# Assignment 3
# Vraj Patel – 261022581

- ## Introduction:

For this assignment, we’re trying to implement offline Reinforcement Learning on two different algorithms such as Imitation learning and Fitted Q-Learning.
For the learning of the RL agent, I’ve decided to use Actor- Critic algorithm from the previous assignment as an expert agent and created a different dataset for different length of the episodes mainly 100, 250 and 500.
For the learning of the RL agent, we’re training on different types of dataset which are only Expert, only random and mixed dataset. The most interesting part of the above dataset is mixed dataset, where we’re taking half number of episodes from expert policy and the other half would be from random policy. This will provide more exploration to the RL agent.

- ## Imitation Learning:

Imitation Learning is simply imitating the behavior of the given policy, we’ve different types of datasets and we’re fitting the dataset into Logistic Regression function to get the action probabilities from that state and taking the action with the highest probability. For the experimentation we’re using Sklearn library to implement the Logistic function.
After the training of the Logistic function of the given dataset, the trained Logistic function is used to predict the action in the current state. For this I’ve decided not to discretize my states into 10 bins but instead I’ve taken the position, velocity, angular momentum, and pole degree as a state and trained my function on those features.

- ## Fitted Q – Learning:

Fitted Q -learning is same as Q – learning, however fitted Q-learning is offline learning meanwhile the later is online learning. As a result, fitted Q-learning is better algorithm than Q-learning since, the former can better approximate the state, action values compare to the later. For this experiment, I’ve use 10 bins as a result each feature of the Cart-pole environment will be discretized in to 10 bins which means in total it has 40 features which means in order to predict the state, action values it has to learn 40 weights.

$$y_i \;= r(s_i, a_i) + \; \gamma \, max_{a_i'} \, Q_{\emptyset}(s_i', a_i')$$

To fit the fitted Q-Learning on the given dataset, we’re using above equation to get the value of the action, value pair of the given state and optimize the weights based on that label.

- ## Hyper parameters:

  For the experiments I've decided to change try different C values in the Logistic Regression function, it affects the accuracy of the model fitting. Lower the values more will be the regularization and faster would be the convergence of the optimization.
  For Fitted Q- learning, I've changed two hyper parameters learning rate as well as number of iterations. Lower the learning rate more number of iteration would require to reach the optimum values of the action, value pairs.

  Below are the different sets of hyper parameters that I've used to tune my models and the results are displayed in the **Results** section.

| **Hyper parameters** | **Fitted Q-learning** | **Logistic Regression** |
|---|---|---|
| C | -- | 1 |
| Learning rate | 1/8 | -- |
| Iterations | 10 | |

**Experiment - 1**

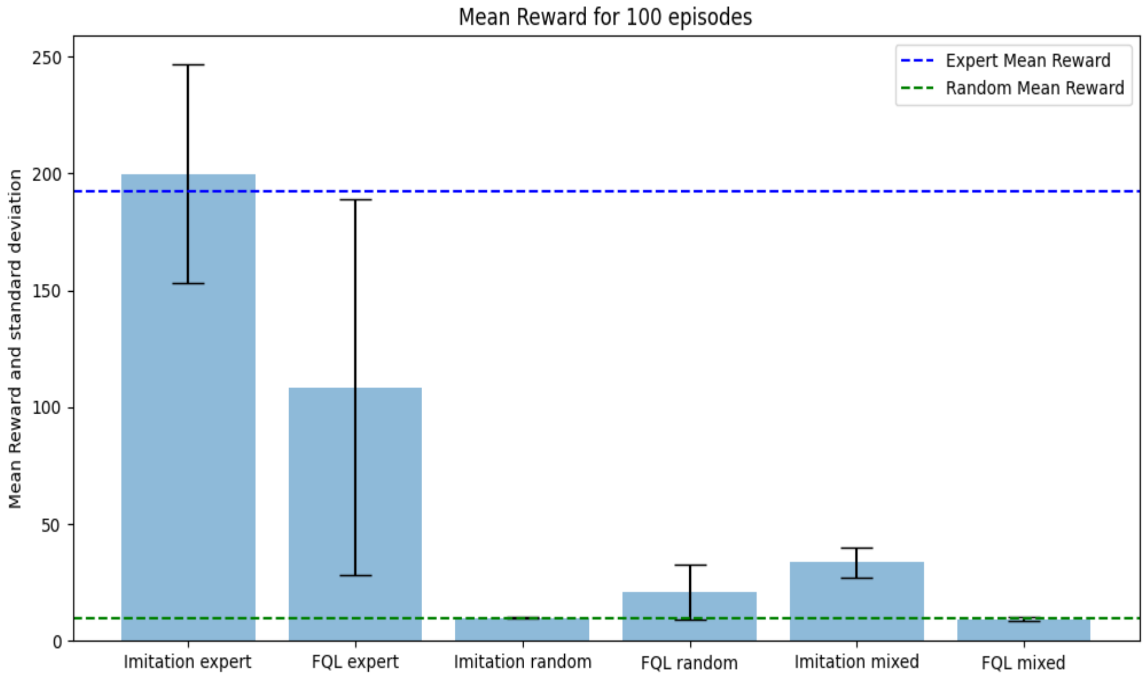
| **Hyper parameters** | **Fitted Q-learning** | **Logistic Regression** |
|---|---|---|
| C | -- | 0.5 |
| Learning rate | 1/16 | -- |
| Iterations | 20 | |

**Experiment – 2**

- **Experiment 1:**

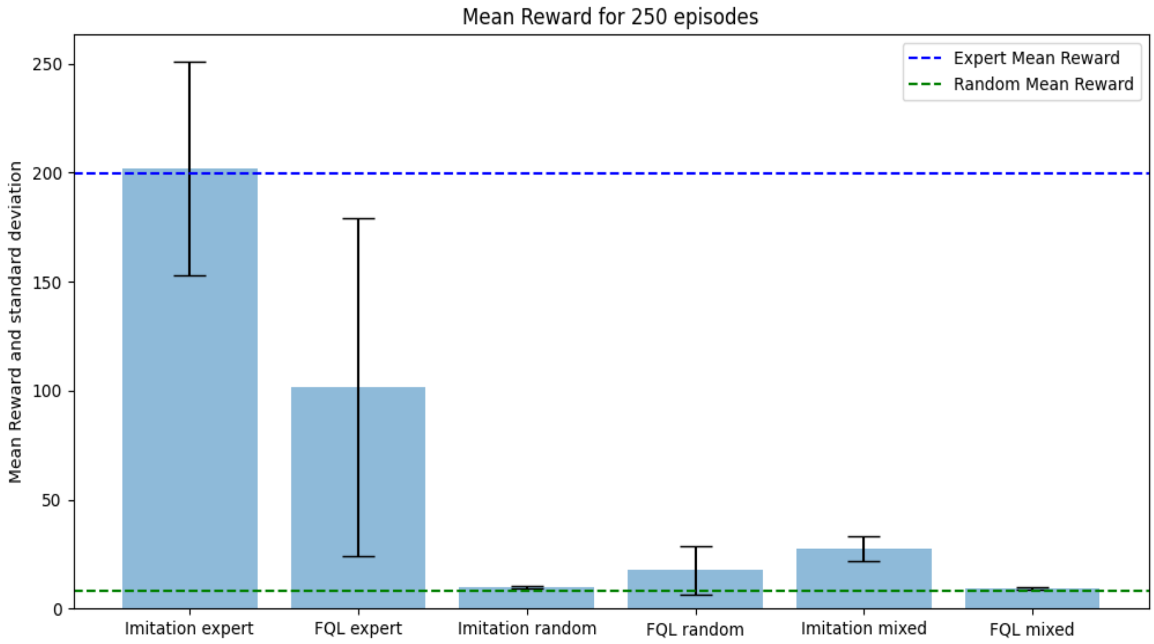
  Here is the results of my experiment – 1 for different sizes of the episodes.

  1. Number of episodes = 100



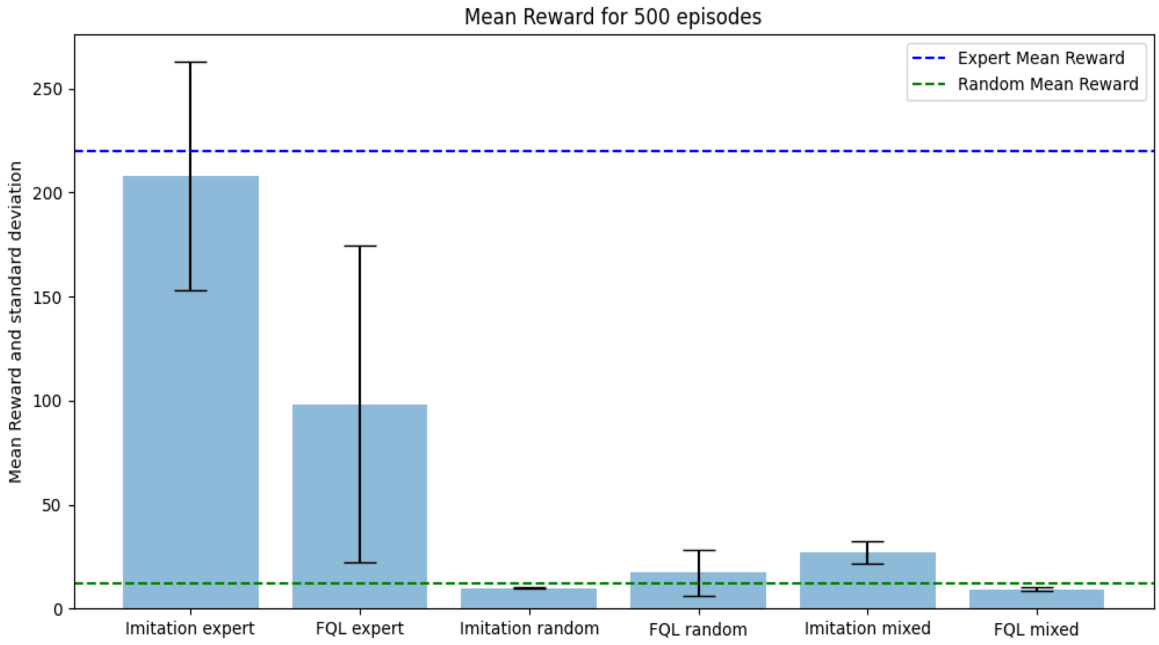
**Average rewards for 100 episodes for Imitation and Fitted Q-Learning (Fig 1.1)**

  2. Number of episodes = 250



**Average rewards for 250 episodes for Imitation and Q – Learning (Fig 1.2)**

**3.** Number of episodes = 500

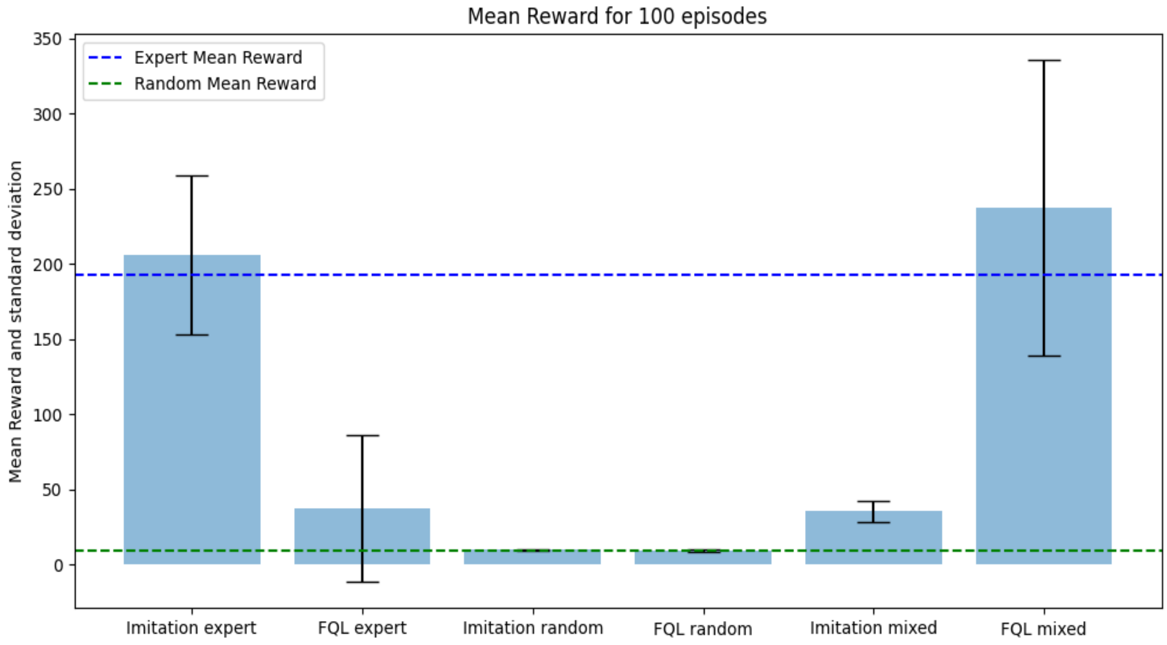


**Average rewards for 500 episodes for Imitation and Q-Learning (Fig 1.3)**

- ## Experiment 2:

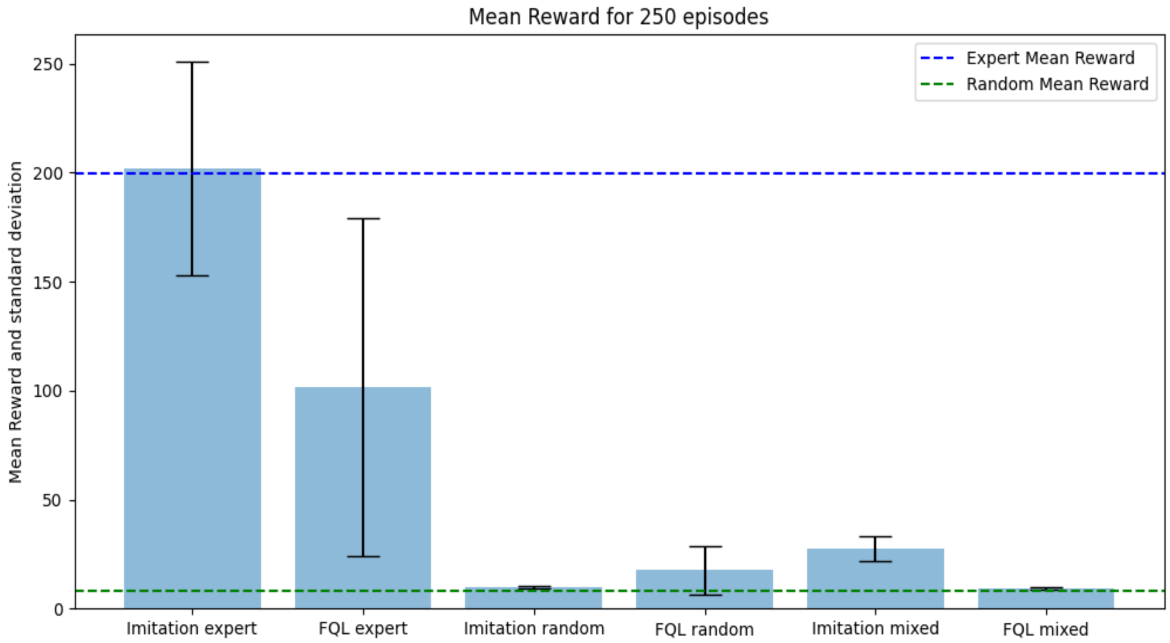
  Here is the results of my experiment – 1 for different sizes of the episodes.

**1.** Number of episodes = 100



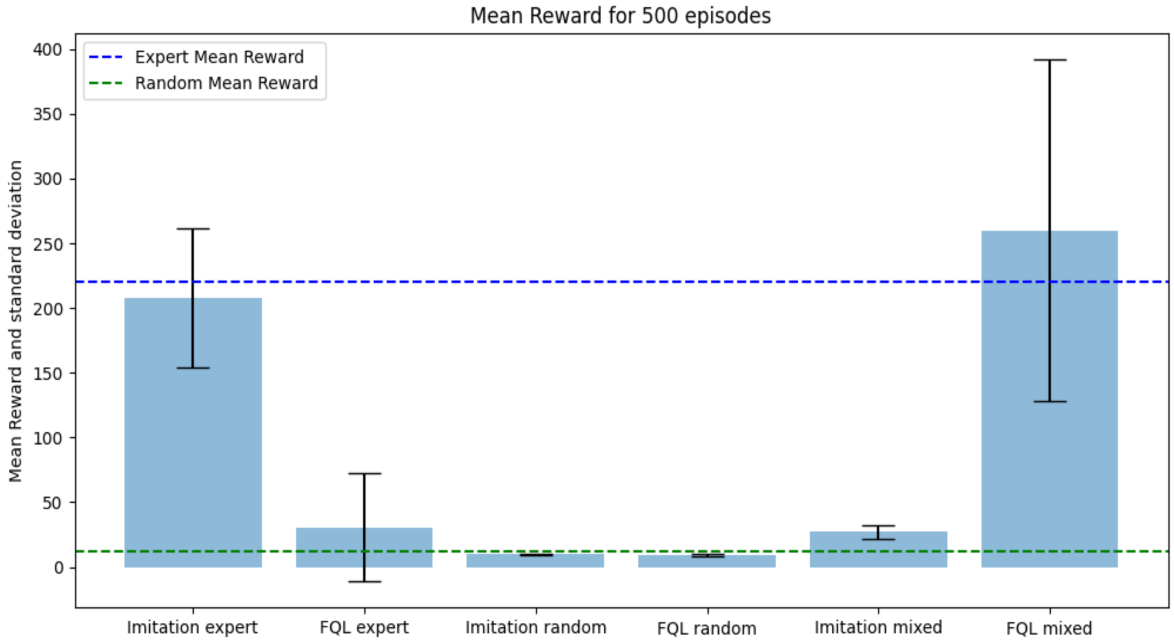
**Average rewards for 100 episodes for Imitation and Q -Learning (Fig 2.1)**

**2.** Number of episodes = 250



**Average rewards for 250 episodes for Imitation and Q-Leaning (Fig 2.2)**

**3.** Number of episodes = 500



**Average rewards for 500 episodes for Imitation and Q-Leaning (Fig 2.3)**

- **Conclusion:**

1. **Stopping criteria:**
   For fitting the weights in the given dataset, I've used two stopping criteria, first number of iterations and tolerance. For Logistic regression my maximum number of iterations are 1000. In the case of Fitted Q – Learning, my error tolerance is $10^{-6}$ and number of iterations are chosen as a hyper parameters

2. **Effect of hyperparameters:**
   For this assignment, I've done two experiments and the results are above, the effect of hyper parameters on Imitation learning is negligible however, we can see that the same is not true for Fitted Q – Learning. In the experiment we can see the average rewards are higher compared to Imitation learning and expert learning in the case of mixed dataset. Although it highly depends on the quality of the data and data size.

3. **Size of the data:**
   Larger the size of the dataset, there will be more data points on which the algorithm can learn but this will give the generalization problem, if model doesn't generalize better than it'll predict the wrong values of state, action pair and as a result average reward would be less.

| Number of Episodes | Expert | Random | Mixed |
|---|---|---|---|
| 100 | 23469 | 2275 | 12871 |
| 250 | 54989 | 5253 | 30120 |
| 500 | 113845 | 10860 | 62352 |

4. **Type of Dataset:**
   As we can see from figures Imitation learning on expert data performs really well however, for random and mixed dataset the average rewards are 9 and 37 respectively for Imitation leaning which shows us that having some randomness in the dataset will results in the lower average rewards since we're just imitating the behavior of the policy, the average rewards are nearly same as optimal policy rewards. Nonetheless, interesting results can be found in the case of fitted Q – learning from fig 2.1 and 2.3 we can see that not only average rewards are higher than Imitation learning but also higher than optimal policy for mixed dataset. This shows that, mixed dataset introduces exploration inside the datasets which forces fitted Q – agent to explore more explored territory and as a result the average rewards are higher than prior algorithm.

5. **Improvement:**
   In order to improve the above results, we can do more hyper parameter tuning in

the case of fitted Q – learning and achieve more higher and consistent results, the quality as well as quantity of the dataset is really important since, the we've to learn 40 weights and need more data so that it can converge to the true value of the weights and quality will give more balance to exploration and exploitation of the agent learning an environment.

- **Reference:**
  The code implementation of this assignment can be found in the following link: https://colab.research.google.com/drive/1cCrkQVvs7gzzxZ0nYS6I2-jVgYDFAGv7?usp=sharing