WINNING SPACE RACE WITH DATA SCIENCE

Tuong Nguyen Vo Cat

31/12/2025

IBM Developer SKILLS NETWORK

TABLE OF CONTENTS

Executive Summary

Methodologies

- **Preprocessing**: filtering, dealing missing values, encoding, scaling
- **Train/Test Split**: Test size = 20%, Random state = 2
- **Models Evaluated**: Logistic Regression, SVM, Decision Tree, KNN
- **Hyperparameter Tuning**: GridSearchCV
- **Evaluation Metrics**: Accuracy, F1-score, Jaccard Score

Results

- All four models were successfully trained and evaluated
- **Decision Tree model achieved the best overall performance** after hyperparameter tuning
- Decision Tree showed superior balance across:
    - Accuracy
    - F1-score
    - Jaccard similarity

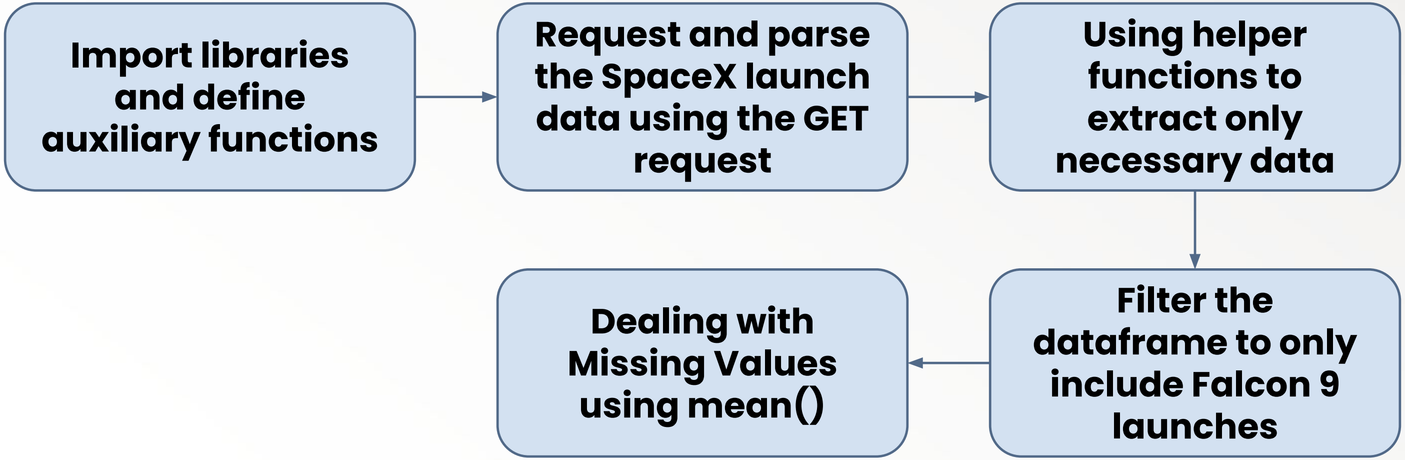
Introduction

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. This project aims to predict whether the first stage will successfully land, enabling competitors to better estimate launch costs and make informed bidding decisions.

3

Methodology

Data Collection – SpaceX API



Data Wrangling

Calculate the number of launches on each site using value_counts()

Calculate the number and occurrence of each orbit

Calculate the number and occurence of mission outcome of the orbits

Create a landing outcome label from Outcome column

EDA with Data Visualization

In this project I used three types of chart: Scatter plot, Bar chart and Line chart for different reasons:

- **Bar charts** were used to compare success rates across categorical variables (e.g., orbit types), making differences in performance easy to interpret.
- **Scatter plots** were used to explore relationships and patterns between numerical and categorical variables, helping identify trends, clusters, and outliers in launch behavior.
- **Line charts** were used to analyze temporal trends, allowing clear visualization of how launch success evolves over time.

EDA with SQL

- Retrieved unique launch sites and sampled launch records to understand overall dataset structure.
- Analyzed payload statistics (total, average, minimum, and maximum) across different customers and booster versions.
- Identified earliest successful ground pad landing to establish historical milestones.
- Filtered missions by landing outcome, payload range, year, and date interval to study performance under specific conditions.
- Examined booster version performance for successful drone ship landings and high-payload missions.
- Aggregated mission and landing outcomes to compare success and failure distributions over time.

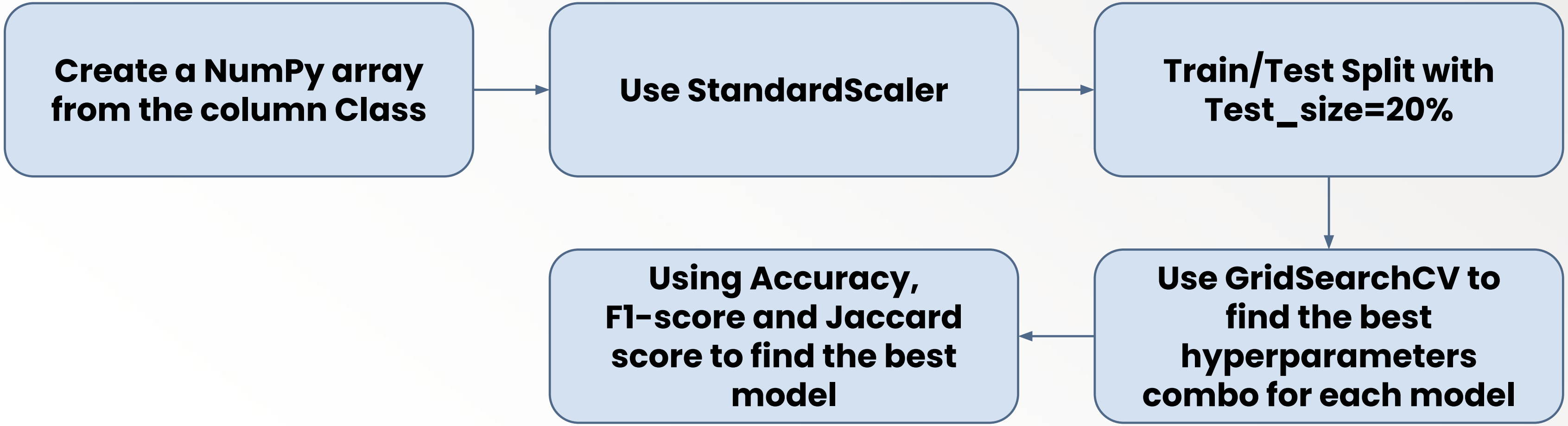
Build an Interactive Map with Folium

- I used circles to represent all launch sites, with the number indicating how many launches occurred at each site. When clicking on a site, red markers indicate failed launches, while green markers represent successful launches.
- Lines were used to clearly visualize the distance between each launch site and nearby locations (such as highways, railways, and cities), with distance labels to help users understand how far these locations are from the launch site.

Build a Dashboard with Plotly Dash

- I added a bar chart to display the number of successful launches across all launch sites. When a specific site is selected, the chart shows the distribution of successful and failed launches for that site. Bar charts are effective for visualizing the distribution of values within a categorical variable.
- I also added a scatter plot to examine the relationship between payload mass and launch success, helping to identify potential correlations and patterns.

Predictive Analysis (Classification)

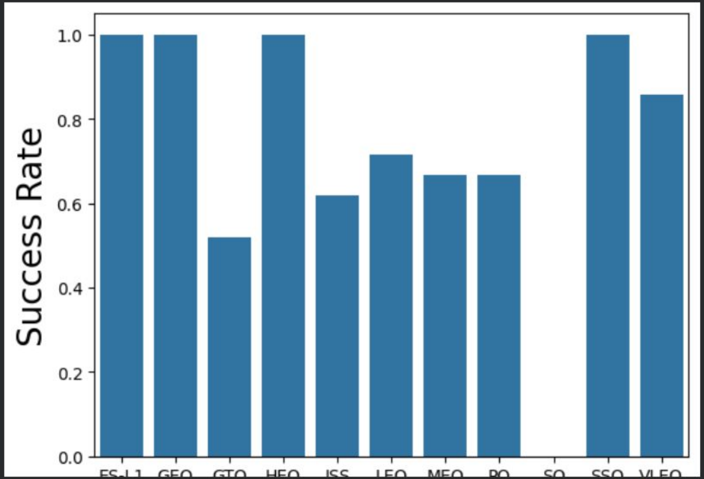


4

Results

EDA with Data Visualization

```
orbit_success_rate = df.groupby('Orbit')['Class'].mean().reset_index()
sns.barplot(x='Orbit', y='Class', data=orbit_success_rate)
plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



EDA with SQL

Commands + Code + Text ▷ Run all Connect

Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
%%sql
SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

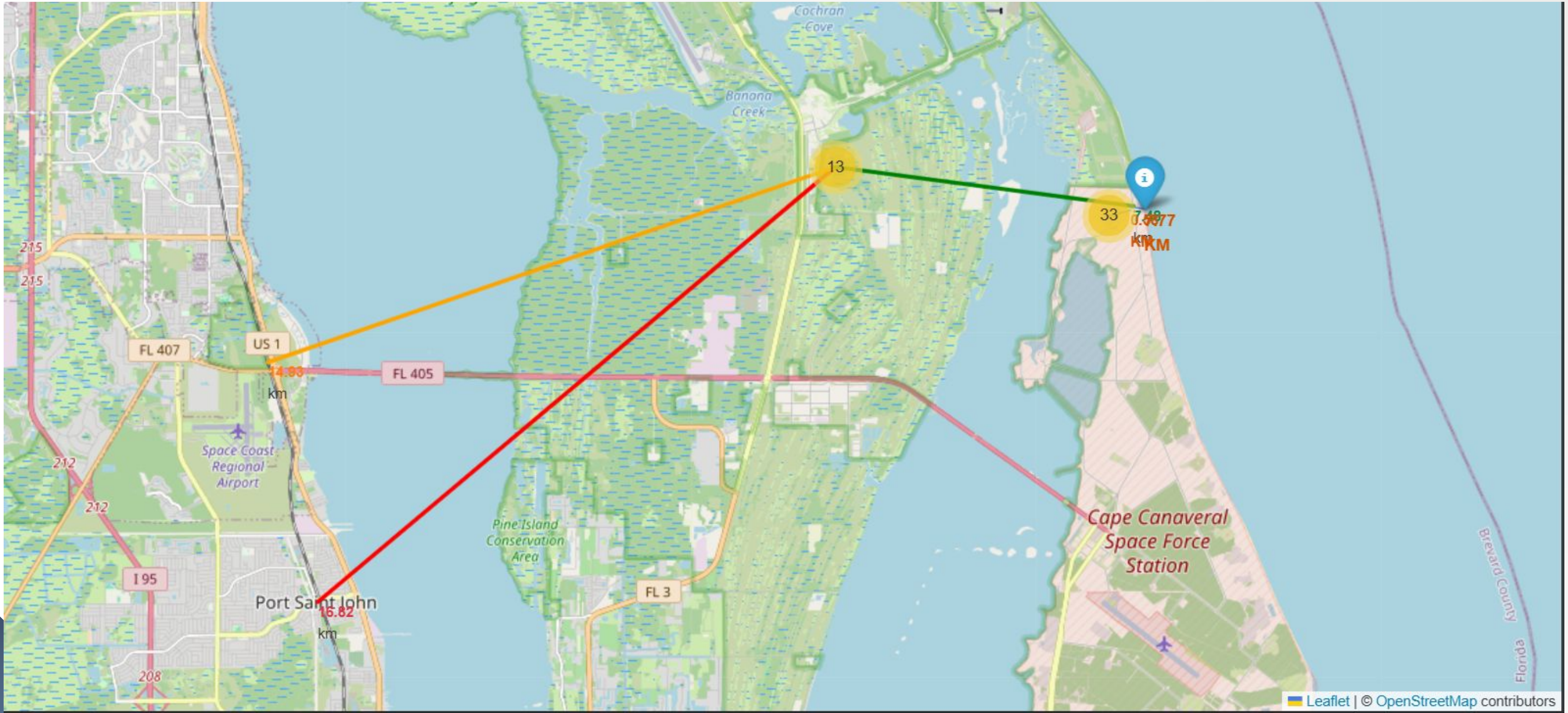
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

Task 3

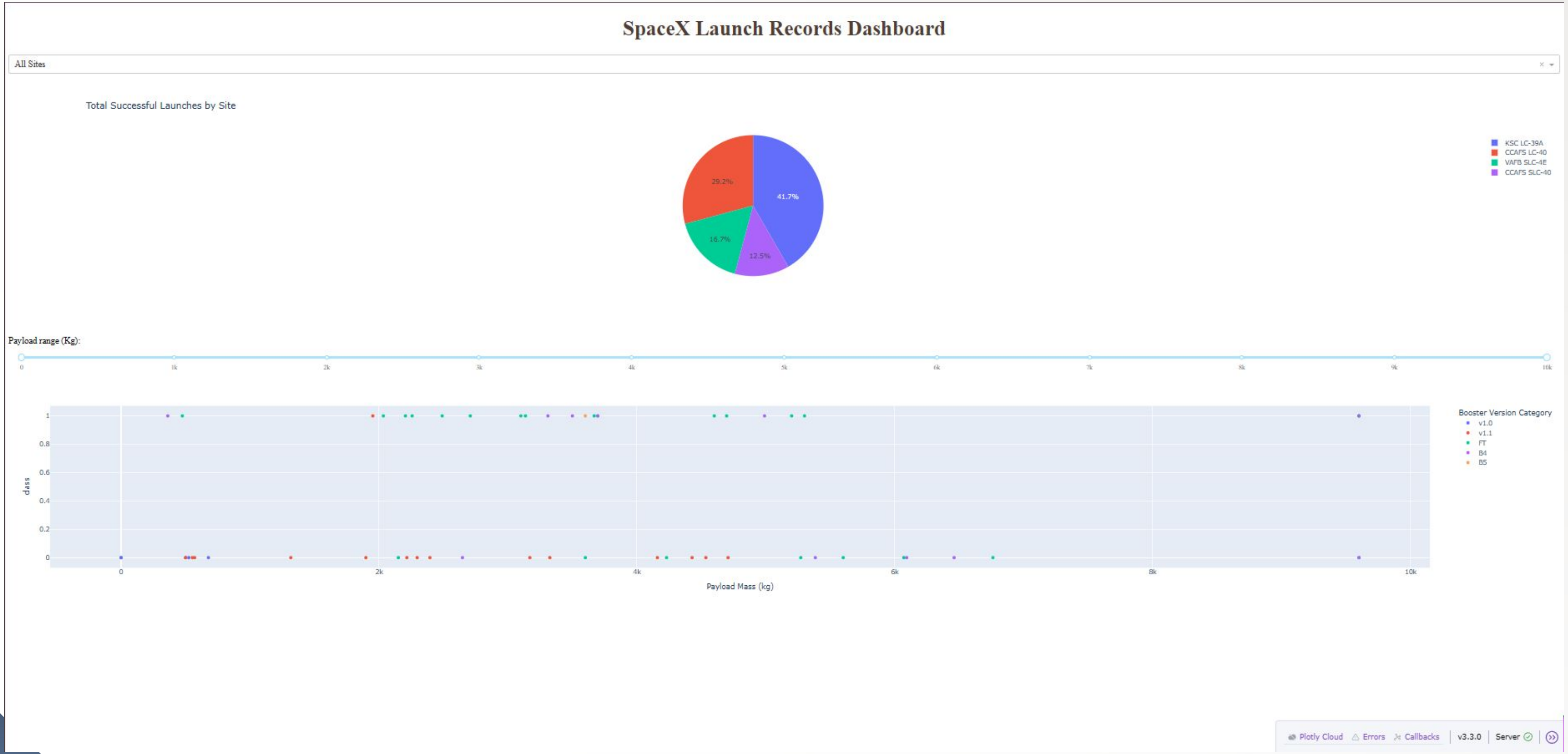
Display the total payload mass carried by boosters launched by NASA (CRS)

{} Variables Terminal

Build an Interactive Map with Folium



Build a Dashboard with Plotly Dash



Predictive Analysis (Classification)

Commands + Code + Text Run all Connect

```
                jaccard_score(Y, tree_cv.predict(X), average='binary'),
                jaccard_score(Y, knn_cv.predict(X), average='binary'),
               ]

f1_scores = [
            f1_score(Y, logreg_cv.predict(X), average='binary'),
            f1_score(Y, svm_cv.predict(X), average='binary'),
            f1_score(Y, tree_cv.predict(X), average='binary'),
            f1_score(Y, knn_cv.predict(X), average='binary'),
           ]

accuracy = [logreg_cv.score(X, Y), svm_cv.score(X, Y), tree_cv.score(X, Y), knn_cv.score(X, Y)]

scores = pd.DataFrame(np.array([jaccard_scores, f1_scores, accuracy]),
                      index=['Jaccard_Score', 'F1_Score', 'Accuracy'],
                      columns=['LogReg', 'SVM', 'Tree', 'KNN'])
scores
```

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.882353	0.819444
F1_Score	0.909091	0.916031	0.937500	0.900763
Accuracy	0.866667	0.877778	0.911111	0.855556

Conclusion

Variables Terminal

Conclusion

- Based on the scores of the Test Set, we can not confirm which method performs best.
- Same Test Set scores may be due to the small test sample size (18 samples). Therefore, we tested all methods based on the whole Dataset.
- The scores of the whole Dataset confirm that the best model is the Decision Tree Model. This model has not only higher scores, but also the highest accuracy.

Github

https://github.com/lilymont145/Applied-Data-Science-Capstone

THANKS!