Division of Work			
No.	Step Description	Category	Performed by
1	Selection and Analysis of Database software that suits this project(Couch DB)	Data Set Acquistion	Chinmaya
2	Database MapViews and Reduce Functions	Data Set Acquistion, Supervised and Unsupervised Learning	Asish Kumar Gaddipati
3	DBAccess.java - this class is used to connect to the database. This works as a wrapper around the LightCouch API for easy database interation.	Data Set Acquistion, Supervised & Unsupervised Learning	Chinmaya
4	Selection and Analysis of Website crawler to grab the content of website (Jaunt java Library)	Data Set Acquistion	Asish Kumar Gaddipati
5	WebScrapper.java class gets the question from the website given its link and stores it in the database	Data Set Acquistion	Asish Kumar Gaddipati
6	TagGrabber.java gets the list of tags that are used for analysis and write them into MasterTagFile.java	Data Set Acquistion	Chinmaya
7	GetQuestions.java gets X no of questions/tag generated in No:4 for each tag	Data Set Acquistion	Asish Kumar Gaddipati
8	FeatureWords.java analyses the training questions database and selects the words that are used as features based upon a word occurance in entire training database as threshold. Its writes selected words as features into file.	Feature Engineering	Chinmaya
9	ComputeIdfs.java computes the Inverse document frequency of the words in No:6 and write them into the database	Supervised Learning	Asish Kumar Gaddipati
10	TfIdfVector.java computes the sparse Feature Vector using Tf-Idf rule of given Question	Supervised Learning	Asish Kumar Gaddipati
11	QvectorGenerator.java transfers the Feature vector having features in the form of words to a sparse feature vector having tf-idf values for each feature using No.8 and write the sparse feature vector into database	Supervised Learning	Asish Kumar Gaddipati
12	SVMFileGenerator.java generates the .train and .test file for each tag having rule as +1 if the questin has that tag -1 otherwise	Supervised Learning	Asish Kumar Gaddipati
13	LearnModels*_*.sh file is used to train the data using SVM light and repective model files are captured into respective features. Similarly Classify*_*.sh file classifies the data using .model files and generates .result files for each tag.	Supervised Learning	Asish Kumar Gaddipati
14	AnalyzeSVM.java analyses the .result files predicts/Suggests top 5 tags for a given question and writes the predictions to database	Supervised Learning	Asish Kumar Gaddipati
15	TagAccuracies.java computes the accuracy of test dataset having Criteria as atleast how many tags are predicted correctly like for How many questions atleast 1/2/3/4/5 tag is predicted correctly?	Accuracy Measures, Supervised and Unsupervised Learning	Chinmaya
16	ComputeTagIdfs.java computes Idf values in unsupervised learning for the feature words. Documents here is a combination of all questions for a particular tag. So, each tag will have a single document. We will compute Idf values on these documents but not using individual questions as documents.	Unsupervised Learning	Chinmaya
17	TfIdfTagVector.java - this computes the TfIdf Vector for a given document.	Unsupervised Learning	Chinmaya
18	TagVectorGenerator.java - This computes Vectors for each Tag in the tags file and inserts into database. (uses TfIdfTagVector)	Unsupervised Learning	Chinmaya
19	CosineSimilarity.java - computes cosine similarity between two given feature vectors	Unsupervised Learning	Chinmaya
20	UnsupervisedTagPrediction.java - predicts the top 5 tags for a each test question and inserts the result into database (uses CosineSimilarity.java for comarison and access test database)	Unsupervised Learning	Chinmaya