For each implemented agent, give 1 pro and 1 con.

1. **Epsilon-greedy bandit**
   Advantage: Very easy and simple to implement: just choose a random arm with epsilon-frequency, otherwise be greedy. Moreover, according to experiments, it shows really good results in practice.
   Disadvantage: In its simplest form the -greedy strategy is sub-optimal because the constant factor prevents the strategy from getting arbitrarily close to the optimal level. Moreover, if it is not well tuned its performance degrades rapidly.

2. **UCB1**
   Advantage: It is optimistic in exploration. It guarantees that it will not be stuck in a local maximum and keep exploiting.
   Disadvantage: UCB algorithm depends critically on the specific choice of upper confidence bound. However, there are similar algorithms like Thompson sampling that depend only on the best possible choice.

3. **BESA**
   Advantage: It has less hyperparameters. It does not need to know a set of distributions in advance, unlike Thompson sampling or KL-UCB and does not even need to know the support, unlike UCB or kl-UCB.
   Disadvantage: Constants are not tight. Also, due to recursive algorithm, and comparing pair by pair all arms, takes much more time than other algorithms.

4. **Softmax**
   Advantage: Compared with epsilon greedy, it does not divide the phases into exploitation and exploration and then just uniformly choose. It provides arms of different chance of being interacted with based on the difference in their current reward averages.
   Disadvantage: When arms have close means, the Softmax algorithm does not seem to be as robust in terms of determining the best arm(compared to eps-Greedy)

5. **Thompson sampling Agent**
   Advantage: It is efficient to implement and has several desirable properties such as small regret for delayed feedback. Also, compared to eps-greedy, it does not get stuck to a suboptimal solution and keep exploiting. This due to its more sophisticated exploration of the variant space.
   Disadvantage: Takes a bit of time until parameters a and b converge to their true rates. So at the beginning, it might choose non-optimal arms often.