Restaurant Recommendation on yelp

Alif Naufal Farrashady A0218302U

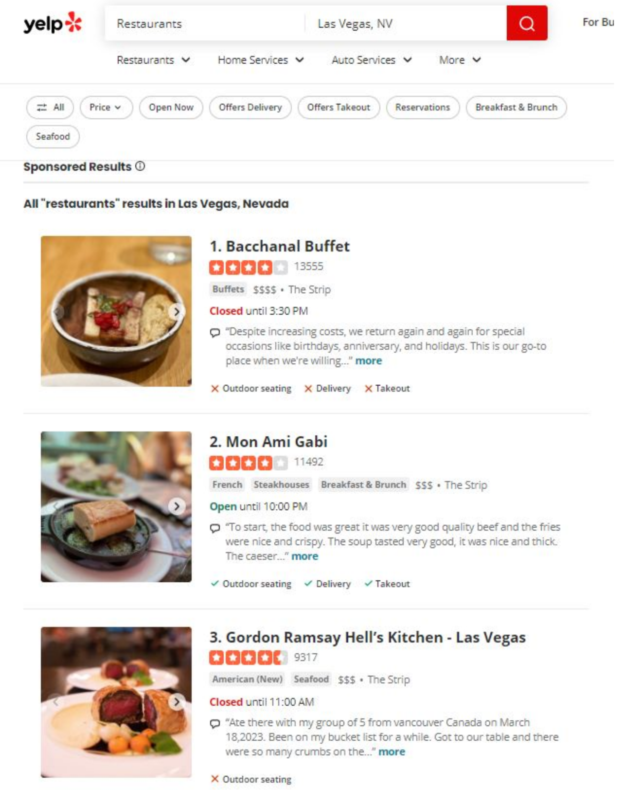
Chng Jin Yang Ray A0218512M

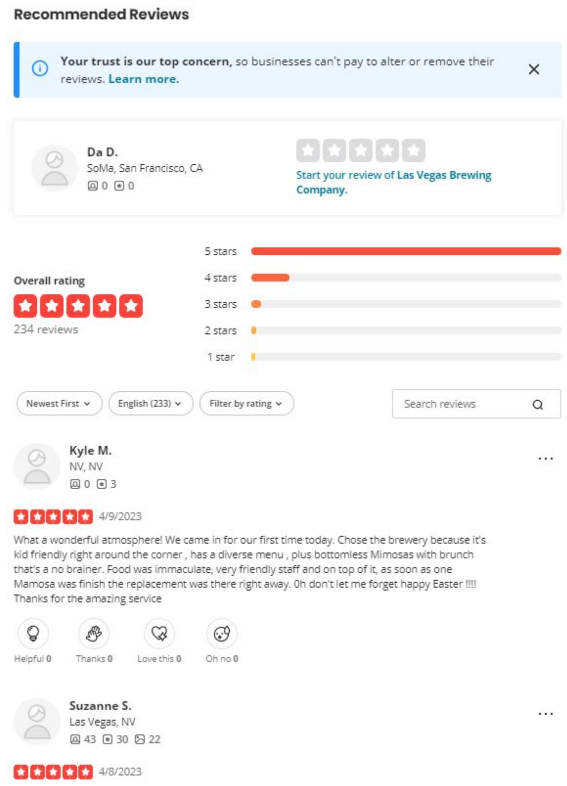
Chua Yee Siong Danyel A0222792A

yelp

Background

Businesses

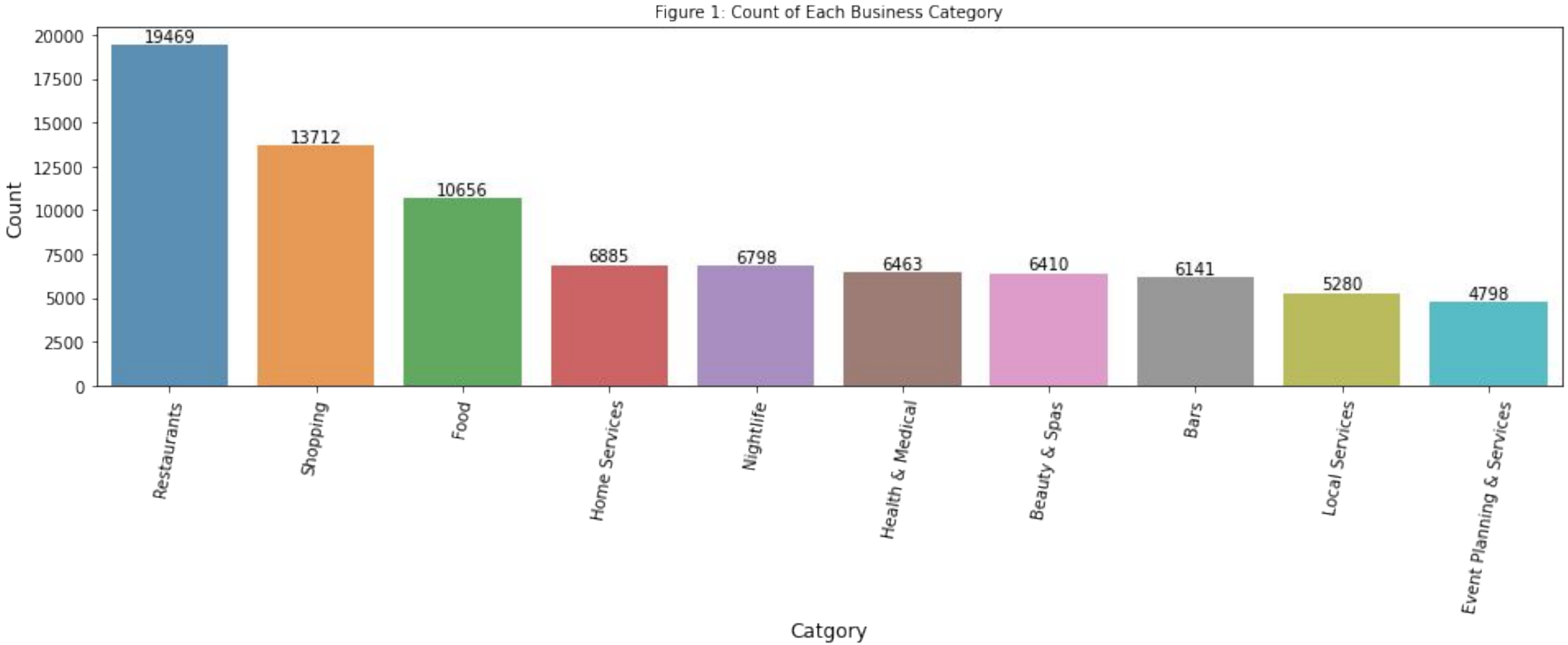




Users

Background

Restaurants on Yelp



Motivation

How do recommendations on Yelp work?

Current recommendation system is:

A

Rigid

Safe but monotonous

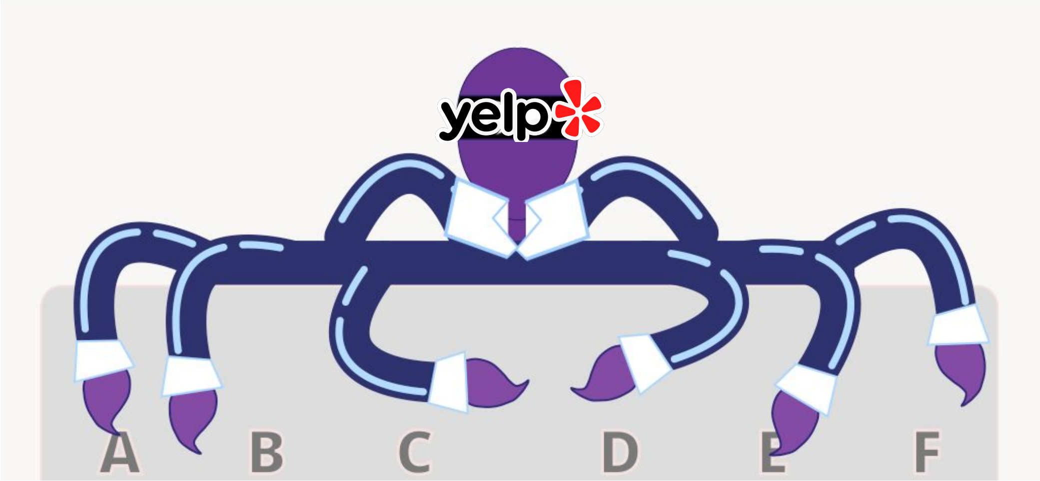
B

Lacks Exploration

Into lesser known businesses

Motivation

Bandit Algorithms



Motivation

01

Recommendations

Spontaneous and varied

02

Engagement

Improved user engagement

03

Retention

Increased user retention

04

Revenue

Higher revenue for Yelp

Problem

Overwhelming number of restaurant options on Yelp and rigidity of recommendations for users

Solution

Using bandit algorithms to provide personalized restaurant recommendations to users

Highlights of Key Findings and Implications

Epsilon-Greedy

Epsilon-Decay

Softmax

Annealing Softmax

Upper Confidence Bound(UCB)

Bayesian UCB

Thompson Sampling

Linear UCB

Contextual Thompson Sampling

Data

Data - files

01 **users.json**

Business data with business attributes

02 **business.json**

User data including metadata

03 **reviews.json**

Full review text data including user and business id

checkin.json **04**

Check Ins on a business

tip.json **05**

Tips written by a user on a business

photo.json **06**

Contains photo data including caption

Data - size

150,346

Businesses

1,987,897

Users

6,990,280

Reviews

Data - filtering

Location

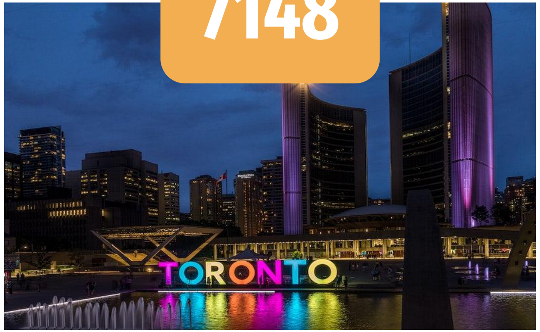
Restaurants Only

Data - filtering

Number of restaurants after filtering

5899

7148



Benchmark number to reduce complexity

20 - 40

Data - filtering

Average Stars per Restaurant

Las Vegas Minimum reviews

2000

Toronto Minimum reviews

500

Data - Final size

38

Restaurants

88854

Reviews

Data - Final size

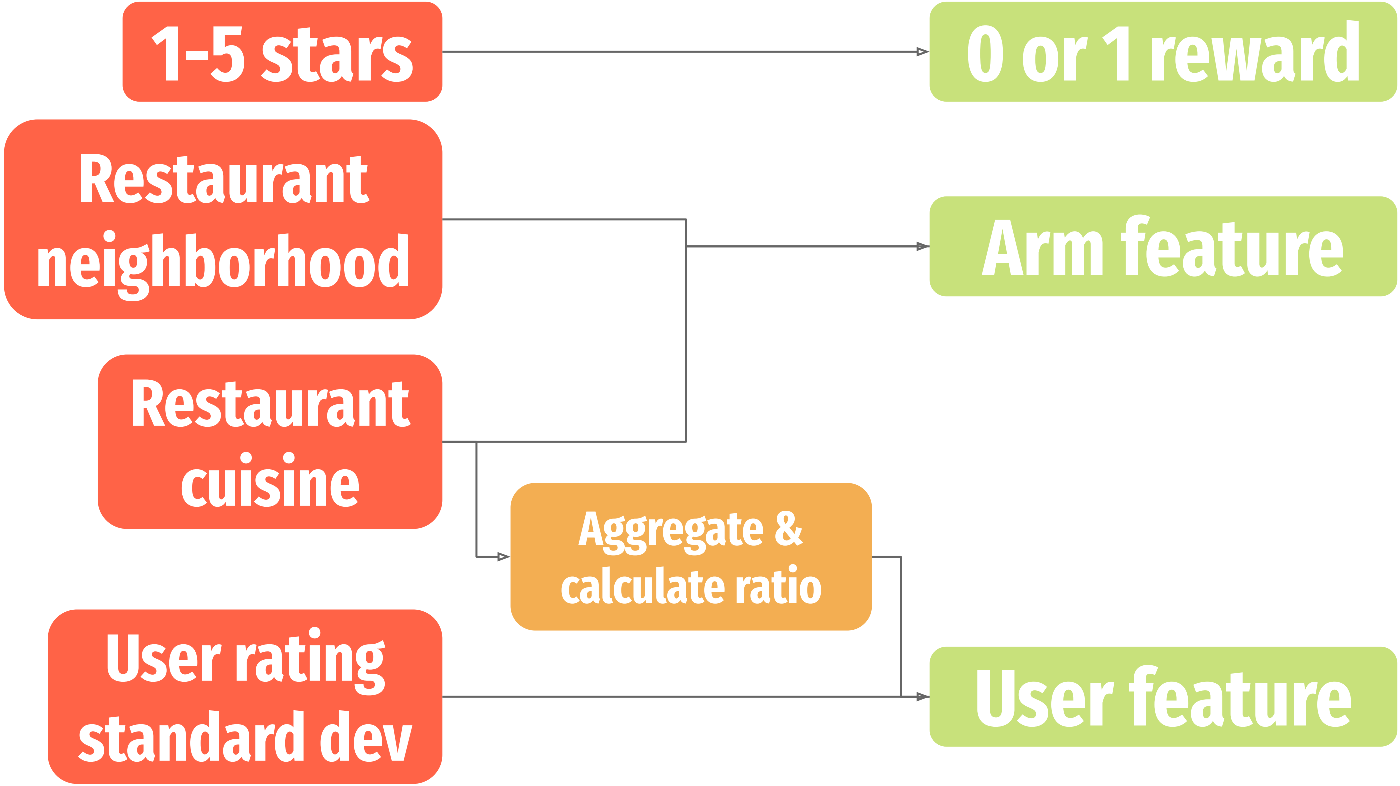
24

Restaurants

15573

Reviews

Feature Engineering



Methodology - Evaluation Replay

Step 01

Set-up

Each restaurant is one arm, at each time step, a review (including features) is shown to the bandit algorithm

Step 02

Select arm

Bandit algorithm will recommend a restaurant, contextual bandits will use features in their recommendations

Step 03

Update

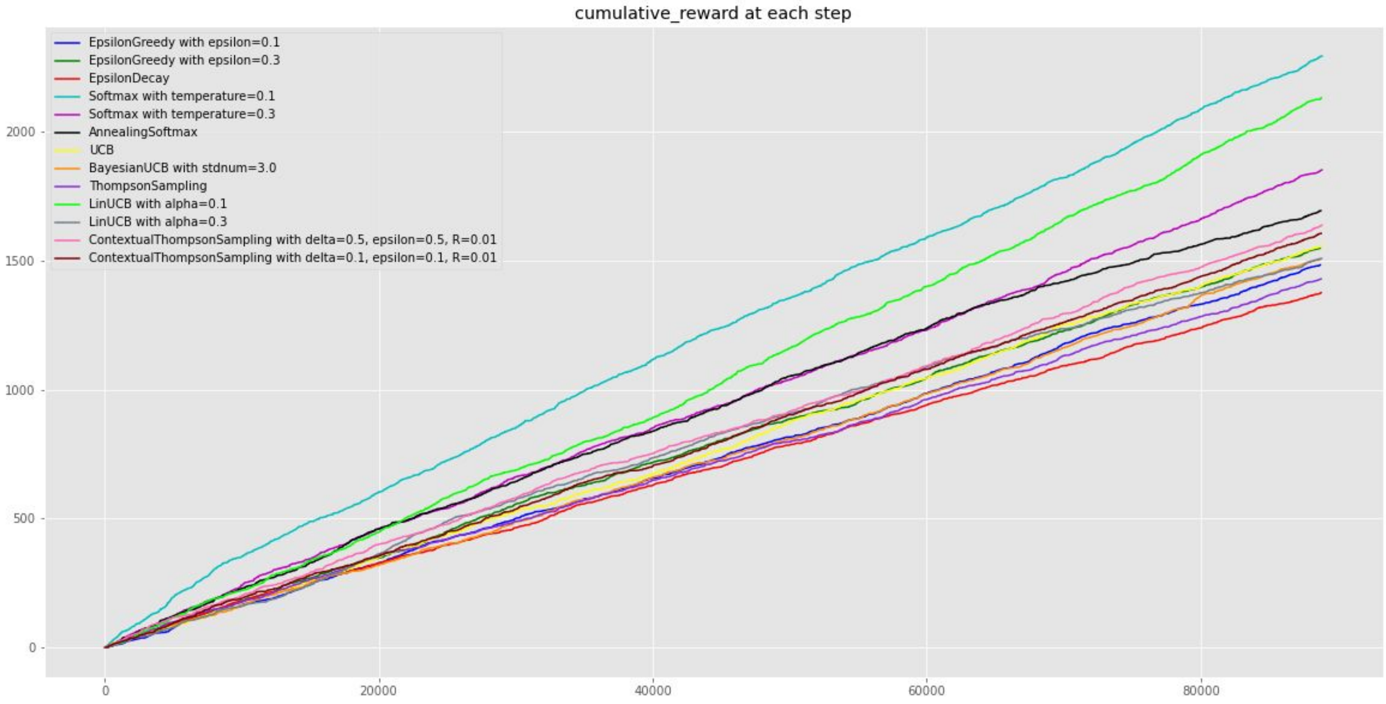
If recommended restaurant matches actual restaurant of review, bandit algorithm will update itself based on the reward of that review

Methodology

Model Name	Parameters Used
Non-contextual Bandit Algorithms	
Epsilon-greedy	epsilon=0.1 epsilon=0.3
Softmax	temperature=0.1 temperature=0.3
Annealing Softmax	
Upper Confidence Bound (UCB)	
Bayesian UCB	stdnum=3
Thompson Sampling	
Contextual Bandit Algorithms	
Linear UCB (LinUCB)	alpha = 0.1 alpha = 0.3
Contextual Thompson Sampling	delta=0.1, epsilon=0.1, R=0.01 delta=0.5, epsilon=0.5, R=0.01

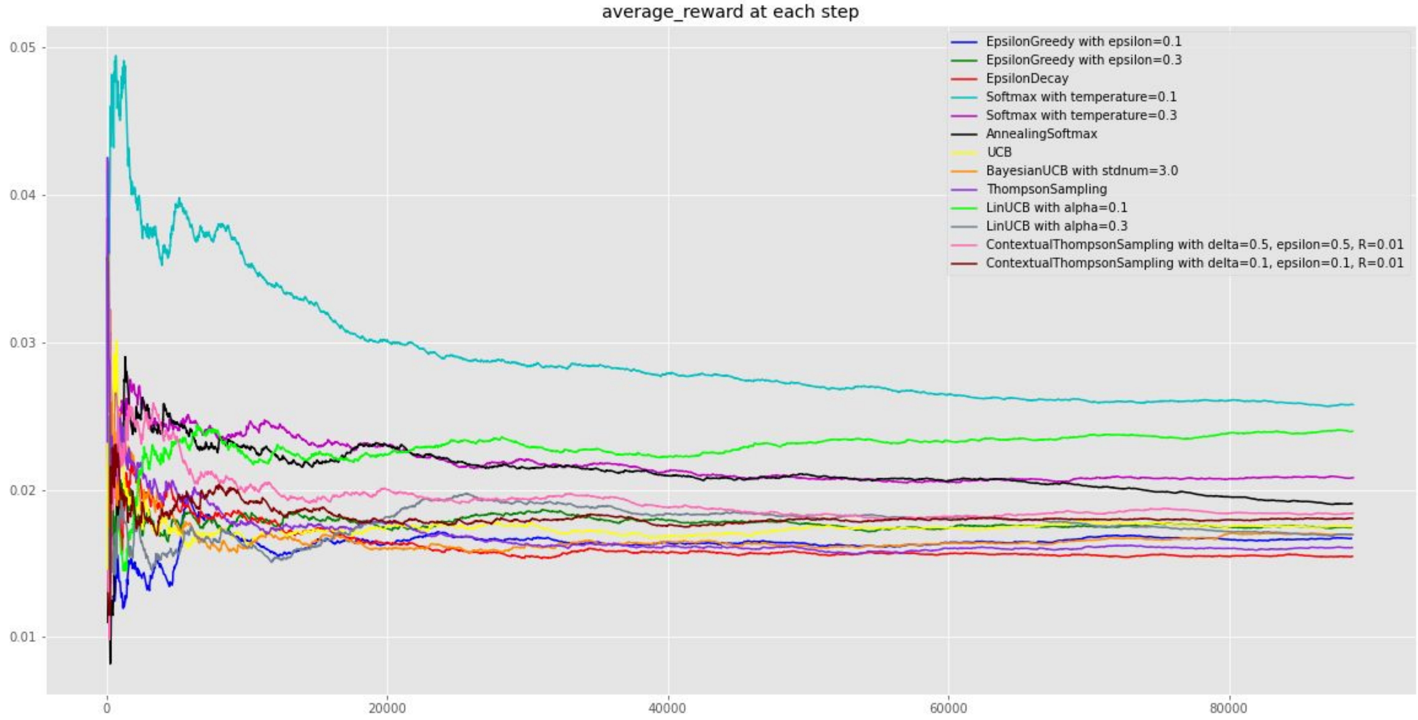
- For Bayesian UCB & Thompson Sampling, reward assumed to follow Bernoulli, Beta distribution as conjugate prior, prior alpha and beta set to 1
- For contextual bandits, model parameters will be disjoint

Las Vegas - Cumulative Reward



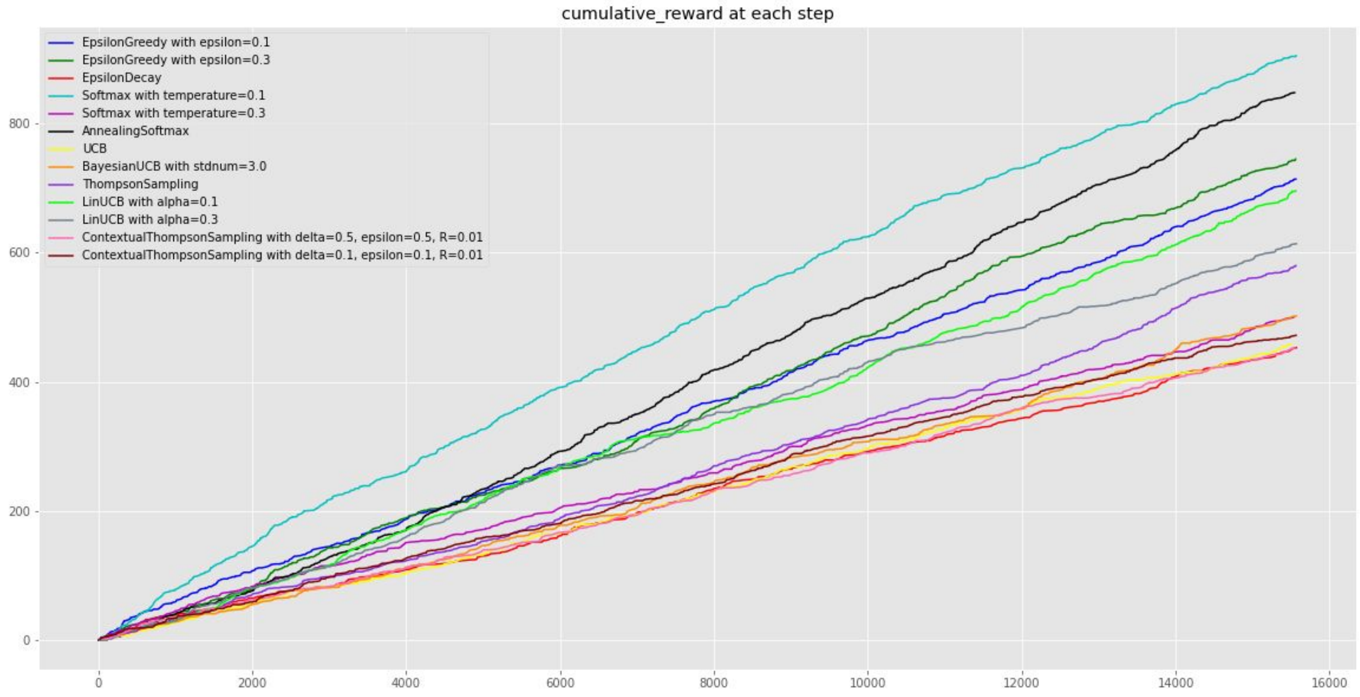
- Softmax (temperature=0.1) was the best performing algorithm, followed by LinUCB(alpha=0.1)
- Epsilon-decay and Thompson Sampling had the lowest cumulative reward
- All 3 Softmax variants did well, outside these 3, the contextual bandits had the highest cumulative reward
- Features were useful in understanding distribution of rewards

Las Vegas - Average Reward



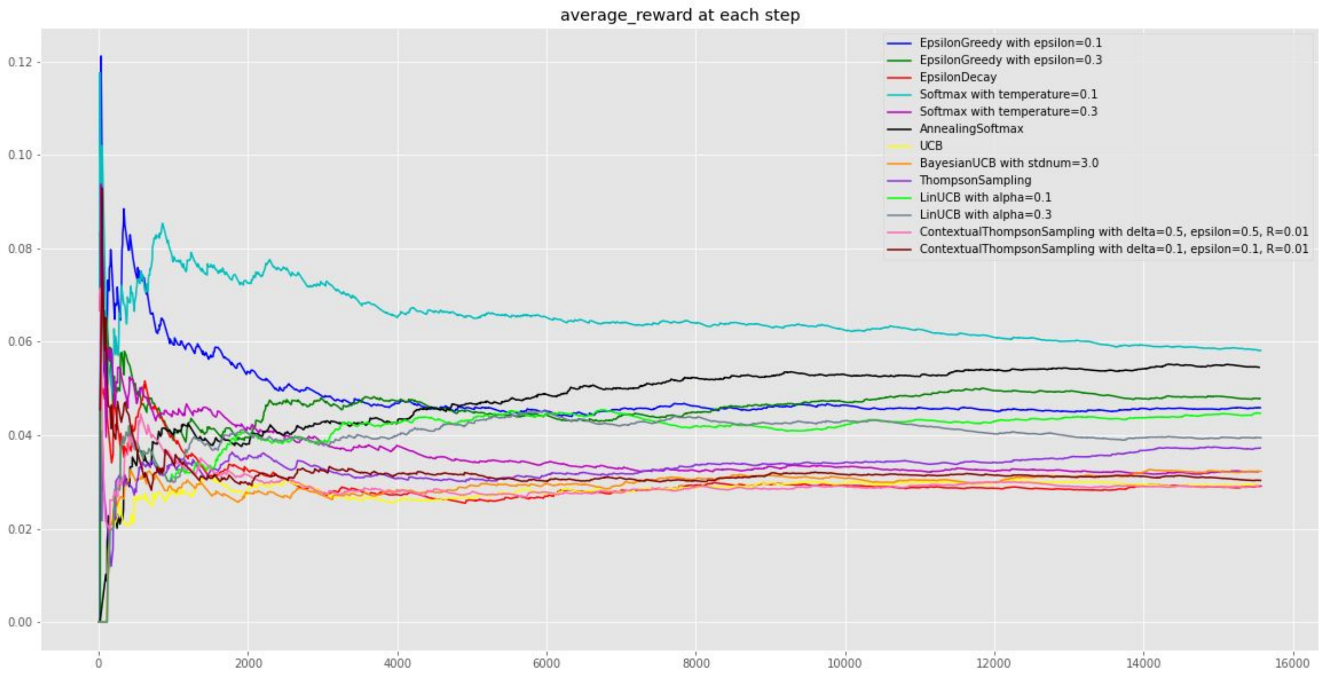
- Same order as cumulative reward
- Softmax (temperature=0.1) takes the longest to converge
- May have allowed it to achieve better long-term performance

Toronto - Cumulative Reward



- **Softmax (temperature = 0.1) still performs the best**
- Followed by Annealing Softmax and EpsilonGreedy algorithms
- Epsilon-decay, UCB and Contextual Thompson Sampling (delta=0.1, epsilon=0.5, R = 0.01) have **lowest cumulative reward**
- Contextual bandit algorithms fell short on performance
- Smaller size of Toronto dataset may have played a role

Toronto - Average Reward



- Same order as cumulative reward as well
- Other algorithms apart from Softmax (temperature = 0.1) not seem to stabilize as quickly
- Likely due to the smaller dataset compared to Las Vegas

Summary of Results

Softmax - best performing algorithm
Non-zero probability for other arms, vs ‘argmax’ approach

Bayesian UCB & Thompson Sampling struggled
Assumption of Bernoulli distribution may not be appropriate

LinUCB outperformed Contextual Thompson Sampling
Usage of multivariate Gaussian may not be suitable

Personalized recommendations possible
LinUCB achieved strong performance, features useful in helping to choose best arm

Limitations

Limited features available

Disjoint vs joint model parameters

Hyperparameter tuning

Conclusion

Softmax

Contextual Bandits

LinUCB

Thank You