Project Proposal
Topic : Prediction of Online News Popularity

Amoli Vani - anv282
Shivani Payal - smp765

- **What is the problem (including motivation and what is the specific outcome)?**

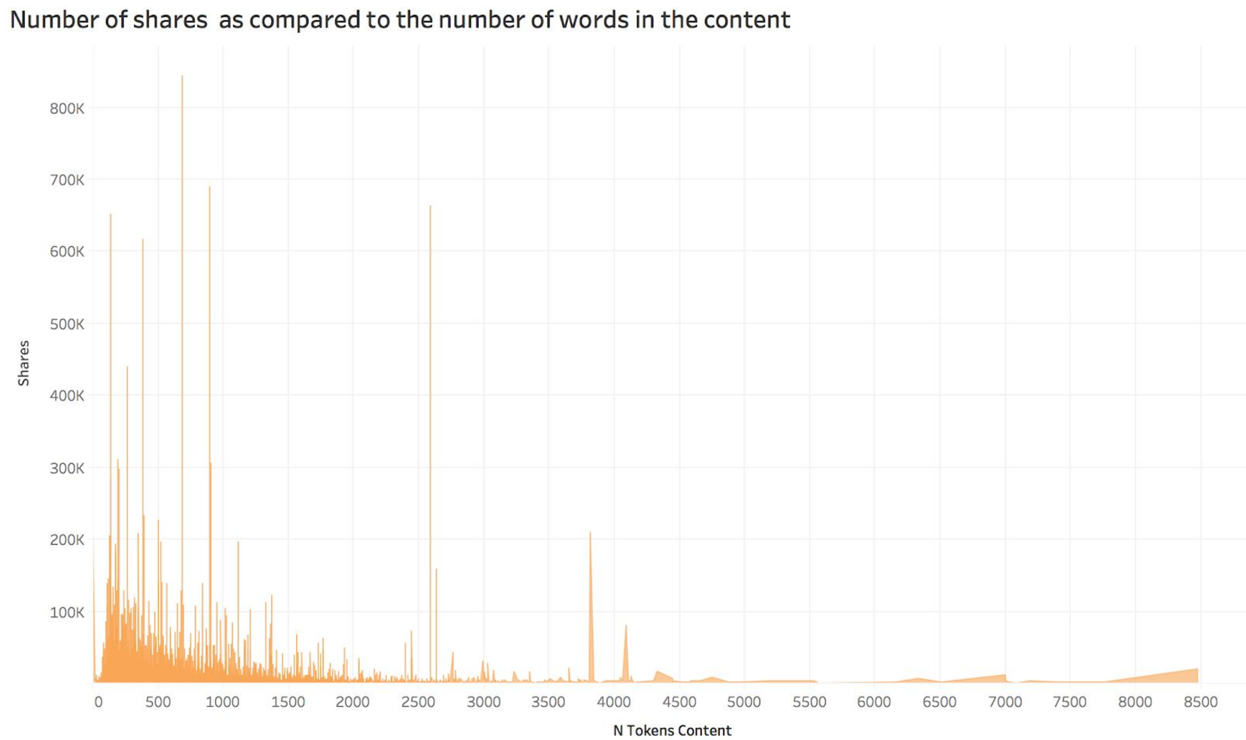
  Editors of digital media, news and blogging websites are flooded daily with hundreds of articles. Their goal is to publish content that will be popular with their readers, so that readers share the articles on various social media platforms, thus increasing advertising revenue. Even if an editor was to read every incoming article, it will be difficult for him/her to guess whether the content will be popular prior to its publication.

  The problem we are addressing is to predict whether an article will be popular or not prior to its publication, by creating a model trained on a historical dataset of articles published over the past two years. The motivation behind tackling this problem is that before investing resources and time into the publication of a new article, we can predict whether or not it is worthy of being published in the first place.

  The outcome of the problem is that given an article that hasn't been published, we will be able to determine whether the article will be popular or not based on a model trained on our training dataset.

- **How will you learn the background? (e.g. are there specific publications that discuss pertinent issues, is there a domain expert you will engage and what is their experience, etc.)**

  We learn about the background by considering the Online News Popularity Dataset available in the UCI Machine Learning Repository. Since the feature engineering has already been performed and the features are mostly numeric or nominal, domain-specific knowledge will not be required in this problem. But we can gain some insight into the background of the problem by performing a simple exploratory analysis on some of the features to see how they are related to the number of shares an article may receive. Like the one below, we can analyse the relation between the number of shares of an article and the number of words in the content using data visualization tool like Tableau. Here we can see that the number of shares are negatively correlated with the number of words in the content.



N Tokens Content vs. Shares.

- **What kinds of data will you use? (describe the data fully including it's temporal and spatial dimensions, features and their types and scales (e.g. numerical or text, ordinal or nominal, etc.))**

  We are using the Online News Popularity Dataset available in the UCI Machine Learning Repository. This dataset consists of a total of 61 attributes, out of which 58 are predictive attributes, 2 are non-predictive attributes, and the target variable is numeric.
  The predictive attributes describe various features of the articles, as shown in the table below. The predictive attributes are either numerical or categorical (nominal in particular). Some of the numeric attributes which represent ratios are continuous-valued and their scale is between 0 and 1, while the others that represent counts are discrete. The nominal attributes are all binary.
  The target variable in the dataset is the number of shares an article received cumulatively on various social media platforms.
  There are 39,797 instances in the dataset. This dataset does not contain any temporal or spatial attributes.
  One point to note is that the dataset consists of the statistical description of text articles, but does not consist of text attributes.

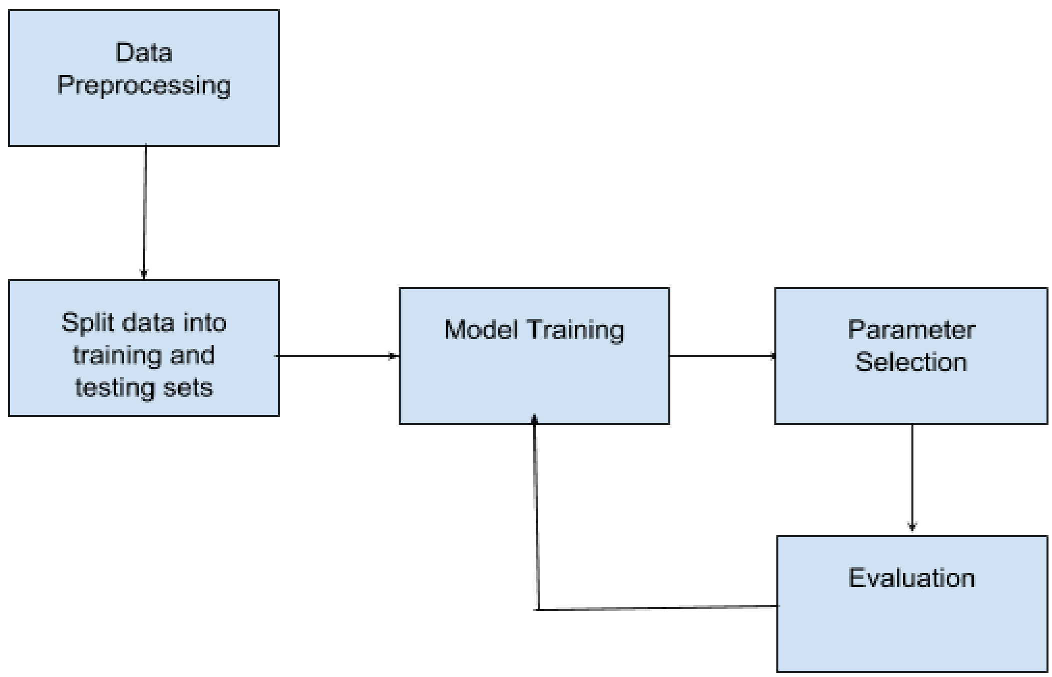
Attribute Information:	Type
0. url: URL of the article (non-predictive)	Numeric
1. timedelta: Days between the article publication and the dataset acquisition (non-predictive)	Numeric
2. n_tokens_title: Number of words in the title	Numeric
3. n_tokens_content: Number of words in the content	Numeric
4. n_unique_tokens: Rate of unique words in the content	Numeric
5. n_non_stop_words: Rate of non-stop words in the content	Numeric
6. n_non_stop_unique_tokens: Rate of unique non-stop words in the content	Numeric
7. num_hrefs: Number of links	Numeric
8. num_self_hrefs: Number of links to other articles published by Mashable	Numeric
9. num_imgs: Number of images	Numeric
10. num_videos: Number of videos	Numeric
11. average_token_length: Average length of the words in the content	Numeric
12. num_keywords: Number of keywords in the metadata	Numeric
13. data_channel_is_lifestyle: Is data channel 'Lifestyle'?	Nominal
14. data_channel_is_entertainment: Is data channel 'Entertainment'?	Nominal
15. data_channel_is_bus: Is data channel 'Business'?	Nominal
16. data_channel_is_socmed: Is data channel 'Social Media'?	Nominal
17. data_channel_is_tech: Is data channel 'Tech'?	Nominal
18. data_channel_is_world: Is data channel 'World'?	Nominal

19. kw_min_min: Worst keyword (min. shares)	Numeric
20. kw_max_min: Worst keyword (max. shares)	Numeric
21. kw_avg_min: Worst keyword (avg. shares)	Numeric
22. kw_min_max: Best keyword (min. shares)	Numeric
23. kw_max_max: Best keyword (max. shares)	Numeric
24. kw_avg_max: Best keyword (avg. shares)	Numeric
25. kw_min_avg: Avg. keyword (min. shares)	Numeric
26. kw_max_avg: Avg. keyword (max. shares)	Numeric
27. kw_avg_avg: Avg. keyword (avg. shares)	Numeric
28. self_reference_min_shares: Min. shares of referenced articles in Mashable	Numeric
29. self_reference_max_shares: Max. shares of referenced articles in Mashable	Numeric
30. self_reference_avg_sharess: Avg. shares of referenced articles in Mashable	Numeric
31. weekday_is_monday: Was the article published on a Monday?	Nominal
32. weekday_is_tuesday: Was the article published on a Tuesday?	Nominal
33. weekday_is_wednesday: Was the article published on a Wednesday?	Nominal
34. weekday_is_thursday: Was the article published on a Thursday?	Nominal
35. weekday_is_friday: Was the article published on a Friday?	Nominal
36. weekday_is_saturday: Was the article published on a Saturday?	Nominal

37. weekday_is_sunday: Was the article published on a Sunday?	Nominal
38. is_weekend: Was the article published on the weekend?	Nominal
39. LDA_00: Closeness to LDA topic 0	Numeric
40. LDA_01: Closeness to LDA topic 1	Numeric
41. LDA_02: Closeness to LDA topic 2	Numeric
42. LDA_03: Closeness to LDA topic 3	Numeric
43. LDA_04: Closeness to LDA topic 4	Numeric
44. global_subjectivity: Text subjectivity	Numeric
45. global_sentiment_polarity: Text sentiment polarity	Numeric
46. global_rate_positive_words: Rate of positive words in the content	Numeric
47. global_rate_negative_words: Rate of negative words in the content	Numeric
48. rate_positive_words: Rate of positive words among non-neutral tokens	Numeric
49. rate_negative_words: Rate of negative words among non-neutral tokens	Numeric
50. avg_positive_polarity: Avg. polarity of positive words	Numeric
51. min_positive_polarity: Min. polarity of positive words	Numeric
52. max_positive_polarity: Max. polarity of positive words	Numeric
53. avg_negative_polarity: Avg. polarity of negative words	Numeric
54. min_negative_polarity: Min. polarity of negative words	Numeric
55. max_negative_polarity: Max. polarity of negative words	Numeric
56. title_subjectivity: Title subjectivity	Numeric
57. title_sentiment_polarity: Title polarity	Numeric

58. abs_title_subjectivity: Absolute subjectivity level	Numeric
59. abs_title_sentiment_polarity: Absolute polarity level	Numeric
60. shares: Number of shares (target)	Numeric

- **What kind of model will you build? (What approach will you take for solving the problem and why not any other approaches, including how data will be cleaned, what specific algorithm(s) and any parameters used, and how you will evaluate your approach – describe a figure/table used to illustrate the evaluation)**



Flow Diagram for the creating the prediction model

Data Preprocessing:
The first step in the data preprocessing for this problem is to filter the dataset such that the instances are between 3 weeks to 2 years (i.e. 730 days) old. This can be done by looking at the second attribute in the dataset (timedelta) which represents the number of days between the article publication and the dataset acquisition. The most recent article in the dataset was 8 days old, while the oldest article was 731 days old from the date of acquisition. Since the date of acquisition is 8th January 2015, the articles range from 7th January 2013 to 8th January 2015. We are going to filter out the instances that were published upto 3 weeks (21 days) before acquisition, as we assume that within such a short period, there will not be enough information available about the number of shares of the article on social media.

The target variable in our dataset represents the number of shares to social media that an article received. We want to convert the target variable into a binary variable so that our problem is reduced to a simple classification problem.

In order to transform the numeric target variable into a binary value, we will need to set a threshold such that any instance with a target variable value below the threshold is categorized as being ‘Not Popular’ (0), whereas any instance with a target variable value greater than equal to the threshold is categorized as being ‘Popular’ (1). There are two approaches to this. One is to set the threshold value to the median value of the distribution of shares. Another approach is to set a threshold that gives maximum accuracy based on a method similar to selecting optimal parameters. This will require manually trying out a number of thresholds on the algorithms we use to see which threshold works the best for the models that we use.

Since the dataset consists of 58 predictive attributes, there is scope for dimensionality reduction using techniques like PCA. This can be done prior to creating the model. There are no missing values in the dataset, thus there is no need to handle the missing values.

Models:
Since the problem has been reduced to a classification problem, we intend to use the best-performing out of the following classification algorithms:
1) Logistic Regression
2) Support Vector Machines (SVM)
3) k – Nearest Neighbor (kNN)
4) Random Forests

Parameters:
We will use cross-validation in order to optimize the parameters for the above algorithms. For example, for SVM, we will compare different kernels, and for K-Nearest Neighbors, we will compare different values of k. We will then run our model using the parameter value that gives us the best accuracy.

Evaluation:
We will create a confusion matrix based on our model. Based on this matrix, we can calculate the accuracy, precision, recall and F-score. Using these metrics, we can then plot the ROC and AUC curves. Despite the data being abundant and cross-validation not being necessary, we will implement it all the same.

- **What assumptions are safe to make? (Explain clearly what the assumptions being made are and why that’s okay, this could be in terms of features considered, potential confounding variables, variable types, etc.)**

One of the assumptions being made is that as the UCI dataset doesn't contain any actual text data of the Mashable.com website, we assume that the statistical data that is available has been processed correctly using sentimental analysis. We also make the assumption that all of the 58 attributes being used are actually useful attributes that can help predict the popularity of an article by considering the number of shares an article will receive.
It would also be logical to assume that the relationship between the age of an article and the number of shares it receives will be positive and linear, i.e. older articles will have more shares. This can create a bias in the model. But we make the assumption that after a month or so, the number of shares that an article will receive will plateau, and the number of shares will not change significantly, thus levelling the number of shares irrespective of age. We have also filtered out instances that were less than 3 weeks old, and this will level out the number of shares irrespective of age.

Things updated and changed:

- After careful evaluation and trying predicting the target value through different approaches as they gave very low accuracy scores, we found the best option for Data preprocessing to be using a threshold on the number of shares target value, which is the median of all the values and then classifying the number of shares as 'popular' if the number of shares are greaters than the median and 'unpopular' if the number of shares are less than the median.
- As there were no missing values, we didn't have to handle the missing values but spaces in the column names were removed and the 2 non predictive variables were dropped.
- Implemented correlation coefficient heatmap and feature importance in order to perform exploratory analysis of variables.
- Implemented the following classification techniques using GridSearchCV for cross-validated parameter optimization:
    1. Logistic Regression
    2. K Nearest Neighbors
    3. Random Forest
    4. Linear Discriminant Analysis
    5. Gaussian Naive Bayes
    6. Decision Trees
- Plotted the ROC curve for all the models for model comparison and evaluation of the models based on their AUC values.
- Although to intended to performing scaling of the features, it took a very long running time for some models - hence we have excluded it from our analysis.
- Implemented Principal Component Analysis to reduce dimension of the data and to select a bucket of features that capture the most information about the dataset.