IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Ismi Ana>
<8th October, 2024>

Outline

Part 1

Executive Summary

Part 2

Introduction

Part 3

Methodology

Part 5

Conclusion

Part 4

Results

Part 6

Appendix

Part

1

Executive Summary

Part 1

Executive Summary

Summary of all results

Exploratory Data Analysis:

- Launch success has improved over time
- KSC LC 39A has the highest success rate among landing sites
- Orbits ES L1, GEO, HEO, and SSO have a 100% success rate

Visualization/ Analytics:

- Most launch sites are near the equator, and all are close to the coast

Predictive Analytics:

- All models performed similarly on the test set. The decision tree model slightly outperformed

Summary of methodologies

The research attempts to identify the factors for a successful rocket landing. To make this determination, the following methodologies where used:

- **Collect** data using SpaceX REST API and web scraping techniques
- **Wrangle** data to create success/fail outcome variable
- Explore data with data visualization techniques, considering the following factors: payload, launch site, flight number and yearly trend
- **Analyze** the data with SQL, calculating the following statistics: total payload, payload range for successful launches, and total # of successful and failed outcomes
- **Explore** launch site success rates and proximity to geographical markers
- **Visualize** the launch sites with the most success and successful payload ranges
- **Build** Models to predict landing outcomes using logistic regression, support vector machine (SVM), decision tree and K nearest neighbor (KNN)

Introduction

- **Project background and context**

SpaceX, a leader in the space industry, strives to make space travel affordable for everyone. Its accomplishments include sending spacecraft to the international space station, launching a satellite constellation that provides internet access and sending manned missions to space. SpaceX can do this because the rocket launches are relatively inexpensive ($62 million per launch) due to its novel reuse of the first stage of its Falcon 9 rocket. Other providers, which are not able to reuse the first stage, cost upwards of $165 million each. By determining if the first stage will land, we can determine the price of the launch. To do this, we can use public data and machine learning models to predict whether SpaceX or a competing company can reuse the first stage.

- **Problems**

1. How payload mass, launch site, number of flights, and orbits affect first stage landing success
2. Rate of successful landings over time
3. Best predictive model for successful landing (binary classification)

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Collect data using SpaceX REST API and web scraping techniques
- Perform data wrangling
  - Wrangle data by filtering the data, handling missing values and applying one hot encoding to prepare the data for analysis and modeling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build Models to predict landing outcomes using classification models. Tune and evaluate models to find best model and parameters

Data Collection - API

- **How data sets were collected:**

1. Request data from SpaceX API (rocket launch data)
2. Decode response using . json () and convert to a dataframe using . json_normalize
3. Request information about the launches from SpaceX API using custom functions
4. Create dictionary from the data
5. Create dataframe from the dictionary
6. Filter dataframe to contain only Falcon 9 launches
7. Replace missing values of Payload Mass with calculated .mean()
8. Export data to csv file

Data Collection - Web Scraping

- **Steps**

1. Request data (Falcon 9 launch data) from Wikipedia
2. Create BeautifulSoup object from HTML response
3. Extract column names from HTML table header
4. Collect data from parsing HTML tables
5. Create dictionary from the data
6. Create dataframe from the dictionary
7. Export data to csv file

Build an Interactive Map with Folium

Markers Indicating Launch Sites

- Added blue circle at NASA Johnson Space Center's coordinate with a popup label showing its name using its latitude and longitude coordinates
- Added red circles at all launch sites coordinates with a popup label showing its name using its name using its latitude and longitude coordinates

Colored Markers of Launch Outcomes

- Added colored markers of successful green) and unsuccessful red) launches at each launch site to show which launch sites have high success rates

Distances Between a Launch Site to Proximities

- Added colored lines to show distance between launch site CCAFS SLC 40 and its proximity to the nearest coastline, railway, highway, and city

Build a Dashboard with Plotly Dash

Dropdown List with Launch Sites

- Allow user to select all launch sites or a certain launch site Dashboard with Plotly Dash

Pie Chart Showing Successful Launches

- Allow user to see successful and unsuccessful launches as a percent of the total Scatter Chart Showing Payload Mass vs. Success Rate by Booster Version

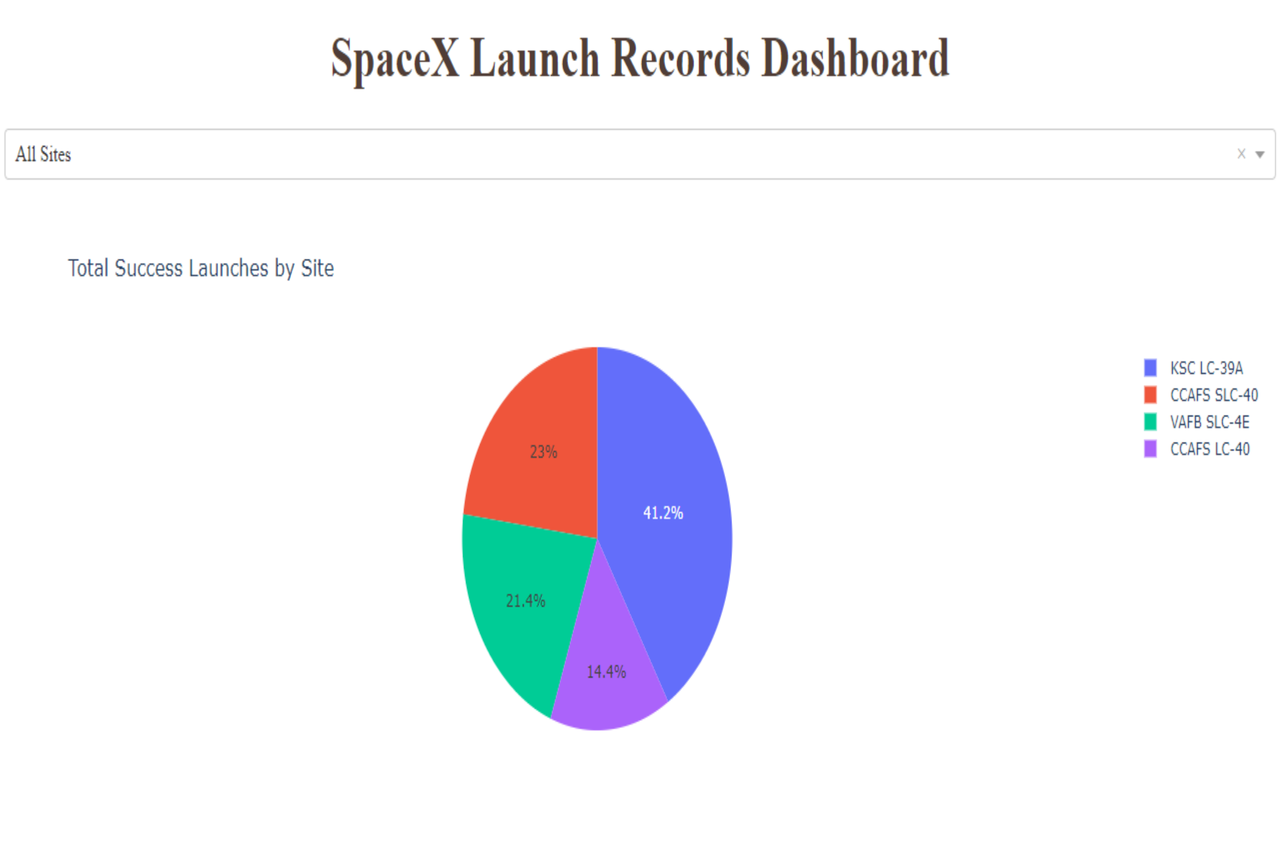
Slider of Payload Mass Range

- Allow user to select payload mass range

Scatter Chart Showing Payload Mass vs. Success Rate by Booster Version

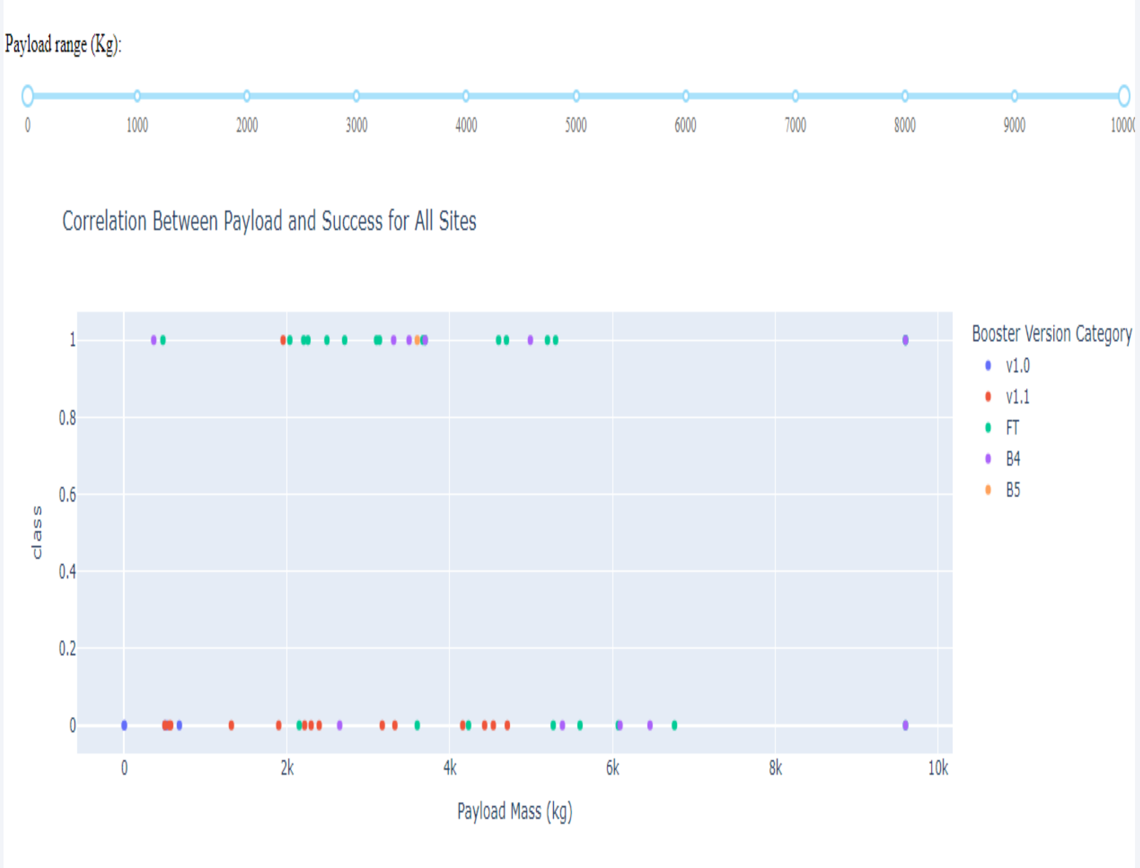
- Allow user to see the correlation between Payload and Launch Success

Build a Dashboard with Plotly Dash



SpaceX Launch Records Dashboard

All Sites



Payload range (Kg):

Predictive Analysis (Classification)

Charts

1. Create NumPy array from the Class column
2. Standardize the data with StandardScaler. Fit and transform the data.
3. Split the data using train_test_split
4. Create a GridSearchCV object with cv=10 for parameter optimization
5. Apply GridSearchCV on different algorithms: logistic regression (LogisticRegression()), support vector machine (SVC()), decision tree
6. (DecisionTreeClassifier()), K Nearest Neighbor (KNeighborsClassifier())
7. Calculate accuracy on the test data using .score() for all models
8. Assess the confusion matrix for all models
9. Identify the best model using Jaccard_Score, F1_Score and Accuracy

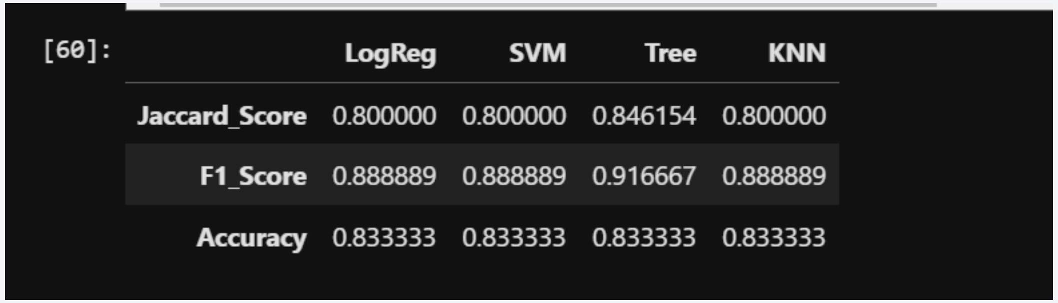
Section 5

Predictive Analysis (Classification)

Classification Accuracy

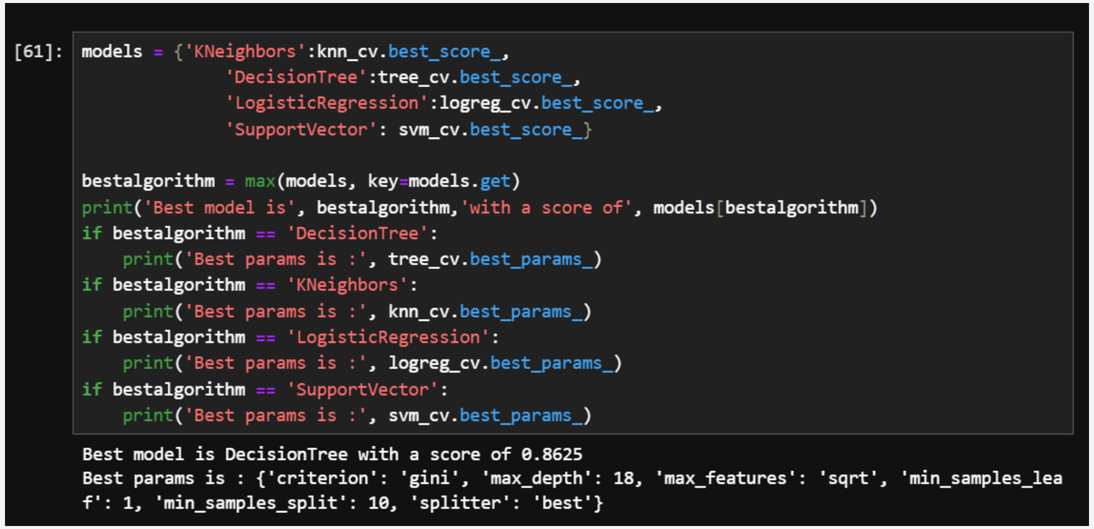
Accuracy

- All the models performed at about the same level and had the same scores and accuracy . This is likely due to the small dataset . The Decision Tree model slightly outperformed the rest when looking at .best_
- .best_score_ is the average of all cv folds for a single combination of the parameters



[60]:

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.846154	0.800000
F1_Score	0.888889	0.888889	0.916667	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333



```
[61]: models = {'KNeighbors':knn_cv.best_score_,
                    'DecisionTree':tree_cv.best_score_,
                    'LogisticRegression':logreg_cv.best_score_,
                    'SupportVector': svm_cv.best_score_}

      bestalgorithm = max(models, key=models.get)
      print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
      if bestalgorithm == 'DecisionTree':
          print('Best params is :', tree_cv.best_params_)
      if bestalgorithm == 'KNeighbors':
          print('Best params is :', knn_cv.best_params_)
      if bestalgorithm == 'LogisticRegression':
          print('Best params is :', logreg_cv.best_params_)
      if bestalgorithm == 'SupportVector':
          print('Best params is :', svm_cv.best_params_)

      Best model is DecisionTree with a score of 0.8625
      Best params is : {'criterion': 'gini', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_lea
      f': 1, 'min_samples_split': 10, 'splitter': 'best'}
```

Confusion Matrix

Performance Summary

A confusion matrix summarizes the performance of a classification algorithm

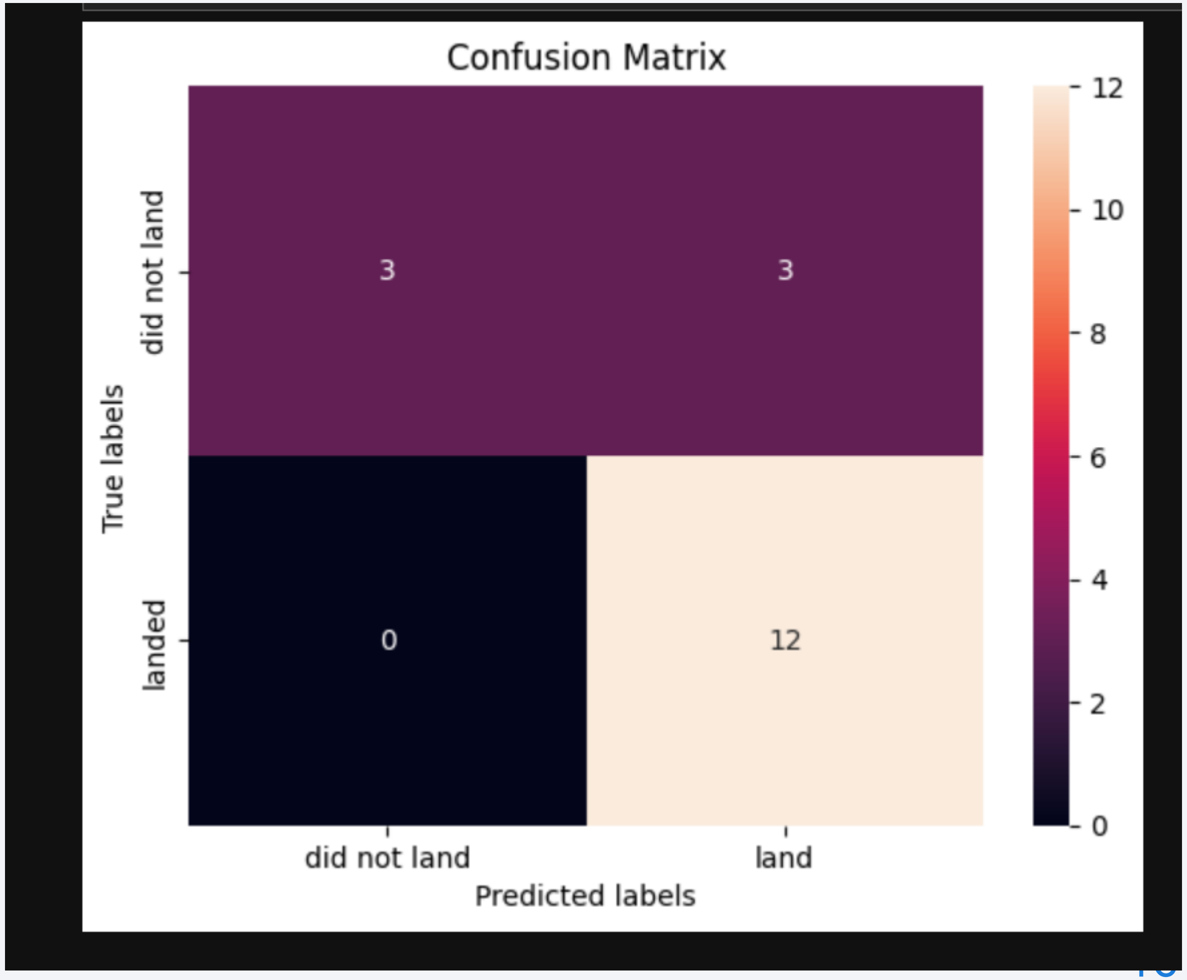
All the confusion matrices were identical

The fact that there are false positives (Type 1 error) is not good

Confusion Matrix Outputs:

12 True positive, 3 True negative, 3 False positive, 0 False Negative

- **Precision** = TP / (TP + FP) = 12 / 15 = .80
- **Recall** = TP / (TP + FN) = 12 / 12 = 1
- **F1 Score** = 2 * (Precision * Recall) / (Precision + Recall) = 2 * (.8 * 1) / (.8 + 1) = .89
- **Accuracy** = (TP + TN) / (TP + TN + FP + FN) = .833



Conclusions

✓ **Model Performance** : The models performed similarly on the test set with the decision tree model slightly outperforming

✓ **Equator** : Most of the launch sites are near the equator for an additional atural boost due to the rotational speed of earth whic h helps save the cost of putting in extra fuel and boosters

✓ **Coast** : All the launch sites are close to the coast

✓ **Launch Success** : Increases over time

✓ **KSC LC 39A** : Has the highest success rate among launch sites. Has a 100% success rate for launches less than 5,500 kg

✓ **Orbits** : ES L1, GEO, HEO, and SSO have a 100% success rate

✓ **Payload Mass** : Across all launch sites, the higher the payload mass (kg), the higher the success rate

Appendix

- https://github.com/izmian/IBM_DataScienceCapstone-SpaceX

Thank you!