AIRLINE TICKET PRICING

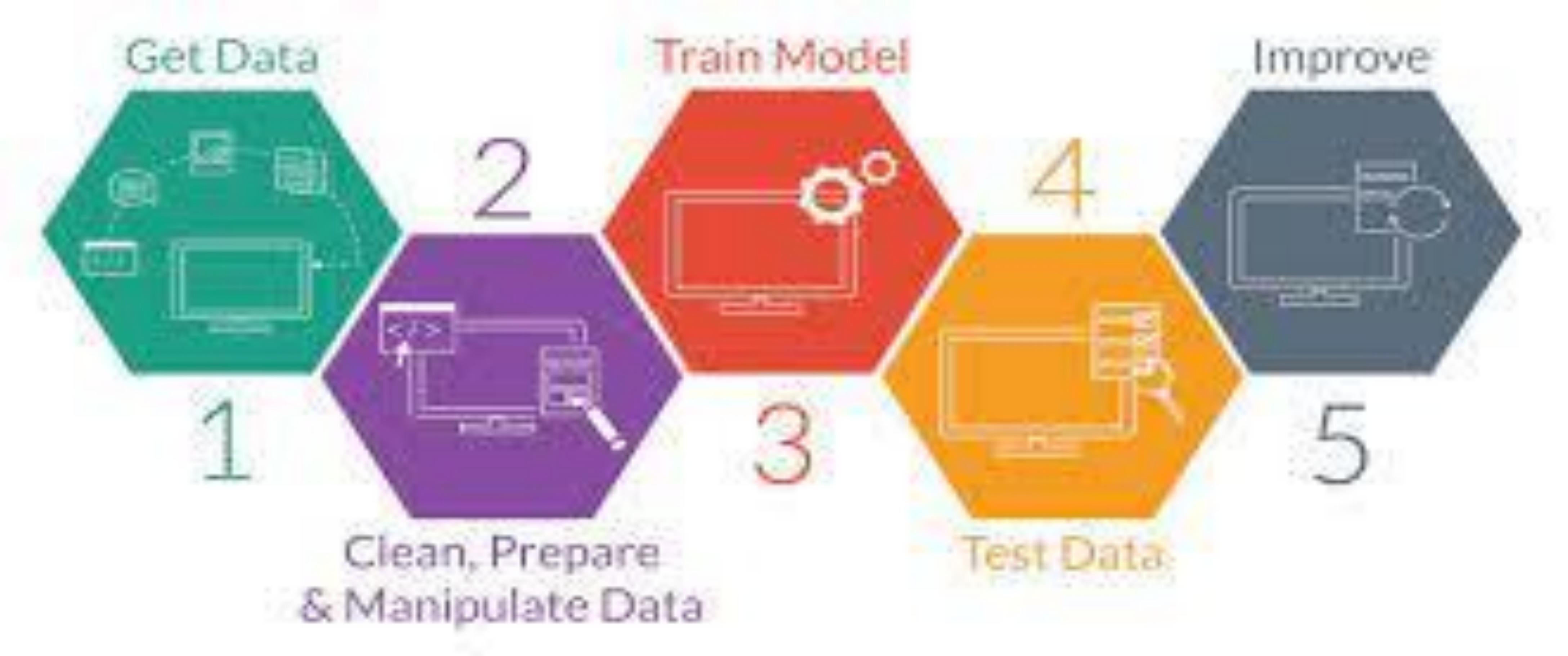
Using REINFORCEMENT LEARNING

INTRODUCTION

- Often , people book airline tickets on irregular basis i.e. on urgent requirement . At that time, they have to pay the same tickets at high price . So there is a need to predict that on which day the prices will be optimal.

- Here, Airline prediction is done using reinforcement leaning to predict the airline ticket prices.

WORKFLOW


Get Data
1
2
Clean, Prepare
& Manipulate Data
Train Model
3
4
Test Data
Improve
5

DATA FROM

- Google flights pricing data in the form of JSON files using the Google QPX Express API service.
- Collected every four hours, date range of 3/2016-3/2017 for one-way flights on the SFO → NYC route.

DATA MANIPULATION AND ANALYSIS

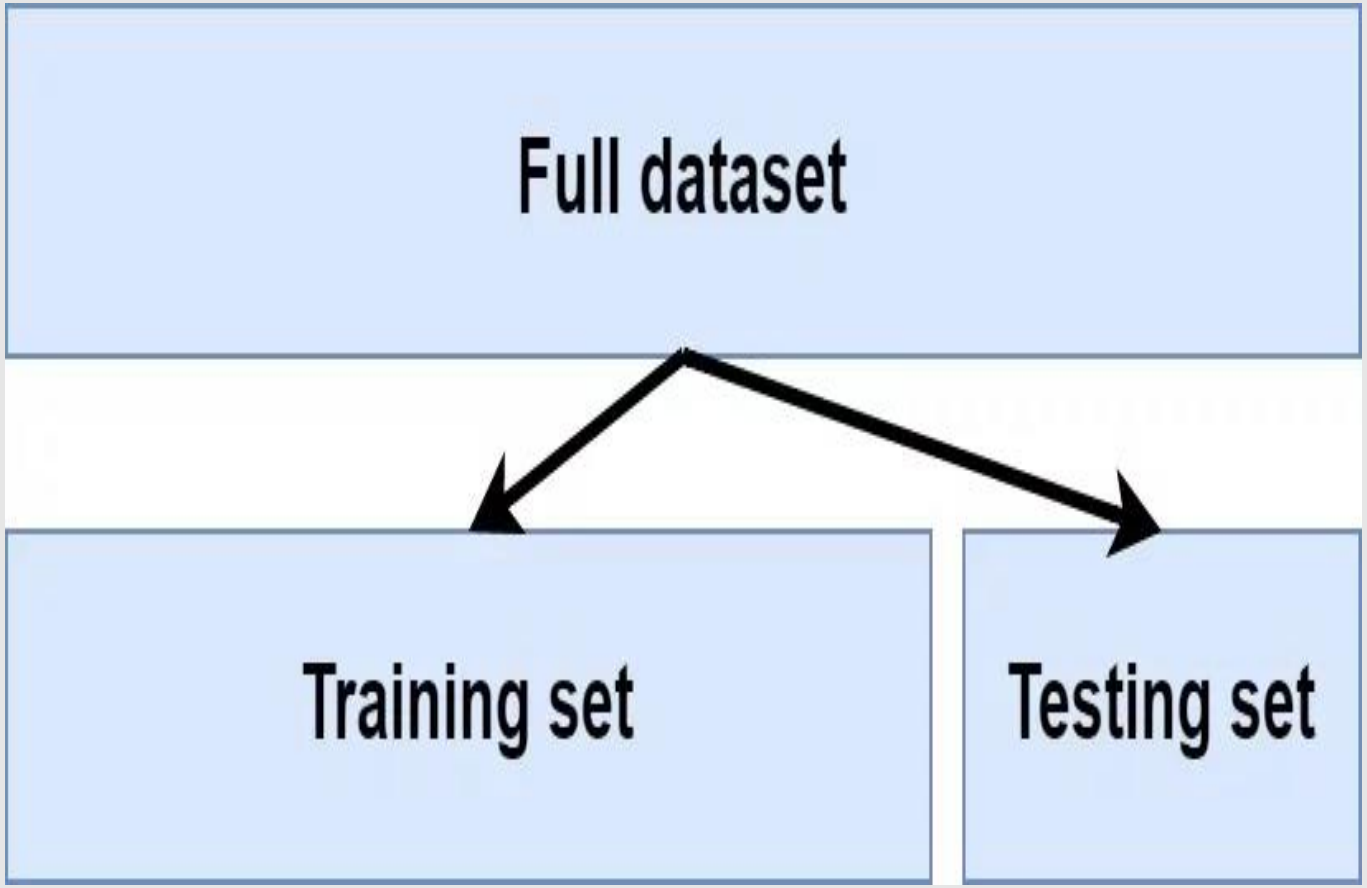
- **Data manipulation**
- Simple Imputer or fillna or drop functions(for handling missing values and noisy data).
- Normalizing the values (using min max techniques or z-score techniques,standardscaler)
- **Data Analysis** is done using matplotlib,plotly, seaborn,folium etc.
- This all is considered as data preprocessing.

FEATURE EXTRACTION

- Extracting important features from high dimensional data i.e. dropping unnecessary features.
- Converting categorical features to numerical values using LabelEncoder, OneHotEncoder etc.

TRAINING AND TESTING DATA

- **Training Data**: 586 flights, 97,848 data points (65% of the flights)
- **Dev Data**: 103 flights, 30,451 data points (10% of the flights)
- **Test Data**: 149 flights, 51,945 data points (25% of flights



MODEL SELECTION

Reinforcement Learning

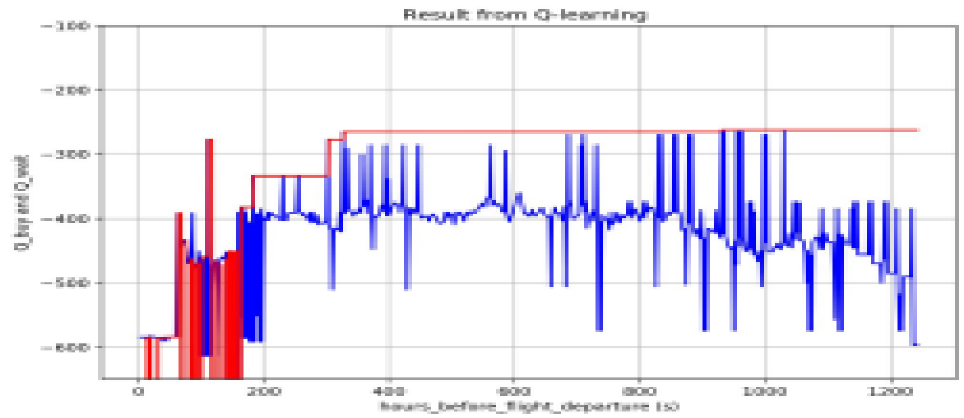
- **Queue Learning-**the sequence of actions that will eventually generate the maximum total reward.
- **Deep Q-Network-**a state-value function in a Q-Learning framework with a neural network.

MODEL EVALUATION

- Model evaluation is done using precision score ,AUC, Recall, Loss , Confusion Matrix, F1 score, Accuracy Score etc.

RESULTS

		Percent Savings	Maximum Savings	% Wait Decisions	% Correct Wait Decisions	Total Tickets	Total Flights
Train	Baseline	0.00%	16.23%	0.00%	0.00%	7402	51
	Q-Learning	-15.44%	16.23%	86.14%	36.34%	7402	51
	DQN	0.00%	16.23%	0.00%	0.00%	7402	51
Dev	Baseline	0.00%	26.25%	0.00%	0.00%	2720	10
	Q-Learning	-2.24%	26.25%	93.46%	33.40%	2720	10
	DQN		26.25%	0.00%	0.00%	2720	10
Test	Baseline	0.00%	23.77%	0.00%	0.00%	3183	10
	Q-Learning	10.17%	23.77%	94.85%	31.00%	3183	10
	DQN	0.00%	23.77%	0.00%	0.00%	3183	10



		Percent Savings	Maximum Savings	% Wait Decisions	% Correct Wait Decisions	Total Tickets	Total Flights
Train	Baseline	0.00%	16.23%	0.00%	0.00%	7402	51
	Q-Learning	-15.44%	16.23%	86.14%	36.34%	7402	51
	DQN	0.00%	16.23%	0.00%	0.00%	7402	51
Dev	Baseline	0.00%	26.25%	0.00%	0.00%	2720	10
	Q-Learning	-2.24%	26.25%	93.46%	33.40%	2720	10
	DQN	0.00%	26.25%	0.00%	0.00%	2720	10
Test	Baseline	0.00%	23.77%	0.00%	0.00%	3183	10
	Q-Learning	10.17%	23.77%	94.85%	31.00%	3183	10
	DQN	0.00%	23.77%	0.00%	0.00%	3183	10

Future Work and Conclusions

- Access to high quality flight data for multiple routes and over long time-periods is incredibly important to be able to train the data. It'll be interesting to extend the work to more routes, with more data.
- Deep Q-Networks can capture more nuanced states, but they are difficult to train and needs more parameter tuning.
- Users generally purchase a ticket on a route, not an individual flight. The current setup doesn't capture the majority use case.
- Currently, the agents interaction doesn't change the environment. However, in actual ticket purchase problems, the agents behavior can lead to tickets being sold out etc.