Machine Learning Final Project

Waszczak *(14081504)*, Ruckensteiner *(13762931)*, Van Rooijen *(13391957)*

Online News Popularity Prediction

1. Introduction

The provided dataset which we will work on contains information about news articles. There are 58 predictive and 2 non predictive features, 60 in total. The main goal is to predict which articles will be popular.

1.1 Research question

What is the most effective method for predicting the popularity of online news articles?

2. Data understanding

All of the articles were provided by the news website Mashable. The data is in csv format and in total there are 61 features and 39.797 rows (articles).

2.1 Predictive features

2. n_tokens_title: Number of words in the title
3. n_tokens_content: Number of words in the content
4. n_unique_tokens: Rate of unique words in the content
…
57. title_sentiment_polarity: Title polarity
58. Abs_title_subjectivity: Absolute subjectivity level
59. Abs_title_sentiment_polarity: Absolute polarity level

2.2 Non-predictive features:

0. url: URL of the article
1. timedelta: Days between the article publication and the dataset acquisition

2.3 Target classification variable

For the purpose of classifying the online news articles by popularity we used the shares feature as our target classification variable. We then calculated the median for the shares of the data for the popularity threshold by labelling articles that have more than 1400 shares as ‘Popular’ and labelling as ‘Unpopular’ articles that have less than or equal to 1,400 shares. We then created a new feature for the data frame with the target label for popularity. According to our threshold the data contains a total of 3,782 unpopular and 5,369 popular articles

3. Supplemental data collection

In addition to utilising the publicly available dataset, we also chose to analyse the content of the news articles themselves. The URLs of each article were provided, allowing us to develop a web scraper that could extract the articles. These articles, which can be found in the file scraped_articles.csv (which has a size of approximately 93 MB), were then analysed using natural language processing techniques.

4. Data preparation and feature engineering

4.1 Data Preprocessing

Predicting the popularity of online news articles is a classification task, which in this case involves predicting the binary outcome ("Popular" or "Unpopular") based on a set of input features. In order to prepare the data for this task, several data processing steps have been carried out in a way that makes it more suitable for the task at hand.

4.1.1. Data cleaning

After the exploratory analysis we checked for missing values and inconsistencies such as articles with no content which had to be removed. We also removed the non-predictive features which did not contribute to the analysis.

4.1.2. Normalising the data

Normalising the data can help to scale the features so that they are on a similar scale, which can help improve the performance of our classification model. After having analysed the distribution of the data, we observed that a normal distribution was not present in the data. To change the entire set of data into a distribution that is as close to normal as possible, a log transformation was applied.

4.1.3. Scaling the data

Since outliers have not been treated in this project, we used the RobustScaler from scikit-learn which can be used to scale the features of the dataset in a way that is less sensitive to outliers. We did not scale the shares column because we use it as the threshold of popularity and therefore needs to be representative to the actual value.

4.2 Dimensionality Reduction: PCA vs t-SNE

Dimensionality reduction was done for visualising high-dimensional data, as it reduces the number of dimensions and allows the data to be plotted on a two-dimensional or three-dimensional graph. Which helped us to better understand the underlying patterns and relationships in the data. For the online news dataset we tested both PCA and t-SNE in order to see which could be more useful for visualising and interpreting the clusters made later.

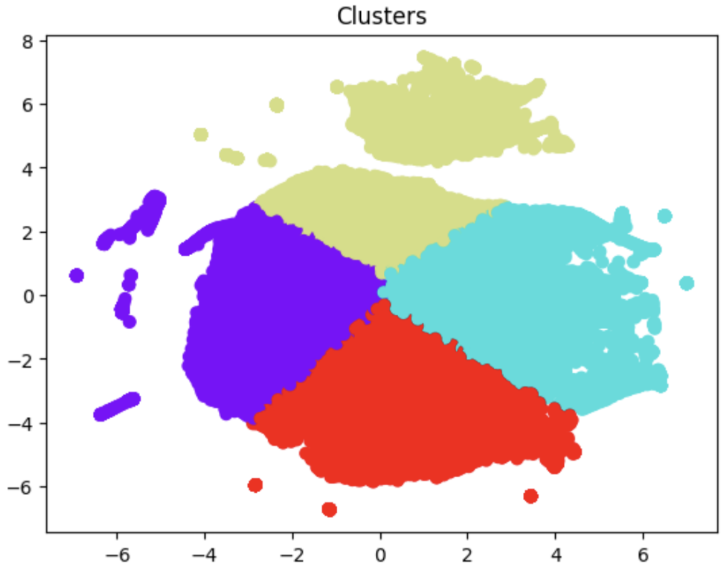
After plotting both methods in a two-dimensional space, we decided that the characteristics of the data were represented better by the t-SNE and it could be used in combination with k-means clustering.

4.3 Data Clustering with K-means

Using the numerical representation of each article in a reduced dimensionality we applied the k-means algorithm to group the articles into clusters. The algorithm iteratively assigns each article to the cluster with the nearest centroid, and then updates the centroids to be the average of all the articles assigned to that cluster. This process continues until the centroids stop changing or a predefined number of iterations is reached. After the algorithm finishes running, we have a set of clusters, each containing a group of similar articles with similar feature representations.

In order to determine the optimal number of clusters for K-means clustering we fitted the K-means model for a range of values for k, and then plotted the cluster sum of squared errors as a function of k. Then we applied the elbow-method to select the value of 4 for k at the "elbow" of the plot, where the sum of squared errors decreased more slowly. After predicting the clusters for the news articles we visualised the clusters in the following graph.

Figure 1: Clusters of online news with K-means



After partitioning the data into the 4 clusters, we extracted the articles in the cluster to make 4 subsets of the data used for feature selection and classification.

4.4 Feature selection

In the context of classifying online news articles based on their popularity, two feature selection techniques were used to identify the most relevant features for predicting popularity. Since we have a classification problem with numerical input and categorical output we used feature selection techniques that are correlation based but take our categorical label into account.

4.4.1. ANOVA F-value

To use ANOVA for feature selection, the F-value is computed for each feature in your dataset and selects the features with the highest F-values which is more likely to be an important predictor of the target variable.

4.4.2. Mutual information

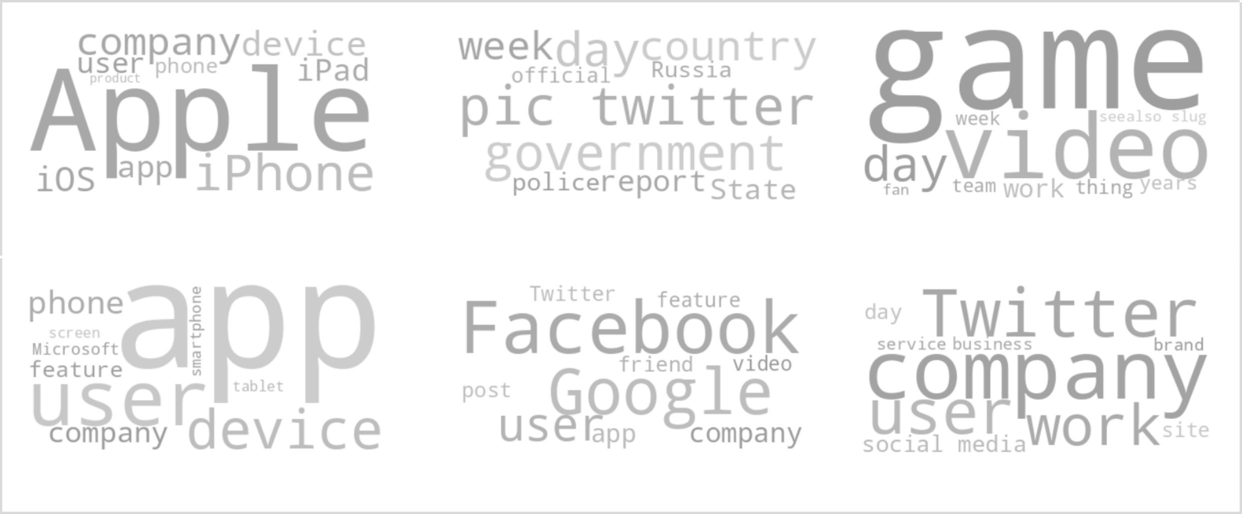
To use mutual information for feature selection, the mutual information score between each feature and the target variable is computed. The features with the highest mutual information values are likely to be the most important predictors of the target variable, indicating that the two variables are highly dependent and that one variable provides a lot of information about the other.

5. Methods

5.1 Topic extraction and clustering using Natural Language Processing

The aim was to generate news cluster topics based on the content of the articles. To achieve this, it was necessary to filter out irrelevant words. To do so, multiple dictionaries containing common English stop words were merged and applied to remove these stop words from the news articles. The resulting data was then processed using the TFIDFVectorizer, which generated a matrix of term frequencies and inverse document frequencies for each word. This matrix was then used by the KMeans++ algorithm to generate a total of 6 clusters, which are represented in the word cloud below.

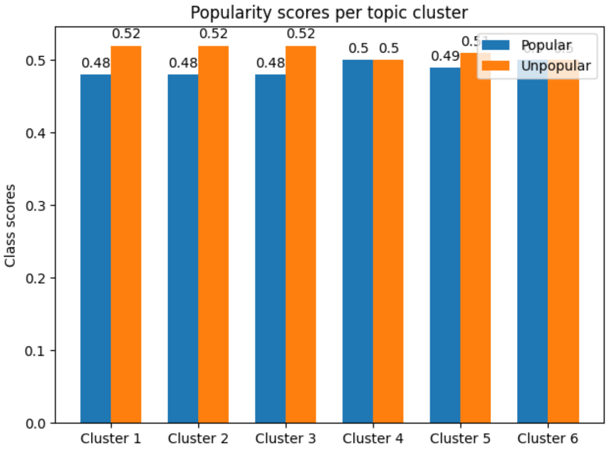
***Figure 2:** Six clusters generated using KMeans++ and TFIDFVectorizer*



5.2 Correlation between topic clusters and popularity

The graph below indicates that there is no significant correlation between the popularity of a topic cluster and the corresponding cluster itself. Given the minimal impact of this feature, it will not be included in the prediction of the target variable. Therefore it will also not be necessary to hypertune the number of clusters.

***Figure 3:** Popularity scores per topic cluster*



5.3. Machine Learning models for classification

Since the mission of this project is to predict weather an online news article is popular or not, rather than to build a regression model that predicted how many times an article would eventually be shared, the problem is treated as a binary classification of popular and unpopular article we used the following 4 models and compared their performance.

5.4. Logistic Regression

This is a simple but powerful linear model that is often used for binary classification tasks. It can be effective for predicting the popularity of online news by learning the relationship between the input features (e.g., headline, content, source, etc.) and the output label (e.g., popular or not popular).

5.5. KNearest neighbours

Nearest Neighbors (KNN) is a simple but powerful machine learning algorithm that can be used for classification tasks. In the KNN algorithm, a sample is classified by a majority vote of its neighbours, with the sample being assigned to the class most common among its k nearest neighbours.

5.6. Decision Trees

These models use a tree-like structure to make predictions based on the values of the input features. They are easy to interpret and can handle both numerical and categorical data.

5.7. Random Forest

These models are an ensemble of decision trees, where each tree is trained on a random subset of the data and the final prediction is made by taking the average of the predictions from all the trees. Random forests are generally more accurate and robust than individual decision trees.

5.8. Support Vector Machines (SVMs)

Powerful linear model that can be used for classification. They work by finding the hyperplane in a high-dimensional feature space that maximally separates the different classes.

6. Experiments and discussion

6.1 Experiments and results

After performing data preprocessing, k-means clustering and feature selection, we used the classification models to predict the popularity of the articles by splitting the data into training and testing data (70% and 30% respectively). The accuracy, precision, recall and f1-scores are included in figure 5.

In order to evaluate the performance of the models we computed the accuracy (the fraction of correct predictions made by the model), precision (the fraction of positive predictions that are actually correct), recall (the fraction of actual positive samples that the model correctly identified), and f1 (balance between precision and recall) metrics and compared the results of each model.

In addition, we also computed the confusion matrix to see how well the models were making the predictions for each class. It is a table that shows the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model. For example, a high number of TP and TN predictions would indicate that the model is accurately classifying popular and non-popular articles. On the other hand, a high number of FP and FN predictions would suggest that the model is not performing well.

Overall, after running these experiments on all 5 classification models, Logistic Regression generally performed best and there was not a significant difference between ANOVA f-value and mutual information feature selection methods, but it could be argued that mutual information performed slightly better.

Moreover, from the predictions of popularity for Logistic Regression we can observe that the online news articles which are similar to cluster 1 have the highest percentage of Popular articles (71%) and the articles similar to cluster 2 have the highest percentage of Unpopular articles (86%).

Figure 4. Popularity predictions for Logistic Regression of each cluster

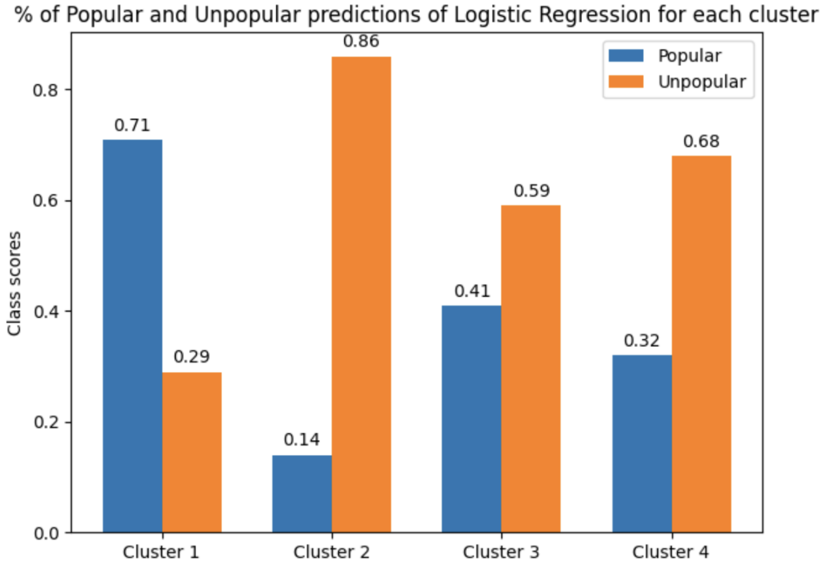
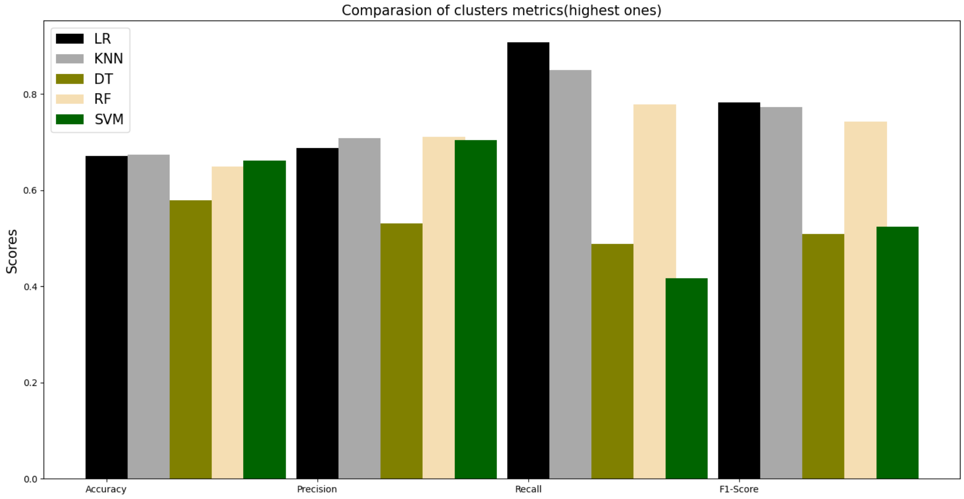


***Figure 5:** Comparison of different metrics from all the models used*



6.2 Discussion

First and foremost we defined the popularity threshold ourselves and this is rather arbitrary. The articles were split using the median, therefore there was around a 1:1 popular to unpopular article ratio. An alternative split could have been used, for instance around the 75th or 90th percentile. This would have led to an imbalance in the dataset wherefor an oversampling or undersampling technique has to be used.

None of the models scored higher than an accuracy of 0.67. This leads to the question: were the models insufficient or is the ε-term relatively high? We believe that the methods used are sufficient to perform the analysis. It must certainly be possible to find a slightly better model but most likely it will not deviate by a lot. There are most likely also other factors that influence the popularity of a news article, but were not included in the provided dataset.

7. Conclusion

This report tried to show to what extent it is possible to predict the popularity of online news articles. Clustering and feature selection was the main mission of this project, it enabled us to make subsets of data of similar articles and their most valuable features to use the machine learning models on them instead of using the models on the whole data set. The highest accuracy is 0.67 and this was achieved using LogisticRegression. Only the second cluster achieved a clear distinction in popular and unpopular articles.

For further analysis it is advisable to increase the number of clusters for higher accuracy and to collect different features and see whether those have a more significant impact on the predicted target variable. Moreover, it would be worth it to compare the results to a model without clustering and feature selection.