PROJECT REPORT P2

Proposed Features:

1. Author's score:
   I have used the author's score attribute present in the datasets for classification. Since the reliability and reputation of an author are important factors to be analyzed for estimating persuasiveness.
2. Readability:
   The complexity of a comment is another significant factor in estimating persuasiveness. If the comment is concise and less complex, it is more likely to be persuasive.
   Here, I compute the readability score of a comment by using a python package[1] with the following command:
   textstat.flesch_reading_ease(lt[i]

Features Implementation:

1. Sentiment score:
   I have implemented VADER in python to get the sentiment score. I have selected this technique because it is specially designed to collect sentiments on social media.[2]
2. Cosine similarity:
   To compute cosine similarity, I have implemented distance function available in scipy[3] package of python.
3. Hedge words:
   I have referenced a list of hedge words[4]. I am reading this file in my code and counting the occurrence of hedge words in each comment.

Feature Sets

1. Feature Set 1(featureSet1.py):
   This feature set contains baseline features which are:
   a. the length of a comment
   b. cosine similarity
   c. sentiment score
   d. hedge words
2. Feature Set 2(featureSet2.py):
   This feature set contains baseline features along with additional proposed features namely:
   a. the length of a comment
   b. cosine similarity
   c. sentiment score
   d. hedge words
   e. author score
   f. readability score

3. Feature Set 3(featureSet3.py):
   This set contains the following subset:
   a. sentiment score
   b. cosine similarity
   c. readability score

Model Performance

1. Feature Set 1:
   With feature set 1, the KNeighbors Classifier has returned the best results for predicting persuasiveness.
   Observations:
   Accuracy: 97.79844
   F1 score: 0.5008465116415092
   Precision score: 0.5235463917713161
   Recall score: 0.5025366834744864
   ROC AUC score: 0.5025366834744863
   Confusion Matrix:
   [[54361 111]
   [1117 8]]
   Analysis:
   Here the false positives and false negatives are at diagonal positions which equate to 111 and 1117. The classifier has predicted comments as persuasive 119 (111+8) times and not persuasive 54472 times.

2. Feature Set 2:
   With feature set 2, Random Forest Classifier has returned the best result.
   Observations:
   accuracy with respect to training set: 97.97650
   F1 score: 0.494889569270185
   Precision score: 0.48988254761947586
   Recall score: 0.5
   ROC AUC score: 0.5
   Confusion Matrix:
   [[54472 0]
   [1125 0]]
   Analysis:
   Here, as we can see the false positive is 0, which indicates that the model has accurately predicted values for non-persuasive comments but it performs poorly for persuasive comments. This could be due to the fact that readability does not necessarily increase persuasiveness.

3. Feature Set 3:
   Random Forest Classifier has returned the best results.
   Accuracy with respect to training set: 97.97650952389517
   F1 score: 0.494889569270185
   Precision score:0.48988254761947586
   Recall score: 0.5
   ROC AUC score: 0.5
   Confusion Matrix:
   [[54472 0]
   [1125 0]]
   The values have remained equal to the previous model indicating higher importance of cosine similarity and semantic score.

References

1. https://pypi.org/project/textstat/
2. https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f
3. https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cosine.html
4. https://github.com/words/hedges/blob/master/data.txt