Success factors - SpaceX

Soonyoul Kwon

Xyz@gmail.com

January 18, 2024

Table of Contents

EXECUTIVE SUMMARY

- Reusing the rockets is key success factor (KSF) of SpaceX
- To increase the cost saving effect, the business should focus its resource more on launching the rockets on the area with higher landing success rate.
- Not all features has had meaningful impact to the landing success.
- There are some factors that had significant impact on the success rate
  - CCAFS SLC 40 when the payload mass over 10,000.
  - The heavier payload mass, LEO, ISS, and PO.
  - KSC LC-39A has the highest success rate.

INTRODUCTION

- SpaceX has become an industry leader based on the comparative cost effectiveness.
- This is due to the reusability of the rockets
- Thus, the successful landing of them is critical
- I would like to discuss some success factors that affected the landing performance.

METHODOLOGY

- Data collection and wrangling
- Exploratory data analysis with SQL & Interactive map with Folium / Plotly Dash dashboard
- Predictive analysis

Data collection & wrangling methodology

- Data collected from SpaceX API
- Data normalization using .jason_normalize()
- Data filtering in pandas DataFrame
- Imputing missing data (i.e. mean value for null PayloadMass)
- Data scrapping using BeautifulSoup
- And Parsing data to dictionary to DataFrame
- Exploring datasets using df.methods

EDA and interactive visual analytics methodology

SQL

- Loading data from csv
- Converting to DataFrame
- Execute sql queries using %sql

Exploratory visualization (plots)

- matplotlib
- seaborn

Folium & Dash

- Locate coordinates in map using folium
- Calculate the distances between coordinates
- Draw lines between coordinates

- Creating a layout including dropdown implementation
- Adding callback functions using inputs and outputs

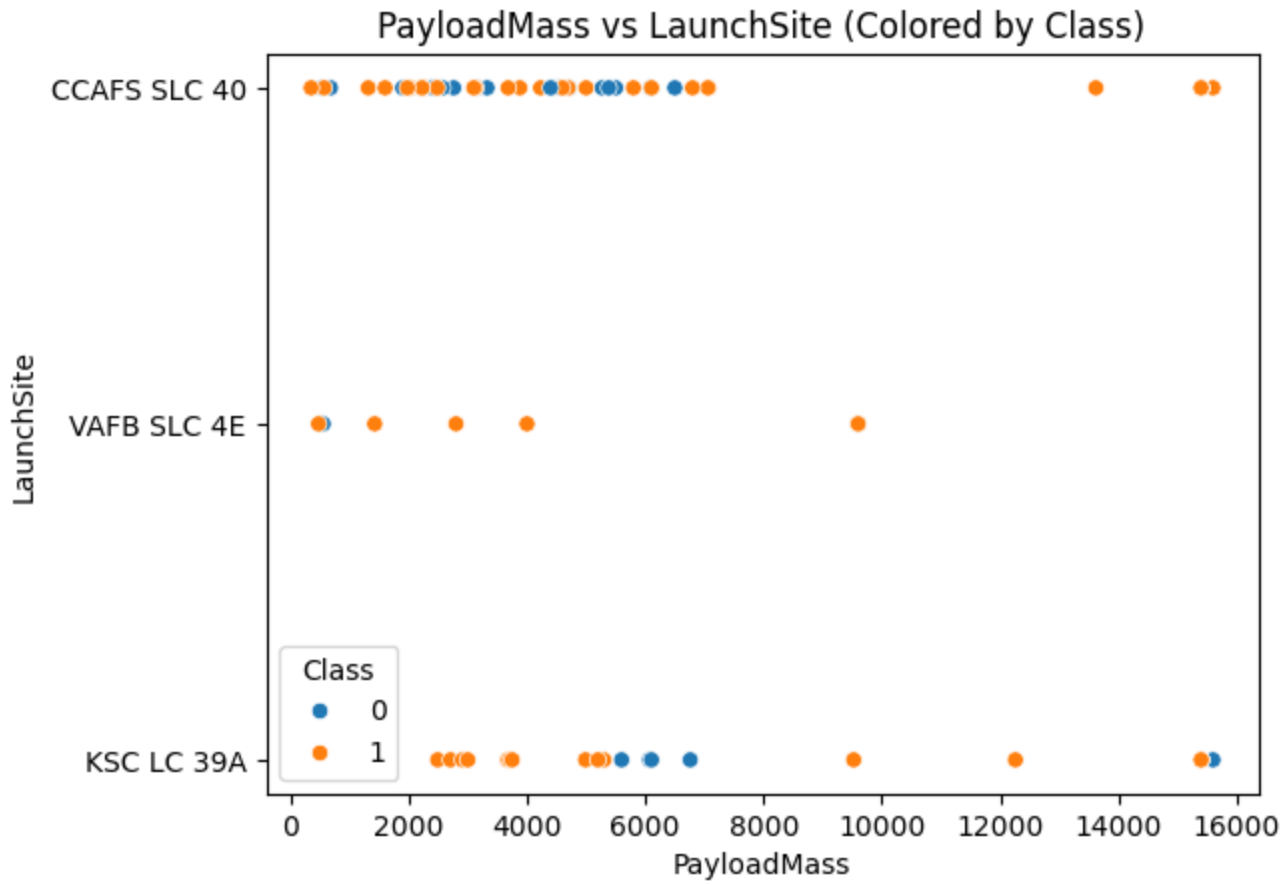
Predictive analysis methodology

- Split train and test data
- For each method
  (logistic regression, support vector machine,
  decision tree classifier, and k nearest neighbors)
  - Create object and fit the object to the train data to get the best parameters and accuracy score
  - Calculate the accuracy on the test data
  - Visualize false-positive/negative in confusion matrix

Results

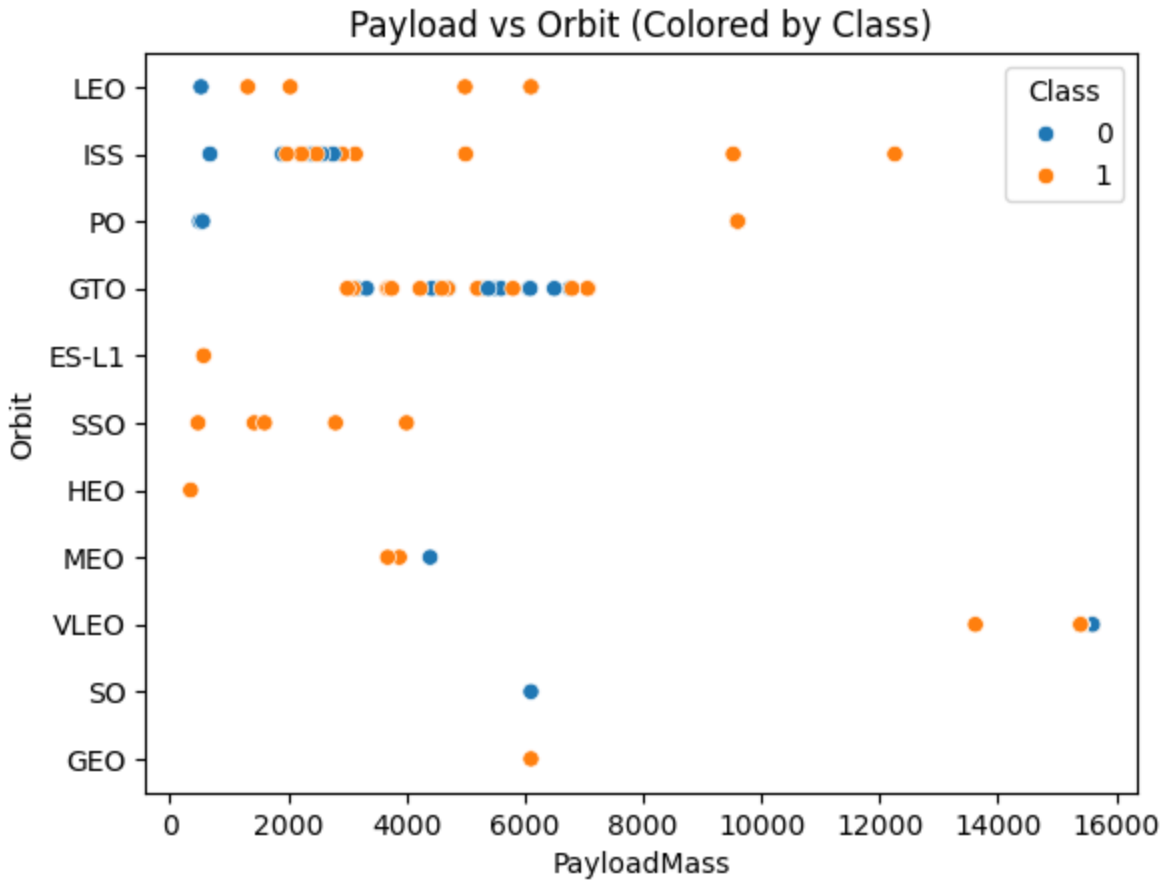
- EDA with visualization
- EDA with SQL
- Interactive map with Folium
- plotly Dash dashboard
- Predictive analysis (classification)

EDA with visualization results



For the launch site CCAFS SLC 40 only, high success rate for the payload mass over 10,000.

EDA with visualization results



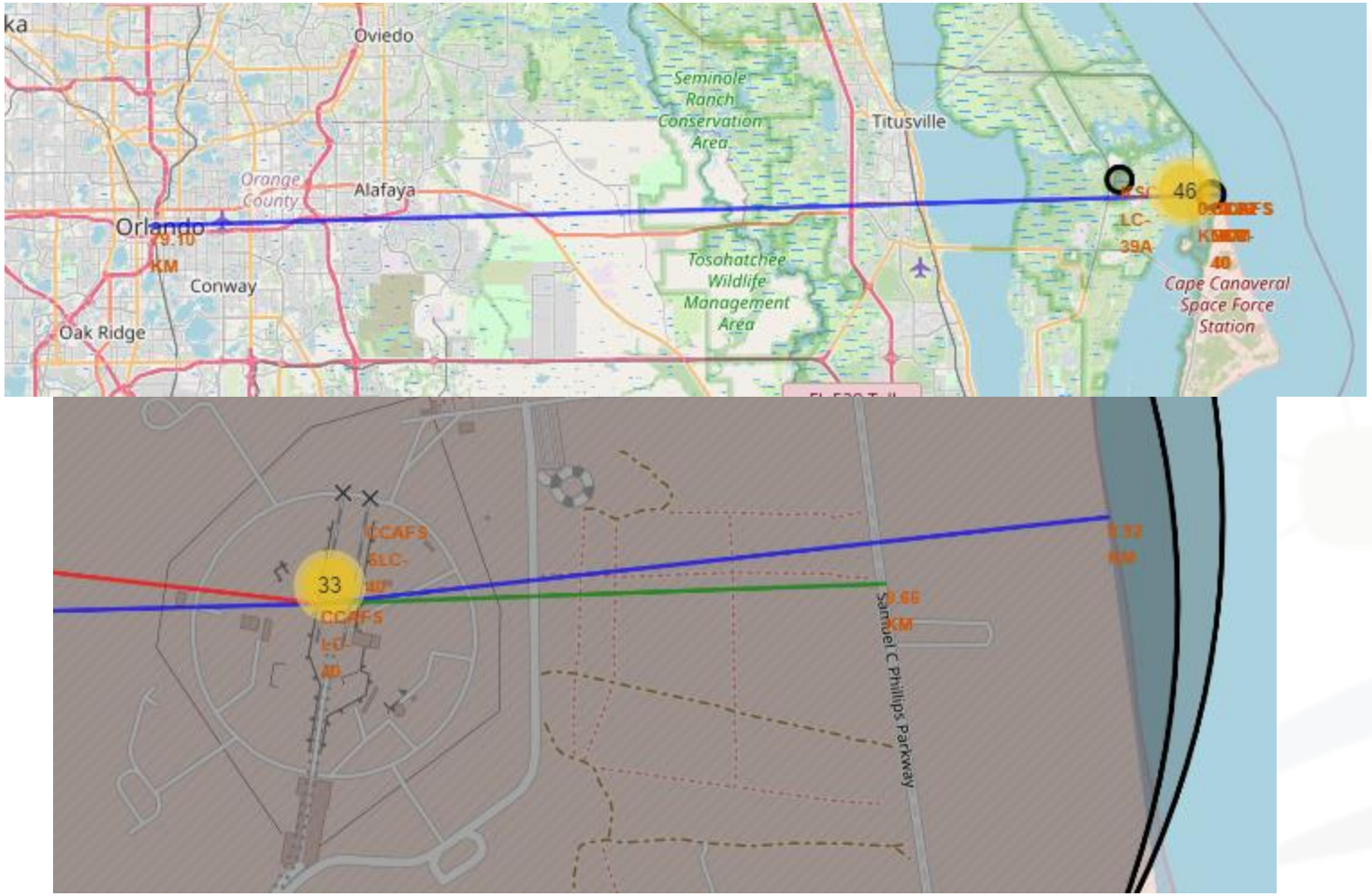
The heavier Payload mass, the higher success rate for the three orbits: LEO, ISS, and PO

EDA with SQL results

Launch_Site	Outcome	Cnt
CCAFS LC-40	Success	6
CCAFS LC-40	Failure	6
CCAFS LC-40	Other	14
CCAFS SLC-40	Success	25
CCAFS SLC-40	Failure	2
CCAFS SLC-40	Other	7
KSC LC-39A	Success	20
KSC LC-39A	Failure	1
KSC LC-39A	Other	4
VAFB SLC-4E	Success	10
VAFB SLC-4E	Failure	1
VAFB SLC-4E	Other	5

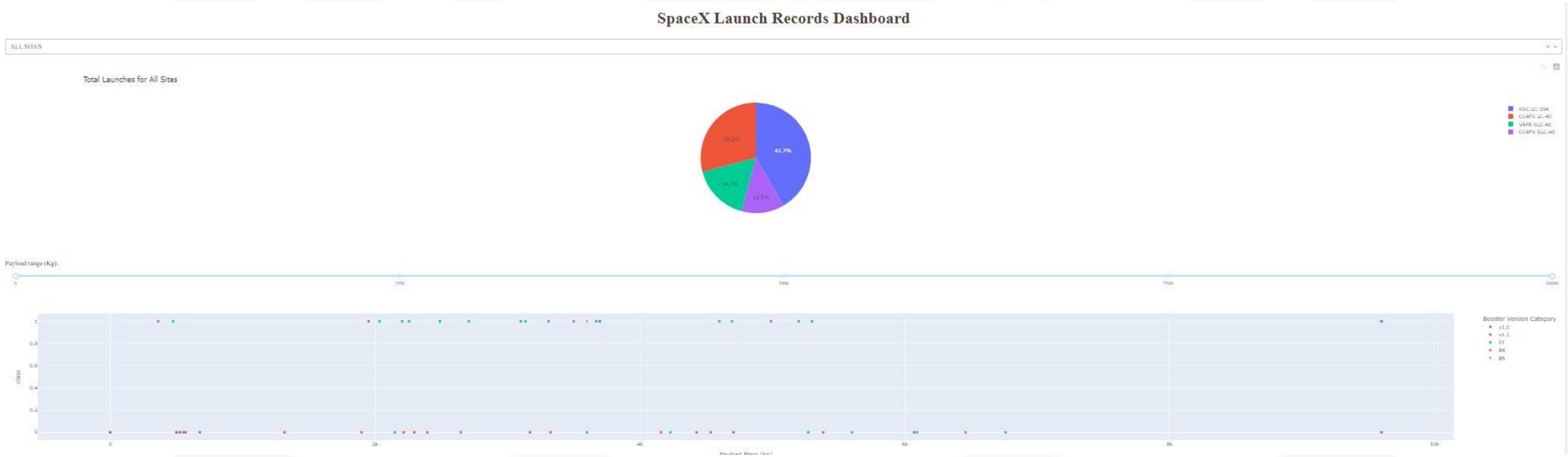
KSC LC-39A has the highest success rate.

Interactive map with Folium results



- Distances from CCAFS LC 40
  - To the closest shore: 0.92km
  - To the closest highway: 0.66km
  - To the closest city (Orlando, FL): 9.10km

plotly Dash dashboard results



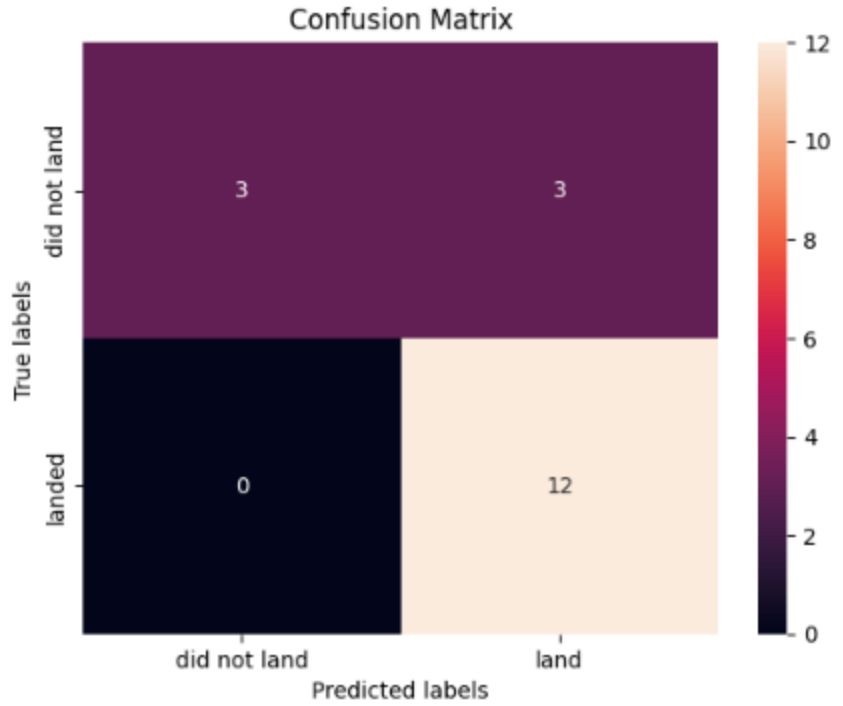
Predictive analysis (classification) results

```
accuracy_on_test_knn = knn_cv.score(X_test, Y_test)
print("Accuracy on Test Data:", accuracy_on_test_knn)
```

```
Accuracy on Test Data: 0.8333333333333334
```

We can plot the confusion matrix

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```




```
parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'p': [1,2]}

KNN = KNeighborsClassifier()
```

```
knn_cv = GridSearchCV(KNN, parameters, cv=10)
knn_cv.fit(X_train, Y_train)
```

```
/lib/python3.11/site-packages/threadpoolctl.py:1019: RuntimeWarning: libc not found. The ctype
too old for this OS.
  warnings.warn(
```

```
GridSearchCV(cv=10, estimator=KNeighborsClassifier(),
             param_grid={'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                         'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
                         'p': [1, 2]})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
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
```

All models tested had three false-negative but a decision tree classifier had one more false-negative than the other models.

K-nearest neighbors(knn) had the highest accuracy score among all models.

DISCUSSION

- Decision-tree classifier
  - The 'max_features' parameter should be set to an integer, a float, a string ('log2' or 'sqrt'), or None. 'auto' that were taught and provided from the instruction is replaced with None.

- Accuracy vs run-time
  - The accuracy of the model was not noticeably different among the models used.
  - Decision tree and kNN took more resource and time for fitting the models to the test data than the logistic regression and SVM.
  - Assuming the quality of dataset stay similar, the trade-off between accuracy and resource consumption should be considered when volume of data is significant.

OVERALL FINDINGS & IMPLICATIONS

Findings

- For the launch site CCAFS SLC 40 only, high success rate for the payload mass over 10,000.
- The heavier Payload mass, the higher success rate for the three orbits: LEO, ISS, and PO.
- KSC LC-39A has the highest success rate.

Put more resources to the combination of the launch site, payload and the orbit that has had higher success rate.

CONCLUSION

- Among many factors (features), some were indifferent with respects to the success rate.
- However, some had definitely influenced more on the success rate of the landing scenario.
- So, the company should focus more on the combination of the success factors by putting more resources to them to increase a chance of reusing the rocket and reduce its cost.

APPENDIX

- Rocket launch data from SpaceX API
  - https://api.spacexdata.com/v4/launches/past
- List of Falcon 9 and Falcon Heavy launches Wikipage updated on 9th June 2021
  - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922