**Thursday, 21st September**

**iOS:**

1. ARC and strong retain cycles - short video : https://www.youtube.com/watch?v=VcoZJ88d-vM

2. Concurrency and threading in iOS: video :https://www.youtube.com/watch?v=iTcq6L-PaDQ

- Not very good, need to read more about GCD and queue switching.

- How can main (serial) queue be async? Dispatch.main.async{}

**Mining:**

1. Similar item sets -> k-Shingling, and Jaccard.

2. Min Hashing to reduce size of set

- Still not clear how Jaccard similarity and min hashing comparison have same prob.

- must read again. Then proceed to locality sensitive hashing(bands)

- Must read again

**Friday, 22nd September**

1. **React for Beginners -** https://www.youtube.com/watch?annotation\_id=annotation\_3776896399&feature=iv&src\_vid=PGUMRVowdv8&v=ZnRFerIP8aA

(Prep for hackathon)

- Interesting, have to learn more.

2. **Re-read MinHashing.**

- Clear about Jaccard and min-hash similarity.

- Will need to re-read the probability calculations before quiz. Most likely source of questions will be choose a appropriate b and r for some threshold.

- Completed Locality sensitive hashing. Clear.

**Saturday, 23rd September and Sunday, 24th September**

**1. ACM Hackathon at USC**

-> Used the Google Maps API quite a bit.

-> very easy to use with minimal ‚ÄúRegister your app and get key‚Äù kind of stuff.

-> More than learning things technically, I learnt more about building a web app in a limited time frame.

-> Need to have server and front end modules at least set up before the event.

-> Need to investigate possible risks at a early stage.

-Learnt this because of the problem of converting Addresses to LatLong, with googles geocode api which had a limit.

-> Prepare for your demo!!!!

-> Prepare to sleep less

-> we got a special mention for our idea being very useful or something.

**Monday, 25th September.**

**1. Tutorial on Flask**. -> Short, will continue part 2 tomorrow.

**2. PCFG(Probabilistic context free grammar)**  -> This is for NLP assignment 3.

-> Read the basics of CFG and PCFG. Seems simple. Have to complete tomorrow and start on assignment.

-> Also, brush up on log-log scales for graphs on the assignment.

**Thursday, 19th October (Haven’t been updating this for a while)**

Brief summary of the last month or so.

**1. NLP**

- Learnt and implemented the CYK parser. Good fun working with dynamic programming. Spent a long time (too long) on the back-tracking part to actually get the solution. Shows I didn’t really understand that well enough.

- Midterm - a lot of dependency parsing, POS tagging, Viterbi Algorithm, Shift reduce parsers. - Will definitely have to revisit for Finals.

**2. Data Mining**

- Implemented the SON algorithm for frequent item sets in Scala and Spark. Good experience. The core algorithm was simple enough, but spent a lot of time on things like writing custom sort for a list of lists.

- Had 3 whole classes on Recommender systems. Very well taught and with the weekly quizzes, its been easy to keep up.

- Now moved onto Clustering and slightly more advanced things. Requires more time.

- Currently implementing a User-User and Item-Item based Collaborative filtering recommender for assignment 3. - Stuck at imputing values for unknown users/items. - with the current pressure, I just want to beat the baseline defined and submit it.

**3.Job search/preparation**

-> A lot of rejects, a lot of no replies.

-> Avvo process is ongoing. I am looking at iOS again. Things are flooding back. Confidence is building but need more prep on things I used to know and also many things I used and didn’t understand.

-> General algorithms etc. going pretty slow. Too much to do with NLP and Mining assignments, exams and quizzes.

**Sunday, 22nd October ( update for last 2 days)**

1. **Implemented User-User and Item-Item Recommender system**. Vanilla algorithm is not good enough for the test case given due to Cold-Start issues.

Hence tried various imputation methods.

-> User Average , Item Average, random number, etc.

For item-item based CF, using the item average is the best till now.

2. **For ALS(Alternating Least Squares) Recommender**, Neither User avg or Item avg made it better. Instead read this paper : https://www.researchgate.net/publication/224587995\_Boosting\_collaborative\_filtering\_based\_on\_missing\_data\_imputation\_using\_item%27s\_genre\_information?enrichId=rgreq-d11a21cbf649652e950d8e18916eddaf-XXX&enrichSource=Y292ZXJQYWdlOzIyNDU4Nzk5NTtBUzoxMDIxNTQ4NDUwMzI0NjBAMTQwMTM2NzAyMjEyOA%3D%3D&el=1\_x\_2&\_esc=publicationCoverPdf

and tried their way.

-> Basically, Check the items genre. Get the average rating of the user on items of that genre and impute with that value.

-> Works pretty well.

**3. Completed NLP assignment 5**. Very easy compared to the last one. Basic Cosine Similarity.

-> Used numpy for the first time. Very similar to Scala.

**4. Read 2 really good articles on design patterns**. iOS Specific but very useful in general as well.

-> https://www.raywenderlich.com/160651/design-patterns-ios-using-swift-part-12

-> https://www.raywenderlich.com/160653/design-patterns-ios-using-swift-part-22

Very well explained, with practical implementation.

Patterns covered ->

Creational: Singleton.

Structural: MVC, Decorator, Adapter, Facade.

Behavioral: Observer, and, Memento

**5. Read a nice article on why you shouldn’t use Notifications (AKA observer) in swift.**

-> https://davidnix.io/post/stop-using-nsnotificationcenter/

-> Agrees with everything my architect at SAP used to say.

**6. Watched tussah roy - how to invert a binary tree (stack + queue).** -> but I don‚Äôt think what he explains is actually inverting a tree..

**7. Searching in a binary tree** -> Recap, but needed it. Forgot so much about simple recursion.

**25th October, Tuesday**

1. **Data Mining:**

* Read clustering: Hierarchical and Point-Based.
* Point Based -> K-means, BFR(distributed) and CURE(distributed)
* Had to re-understand normal-distributions, Standard deviations etc.
* Learnt about distances in multi-dimensional space and Mahalanobis distance measure.
* Ready to tackle the next assignment.

1. **Data-Structures: Binary Search Trees.**

* Re-learnt BSTs.
* Creating, Searching, traversing and removing nodes.
* Implemented in Java(!) and uploaded to Github.