IIITV-ICD

B.Tech. Year – III, 5th Semester, Academic Year (2022-23)

Subject Code: CS335

Subject Name: Machine Learning

Project Report

Group member: Alex Tamboli(202011071), Perin Mangukiya(202011055)
Title: SpaceX Falcon9 First Stage Landing Prediction

1. Introduction

Until the first successful landing of Falcon 9, rockets' first stage was not reusable. But the Falcon 9 landing requires precise accuracy so predicting before the launch whether the rocket would land accurately on the landing pad would be very beneficial. In this project, we will predict if the Falcon 9 first stage will land successfully. This pre-known data of rockets' first stage performance is beneficial to the engineering team as well. The prediction here is done on the data of previous rocket launches, and not on rocket science variables, so insights gained from this prediction and data analysis could help better the rocket's engineering aspects.

We tried so many permutations of algorithms to approach our problem to get better accuracy and successful prediction of the rocket's launch. Some of the algorithms used are Decision Tree, Support Vector Machine, Logistic and K-Nearest Neighbour. But the best accuracy we obtained from K-Nearest Neighbour, greater than 90%.

SpaceX 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 SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

2. Problem definition

Our model predicts the Falcon 9 landing on the basis of the data which consists of all past rockets data. Our model is trained with this data and all the prediction is done on the basis of this. Model requires some input data for its prediction such as payload mass (Amount of load rocket carries during its flight), orbit (eg: lower earth orbit, middle earth orbit), launch site (location of the landing), landing pad (Available locations), etc. Based on these inputs and the data on which model is trained after feature engineering, the model predicts whether the rocket will land successfully or not.

3. The learning system (K-Nearest Neighbour)

The K-Nearest Neighbour algorithm, also known as KNN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

While kNN is a universal function approximator under certain conditions, the underlying concept is relatively simple. kNN is an algorithm for supervised learning that simply stores the labeled training examples

$$\langle \mathbf{x}^{[i]}, y^{[i]} \rangle \in \mathcal{D} \quad (|\mathcal{D}| = n)$$

during the training phase. For this reason, kNN is also called a lazy learning algorithm.

What it means to be a lazy learning algorithm is that the processing of the training examples is postponed until making predictions 1 – again, the training consists literally of just storing the training data. Then, to make a prediction (class label or continuous target), the kNN algorithms find the k nearest neighbors of a query point and compute the class label (classification) or continuous target (regression) based on the k nearest (most "similar") points. The exact mechanics will be explained in the next sections. However, the overall idea is that instead of approximating the target function f(x) = y globally, during each prediction, kNN approximates the target function locally. In practice, it is easier to learn to approximate a function locally than globally.

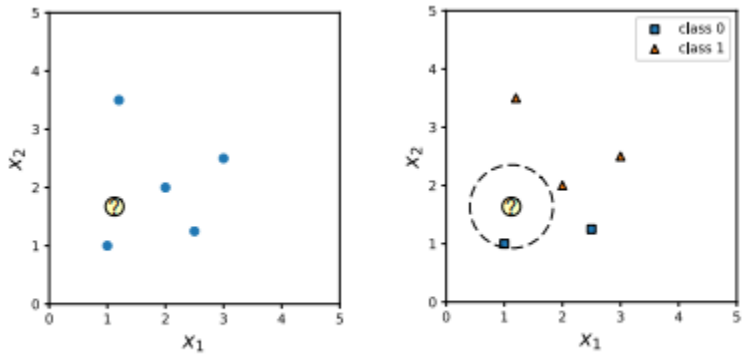


Figure 1: Illustration of the nearest neighbor classification algorithm in two dimensions

a. Representation

All the data that we collected from SpaceX API is transformed from text to categorical data on the basis of the range selected.

```
transform = preprocessing.StandardScaler()
x_scaled = transform.fit_transform(X)
```

X = pd.DataFrame(x_scaled)

Above code is used to transfrom the data, first StandardScaler function is used to scale the data in a given range then the data is transformed and stored in x_scaled, now it's data frame is stored in X.
– above code scales data to a selected range

b. System structure

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

c. The learning algorithm

Training Algorithm:

for i = 1, …, n in the n-dimensional training dataset D (|D| = n):

- store training example $\langle \mathbf{x}^{[i]}, f(\mathbf{x}^{[i]}) \rangle$

Prediction Algorithm:

closest_point := None
closest_distance := ∞

- for i = 1, …, n:
  - current_distance := d($x^{[i]}$+$x^{[q]}$)
  - current_distance < closest_distance:
    - closest_distance := current_distance
    - closest_point := $x^{[i]}$

prediction h($x^{[q]}$) is the target value of closest_point.

The default distance metric (in the context of this lecture) of nearest neighbor algorithms is the Euclidean distance (also called L^2 distance), which computes the distance between two points, $x^{[a]}$ and $x^{[b]}$:

$$d(\mathbf{x}^{[a]}, \mathbf{x}^{[b]}) = \sqrt{\sum_{j=1}^{m} \left(x_j^{[a]} - x_j^{[b]} \right)^2}$$

4. Experimental Evaluation/tests /Results

a. The data sets

Data is collected from Space RESTful API, which is recieved in form of JSON. Recived data is than converted into Pandas Dataframe. This data contains ID as information which API endpoint of that data (like getBoosterVersion() function). All the data is converted into text, all the data of Falcon 1 is removed as it's now reusable and our project is on reusable rockets. This process if followed by Data Wrangling in which all the noise data or null data is replaced by mean of the column of that attribute.

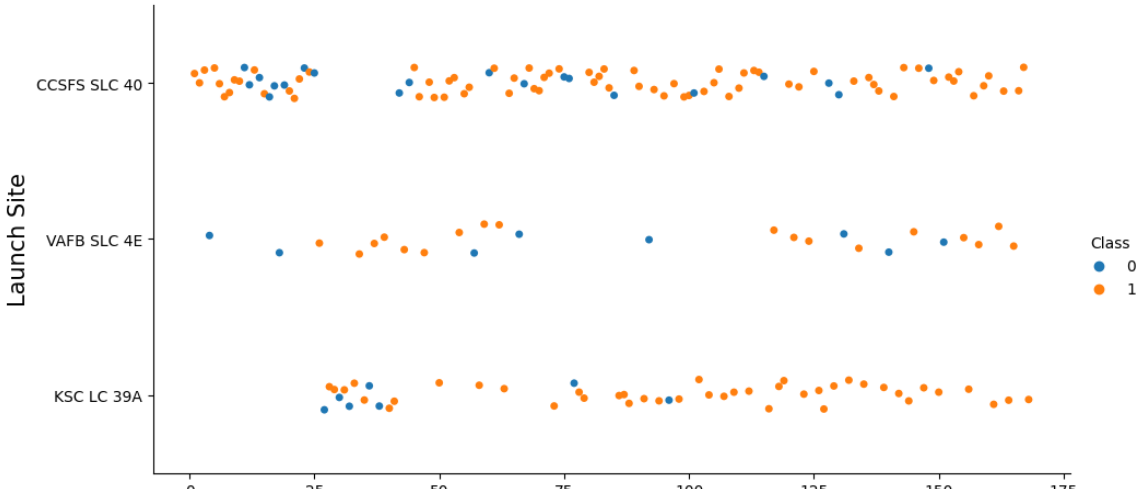
FlightNum	Date	BoosterVe	PayloadM	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPa	Block	ReusedCo	Serial	Longitude	Latitude
1	########	Falcon 9	8191.079	LEO	CCSFS SLC	None Non	1	FALSE	FALSE	FALSE		1	0	B0003	-80.5774	28.56186
2	########	Falcon 9	525	LEO	CCSFS SLC	None Non	1	FALSE	FALSE	FALSE		1	0	B0005	-80.5774	28.56186
3	########	Falcon 9	677	ISS	CCSFS SLC	None Non	1	FALSE	FALSE	FALSE		1	0	B0007	-80.5774	28.56186
4	########	Falcon 9	500	PO	VAFB SLC 4	False Ocea	1	FALSE	FALSE	FALSE		1	0	B1003	-120.611	34.63209
5	########	Falcon 9	3170	GTO	CCSFS SLC	None Non	1	FALSE	FALSE	FALSE		1	0	B1004	-80.5774	28.56186
6	########	Falcon 9	3325	GTO	CCSFS SLC	None Non	1	FALSE	FALSE	FALSE		1	0	B1005	-80.5774	28.56186
7	########	Falcon 9	2296	ISS	CCSFS SLC	True Ocea	1	FALSE	FALSE	TRUE		1	0	B1006	-80.5774	28.56186
8	########	Falcon 9	1316	LEO	CCSFS SLC	True Ocea	1	FALSE	FALSE	TRUE		1	0	B1007	-80.5774	28.56186
9	########	Falcon 9	4535	GTO	CCSFS SLC	None Non	1	FALSE	FALSE	FALSE		1	0	B1008	-80.5774	28.56186
10	########	Falcon 9	4428	GTO	CCSFS SLC	None Non	1	FALSE	FALSE	FALSE		1	0	B1011	-80.5774	28.56186
11	########	Falcon 9	2216	ISS	CCSFS SLC	False Ocea	1	FALSE	FALSE	FALSE		1	0	B1010	-80.5774	28.56186
12	########	Falcon 9	2395	ISS	CCSFS SLC	False ASDS	1	TRUE	FALSE	TRUE	5e9e3032	1	0	B1012	-80.5774	28.56186
13	########	Falcon 9	570	ES-L1	CCSFS SLC	True Ocea	1	TRUE	FALSE	TRUE		1	0	B1013	-80.5774	28.56186
14	########	Falcon 9	1898	ISS	CCSFS SLC	False ASDS	1	TRUE	FALSE	TRUE	5e9e3032	1	0	B1015	-80.5774	28.56186
15	########	Falcon 9	4707	GTO	CCSFS SLC	None Non	1	FALSE	FALSE	FALSE		1	0	B1016	-80.5774	28.56186
16	########	Falcon 9	2477	ISS	CCSFS SLC	None ASD	1	TRUE	FALSE	TRUE	5e9e3032	1	0	B1018	-80.5774	28.56186
17	########	Falcon 9	2034	LEO	CCSFS SLC	True RTLS	1	TRUE	FALSE	TRUE	5e9e3032	1	0	B1019	-80.5774	28.56186
18	########	Falcon 9	553	PO	VAFB SLC 4	False ASDS	1	TRUE	FALSE	TRUE	5e9e3033	1	0	B1017	-120.611	34.63209
19	########	Falcon 9	5271	GTO	CCSFS SLC	False ASDS	1	TRUE	FALSE	TRUE	5e9e3032	1	0	B1020	-80.5774	28.56186
20	########	Falcon 9	3136	ISS	CCSFS SLC	True ASDS	1	TRUE	FALSE	TRUE	5e9e3032	2	1	B1021	-80.5774	28.56186
21	########	Falcon 9	4696	GTO	CCSFS SLC	True ASDS	1	TRUE	FALSE	TRUE	5e9e3032	2	0	B1022	-80.5774	28.56186
22	########	Falcon 9	3100	GTO	CCSFS SLC	True ASDS	1	TRUE	FALSE	TRUE	5e9e3032	2	1	B1023	-80.5774	28.56186

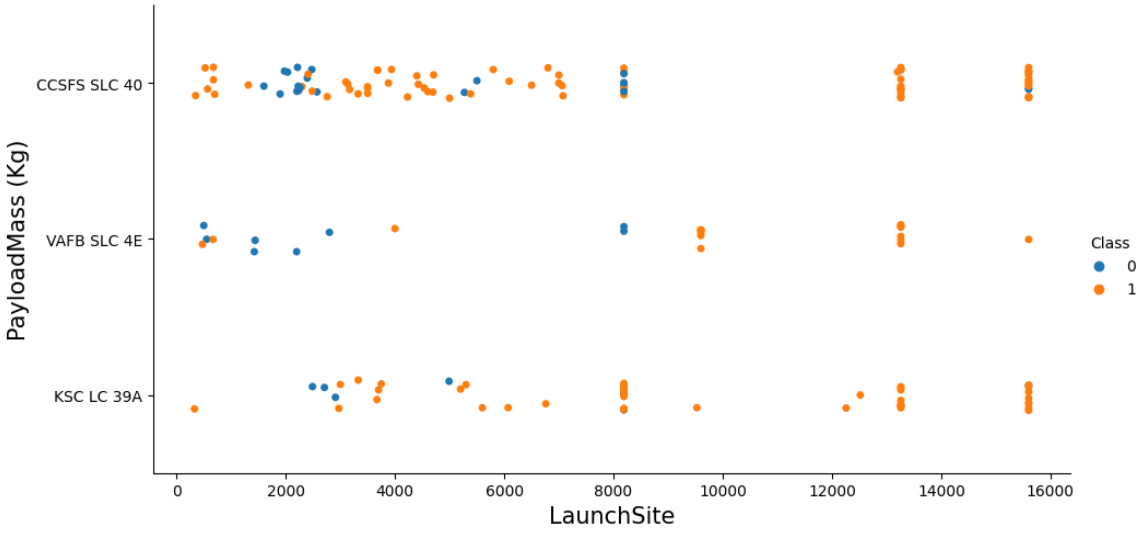
In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

So basically those outcomes are converted into training labels in which 1 means the booster will successfully land and 0 means it will land unsuccessfully. To achieve this we have used method .value_counts() on the column outcome to determine the number of landing_outcomes. Then assigning it to a variable landing _outcomes. While training the model the data is splited into 80% training set and 20% test set.

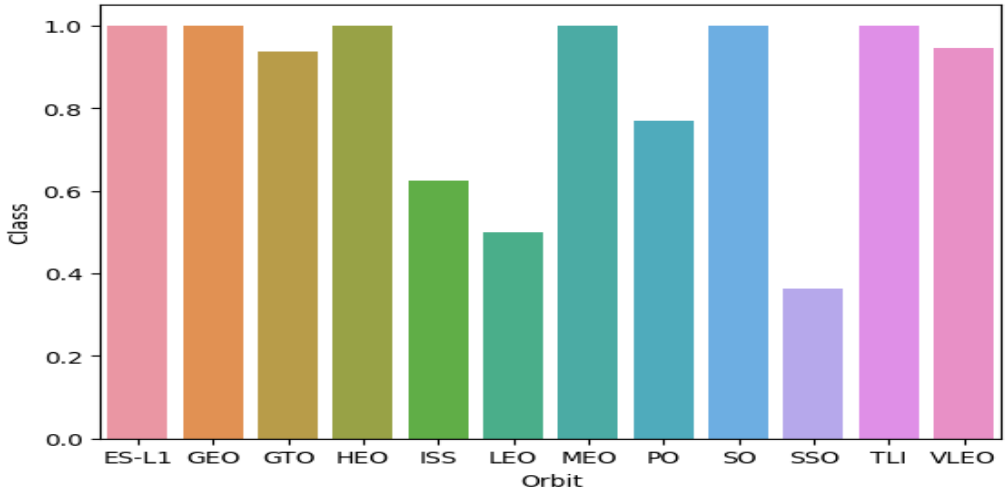
b. Data analysis

Flight Number vs Launch Site

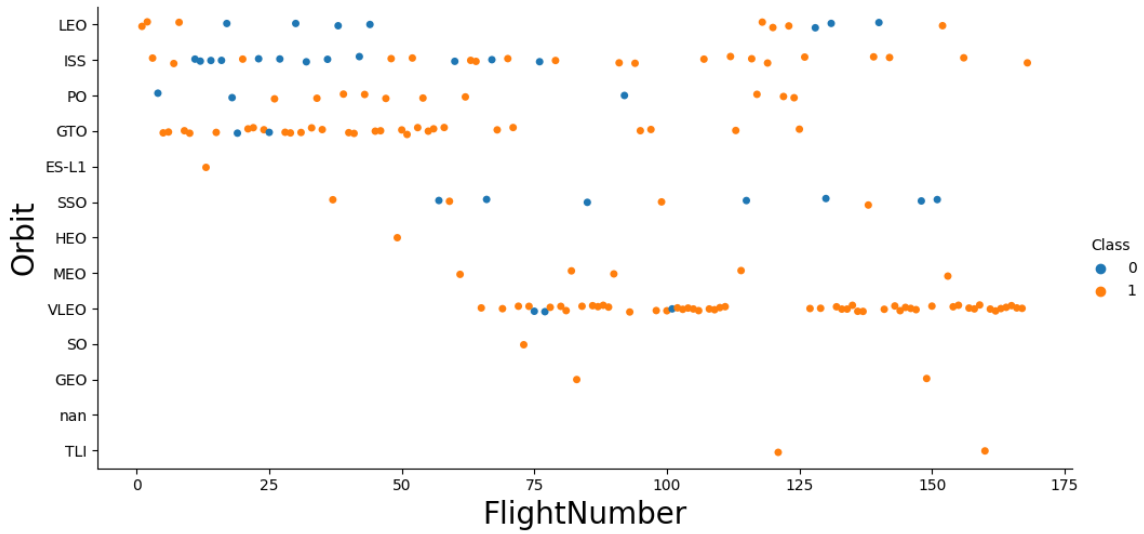




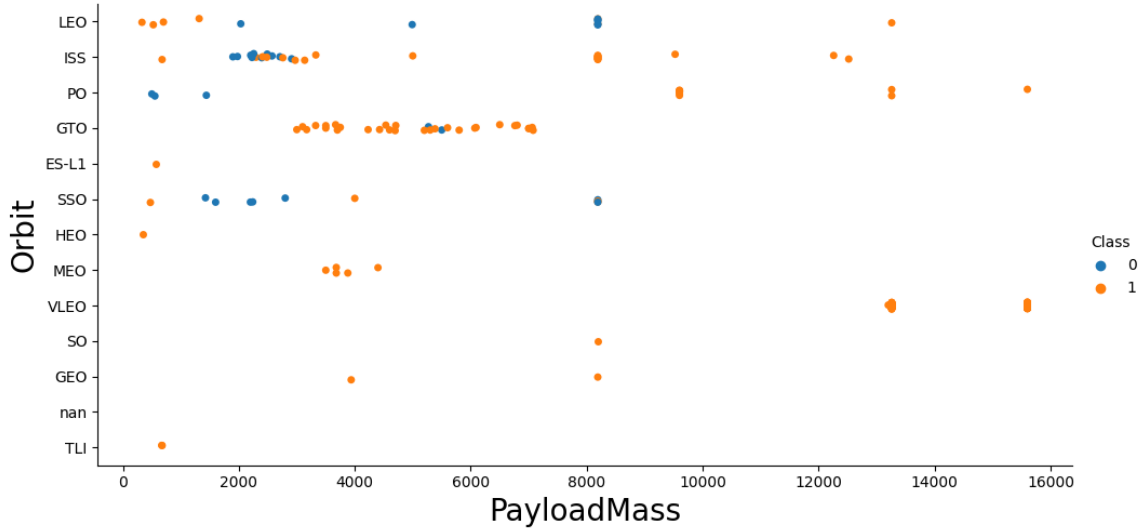
Orbit success rate



Orbit vs Flight Number



Orbit vs Payload Mass



After Feature Engineering

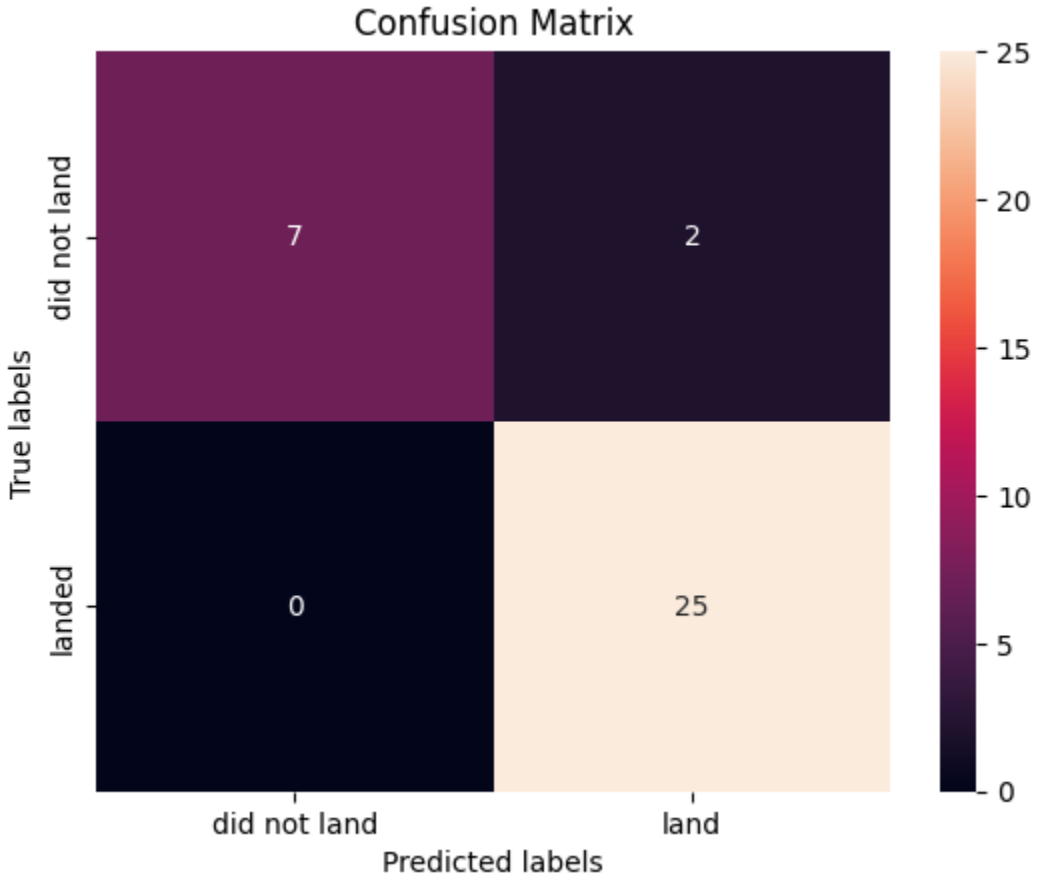
FlightNumber: It is count of the flight which has been taken till now.
PayloadMass: It is total extra weight it can carry into the orbit.
Orbit: It is the orbit name in which rocket is to be sent.
Launch Site: It's the location from which flight will
Grid Fins: They are the fins which are attached for landing to every booster.

Reused: It's a boolean value which tells whether booster are reused or not.
LandingPad: It is the name of the landing pad which is to be used while landing.
ReusedCount: Number of times booster is reused.
Serial: It is the serial number of the rocket.

c. Learning

Our data is splitted 80% to train our model and 20% to test our model. All out data is standardized and normalized using gridsearchCV. This process is done in completed in one minute. We have used MSE error and R^2. the mean squared error (MSE) or mean squared deviation (MSD) of an estimator measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. R-squared represents the fraction of variance of the actual value of the response variable captured by the regression model rather than the MSE which captures the residual error. Accuracy of our model is 94%.

d. Results



By using KNN algorithm we reached accuracy of 94%. When user input is given it is stored as last input and after applying KNN algorithm on we get the above confusion matrix. Random user input to test data accuracy is 88% so it is benefited to spacex engineers.

7. Comparing learning systems

To approach this problem we have applied KNN, SVM, Decision Tree, and Logistic regression. From all this approaches we got maximum accuracy in KNN. I

Algorithms	Accuracy obtained	Time taken (approx)
KNN	90% - 94%	1 minute
SVM	85% - 86%	7.5 hours
Logistic regression	88% - 84%	1 minute
Decision Tree	92% - 88%	1 minute

8. Concluding Remarks & Future work

We have successfully done for Falcon 9 rockets and our future plans is to apply the same for Falcon 1 rockets flight prediction. We will experiment to use rocket science and engineering features instead of past launches data parameters which are using for this project. Improvement of classification by using neural networks in future when data gets much bigger.

9. References

[1] Cognitive.ai - Data Science capstone

[2] H. Wang.: Nearest Neighbours without k: A Classification Formalism based on Probability, technical report, Faculty of Informatics, University of Ulster, N.Ireland, UK (2002)