4.1 Dataset Analysis

Geomotions.json has a size file of 21.7 MB (22,766,504 bytes) and contains objects of three categories: Reddit Posts, Emotions and Sentiments. Emotions and Sentiments are used as our classes, while the Posts themselves are used as our baseline dataset. The file itself contains posts with different characters such as words (upper case and lower case), special characters (exclamation points, periods, colons, underscores, etc.) as well as emojis, which was made more apparent during section 3 where we were tasked with tokenizing a post. Without the use of particular models, it would be difficult to properly classify each post. The entire .json contains 2,612,946 words and has 859,103 lines. This dataset is relatively large, thus explaining the scores obtained in part 2 of the project, where for Base-Multinomial NB and Base-Decision Tree, using Emotions (which has more classifier options than Sentiments), the accuracies were below 50%, while using Sentiments, the accuracies ranged 50% and more. Using Gridsearch did improve some scores, but not by much, while TfidfTransformer showed greater scores for Sentiments than Gridsearch, but scored lower for Emotions. MLP classifier seemed to have performed the best compared to the other two for Emotions as it is the best classifier for large datasets and can make faster predications post training. Overall, the dataset was relatively messy and unclean, which could explain the reasoning behind the poor scores. Had the dataset been cleaner, had fewer special characters and emojis, the scores would have most likely increased overall.

4.2 Analysis of the Results of all Models

Models	Precision	Recall	F-Mesure
Model 1	0.25	0.5	0.33
Model 1 GridSearch params={'activation':['identity', 'logistic', 'tanh', 'relu'], 'solver':['adam', 'sgd']}	0.25	0.5	0.33
Model 2	0.25	0.5	0.33
Model 2 GridSearch params={'activation':['identity', 'logistic', 'tanh', 'relu'], 'solver':['adam', 'sgd']}	0.25	0.5	0.33
Model 3	0.25	0.5	0.33
Model 3 GridSearch params={'activation':['identity', 'logistic', 'tanh', 'relu'], 'solver':['adam', 'sgd']}	0.25	0.5	0.33

(Table 1. Models performance results)

In Embeddings as Features approach to text classification of our Mini project we used Google News 2013 trained model with the vocabulary size of 2883863 words for our Model 1. We trained and test Multi-Layered Perceptron classifier with the default hyper-parameters as well as MLP using the GridSearch with the following parameters: 'activation': ['identity', 'logistic', 'tanh', 'relu'], 'solver':['adam', 'sgd']. As you can see from Table 1, the performance for Model 1 is 0.25, 0.5, and 0.33 for Precision, Recall and F-Mesure respectively. The same model shows the same performance results while using 'activation': 'identity', 'solver': 'adam' hyper-parameters. For our Model 2 we used English Wikipedia Dump of February 2017 pre-trained model with 302866 words of vocabulary. Validation results on testing data shows the same level of performance of Precision=0.25, Recall=0.5,

F-Mesure=0.33 as in Model 1. Finally, we run our last model 3 with Gigaword 5th Edition English pretrained embedding model. As with the previous model 1 and 2, we trained Multi-Layered Perceptron classifier with default parameters and custom 'activation' and 'solver' hyper-parameters. The performance results for model 3 are completely identical to our two previous models: Precision=0.25, Recall=0.5, F-Mesure=0.33.

4.3 Task Separation, Roles and Responsabilities

Written by **Souvik**

Task 1: Dataset Preparation & Analysis

Task done by all members, submitted file used **Andrei**'s contribution.

Task 2: Words as Features

Argiro took responsibility for Base-MNB & Top-MNB.

Andrei took responsibility for Base-DT & Top-DT.

Souvik took responsibility for Base-MLP & Top-MLP.

All members kept their respective responsibilities when doing task 2.5.

All members used TfidfTransformer() to redo substeps of task 2.3.

Task 3: Embeddings as Features

All tasks completed by **Argiro** & **Souvik**

Task 4: Analysis

Analysis of the dataset given on Moodle was written by **Argiro**.

Analysis of the results of all the models for both classification tasks was written by **Andrei**.