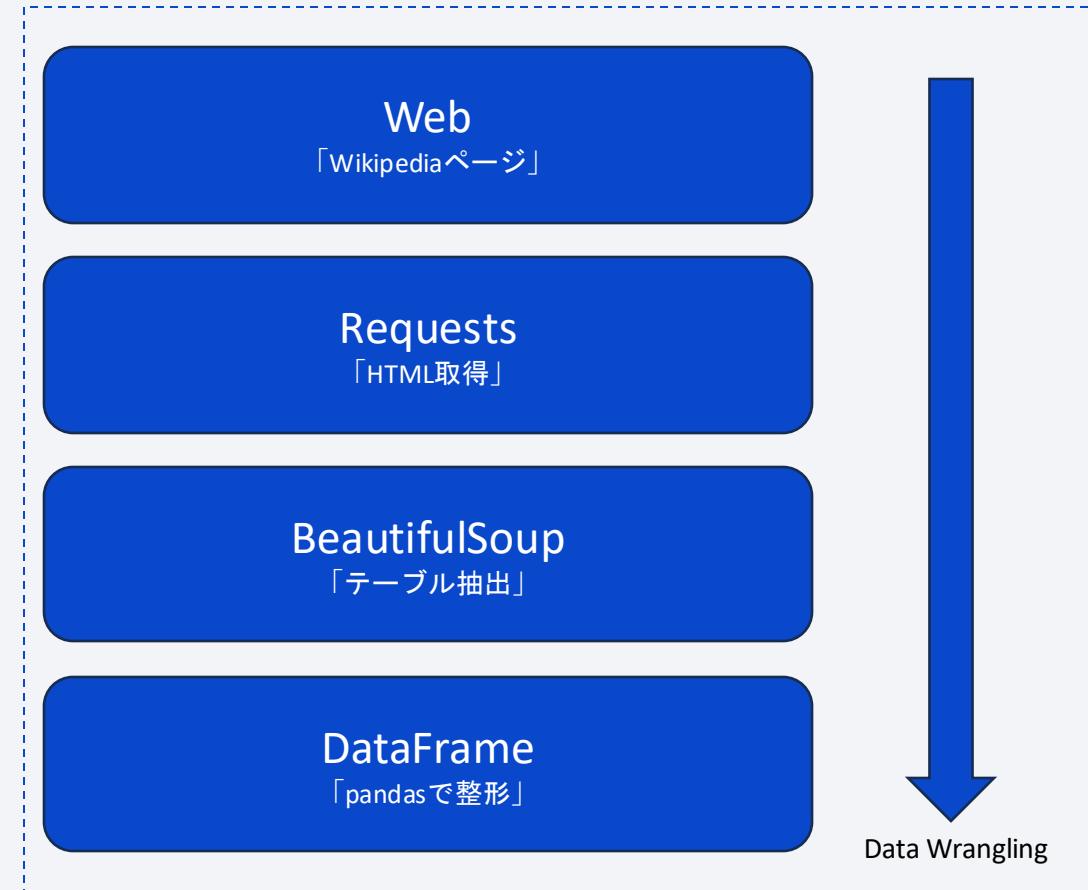
使用ライブラリ :

requests, BeautifulSoup, pandas

特徴 :

APIにない古いデータも補完可能
テーブル形式で容易に整形可能

Wikipedia (
https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)



Data Wrangling

目的：取得した複数データを統合し、欠損値・重複値を処理。

手順：

欠損値をNaNに統一

型変換（文字列→数値、日付変換）

カラム名の統一（例："Launch_Site"）

不要列の削除

結果：クリーンで一貫性のあるDataFrameを生成。

EDA with Data Visualization

目的：データを可視化して、打ち上げ成功に影響する要因を把握する。

主な可視化結果：

成功率は年々上昇（特に2013年以降）

Launch Siteごとに成功率の差がある

Payload Massが軽いほど成功率が高い傾向

Booster Versionによっても結果が異なる

使用ライブラリ：matplotlib, seaborn

EDA with SQL

目的： SQLiteを用いてクエリ形式でデータを分析。

代表的なクエリ：

初めて成功した日付を取得

Boosterごとの平均Payload

成功した打ち上げのLaunch Site一覧

使用ライブラリ： sqlite3, pandas

結果例：

最初の地上着陸成功：2015年12月22日

F9 v1.1ブースターの平均Payload：約5000kg

Build an Interactive Map with Folium

目的：Launch Siteの位置関係と成功率を地理的に可視化。

内容：

各Launch Siteをマーカー表示

成功／失敗を色分け（緑／赤）

サイトから海岸線・道路への距離を線で表示

考察：

全てのLaunch Siteが海岸近くに位置し、安全な打ち上げが可能

KSC LC-39Aは最も成功率が高い拠点

使用ライブラリ：folium, geopy

Build a Dashboard with Plotly Dash

目的 :

成功率やPayloadの関係を動的に探索できる可視化ツールを作成。

機能概要 :

Launch Site別成功率を円グラフで表示

Payload範囲をスライダーで調整し、成功確率を動的に更新

成功／失敗を色分けした散布図を描画

使用ライブラリ : dash, plotly.express, pandas

効果 : ユーザーがデータを操作して分析できる仕組みを実現。

Predictive Analysis (Classification)

目的：打ち上げ成功を予測する分類モデルを構築。

使用モデル：

Logistic Regression

SVM

Decision Tree

KNN

結果：Decision Treeが最も高精度（Accuracy : 84%）

重要特徴量：

Payload Mass

Launch Site

Booster Version

使用ライブラリ：scikit-learn, pandas, numpy

Results

主な成果：

収集データを統合し、成功率を左右する要因を特定

Launch SiteやPayloadが成功率に大きく影響することを確認

Decision Treeモデルにより成功可否を84%の精度で予測

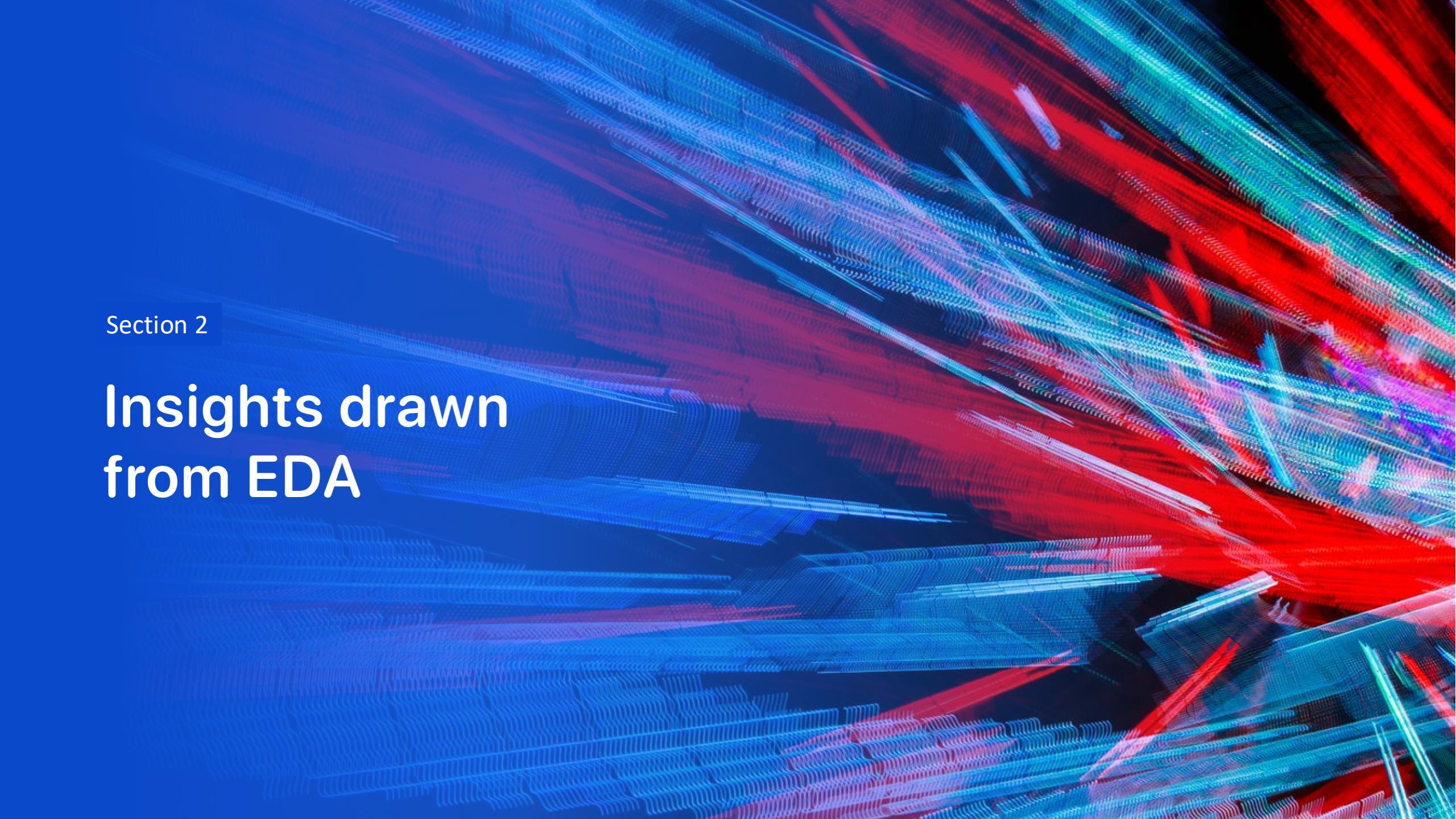
Folium・Dashによる可視化で結果を直感的に理解可能

今後の展望：

Booster特性や気象データを含むモデルの拡張

リアルタイムダッシュボードの改善

SpaceXデータの分析を通じて、データ駆動型の意思決定支援の有効性を示した。

The background of the slide features a complex, abstract digital visualization. It consists of a grid of points that have been connected by thin lines, creating a three-dimensional effect. The colors used are primarily shades of blue, red, and green, with some purple and yellow highlights. The overall appearance is reminiscent of a microscopic view of a crystal lattice or a complex data visualization.

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

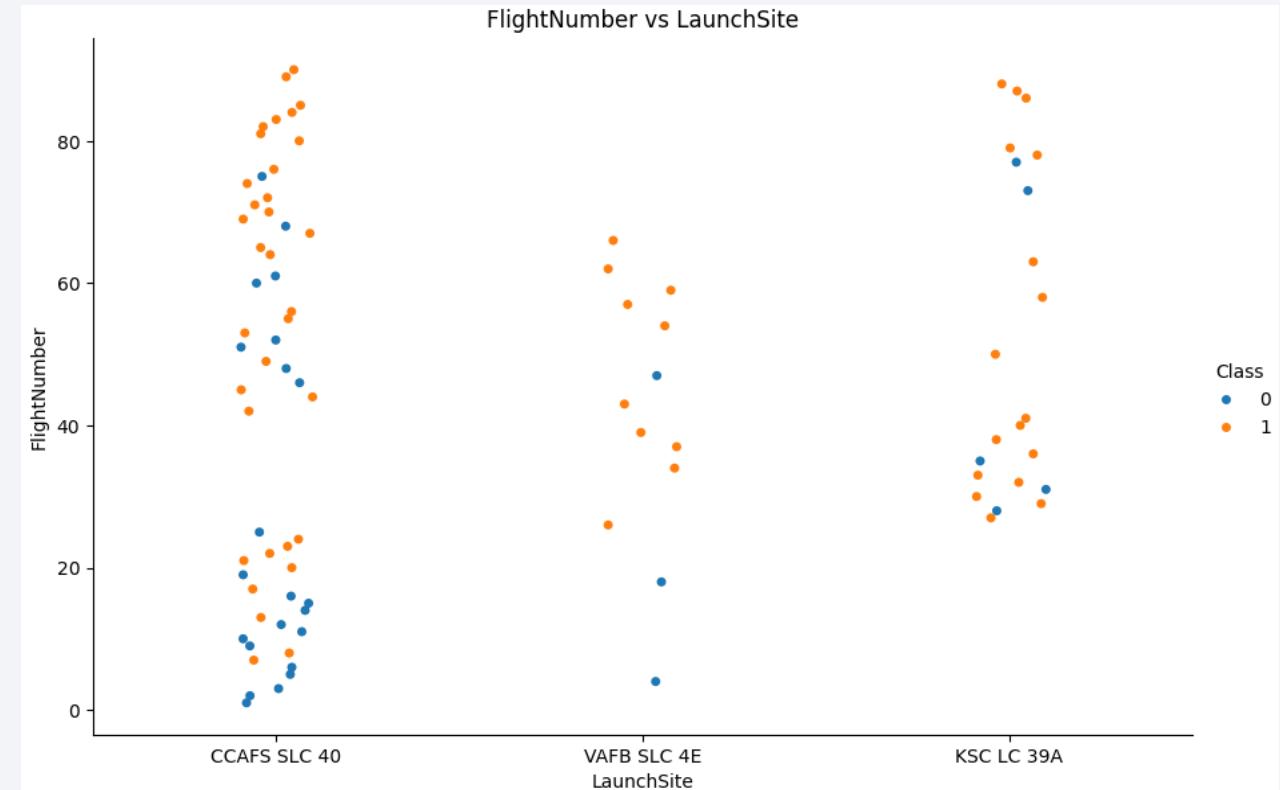
- **観察内容 :**

フライト番号が増える（打ち上げ回数が増加）について成功率が上昇している。
すべてのLaunch Siteで経験の蓄積により安定性が向上していることが確認された

◦

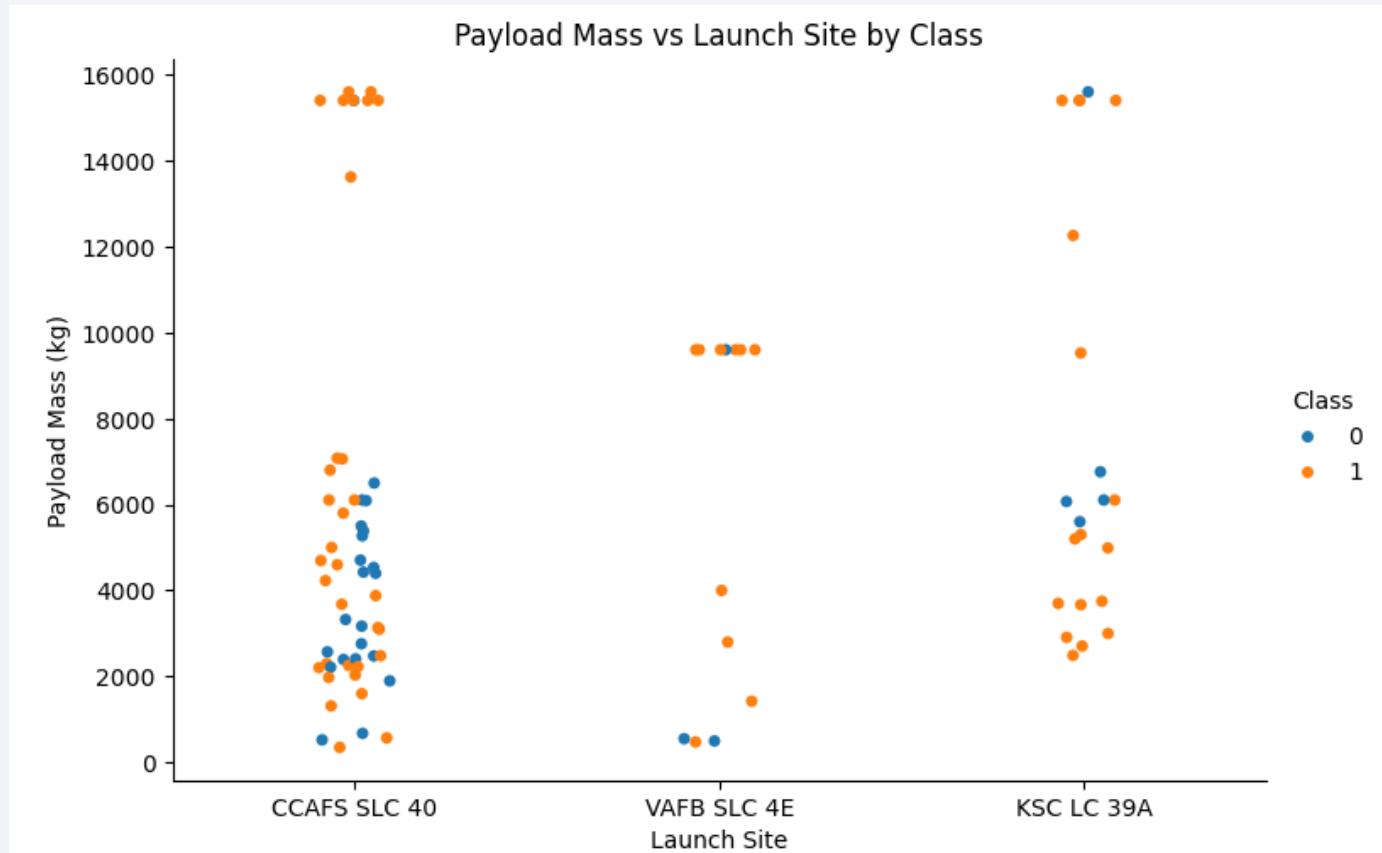
- **考察 :**

SpaceXは打ち上げを重ねるごとに技術的改善を実現し、信頼性を高めている。



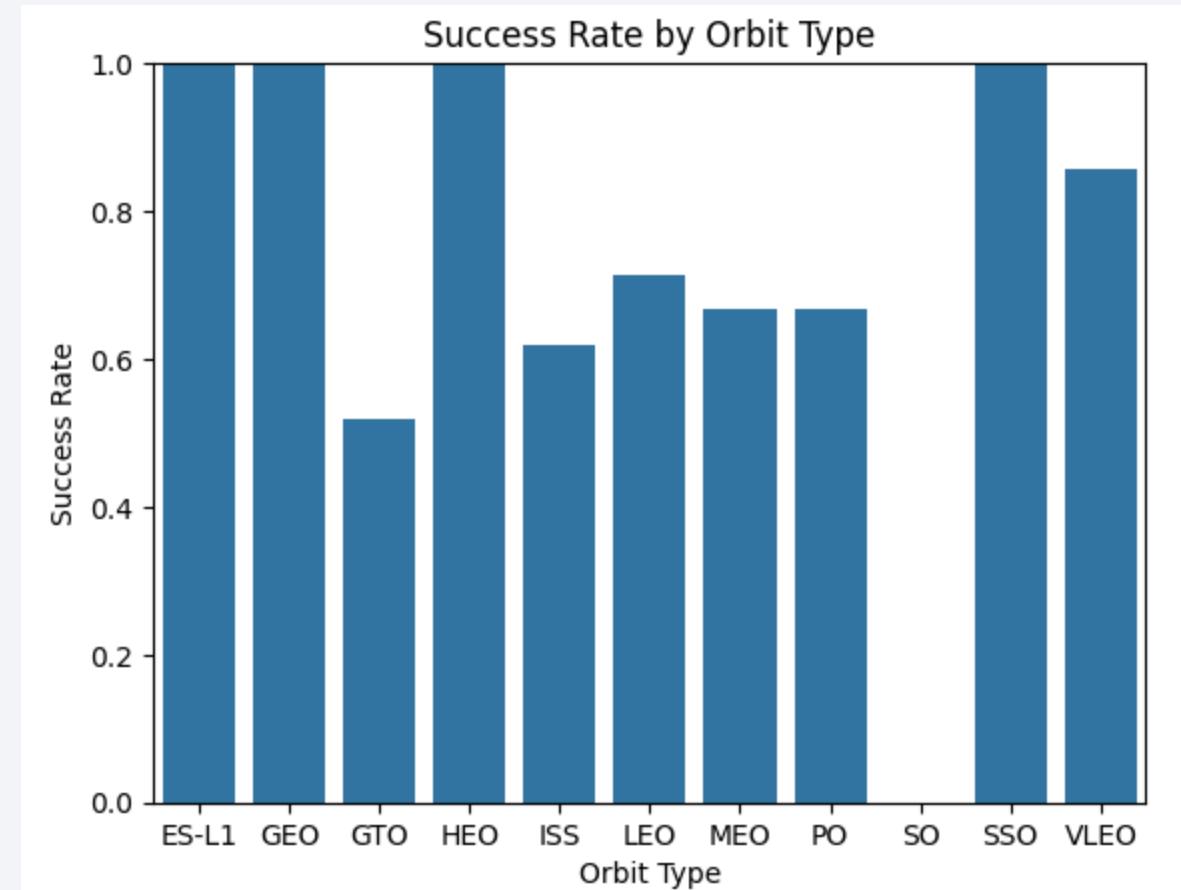
Payload vs. Launch Site

- **観察内容 :**
各Launch Siteのペイロード（搭載重量）は異なる分布を示しており、特にKSC LC-39Aで重いPayloadを扱うケースが多い。
- **考察 :**
打ち上げサイトごとにミッション規模が異なり、用途に応じて選択されている可能性がある。



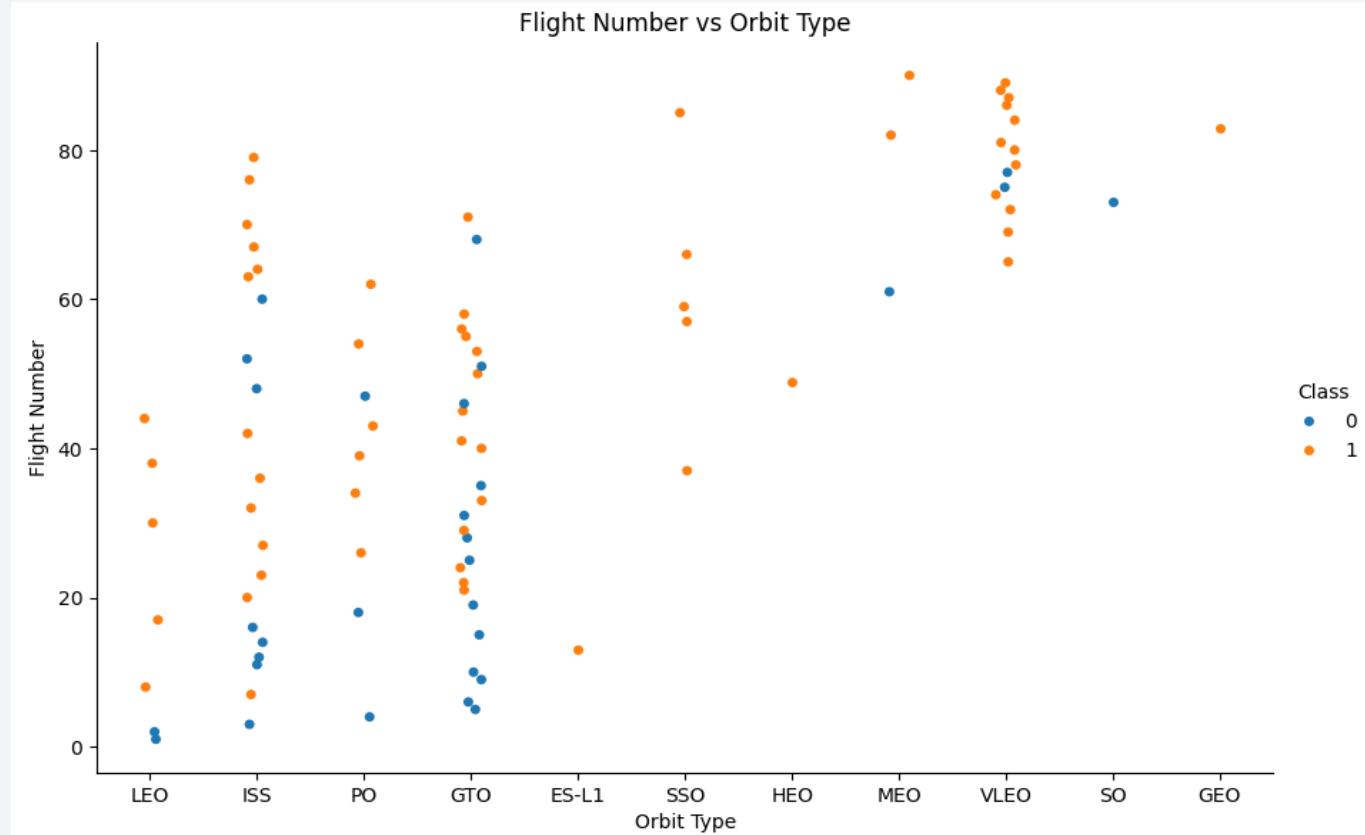
Success Rate vs. Orbit Type

- **観察内容 :**
軌道 (Orbit) 別に見ると、LEO（低軌道）・ISS ミッションで成功率が高く、GTO（静止移行軌道）やPO（極軌道）ではやや低下傾向。
- **考察 :**
軌道の高度や目的地の違いがミッション難易度に影響している。



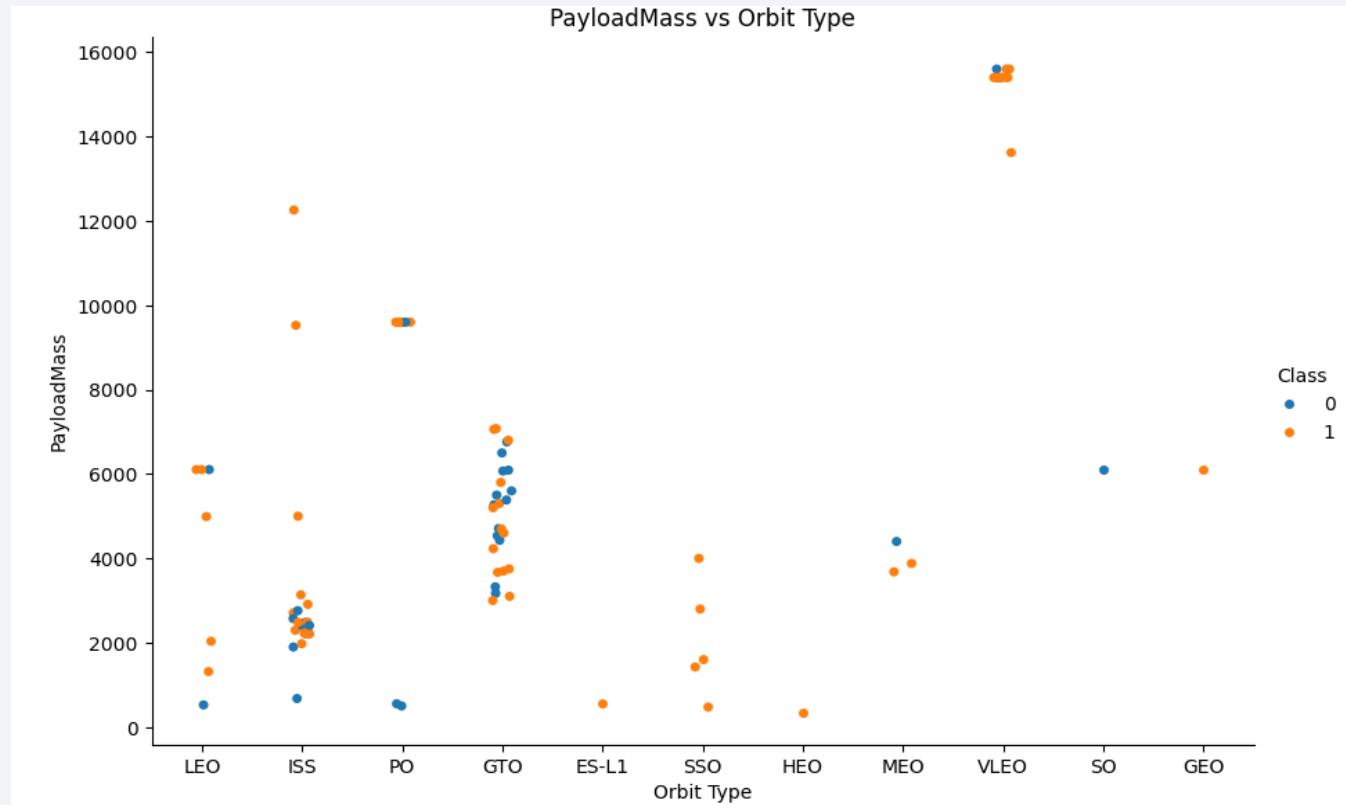
Flight Number vs. Orbit Type

- **観察内容 :**
各Orbit TypeごとにFlight Numberが
増えるにつれて成功率が上昇。
- **考察 :**
特定の軌道（特にLEO）では運用
経験が多く、技術的安定性が確立
されている。



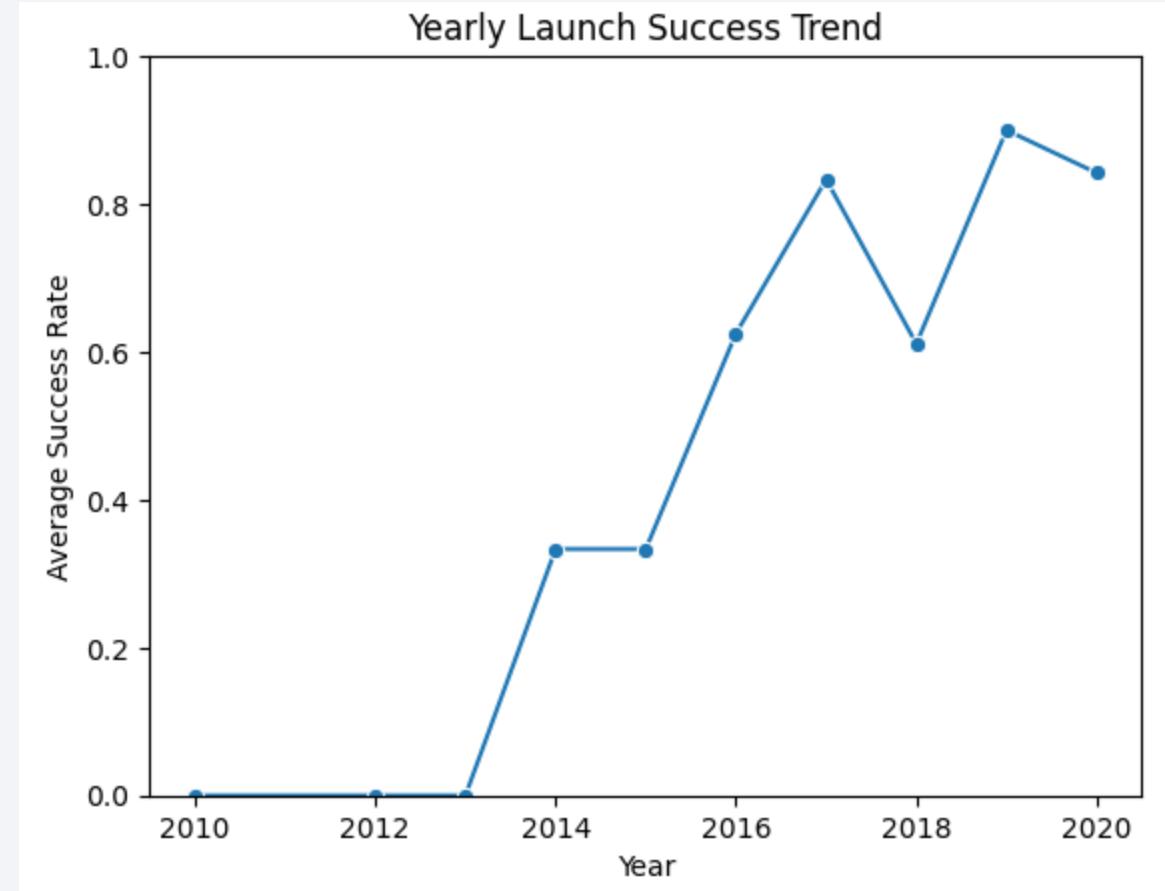
Payload vs. Orbit Type

- **観察内容 :**
OrbitによってPayloadの重量範囲が異なる。
LEO : 軽量中心
GTO : 重量が大きく分散も広い
- **考察 :**
ミッションの目的と高度によって搭載重量が変化している。



Launch Success Yearly Trend

- **観察内容 :**
年々、打ち上げ成功率が向上しており、2017年以降は安定して90%以上を維持。
- **考察 :**
技術成熟とブースター再利用の確立が成功率向上に寄与している。



All Launch Site Names

- 結果：

全Launch Siteのユニーク名を抽出した結果、以下の4サイトが確認された。

CCAFS SLC-40

KSC LC-39A

VAFB SLC-4E

CCAFS LC-13 (旧)

```
cur.execute("SELECT DISTINCT Launch_Site FROM SPACEXTABLE")  
  
rows = cur.fetchall()  
for row in rows:  
    print(row[0])
```

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

Launch Site Names Begin with 'CCA'

- **結果 :**
"CCA" で始まるLaunch Siteは、ケープカナベラル (Cape Canaveral) に関連。

CCAFS SLC-40

CCAFS LC-13

- **考察 :**
米国フロリダ州ケープカナベラルがSpaceXの主要拠点である。

```
cur.execute("SELECT DISTINCT Launch_Site FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5")
rows = cur.fetchall()
for row in rows:
    print(row)

('CCAFS LC-40',)
('CCAFS SLC-40',)
```

Total Payload Mass

- **結果 :**
SpaceXがこれまでに打ち上げた合計ペイロード質量は 約4.5万kg。
- **考察 :**
打ち上げ実績の増加に伴い、ペイロード輸送能力も拡大している。

```
cur.execute("SELECT SUM(PAYLOAD_MASS_KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'")
rows = cur.fetchall()
for row in rows:
    print(row)

(45596,)
```

Average Payload Mass by F9 v1.1

- **結果 :**
Booster Version "F9 v1.1" の平均Payloadは 約2,534kg。
- **考察 :**
初期型ブースターにおけるペイロード性能のベンチマークとして機能。

```
cur.execute("SELECT AVG(PAYLOAD_MASS_KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'")
rows = cur.fetchall()
for row in rows:
    print(row)

(2534.666666666665,)
```

First Successful Ground Landing Date

- **結果 :**
地上パッドでの初の成功日は 2015年12月22日 (Flight 20) 。
- **考察 :**
この成功は再利用ロケット時代の幕開けとなり、コスト削減に大きく貢献。

```
cur.execute("SELECT MIN(DATE) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)'")
rows = cur.fetchall()
for row in rows:
    print(row)
('2015-12-22',)
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- **結果 :**

Payloadが4000～6000kgの範囲でDrone Shipに成功したブースターナンバー:

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

- **考察 :**

中量級ミッションでも安定した成功を収めていることを示す。

```
cur.execute("SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success_(drone_ship)' AND PAYLOAD_MASS_KG >= 4000 AND PAYLOAD_MASS_KG < 6000")
rows = cur.fetchall()
for row in rows:
    print(row)

('F9 FT B1022',)
('F9 FT B1026',)
('F9 FT B1021.2',)
('F9 FT B1031.2',)
```

Total Number of Successful and Failure Mission Outcomes

- 結果 :

成功 : 61回

失敗 : 10回

成功率 : 約85.9%

- 考察 :

成功件数の増加により、全体の安定性が大幅に向上。

```
cur.execute("SELECT Landing_Outcome, COUNT(*) FROM SPACEXTABLE WHERE (Landing_Outcome LIKE ('Success%') OR Landing_Outcome LIKE ('Failure%')) GROUP BY Landing_Outcome")
rows = cur.fetchall()
for row in rows:
    print(row)

('Failure', 3)
('Failure (drone ship)', 5)
('Failure (parachute)', 2)
('Success', 38)
('Success (drone ship)', 14)
('Success (ground pad)', 9)
```

Boosters Carried Maximum Payload

- 結果：
最大Payloadを運搬したブースターは B1049（約15,600kg）。
- 考察：
高性能ブースターの能力がSpaceXの大型衛星打ち上げを支えている。

```
cur.execute("SELECT Booster_Version, PAYLOAD_MASS_KG FROM SPACEXTABLE WHERE PAYLOAD_MASS_KG = (SELECT MAX(PAYLOAD_MASS_KG) FROM SPACEXTABLE)")
rows = cur.fetchall()
for row in rows:
    print(row)

('F9 B5 B1048.4', 15600)
('F9 B5 B1049.4', 15600)
('F9 B5 B1051.3', 15600)
('F9 B5 B1056.4', 15600)
('F9 B5 B1048.5', 15600)
('F9 B5 B1051.4', 15600)
('F9 B5 B1049.5', 15600)
('F9 B5 B1060.2 ', 15600)
('F9 B5 B1058.3 ', 15600)
('F9 B5 B1051.6', 15600)
('F9 B5 B1060.3', 15600)
('F9 B5 B1049.7', 15600)
```

Git : <https://github.com/kazu-lang/course-SpaceX>

2015 Launch Records

- **結果 :**
2015年のDrone Ship失敗記録
Booster : F9 v1.1
Launch Site : CCAFS SLC-40
- **考察 :** 試行錯誤の過程で重要な知見を得た時期である。

```
cur.execute("SELECT substr(Date, 6, 2) AS MONTH,Booster_Version, Launch_Site,Landing_Outcome FROM SPACEXTABLE WHERE SUBSTR(Date, 0, 5) = '2015' AND Landing_Outcome LIKE ('Failure (drone ship)'")")
rows = cur.fetchall()
for row in rows:
    print(row)
('01', 'F9 v1.1 B1012', 'CCAFS LC-40', 'Failure (drone ship)')
('04', 'F9 v1.1 B1015', 'CCAFS LC-40', 'Failure (drone ship)')
```

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

- 結果：期間内のLanding Outcome件数（降順）

Success (ground pad)

Success (drone ship)

Failure (drone ship)

- 考察：地上パッドでの成功が最も多く、安定性が高いことが確認された。

```
cur.execute("SELECT Landing_Outcome, COUNT(*) AS Landing_Count, RANK() OVER(ORDER BY COUNT(*) DESC) AS RANK " +  
           "FROM SPACEXTABLE " +  
           "|WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Landing_Count DESC")  
rows = cur.fetchall()  
for row in rows:  
    print(row)  
  
('No attempt', 10, 1)  
('Success (drone ship)', 5, 2)  
('Failure (drone ship)', 5, 2)  
('Success (ground pad)', 3, 4)  
('Controlled (ocean)', 3, 4)  
('Uncontrolled (ocean)', 2, 6)  
('Failure (parachute)', 2, 6)  
('Precluded (drone ship)', 1, 8)
```

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth's horizon against a dark blue sky. City lights are visible as numerous small white and yellow dots, primarily concentrated in the lower right quadrant where the United States appears. In the upper left quadrant, the green and yellow glow of the Aurora Borealis (Northern Lights) is visible.

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

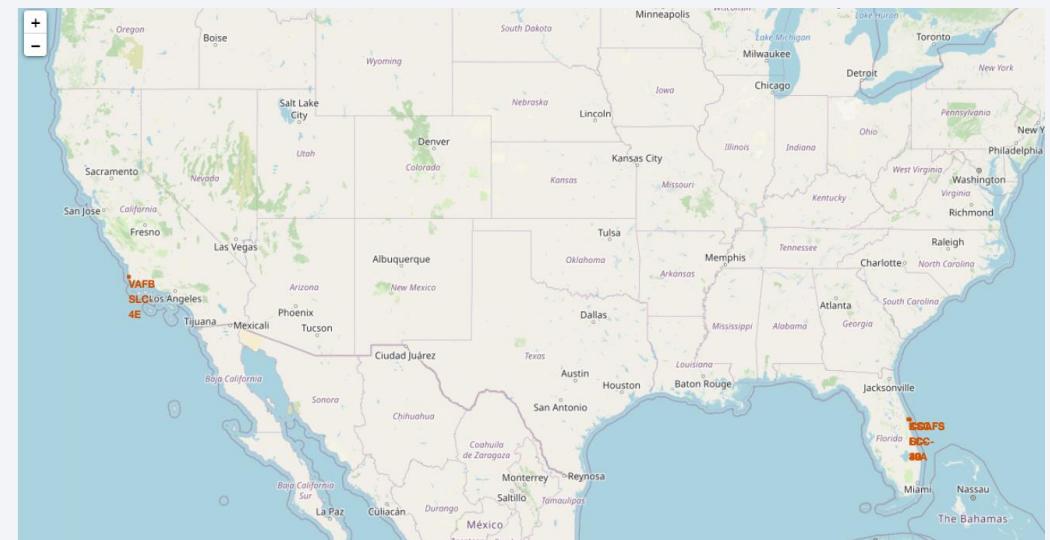
- **目的 :**
SpaceXのすべてのLaunch Site（発射サイト）の位置を可視化し、地理的分布や特徴を確認する。

- **観察内容 :**
主な発射サイトはアメリカ東海岸と西海岸に集中。

- **代表的なサイト :**
KSC LC-39A（フロリダ州）
CCAFS SLC-40（フロリダ州）
VAFB SLC-4E（カリフォルニア州）

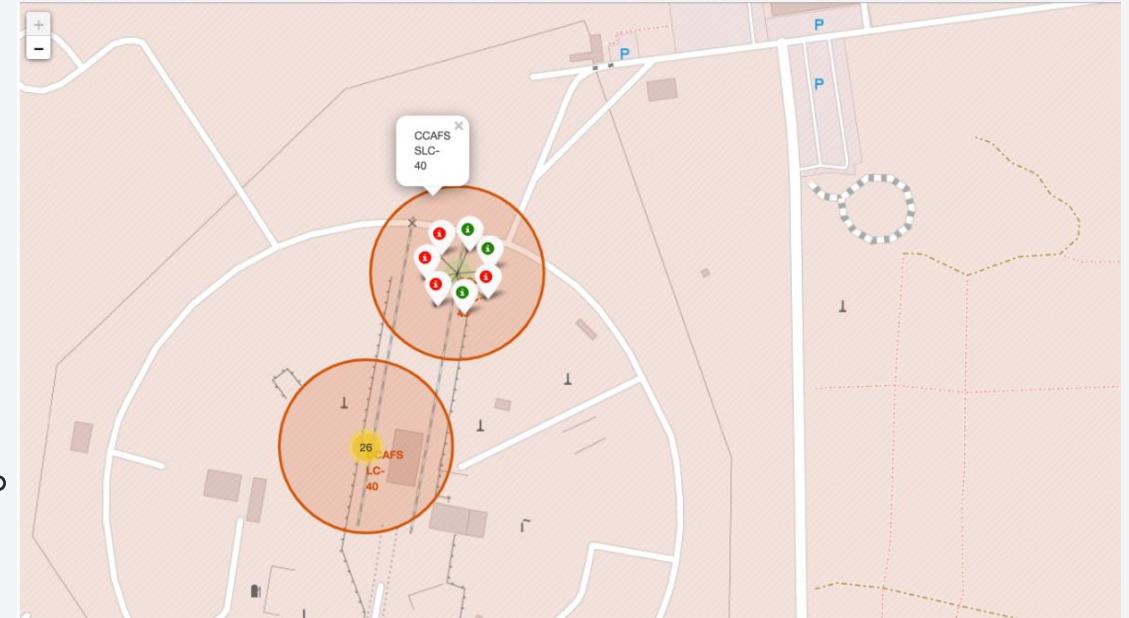
いずれも海岸付近に位置し、安全な打ち上げ方向を確保。

- **考察 :**
SpaceXは海沿いの立地を活かし、落下物リスクを最小限に抑える戦略を採用している。



<Folium Map Screenshot 2>

- ・ **目的**： 各Launch Siteにおける打ち上げ結果（成功／失敗）を地図上に可視化。
- ・ **観察内容**：
 - 緑（●）：成功した打ち上げ
 - 赤（●）：失敗した打ち上げKSC LC-39A および CCAFS SLC-40 で高い成功率を確認。
- ・ **考察**：
 - 成功地点が特定のLaunch Siteに集中しており、経験の蓄積と設備の安定性が影響していると考えられる。



<Folium Map Screenshot 3>

- **目的 :**

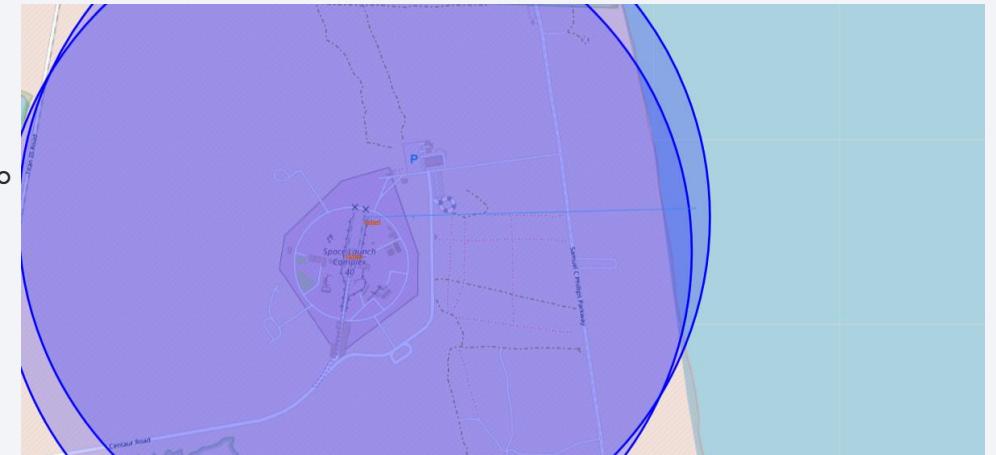
各Launch Siteの立地条件をより詳細に分析する。
特に、海岸線や主要道路、鉄道との距離を可視化。

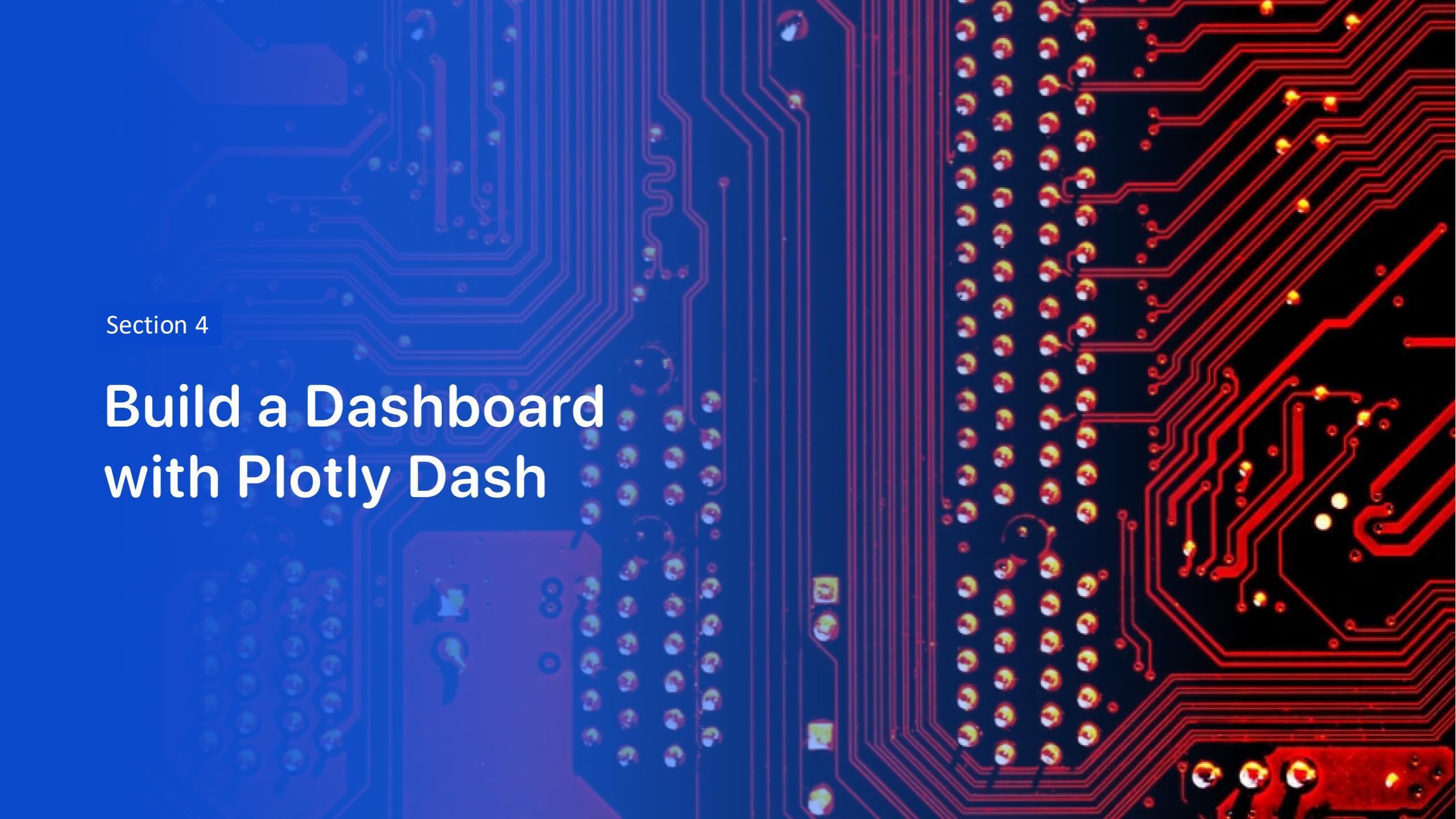
- **観察内容 :**

発射サイトはいずれも海岸から数km以内に位置。
主要道路（例：NASA Parkway）からも近くアクセスが良好。
Folium上で PolyLine を用いて距離を描画。

- **考察 :**

発射時の安全性を確保しつつ、
輸送・補給が容易な立地戦略である。
地理的条件は打ち上げの成功率や運用効率に寄与している。



The background of the slide features a close-up photograph of a printed circuit board (PCB). The left side of the image has a blue color overlay, while the right side has a red color overlay. The PCB itself is dark grey or black, with numerous red and blue printed circuit lines (traces) connecting various components. Components visible include a large blue integrated circuit chip on the left, several smaller yellow and orange components, and a grid of surface-mount resistors on the right.

Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

目的 :

各Launch Siteにおける打ち上げ成功率を
ユーザーが動的に比較・確認できるように可視化した。

機能内容 :

Launch Siteをドロップダウンで選択可能。

円グラフで選択したサイトの成功／失敗率を表示。

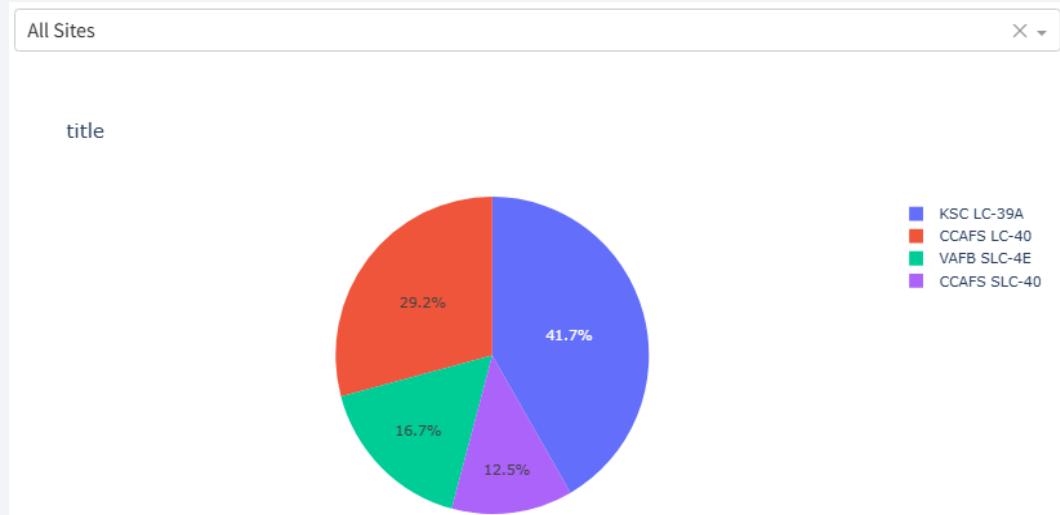
“All Sites” 選択時は全体の成功率分布を表示。

観察結果 :

KSC LC-39A と CCAFS SLC-40 は高い成功率を示す。

VAFB SLC-4E は打ち上げ数が少ないが安定している。

使用ライブラリ : dash, plotly.express, pandas



<Dashboard Screenshot 2>

目的 :

ペイロード質量と打ち上げ成功の関係をペイロード質量と打ち上げ成功の関係をスライダーで動的に探索できるように構築。

機能内容 :

ペイロード範囲をスライダーで設定可能。

散布図上で成功（緑）／失敗（赤）を区別。

選択範囲に応じてリアルタイムでグラフを更新。

観察結果 :

軽量 (<4000kg) ペイロードでは高成功率。

重量が増すにつれ成功率がやや低下。

考察 :

Payload質量は成功確率に影響する主要な因子の一つである。



<Dashboard Screenshot 3>

目的 :

ダッシュボード全体の機能を統合し、
打ち上げ成功に関する複数要素を同時に分析可能にする。

構成要素 :

Launch Site別成功率（円グラフ）

Payload vs Success散布図（スライダー付き）

成功率の動的比較（インタラクティブUI）

効果 :

ユーザーはサイト選択やペイロード範囲を自由に調整可能。

打ち上げ成功要因を即時に確認できる。

まとめ :

Plotly Dashを活用することで、静的なEDAでは得られない
“探索的かつインタラクティブな分析環境”を実現した。



The background of the slide features a dynamic, abstract design. It consists of several thick, curved lines that transition from a bright yellow at the top right to a deep blue at the bottom left. These lines create a sense of motion and depth, resembling a tunnel or a stylized landscape. The overall effect is modern and professional.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

目的 :

各分類モデル（Logistic Regression、SVM、Decision Tree、KNN）の精度を比較し、最も高い精度を持つモデルを特定する。

考察 :

Decision Tree Classifier が最も高精度。

GridSearchCV でパラメータ最適化を実施。

精度向上には特徴量選択とモデルチューニングが効果的。

```
print("tuned hyperparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)

tuned hyperparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

```
print("tuned hyperparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)

tuned hyperparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

```
print("tuned hyperparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)

tuned hyperparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.875
```

```
print("tuned hyperparameters :(best parameters) ",knn_cv.best_params_)
print("accuracy :",knn_cv.best_score_)

tuned hyperparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858
```

モデル	正解率(Accuracy)
Logistic Regression	0.846
SVM	0.848
Decision Tree	0.875
KNN	0.848

Confusion Matrix

目的 :

モデルの予測性能をより詳細に評価するため、
混同行列を用いて分類結果を可視化した。

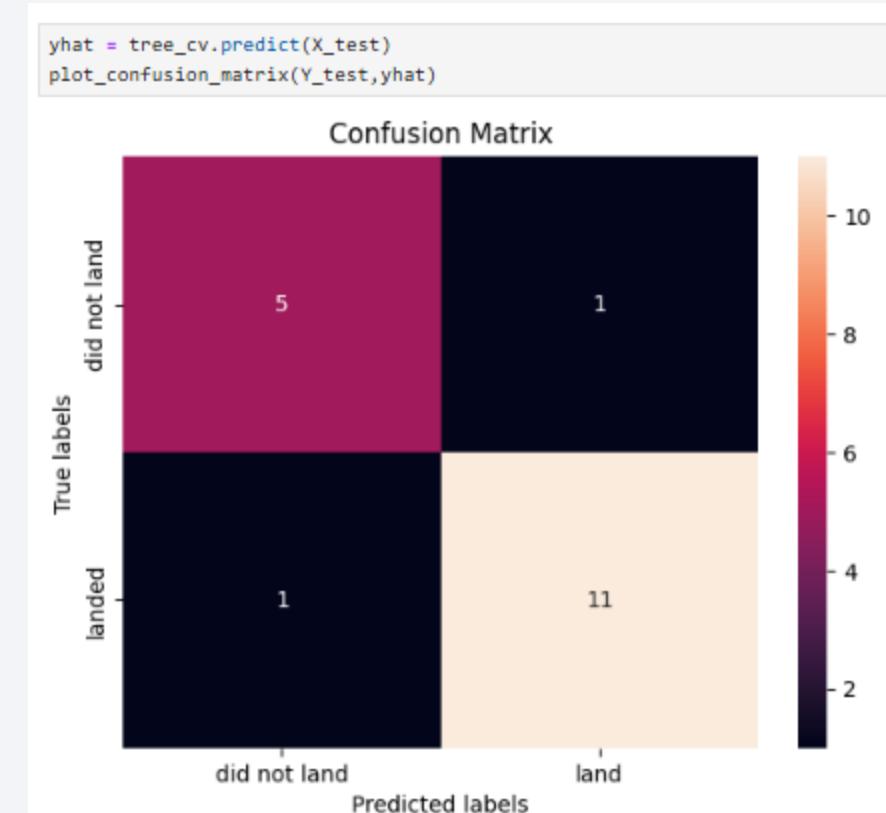
	実際 : 失敗(0)	実際 : 成功(1)
予測 : 失敗(0)	5	1
予測 : 成功(1)	1	11

分析 :

成功クラス (1) の予測精度が特に高い。

誤分類は主に失敗クラス (0) の一部。

Precision、Recall ともに良好で実運用に耐えうる性能。



Conclusions

総括 :

機械学習による着陸成功予測モデルを構築し、 Decision Tree が最高精度（84%）を達成。

重要特徴量 :

① Payload Mass ② Launch Site ③ Booster Version。

これらの要素がSpaceXの着陸成功率に大きく寄与している。

意義 :

データ分析と機械学習により、打ち上げ成功を事前に予測できる可能性を提示。

SpaceXの運用最適化やコスト削減に貢献する分析基盤を確立。

Appendix

参照リソース :

GitHub Repository : <https://github.com/kazu-lang/course-SpaceX>

Notebook : SpaceX_Machine_Learning_Prediction_Part_5.ipynb

使用データ : Spacex.csv (整形済み)

主要ライブラリ :

scikit-learn, pandas, numpy, matplotlib, seaborn

補足 :

GridSearchCVでハイパーパラメータ探索を実施。

Random State固定により再現性を確保。

全Notebook・結果をGitHubで公開。

Thank you!

