Project Timeline:

May 23 - June 3:

1. Introduce code from v-hasu's branch into non-machine learning classes and methods (e.g. FeatureManager, FeatureVector and Features classes) and refactor the code.

** Keeping week 1 light on work due to ongoing final exams which end on May 30th. I'm moving back from college to home between June 1-2. The remaining of GSoC period should be free from any more interruptions.

June 4 - June 12:

1. Put the test harness in place and write tests for modules completed so far.
2. Take feedback on tests written. Review and merge.

June 13 - June 20:

1. Integrate ListNET and NDCGScore

June 21 - June 26:

1. Write unit tests
2. Refactoring and creating a stable API till ListNet and NDCGScore.
3. ~~Prepare rst topic documentation for ListNET~~
4. ~~Review and merge~~

Mid-term evaluation

June 27 - July 3: Continued from last week

1. Write unit tests
2. Refactoring and creating a stable API till ListNet and NDCGScore.
3. Prepare rst topic documentation
4. ~~Test performance against INEX dataset~~
5. Review and merge

July 4 - July 17:

1. Integrate SVMRanker
2. Write unit tests
3. Prepare rst topic documentation
4. ~~Test performance against INEX dataset~~
5. Review and merge

July 18 - July 24:

1. Integrate ListMLE
2. Write unit tests
3. Prepare rst topic documentation

4. ~~Test performance against INEX dataset~~
5. Review and merge

July 25 - July 31:

1. Integrate Adarank
2. Write unit tests
3. Prepare rst topic documentation
4. ~~Test performance against INEX dataset~~
5. Review and merge

August 1 - August 7:

1. Integrate ERR
2. Link with all rankers and test
3. Prepare practical code examples for Getting Started guide

August 8 - August 24: (buffer time)

1. Prepare final evaluation report upon testing performance against INEX dataset
2. Take feedback on documentation and practical code examples
3. Discuss and work on “Stretch Goals”, or identify any segments that need improvement

End-term evaluation

Stretch Goals:

1. Write Python bindings for xapian-letor. Add relevant code examples to getting started guide.
2. Add a nonlinear ranking algorithm e.g. LambdaMART