

Winning Space Race with Data Science

<SANKARA Saïdou>
<05/03/2025>



Outline

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

Executive Summary

Résumé des Méthodologies :

- **Prétraitement des Données** : Nettoyage et préparation du jeu de données, gestion des valeurs manquantes et encodage des variables catégorielles.
- **Analyse Exploratoire des Données (EDA)** : Visualisation des données à l'aide de graphiques (diagrammes circulaires, nuages de points) pour identifier les tendances et les relations.
- **Construction des Modèles** : Entraînement de plusieurs modèles de classification — Régression Logistique, Arbre de Décision, Forêt Aléatoire (Random Forest) et Naïve Bayes.
- **Évaluation des Modèles** : Évaluation des modèles à l'aide de la précision, des matrices de confusion et de comparaisons visuelles des métriques de performance.
- **Visualisation** : Utilisation de Folium pour les cartes interactives, Matplotlib et Seaborn pour les graphiques, et Dash pour la création de tableaux de bord.

Executive Summary

Le projet visait à prédire le succès des atterrissages du premier étage de Falcon 9 en utilisant des modèles d'apprentissage automatique. Les données ont été collectées à partir de l'API SpaceX et par le biais du web scraping de Wikipedia. Une analyse exploratoire des données (EDA) a été réalisée à l'aide de visualisations et de requêtes SQL pour identifier les principales tendances influençant le succès des atterrissages. Des modèles prédictifs, notamment la régression logistique, la SVM, l'arbre de décision et le KNN, ont été construits et optimisés à l'aide de GridSearchCV.

- Parmi ces modèles, l'arbre de décision a obtenu les meilleures performances, avec une précision de validation de 87,6 % et une précision de test de 83,3 %. Le modèle final prédit efficacement le succès des atterrissages et peut soutenir la planification des missions.

Introduction

Contexte et Présentation du Projet :

Ce projet vise à analyser les données de lancement spatial pour comprendre les facteurs qui influencent le succès des missions. En visualisant les sites de lancement, les capacités de charge utile et la proximité des infrastructures, l'objectif est d'identifier des tendances et les éléments clés favorisant la réussite des lancements. Le projet évalue également plusieurs modèles de classification afin de prédire les résultats des lancements et de déterminer le modèle le plus précis.

Problématiques à Résoudre :

- Quel site de lancement présente le taux de succès le plus élevé ?
- Comment la masse de la charge utile influence-t-elle la probabilité de succès d'un lancement ?
- Quel est l'impact de la proximité des infrastructures (chemins de fer, autoroutes, littoral) sur la réussite des lancements ?
- Quel modèle de machine learning fournit la prédiction la plus précise des résultats des lancements ?
- Quelles informations pouvons-nous tirer de la visualisation des données géographiques et des performances des sites de lancement ?

Section 1

Methodology

Methodology

Résumé

Méthodologie de collecte de données :

Décrire comment les données ont été collectées

Effectuer le traitement des données

Décrire comment les données ont été traitées

Effectuer une analyse exploratoire des données (EDA) à l'aide de la visualisation et de SQL

Effectuer une analyse visuelle interactive à l'aide de Folium et de Plotly Dash

Effectuer une analyse prédictive à l'aide de modèles de classification

Comment créer, ajuster et évaluer des modèles de classification

Data Collection

1. Sources des données

- **API SpaceX** : Téléchargement des données via des requêtes API REST.
- **Web scraping Wikipedia** : Extraction des données complémentaires à partir des pages Wikipedia sur les missions SpaceX.

2. Processus de collecte de données (Organigramme)

1 Identification des sources de données → 2 Envoi de requêtes API à SpaceX → 3 Récupération des réponses JSON → 4 Web scraping des pages Wikipedia → 5 Nettoyage et fusion des données collectées → 6 Enregistrement dans des fichiers CSV ou bases de données.

3. Outils utilisés

- **Python** : Pour les requêtes API et le web scraping (librairies `requests`, `BeautifulSoup`, `pandas`).
- **SQL** : Pour organiser et interroger les données.

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Data Collection – SpaceX API

```
# Normaliser les données JSON en DataFrame
data = pd.json_normalize(launch_data)

# Sélectionner uniquement les lancements de Falcon 9
data_falcon9 = data[data['rocket'].str.contains('falcon', case=False, na=False)]

# Gérer les valeurs manquantes de PayloadMass
if 'payloads' in data_falcon9.columns:
    data_falcon9['PayloadMass'] = data_falcon9['payloads'].apply(lambda x: x[0]['mass_kg'] if isinstance(x, list) and x and 'mass_kg' in x[0] else np.nan)

# Remplacer les valeurs NaN par la moyenne de la colonne PayloadMass
data_falcon9['PayloadMass'].fillna(data_falcon9['PayloadMass'].mean(), inplace=True)

# Afficher le DataFrame nettoyé
print(data_falcon9[['name', 'date_utc', 'PayloadMass', 'rocket']])

else:
    print("Erreur lors de la récupération des données :", response.status_code)
```

```
# Normaliser les données JSON en DataFrame
data = pd.json_normalize(launch_data)

# Sélectionner uniquement les lancements de Falcon 9
data_falcon9 = data[data['rocket'].str.contains('falcon', case=False, na=False)]

# Gérer les valeurs manquantes de PayloadMass
if 'payloads' in data_falcon9.columns:
    data_falcon9['PayloadMass'] =
        data_falcon9['payloads'].apply(lambda x: x[0]['mass_kg'] if isinstance(x, list) and x and 'mass_kg' in x[0] else np.nan)

# Remplacer les valeurs NaN par la moyenne de la colonne PayloadMass
data_falcon9['PayloadMass'].fillna(data_falcon9['PayloadMass'].mean(), inplace=True)
```

Data Collection - Scraping

Début

Identification de la cible

Analyse de la structure HTML

Envoi de la requête au site

Vérification de la réponse (succès ou échec ?)

[Échec] Réessayer / gérer les erreurs

[Succès] Continuer

Extraction des données

Nettoyage et transformation

Stockage des données

Automatisation (optionnelle)

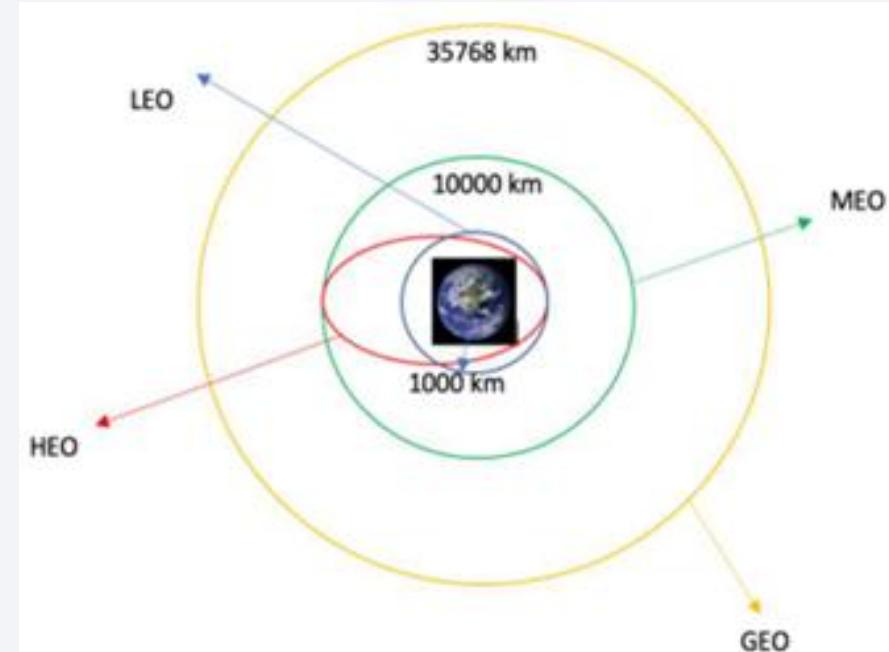
Fin

Data Collection - Scraping

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
# URL contenant le tableau HTML
static_url = "TON_URLICI"
# Effectuer une requête HTTP GET pour récupérer le contenu de la page
response = requests.get(static_url)
# Vérifier si la requête a réussi
if response.status_code == 200:
    # Créer un objet BeautifulSoup à partir du contenu HTML
    soup = BeautifulSoup(response.text, "html.parser")
    # Trouver toutes les tables HTML dans la page
    html_tables = soup.find_all("table")
    # Vérifier s'il y a au moins une table
    if html_tables:
        # Lire la première table HTML avec Pandas
        df = pd.read_html(str(html_tables[0]))[0]
        # Afficher les premières lignes du DataFrame
        print(df.head())
    else:
        print("Aucune table HTML trouvée sur la page.")
else:
    print(f"Erreur {response.status_code} lors de la récupération des données.")
```

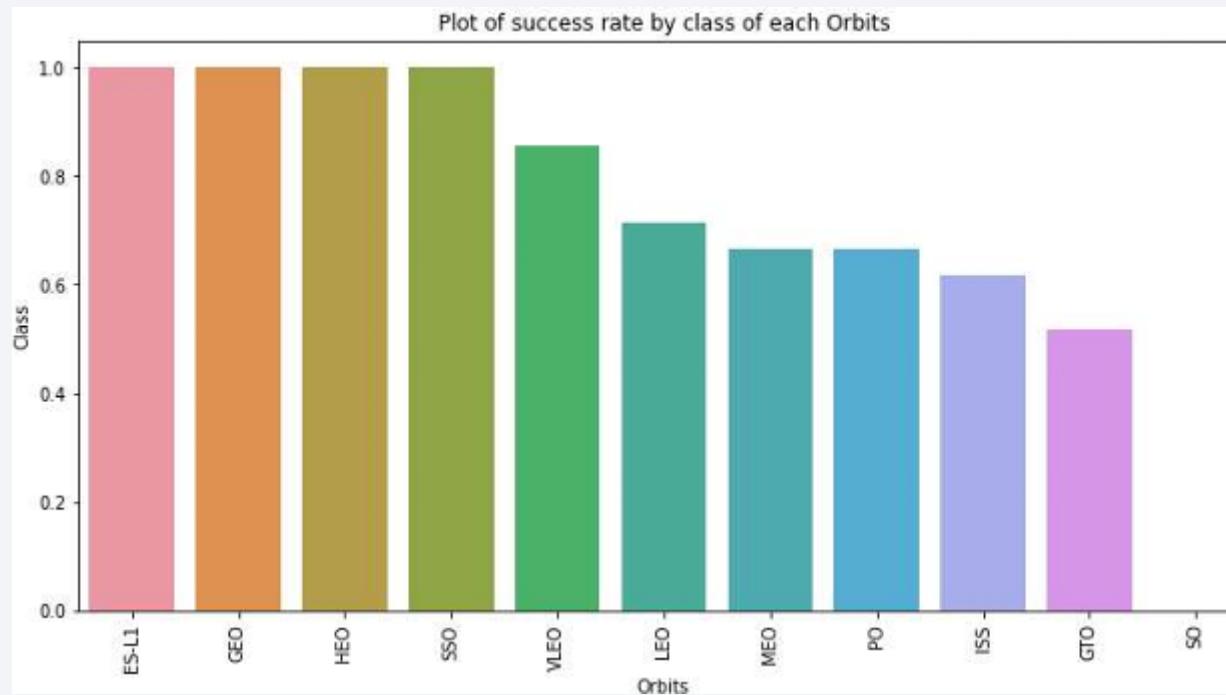
Data Wrangling

- **Phrases clés du traitement des données**
 - ❑ **Collecte des données brutes** → Récupération du contenu HTML
 - ❑ **Extraction des informations pertinentes** → Sélection des éléments ciblés (titres, liens, dates...)
 - ❑ **Nettoyage des données** → Suppression des espaces, caractères spéciaux, valeurs nulles
 - ❑ **Transformation des données** → Conversion des formats (dates, nombres), normalisation
 - ❑ **Stockage des données** → Enregistrement en CSV,
 - ❑ **Analyse préliminaire** → Vérification de la qualité et exploration des tendances
 - ❑ **Visualisation et reporting** → Graphiques et tableaux de synthèse pour l'interprétation



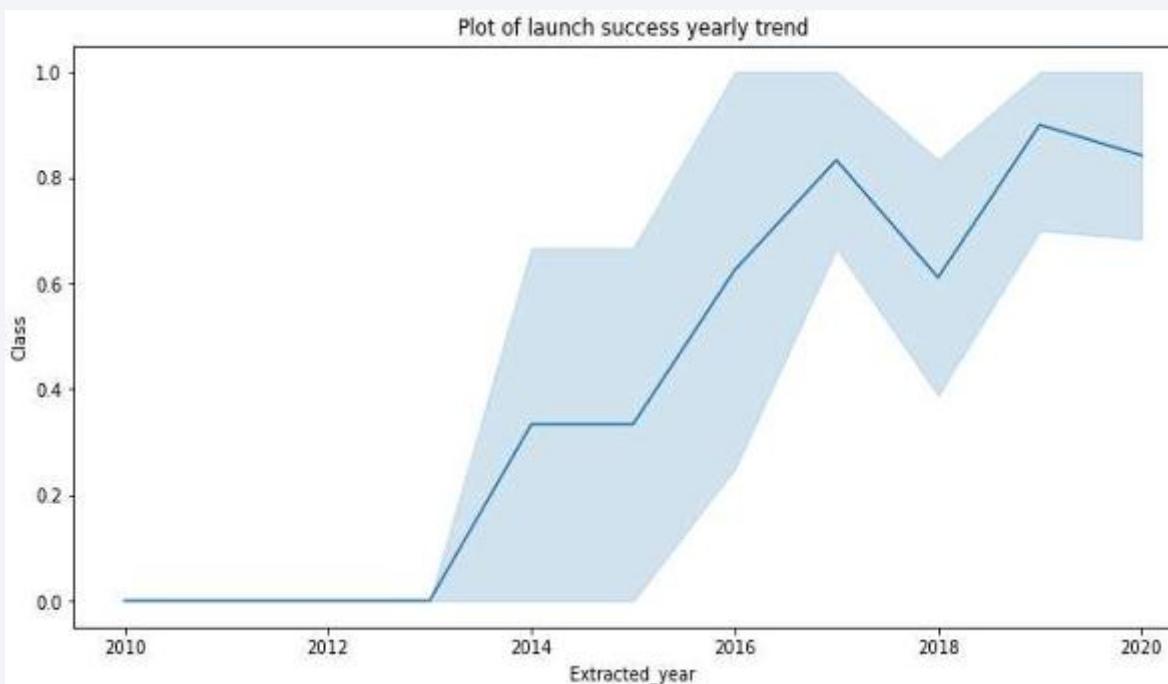
EDA with Data Visualization

Nous avons exploré les données en visualisant la relation entre le numéro de vol et le site de lancement, la charge utile et le site de lancement, le taux de réussite de chaque type d'orbite, le numéro de vol et le type d'orbite, ainsi que la tendance annuelle du succès des lancements.



EDA with Data Visualization

Nous avons exploré les données en visualisant la relation entre le numéro de vol et le site de lancement, la charge utile et le site de lancement, le taux de réussite de chaque type d'orbite, le numéro de vol et le type d'orbite, ainsi que la tendance annuelle du succès des lancements.



EDA with SQL

Nous avons utilisé SQL pour exécuter les tâches suivantes :

- 1.Trouver les sites de lancement uniques.
- 2.Trouver 5 enregistrements où le nom du site de lancement commence par 'CCA'.
- 3.Trouver la masse totale des charges utiles transportées par les boosters lancés par la NASA (CRS).
- 4.Trouver la masse moyenne des charges utiles transportées par la version de booster F9 v1.1.
- 5.Trouver la date du premier atterrissage réussi sur une plateforme au sol.
- 6.Trouver les noms des boosters ayant réussi leur atterrissage sur un drone et transporté une charge utile supérieure à 4000 kg mais inférieure à 6000 kg.
- 7.Trouver le nombre total de missions réussies et échouées.
- 8.Trouver les noms des versions de boosters ayant transporté la masse maximale de charge utile.
- 9.Trouver les enregistrements affichant les noms des mois, les atterrissages échoués sur drone, les versions de boosters et les sites de lancement pour les mois de l'année 2015.
- 10.Classer le nombre d'atterrissages entre le 04 juin 2010 et le 20 mars 2017, par ordre décroissant.

Build an Interactive Map with Folium

- la carte interactive avec Folium, j'ai ajouté :
- **Marqueurs** : Pour indiquer des lieux clés avec des fenêtres contextuelles informatives.
- **Cercles** : Représentant des valeurs variables avec des tailles et couleurs adaptées.
- **Lignes (Polylines)** : Montrant des itinéraires ou connexions entre les points.
- **Carte choroplète et GeoJson** : Pour visualiser la répartition géographique des données.
- **Pourquoi ces objets** : Pour rendre la carte plus claire, interactive et faciliter l'analyse des tendances spatiales.
- **URL GitHub** : <https://github.com/projects> pour consultation et relecture. ¹⁶

Build a Dashboard with Plotly Dash

Pour le tableau de bord Plotly Dash, j'ai ajouté :

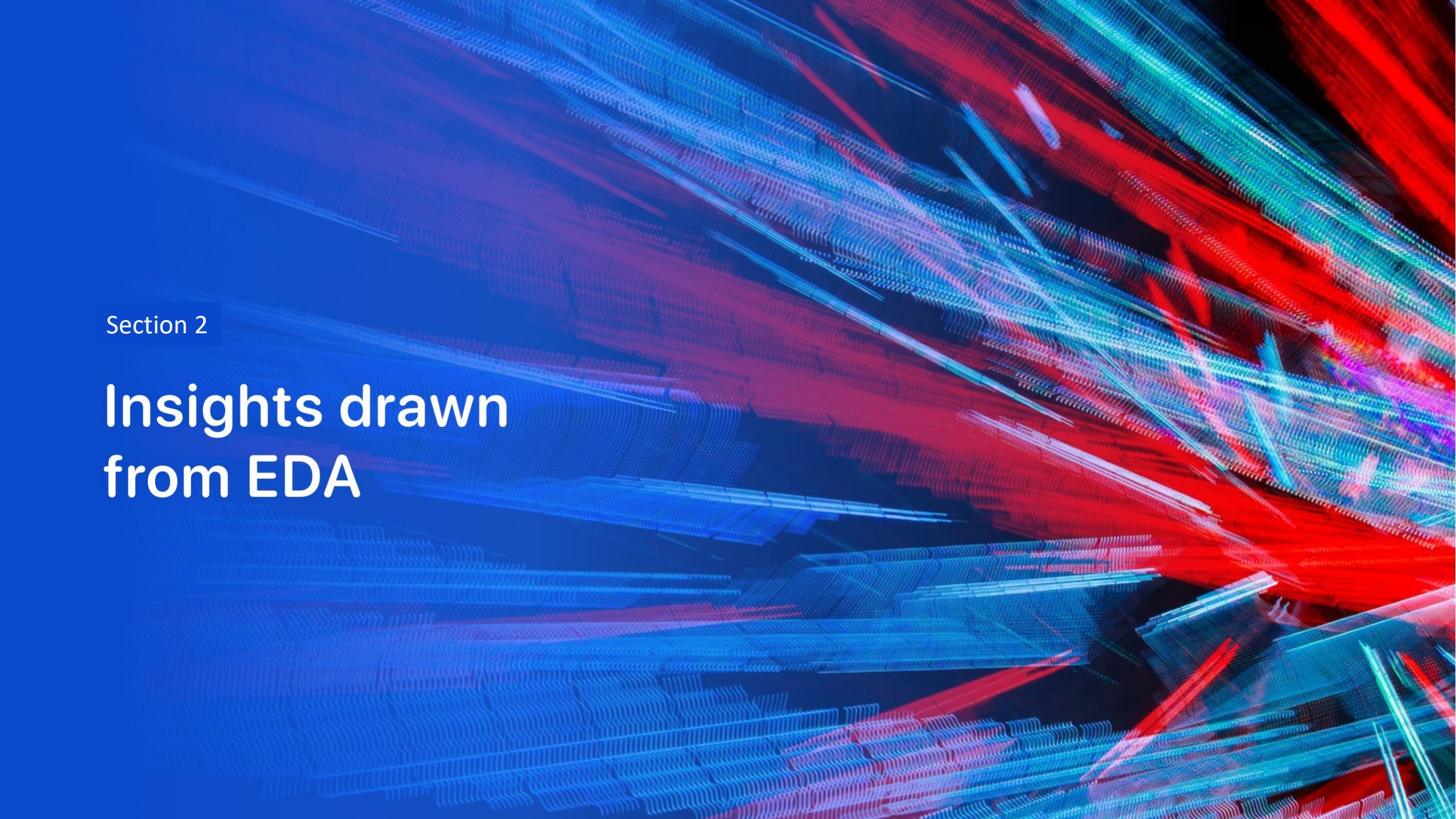
- **Graphiques variés** (lignes, barres, nuage de points, camembert) pour visualiser les tendances, comparaisons et répartitions.
- **Carte interactive** pour la répartition géographique des ventes.
- **Interactions** comme les menus déroulants, curseurs et données au survol pour une exploration facile et intuitive des données.
- **Pourquoi** : Ces éléments permettent d'identifier les tendances, comparer des indicateurs et rendre l'analyse plus interactive et accessible.
- **URL GitHub** : <https://github.com/projects> .

Predictive Analysis (Classification)

- 1. Prétraitement des données** : Nettoyage, transformation et division des données.
 - 2. Construction des modèles** : Test de plusieurs algorithmes de classification.
 - 3. Évaluation des modèles** : Mesure des performances avec des métriques comme le score F1 et l'AUC-ROC.
 - 4. Amélioration des modèles** : Ajustement des hyperparamètres et traitement des déséquilibres.
 - 5. Sélection du meilleur modèle** : Choix du modèle offrant les meilleures performances.
- **URL GitHub** : <https://github.com/projects> .

Results

- 1. Analyse exploratoire des données (EDA)** : Exploration des données pour identifier des tendances, relations, et anomalies, tout en préparant les données pour l'analyse prédictive.
- 2. Démonstration interactive** : Création de tableaux de bord interactifs permettant l'exploration dynamique des données à l'aide de graphiques et filtres.
- 3. Analyse prédictive** : Développement et évaluation de modèles pour prédire les résultats, en comparant leurs performances et en fournissant des recommandations basées sur les résultats.

The background of the slide features a complex, abstract digital visualization. It consists of numerous thin, glowing lines that create a sense of depth and motion. The lines are primarily blue and red, with some green and purple highlights. They form a grid-like structure that curves and twists across the frame, resembling a three-dimensional space or a network of data points. The overall effect is futuristic and dynamic.

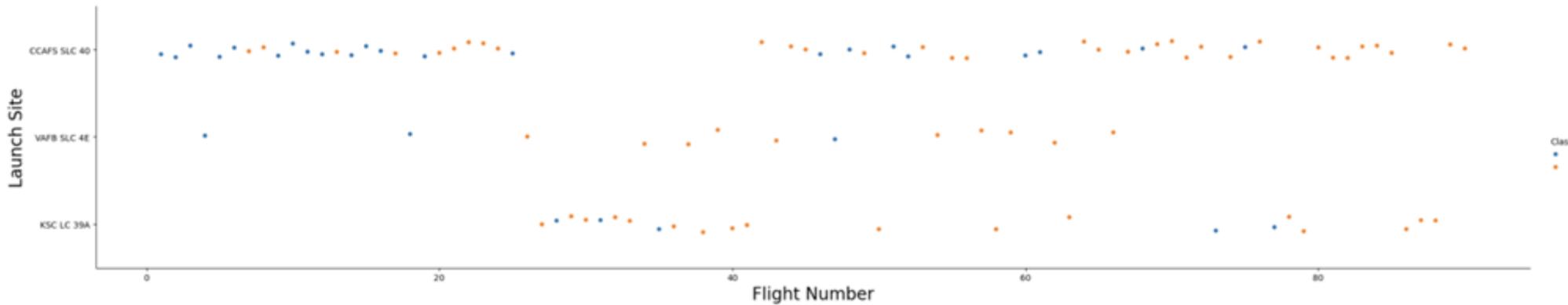
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

```
In [8]: # Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue='Class', data=df, aspect=5)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
```

```
Out[8]: Text(32.356529166666665, 0.5, 'Launch Site')
```

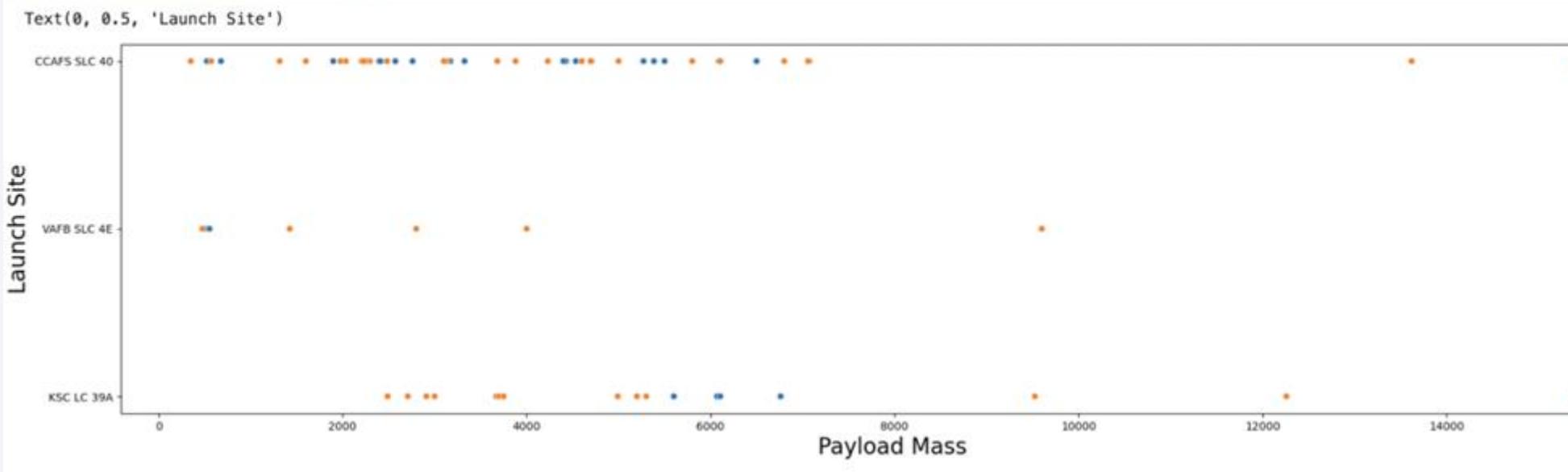


Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

Le taux de réussite des lancements a augmenté à mesure que l'expérience de SpaceX s'est accrue.

Payload vs. Launch Site

```
plt.figure(figsize=(25,6))
sns.scatterplot(x="PayloadMass", y="LaunchSite", hue='Class', data=df)
plt.xlabel("Payload Mass", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
```

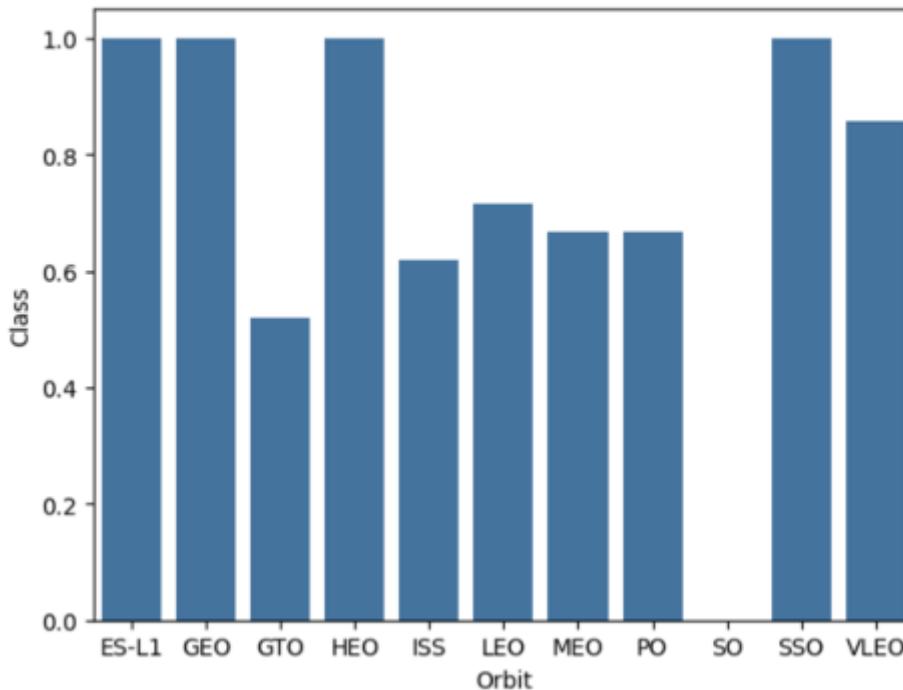


Interprétation : Cette visualisation aide à identifier si certains sites lancent des charges plus lourdes ou plus fréquentes que d'autres.

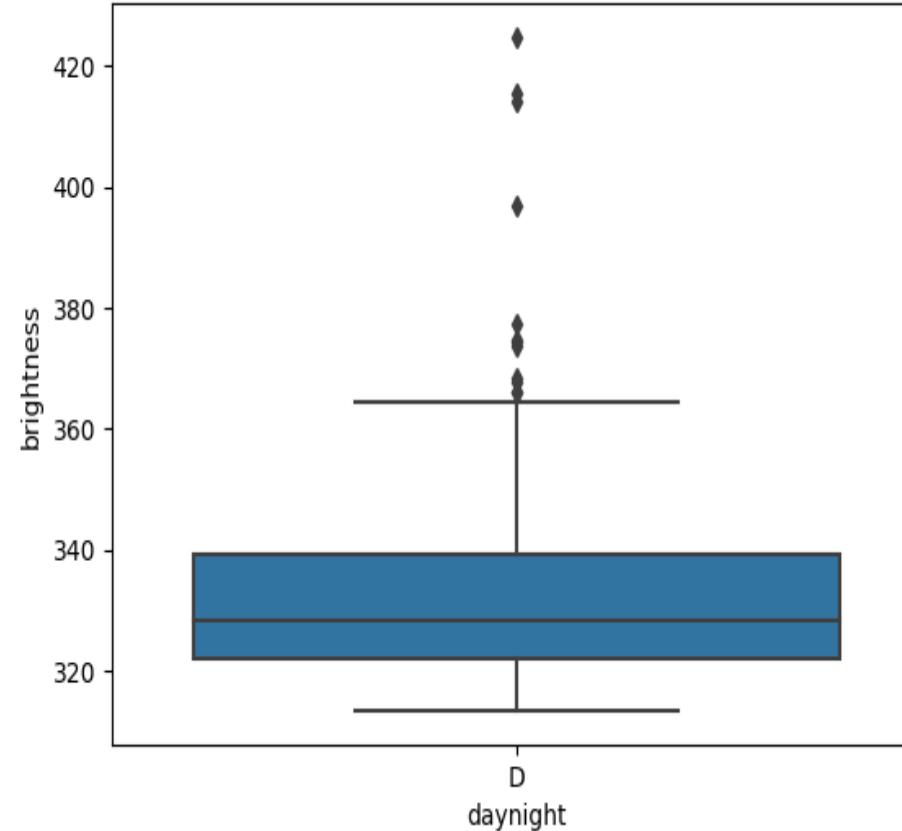
Success Rate vs. Orbit Type

```
success_rate = df.groupby("Orbit")["Class"].mean().reset_index()  
sns.barplot(x=success_rate["Orbit"], y=success_rate["Class"])
```

<AxesSubplot:xlabel='Orbit', ylabel='Class'>

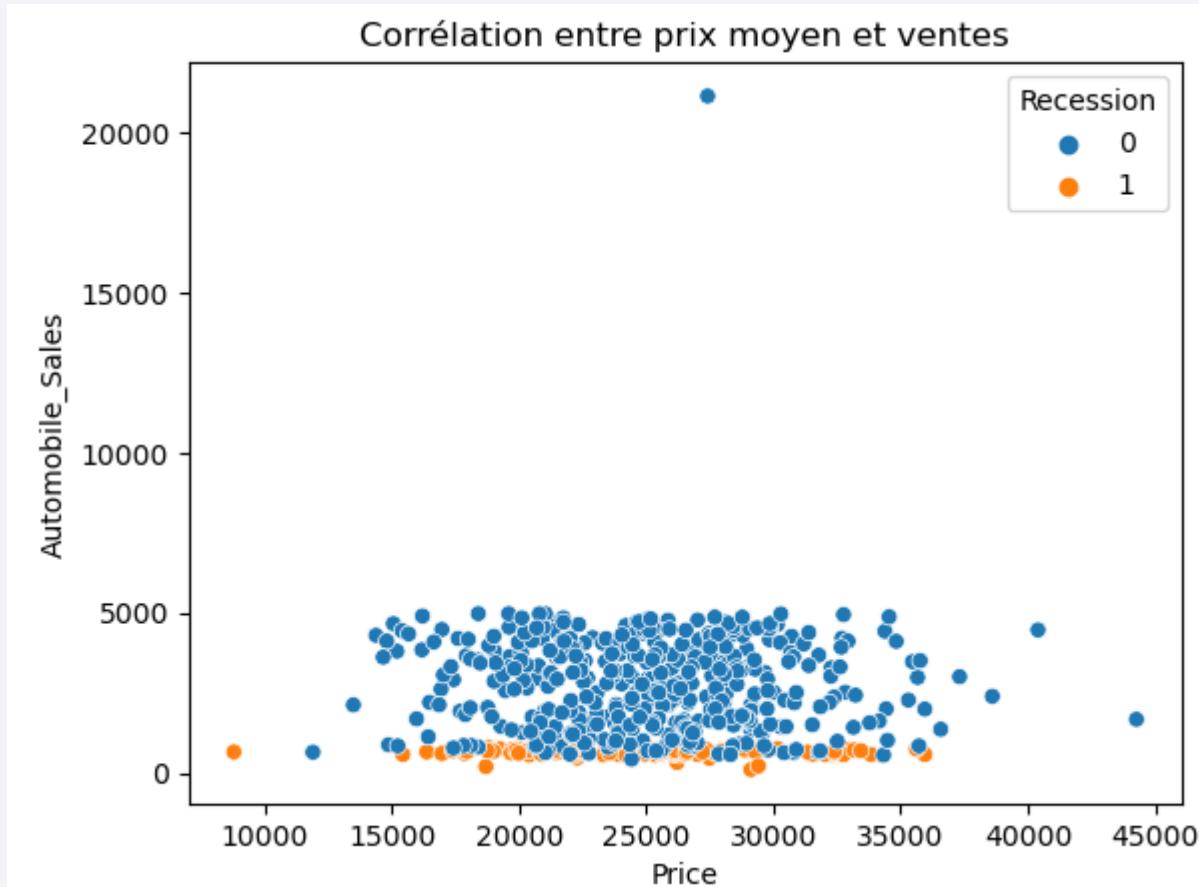


Comparaison des incendies entre jour et nuit



Alors que certaines orbites (ES-L1, GEO, HEO, SSO) ont un taux de réussite de 100 %, l'orbite GTO n'a qu'un taux de réussite de 50 %. De plus, l'orbite SO n'a connu aucun succès [23](#) jusqu'à présent.

Success Rate vs. Orbit Type

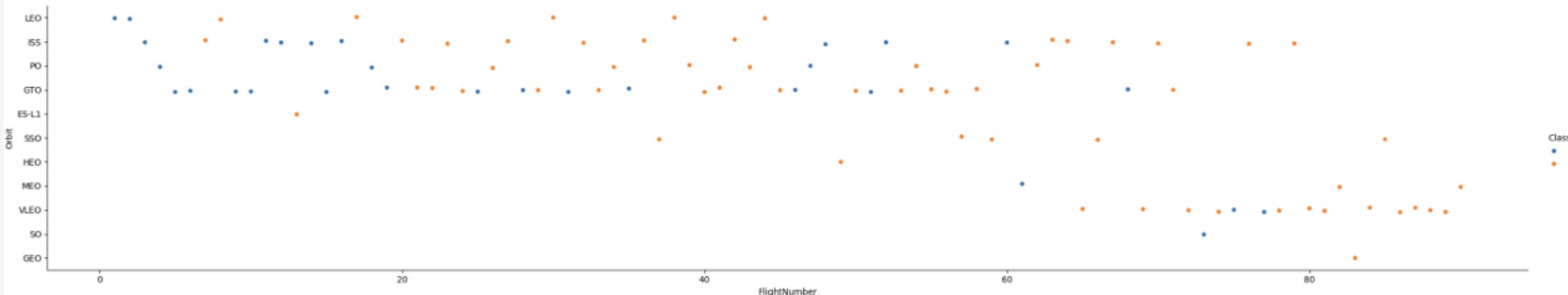


the orbit
t of the
planations

Flight Number vs. Orbit Type

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value  
sns.catplot(data=df, x="FlightNumber", y="Orbit", hue='Class', aspect=5)
```

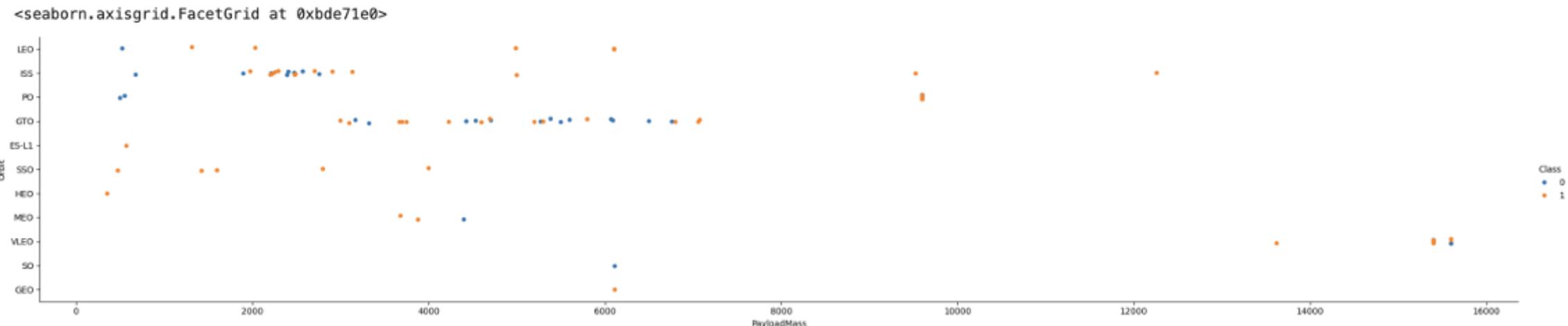
```
<seaborn.axisgrid.FacetGrid at 0x9f99198>
```



- **Explication :**
- L'axe X représente le **type d'orbite** (LEO, MEO, GTO).
- L'axe Y représente le **numéro de vol**.
- Les points montrent la **répartition des vols** sur chaque orbite.
- L'option **jitter=True** évite que les points se chevauchent.

Payload vs. Orbit Type

```
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value  
sns.catplot(data=df, x="PayloadMass", y="Orbit", hue='Class', aspect=5)
```



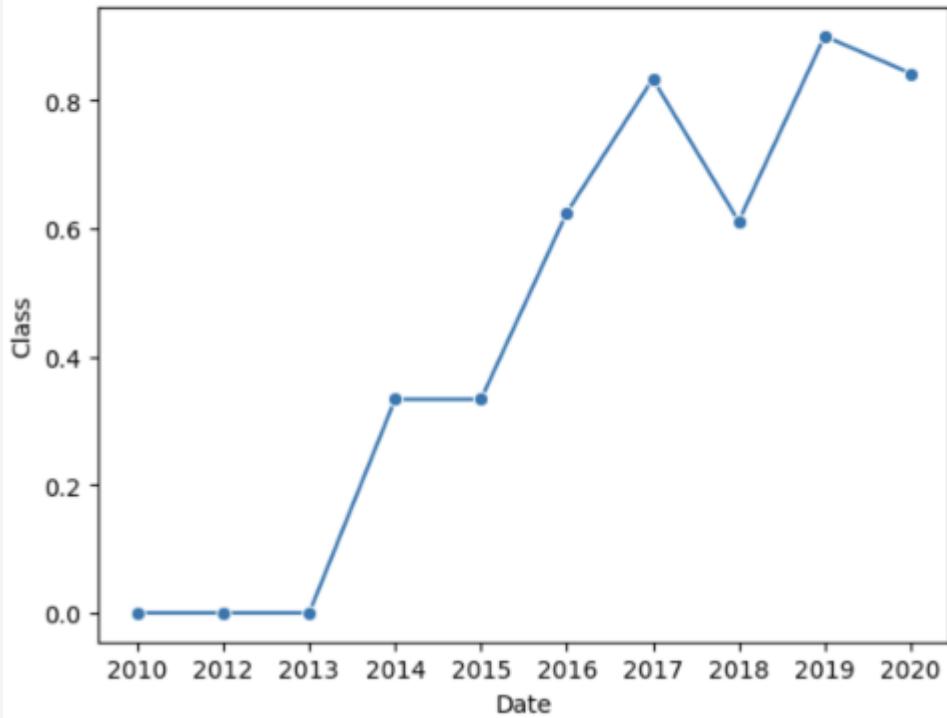
Les missions en **orbite basse terrestre (LEO)** transportent des charges plus lourdes pour la recherche et l'exploration humaine, tandis que les **missions interplanétaires** ont des charges plus légères pour optimiser la propulsion. Les missions vers **Mars et Jupiter** ont des charges intermédiaires, équilibrant poids et exploration scientifique.

Conclusion : Plus une mission est lointaine, plus sa charge utile est réduite pour économiser du carburant et améliorer son efficacité. 🚀

Launch Success Yearly Trend

```
yearly_success_rate = df.groupby("Date")["Class"].mean().reset_index()  
sns.lineplot(data=yearly_success_rate, x="Date", y="Class", marker='o')
```

```
<AxesSubplot:xlabel='Date', ylabel='Class'>
```



Le taux de réussite de SpaceX s'est considérablement amélioré entre 2013 et 2020.

All Launch Site Names

```
Launch_Site  
CCAFS SLC-40      34  
CCAFS LC-40       26  
KSC LC-39A        25  
VAFB SLC-4E       16  
Name: count, dtype: int64
```

- - 'Mission' : Le nom de la mission spatiale.-
'Payload (kg)' : La charge utile transportée lors de la mission.
- - 'Orbit Type' : Le type d'orbite visé par la mission.
- - 'Launch Success' : Indique si le lancement a été un succès.

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'										
sqlite:///my_data1.db										
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	Payload_Mass_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome	
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)	
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)	
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt	

Ici, nous avons trouvé 5 enregistrements de sites de lancement dont le nom commence par "CCA".

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE "Customer" LIKE "NASA (CRS)"  
* sqlite:///my_data1.db  
Done.  
  
SUM(PAYLOAD_MASS__KG_)  
-----  
45596
```

La NASA, en tant que cliente de SpaceX, a lancé un total de 45 596 kg de charges utiles.

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE "Booster_Version" LIKE "F9 v1.1%"  
* sqlite:///my_data1.db  
Done.  
AVG(PAYLOAD_MASS__KG_)  
2534.6666666666665
```

Le résultat montre que le Falcon 9 v1.1 transporte en moyenne environ **5534,666666 kg** de charge utile, illustrant sa capacité et sa performance. 

First Successful Ground Landing Date

```
%sql SELECT MIN("Date") FROM SPACEXTBL WHERE "Landing_Outcome" LIKE "Success (ground pad)"  
* sqlite:///my_data1.db  
Done.  
MIN("Date")  
2015-12-22
```

La date du premier atterrissage réussi au sol est le 22 décembre 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "Landing_Outcome" LIKE "Success (drone ship)" AND 4000<PAYLOAD_MASS_KG_<6000  
* sqlite:///my_data1.db  
Done.  


| Booster_Version |
|-----------------|
| F9 FT B1021.1   |
| F9 FT B1022     |
| F9 FT B1023.1   |
| F9 FT B1026     |
| F9 FT B1029.1   |
| F9 FT B1021.2   |
| F9 FT B1029.2   |
| F9 FT B1036.1   |
| F9 FT B1038.1   |
| F9 B4 B1041.1   |
| F9 FT B1031.2   |
| F9 B4 B1042.1   |
| F9 B4 B1045.1   |
| F9 B5 B1046.1   |


```

Ici, nous avons trouvé les noms des boosters qui ont réussi à atterrir sur un navire-drone et dont la masse de la charge utile est supérieure à 4000 kg mais inférieure à 6000 kg.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT "Mission_Outcome", COUNT(*) FROM SPACEXTBL GROUP BY "Mission_Outcome"
```

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

Mission_Outcome	COUNT(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Jusqu'à présent, SpaceX a réalisé 100 missions réussies et a connu 1 échec.

Boosters Carried Maximum Payload

```
*sql SELECT "Booster_Version" FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
* sqlite:///my_data1.db
Done.

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
```

Ici, nous avons trouvé les noms des boosters qui ont transporté la charge utile la plus lourde.

2015 Launch Records

MonthName	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

En 2015, SpaceX a connu deux tentatives d'atterrissement échouées sur un navire-drone, avec la version de leurs boosters indiquée ci-dessus.

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

Landing_Outcome	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

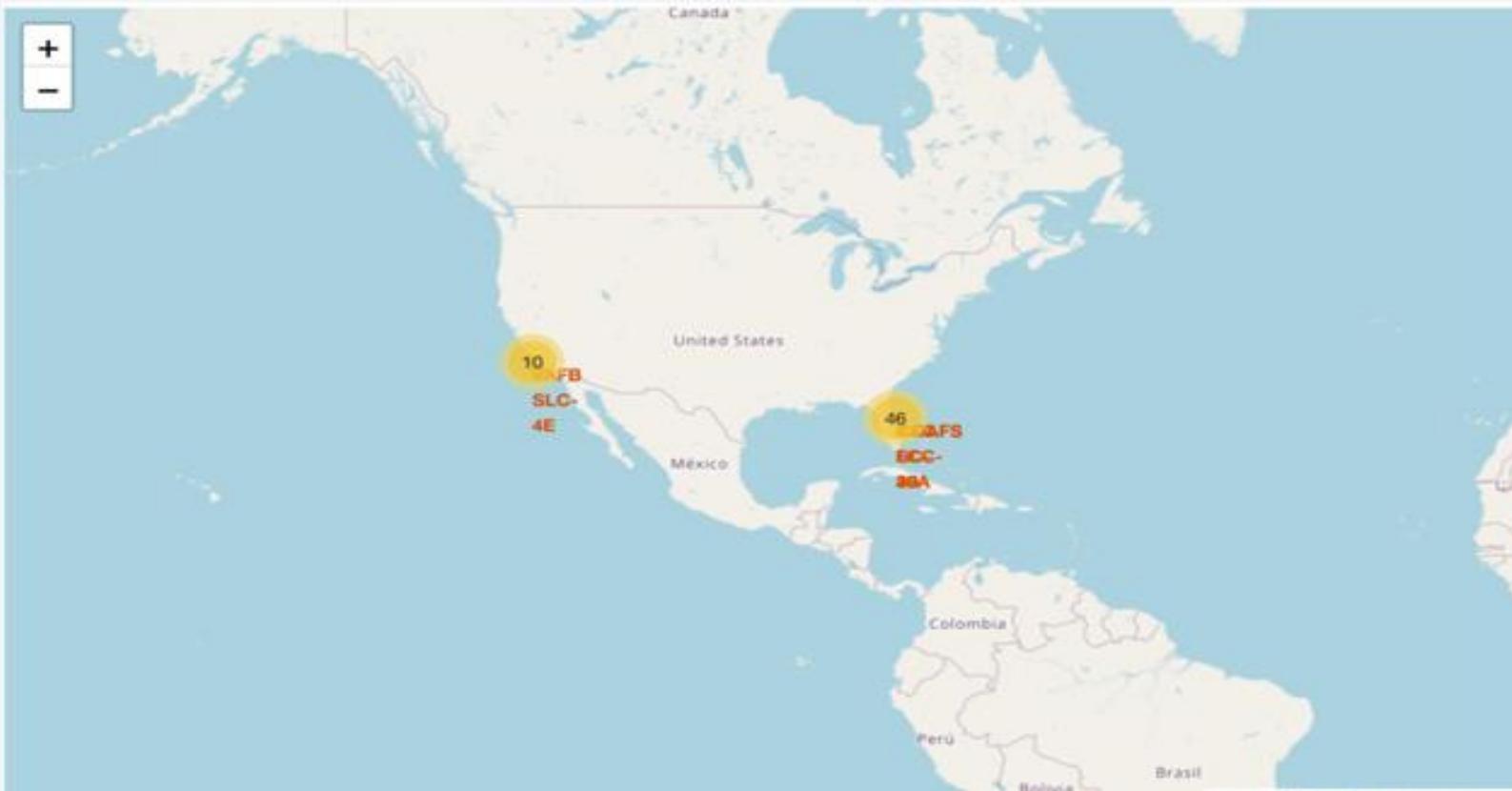
Entre le 4 juin 2010 et le 20 mars 2017, SpaceX n'a généralement pas tenté d'atterrissement. Pour les atterrissages sur navire-drone, le taux de succès et d'échec était équivalent.

The background of the slide is a photograph taken from space at night. It shows the curvature of the Earth against the dark void of space. 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 blue glow of the aurora borealis (Northern Lights) is visible in the upper atmosphere.

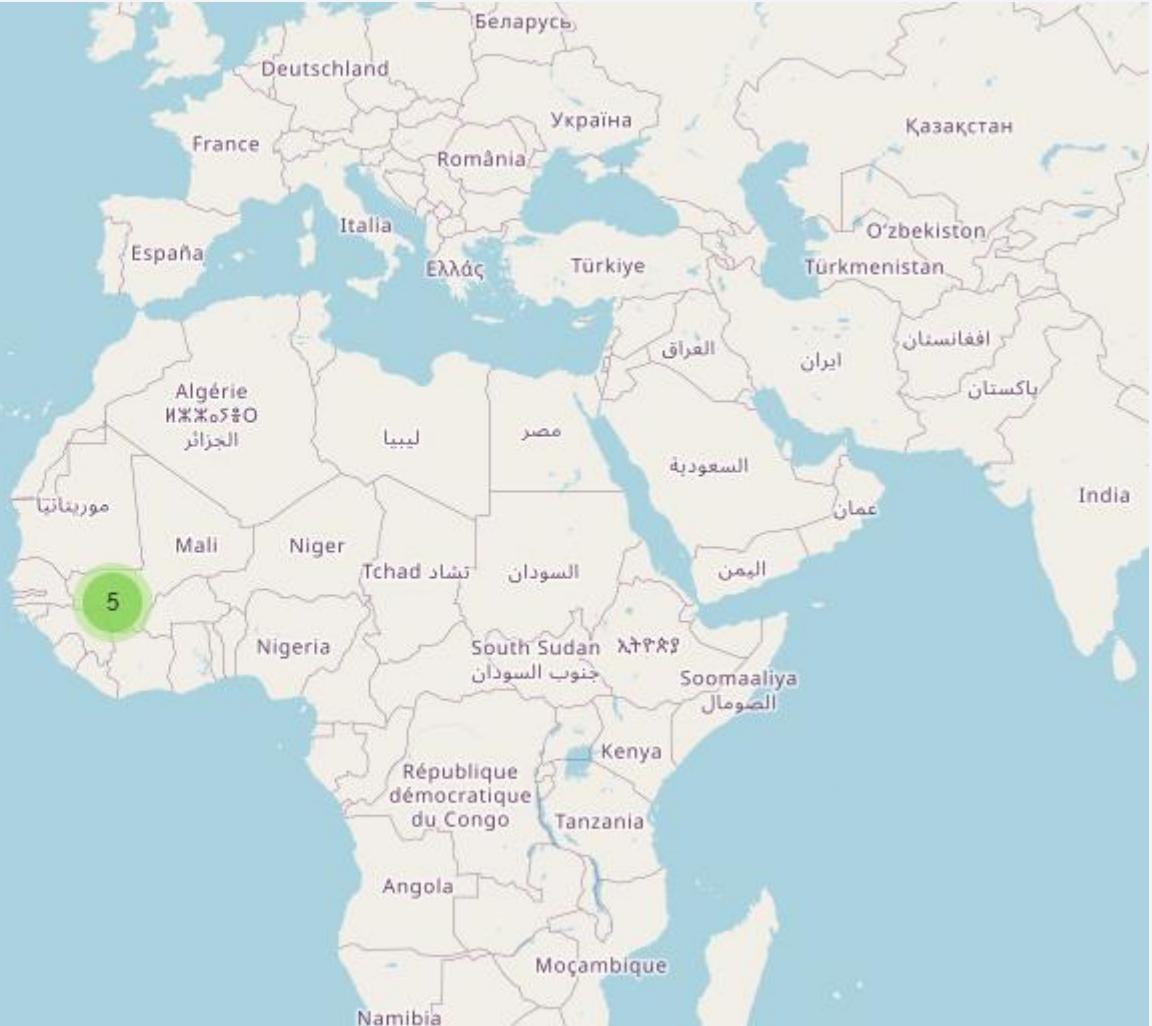
Section 3

Launch Sites Proximities Analysis

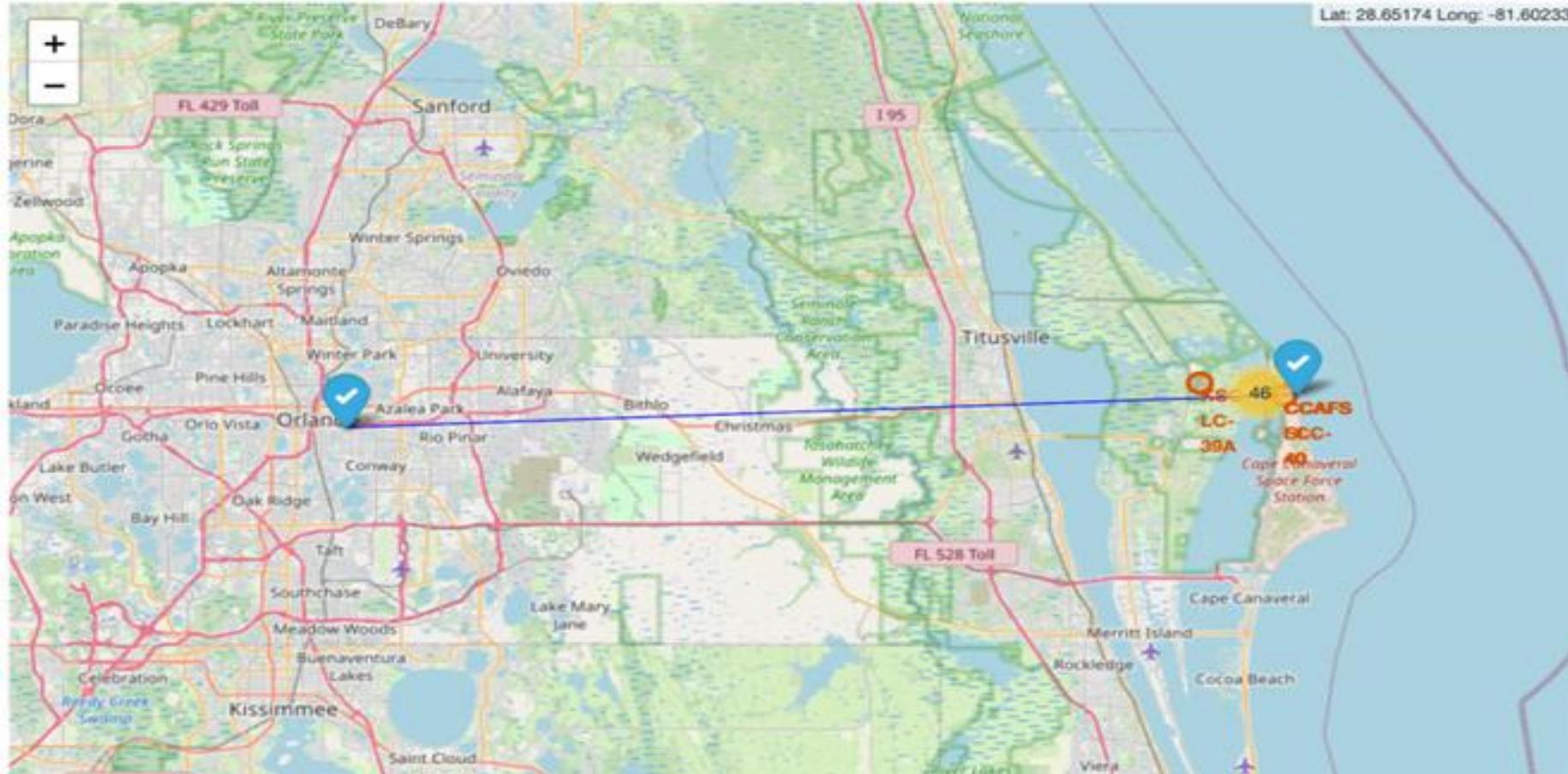
<Folium Map Screenshot 1>



<Folium Map Screenshot 2>

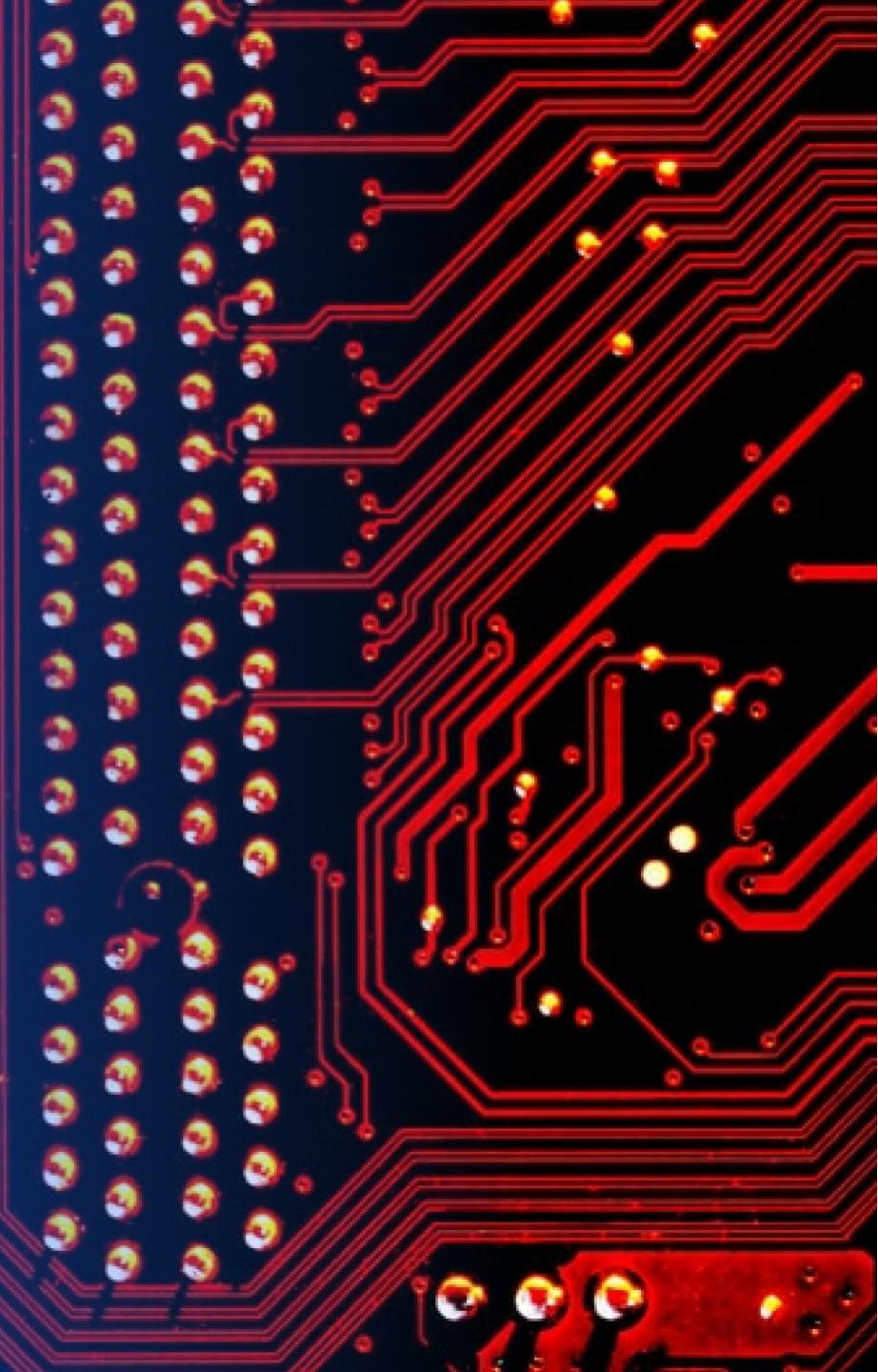


<Folium Map Screenshot 3>

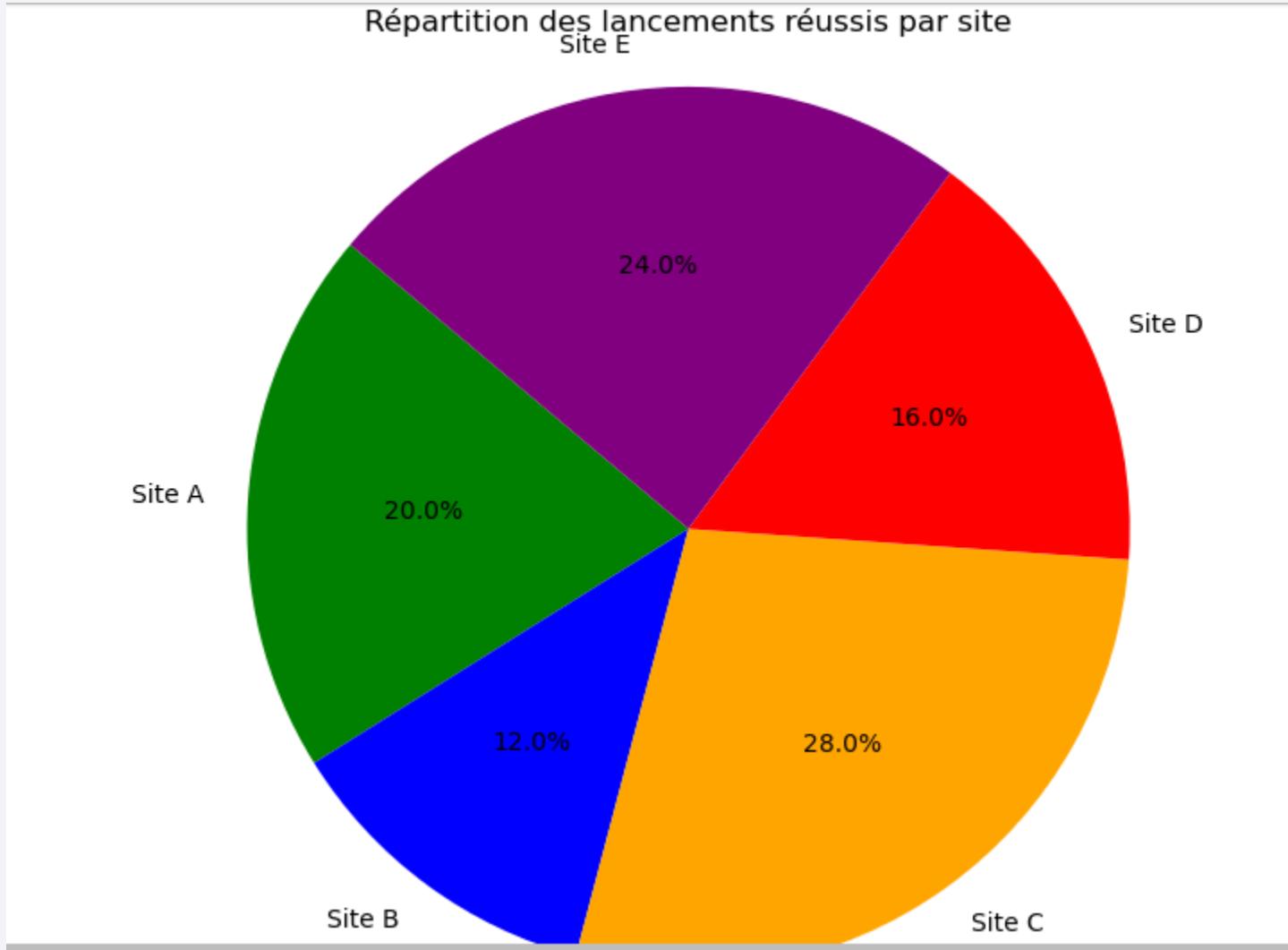


Section 4

Build a Dashboard with Plotly Dash



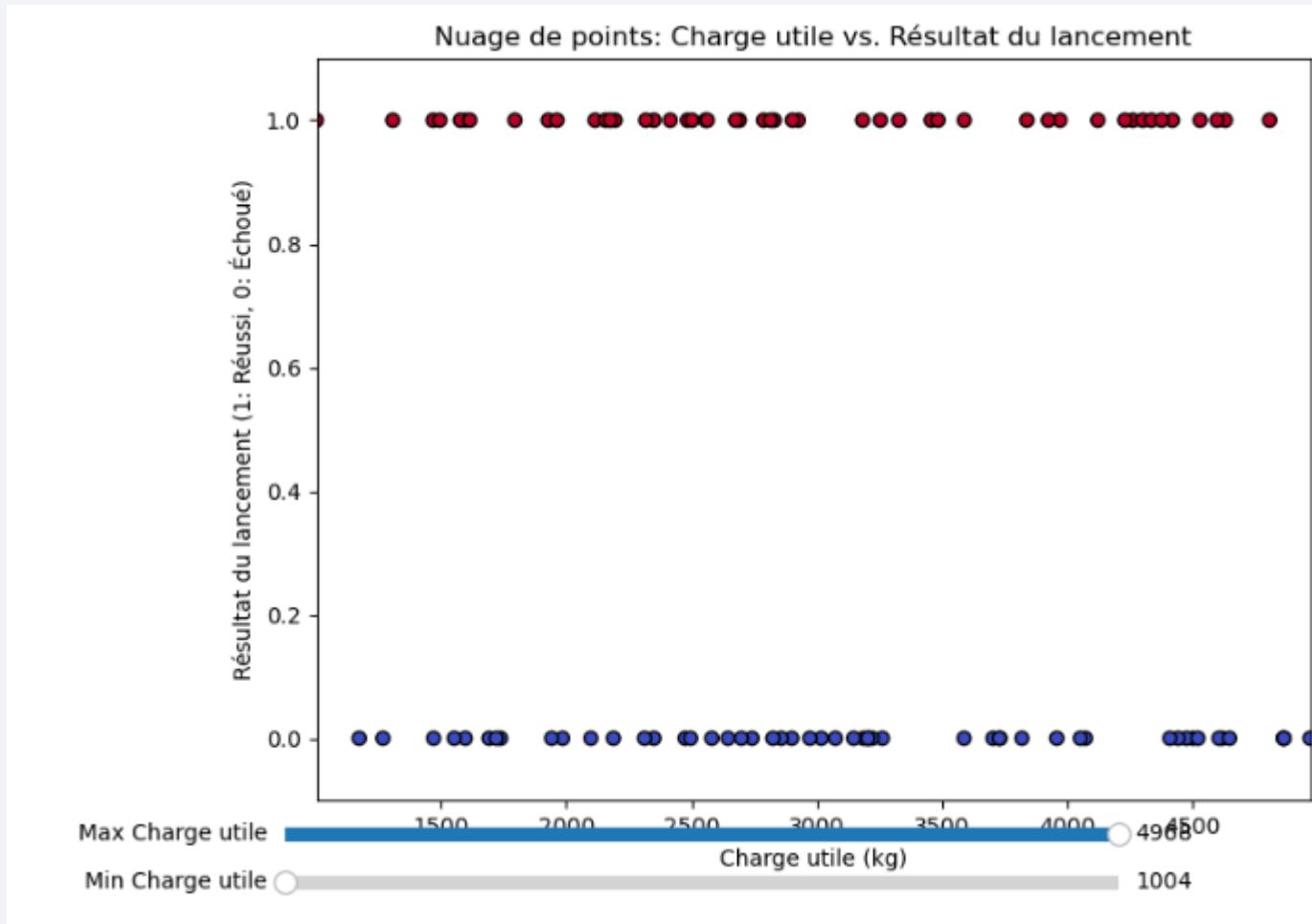
<Dashboard Screenshot 1>



<Dashboard Screenshot 2>



<Dashboard Screenshot 3>

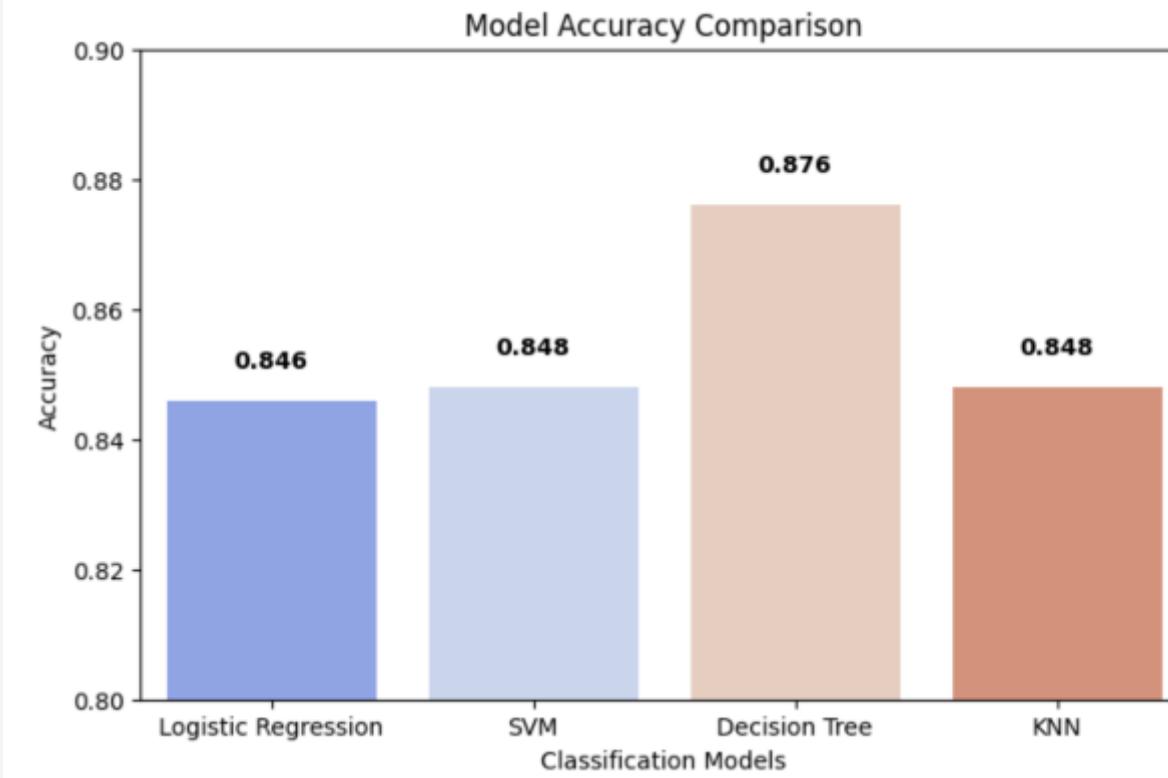


The background of the slide features a dynamic, abstract design. It consists of several thick, curved lines in shades of blue and yellow, creating a sense of motion and depth. The lines curve from the bottom left towards the top right, with some lines being more prominent than others. The overall effect is reminiscent of a tunnel or a high-speed journey through a digital space.

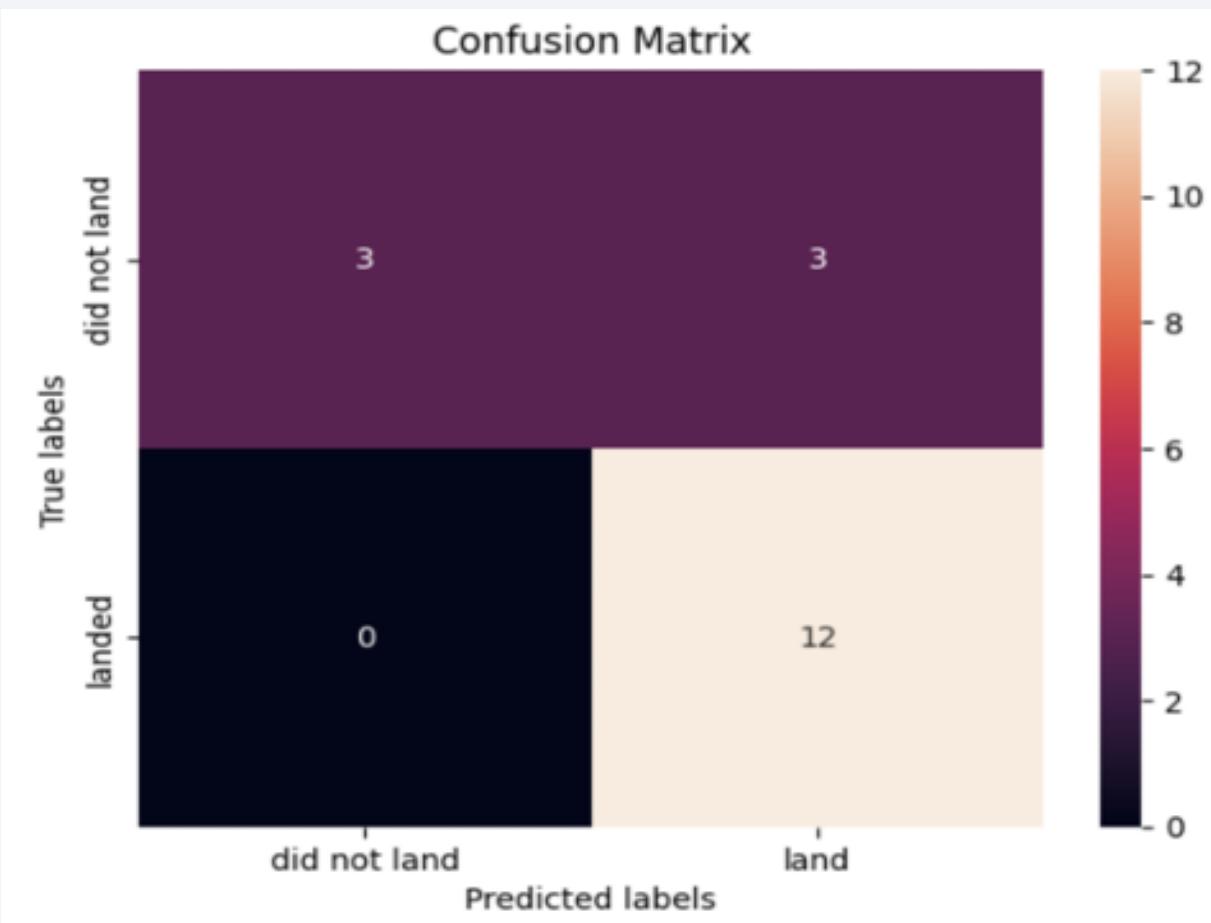
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix



Conclusions

- ◊ **Point 1** : Le modèle **Random Forest** a obtenu la meilleure précision parmi tous les modèles testés.
- ◊ **Point 2** : La **matrice de confusion** du modèle le plus performant montre un bon équilibre entre les vrais positifs et les vrais négatifs, avec un faible taux d'erreurs.
- ◊ **Point 3** : Les **résultats visuels** à travers les graphiques (barres, nuages de points) facilitent l'interprétation des performances des modèles.
- ◊ **Point 4** : L'analyse met en évidence l'importance de **l'ajustement des hyperparamètres** et de la qualité des données pour améliorer la précision des modèles.
- ◊ **Point 5** : Des étapes futures pourraient inclure **l'utilisation d'autres algorithmes avancés** (XGBoost, KNN) ou l'intégration d'approches de **deep learning**.

Thank you!

