SAI
UNIVERSITY

# Machine Learning

## Course Project Report (Classification)

**(Draft - 03, Team No: 02)**

**Title of the project:** Online News Popularity

**Student 1 :** Kavin Fidel, kavin.j-24@sas.saiuniversity.edu.in
**Student 2 :** Ishaan Reddy, ishaan.r-25@scds.saiuniversity.edu.in

**ML Category:** Classification

## 1. Introduction

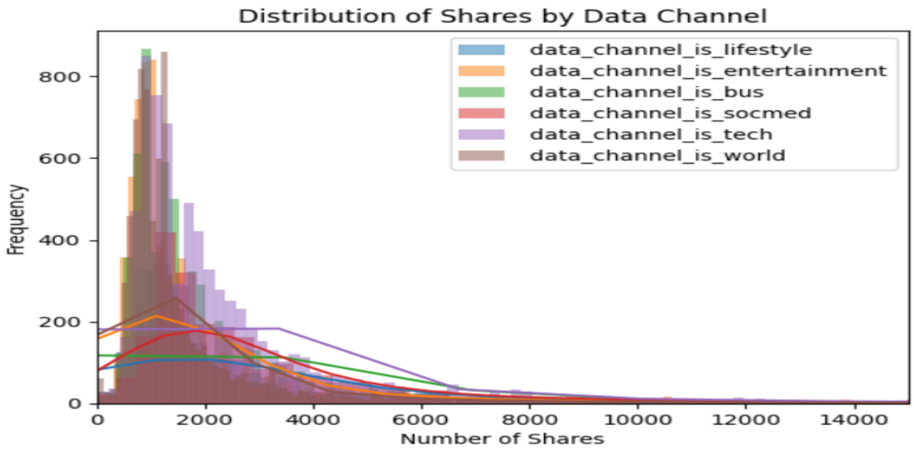
This dataset summarises a heterogeneous set of features about articles published by Mashable in a period of two years. The analysis is done on them to find out which feature/attribute contributes maximum in the popularity and predict the popularity of future news articles in advance before they are published online. This project will try to find the best classification algorithm to predict the popularity of the news.

## 2. Dataset and Features

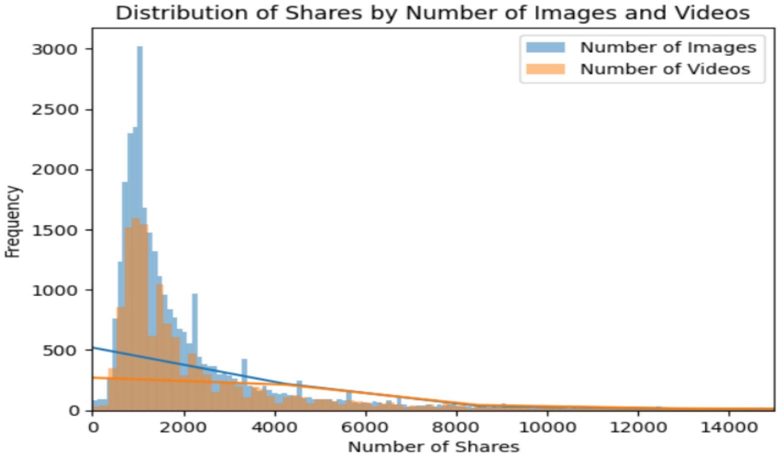
The Online News Popularity dataset contains 61 features and each of them has 39644 rows of data. 'Shares' is the most important feature in the dataset and is used to characterise the popularity. If the number of shares is higher than a predefined threshold then it is categorised as popular, otherwise it is labelled as unpopular.

### Exploratory Data Analysis::

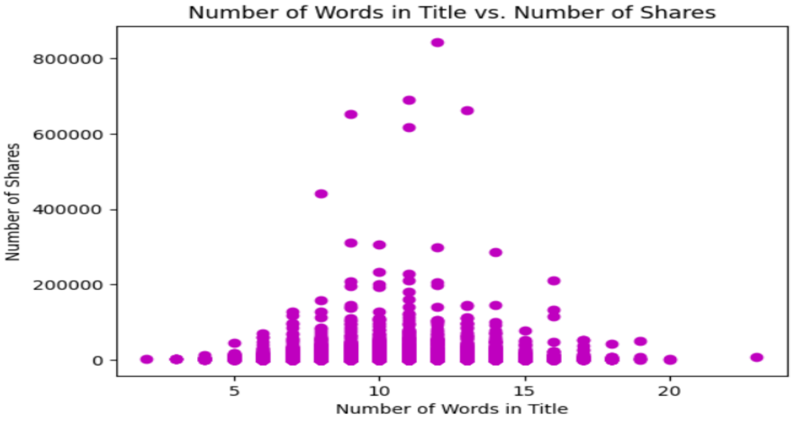
Data Channels vs Number of Shares



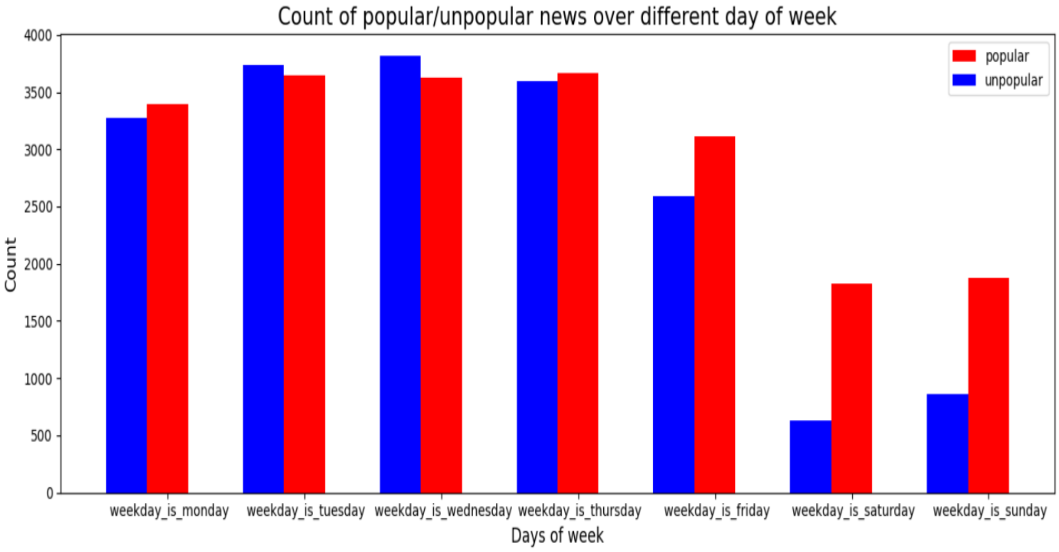
## Number of Videos and Images vs Number of Shares



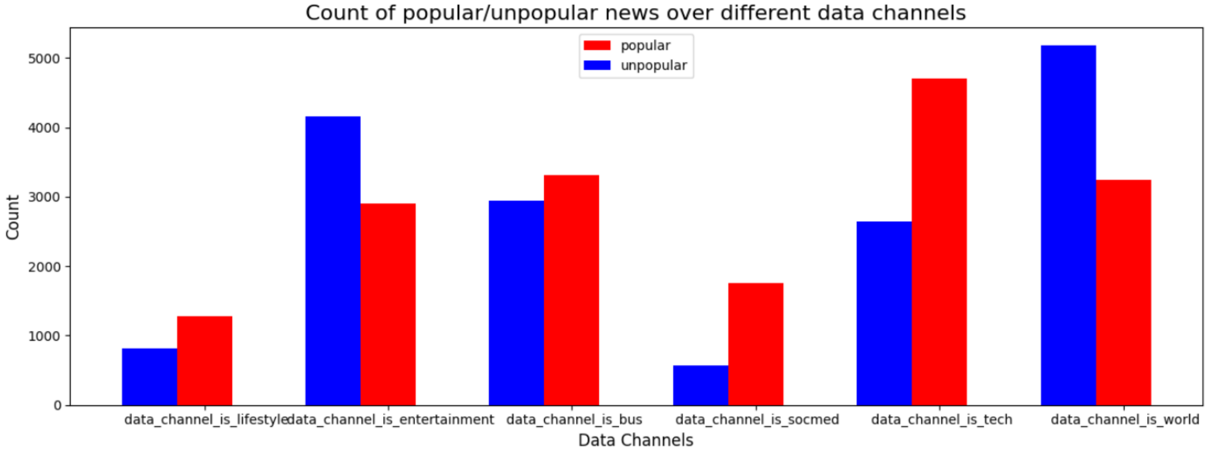
## Number of words in Title vs Number of Shares



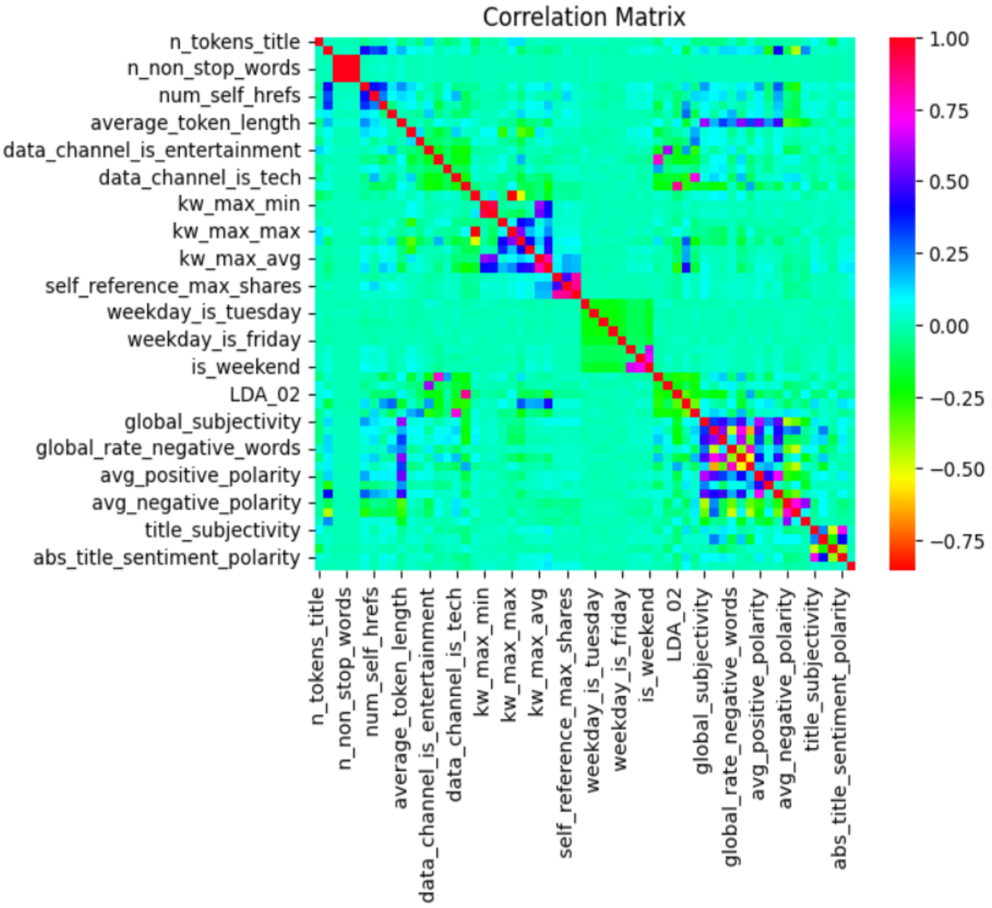
## Popularity vs Days of the week



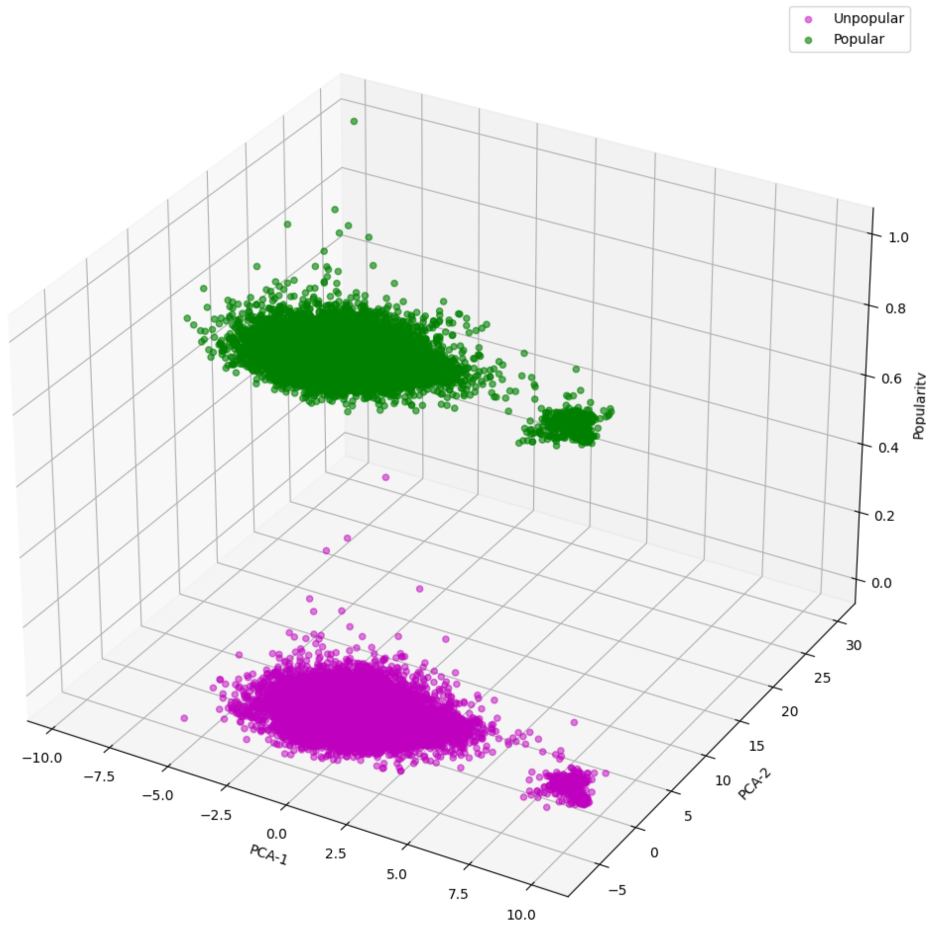
## Popularity vs Data Channels



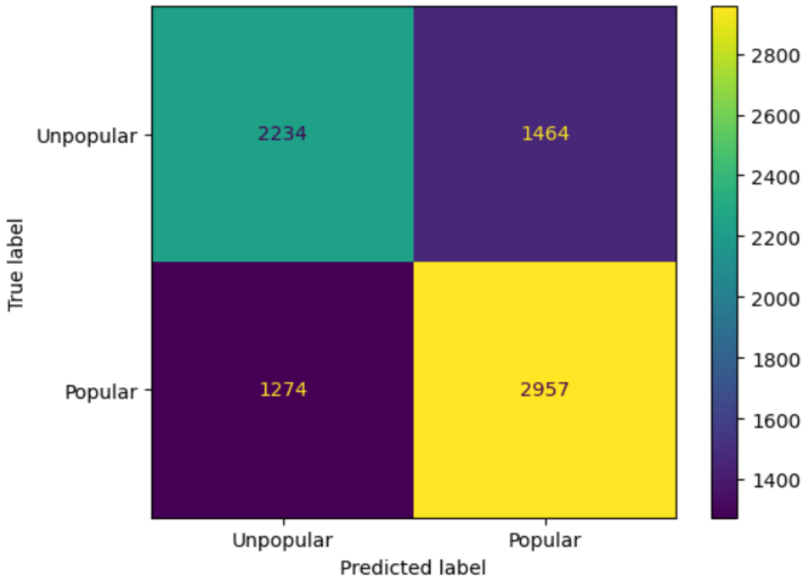
## Correlation Matrix Heatmap



## Principal component analysis



## Confusion Matrix



# 3. Methods

## 3.1 Baseline - Logistic/Softmax Regression

Logistic regression is a linear model for classification. It is an example of supervised learning. It is used to predict the probability of a binary event. The advantage of logistic regression is that the training and predicting speed is very fast. In this case, the binary classification is either popular or unpopular.

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

## 3.2 Support Vector Machine

The use of support vectors to draw a straight line(Hyper-Plane) for classification is called  Support Vector Machine Classification.

Linear SVM:

Linear SVM uses the linear kernel function to the hyperplane that best fits the data.

Kernel SVM:

In kernel SVM  the input data is transformed into a higher dimensional space using the kernel function. This helps in capturing the non-linear relationship between the input features and the target variable(“Shares”). The radial functions could be linear,polynomial  or radial basis function kernels.

The Results of the SVM  models are in section 4.

- Polynomial kernel function takes the dot product of the input data points and adds a constant to the result, which is raised to a power specified by the degree parameter of the function. The result of this transformation is a set of new features that capture the non-linear relationships between the input data.
- Radial Basis Function Kernel is a very powerful kernel used in SVM. Unlike linear or polynomial kernels, RBF is more complex and efficient at the same time that it can combine multiple polynomial kernels multiple times of different degrees to project the

non-linearly separable data into higher dimensional space so that it can be separable using a hyperplane.

## 3.3 Decision Tree

### Classification Tree:

A classification tree is a type of decision tree used for solving classification problems. It represents a hierarchical structure where each internal node represents a feature or attribute, and each leaf node represents a class label or a prediction. The tree is built by recursively partitioning the data based on the values of the features. At each internal node, a decision rule is applied to determine the next feature to split on, aiming to create subsets that are more homogeneous with respect to the target variable.

We use a classification tree to create a model that would be able to accurately predict(classify) the class labels of novel instances by using the decision tree path from the root node to a leaf node.

### CART Algorithm:

The CART (Classification and Regression Trees) algorithm is a popular method used to construct decision trees. It is a greedy algorithm that builds a tree by repeatedly partitioning the data based on the values of the features.
The algorithm starts with the entire dataset and selects the best feature and corresponding splitting point that optimises a specified criterion for class separation.

### Genie Measure:

The Genie measure is an impurity measure used in decision tree algorithms, specifically in the CART algorithm. It calculates the impurity of a node and evaluates the quality of a split. The goal of the Genie measure is to find splits that minimise impurity and increase the homogeneity of the resulting subsets.

## 3.4 Random Forest Classifier:

Random Forest Classifier is an ensemble of decision trees, where each tree is trained on a random subset of the data and features. It combines the predictions of these individual trees to make accurate predictions, utilising the diversity and collective decision-making power of the ensemble. Ensemble learning works by creating an ensemble of trees trained on different subsets of the training data; the subsets are chosen through a bootstrapping process.

## 3.5 AdaBoost Classifier:

AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get a high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and train the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as a base classifier if it accepts weights on the training set.

## 3.6 GradientBoost Classifier:

Gradient boosting classifiers are the AdaBoosting method combined with weighted minimization, after which the classifiers and weighted inputs are recalculated. The objective of Gradient Boosting classifiers is to minimise the loss, or the difference between the actual class value of the training example and the predicted class value.

## 3.7 Hyperparameter Tuning:

In Hyperparameter Tuning, the hyperparameters cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn. Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. The two best strategies for Hyperparameter tuning are:

- GridSearchCV- In GridSearchCV approach, the machine learning model is evaluated for a range of hyperparameter values. This approach is called GridSearchCV, because it searches for the best set of hyperparameters from a grid of hyperparameters values.
- RandomizedSearchCV- RandomizedSearchCV solves the drawbacks of GridSearchCV, as it goes through only a fixed number of hyperparameter settings. It moves within the grid in a random fashion to find the best set of hyperparameters. This approach reduces unnecessary computation.

# 4. Experiments & Results

## 4.1 Protocol

Pre-Processing:

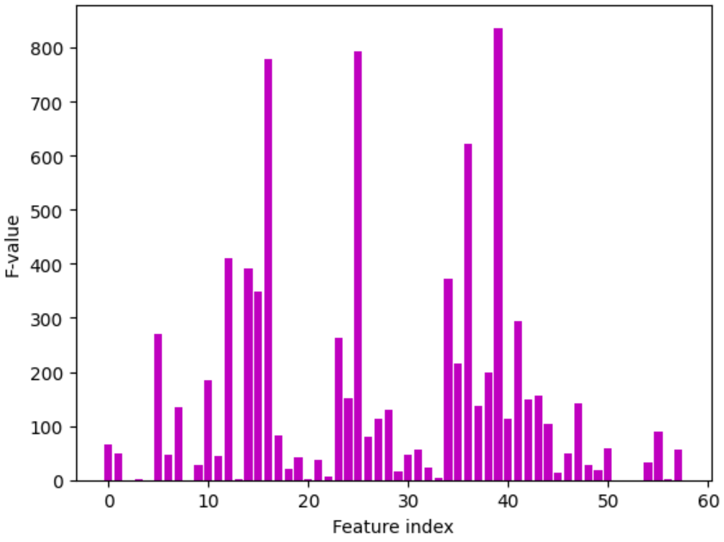
The dataset underwent 2 Preprocessing methods:

- Feature Selection

- Feature Scaling

## Feature Selection:

- Feature selection is a process that chooses a subset of features from the original dataset so that in effect the feature space is optimally reduced.
- For this dataframe the Chi-square method was used to select the features.
- Using the feature selection method, the top 50 most relevant features were copied into the 'dataframe'.



## Feature Scaling:

In order to normalise the range of features in the dataset we use what is called "Feature Scaling". In this model, we make use of Feature standardisation.

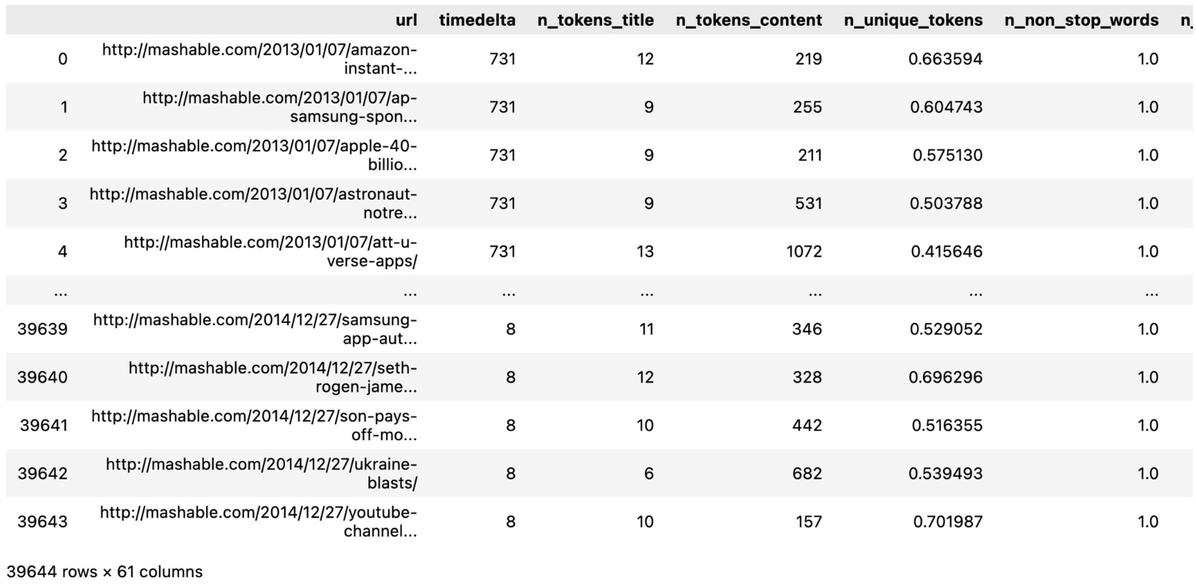
Feature standardisation works by determining the mean and standard deviation for each feature and calculating the new data point by the formula.

$$\tilde{x} = \frac{x-\mu}{\sigma}$$

σ: Standard Deviation

μ: Mean

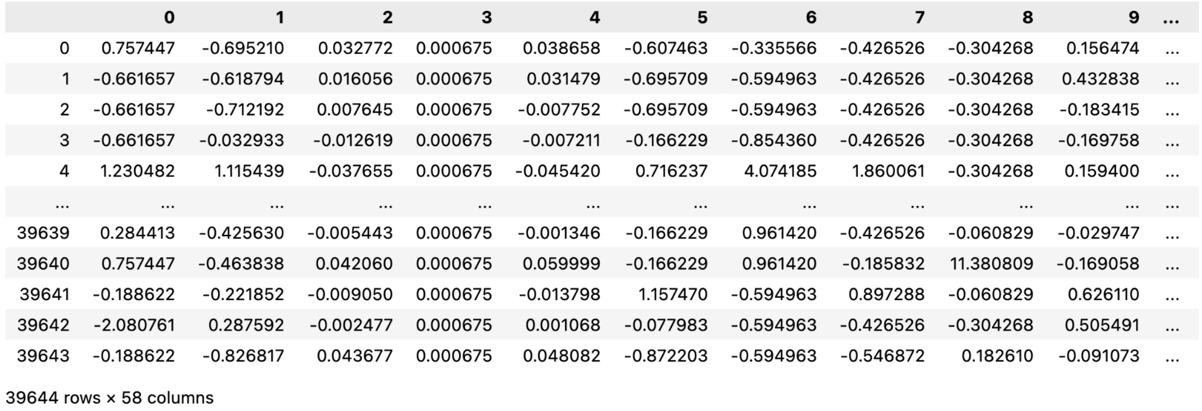
## Dataset before standardisation:



| | url | timedelta | n_tokens_title | n_tokens_content | n_unique_tokens | n_non_stop_words | n_ |
|---|---|---|---|---|---|---|---|
| 0 | http://mashable.com/2013/01/07/amazon-instant-... | 731 | 12 | 219 | 0.663594 | 1.0 | |
| 1 | http://mashable.com/2013/01/07/ap-samsung-spon... | 731 | 9 | 255 | 0.604743 | 1.0 | |
| 2 | http://mashable.com/2013/01/07/apple-40-billio... | 731 | 9 | 211 | 0.575130 | 1.0 | |
| 3 | http://mashable.com/2013/01/07/astronaut-notre... | 731 | 9 | 531 | 0.503788 | 1.0 | |
| 4 | http://mashable.com/2013/01/07/att-u-verse-apps/ | 731 | 13 | 1072 | 0.415646 | 1.0 | |
| ... | ... | ... | ... | ... | ... | ... | |
| 39639 | http://mashable.com/2014/12/27/samsung-app-aut... | 8 | 11 | 346 | 0.529052 | 1.0 | |
| 39640 | http://mashable.com/2014/12/27/seth-rogen-jame... | 8 | 12 | 328 | 0.696296 | 1.0 | |
| 39641 | http://mashable.com/2014/12/27/son-pays-off-mo... | 8 | 10 | 442 | 0.516355 | 1.0 | |
| 39642 | http://mashable.com/2014/12/27/ukraine-blasts/ | 8 | 6 | 682 | 0.539493 | 1.0 | |
| 39643 | http://mashable.com/2014/12/27/youtube-channel... | 8 | 10 | 157 | 0.701987 | 1.0 | |

39644 rows × 61 columns

## Dataset after standardisation:



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.757447 | -0.695210 | 0.032772 | 0.000675 | 0.038658 | -0.607463 | -0.335566 | -0.426526 | -0.304268 | 0.156474 | ... |
| 1 | -0.661657 | -0.618794 | 0.016056 | 0.000675 | 0.031479 | -0.695709 | -0.594963 | -0.426526 | -0.304268 | 0.432838 | ... |
| 2 | -0.661657 | -0.712192 | 0.007645 | 0.000675 | -0.007752 | -0.695709 | -0.594963 | -0.426526 | -0.304268 | -0.183415 | ... |
| 3 | -0.661657 | -0.032933 | -0.012619 | 0.000675 | -0.007211 | -0.166229 | -0.854360 | -0.426526 | -0.304268 | -0.169758 | ... |
| 4 | 1.230482 | 1.115439 | -0.037655 | 0.000675 | -0.045420 | 0.716237 | 4.074185 | 1.860061 | -0.304268 | 0.159400 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 39639 | 0.284413 | -0.425630 | -0.005443 | 0.000675 | -0.001346 | -0.166229 | 0.961420 | -0.426526 | -0.060829 | -0.029747 | ... |
| 39640 | 0.757447 | -0.463838 | 0.042060 | 0.000675 | 0.059999 | -0.166229 | 0.961420 | -0.185832 | 11.380809 | -0.169058 | ... |
| 39641 | -0.188622 | -0.221852 | -0.009050 | 0.000675 | -0.013798 | 1.157470 | -0.594963 | 0.897288 | -0.060829 | 0.626110 | ... |
| 39642 | -2.080761 | 0.287592 | -0.002477 | 0.000675 | 0.001068 | -0.077983 | -0.594963 | -0.426526 | -0.304268 | 0.505491 | ... |
| 39643 | -0.188622 | -0.826817 | 0.043677 | 0.000675 | 0.048082 | -0.872203 | -0.594963 | -0.546872 | 0.182610 | -0.091073 | ... |

39644 rows × 58 columns

## Splitting the dataset:

After the features in the dataset were  filtered and scaled, the dataset was split into a training set and testing set with the 80-20 percent proportion respectively. The random_state parameter was set to 42 to preserve the randomness of the split.

## 4.2 Results

## Logistic Regression:

Logistic Regression yielded a score value of : 0.65

## Cross Validation:

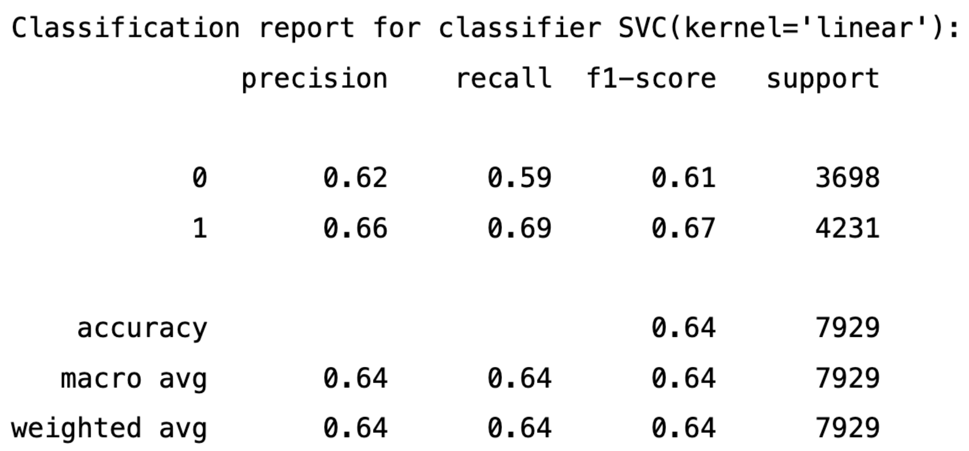
The mean and standard deviation of the cross validation is as follows:

$$0.682 + / - 0.00489$$

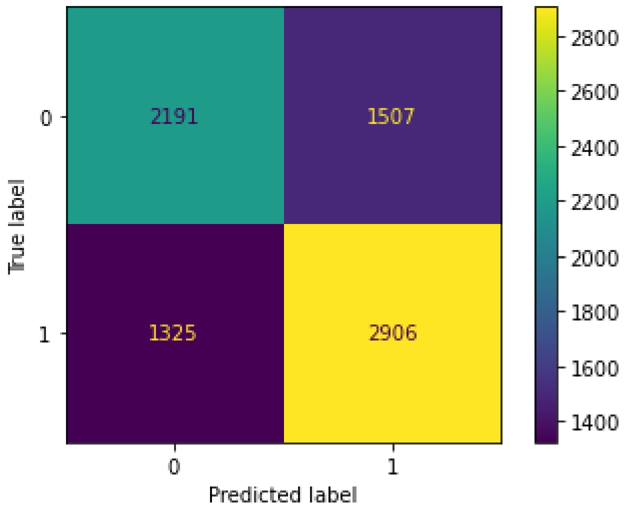
## SVM Classification:

### Linear:

Linear SVM gave an accuracy score of : 0.6486

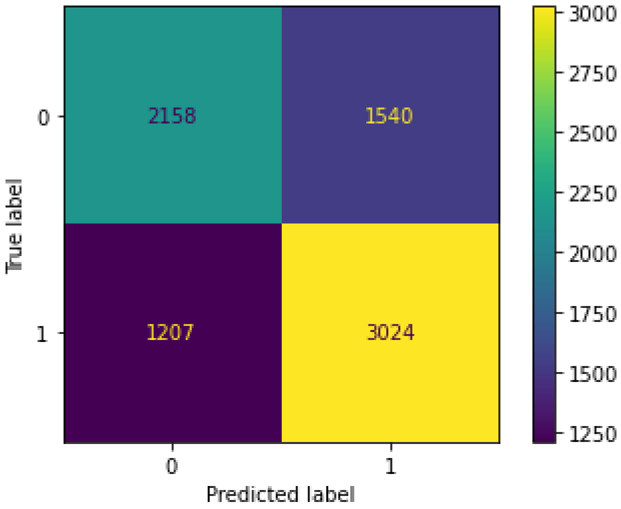


```
Classification report for classifier SVC(kernel='linear'):
              precision    recall  f1-score   support

           0       0.62      0.59      0.61      3698
           1       0.66      0.69      0.67      4231

    accuracy                           0.64      7929
   macro avg       0.64      0.64      0.64      7929
weighted avg       0.64      0.64      0.64      7929
```



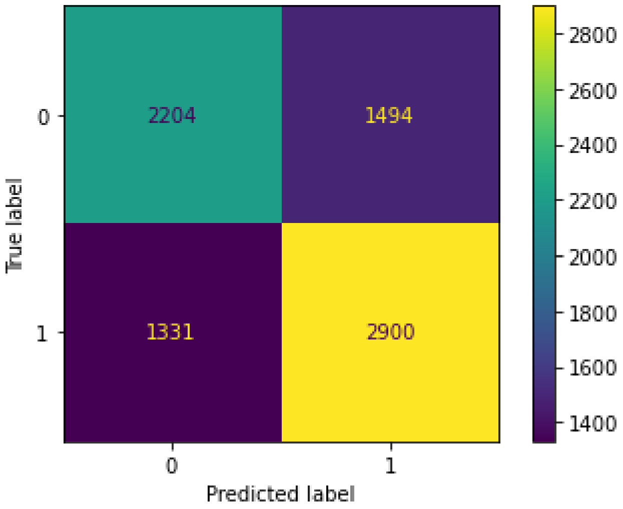
RBF:

```
Classification report for classifier SVC():
              precision    recall  f1-score   support

           0       0.64      0.58      0.61      3698
           1       0.66      0.71      0.69      4231

    accuracy                           0.65      7929
   macro avg       0.65      0.65      0.65      7929
weighted avg       0.65      0.65      0.65      7929
```



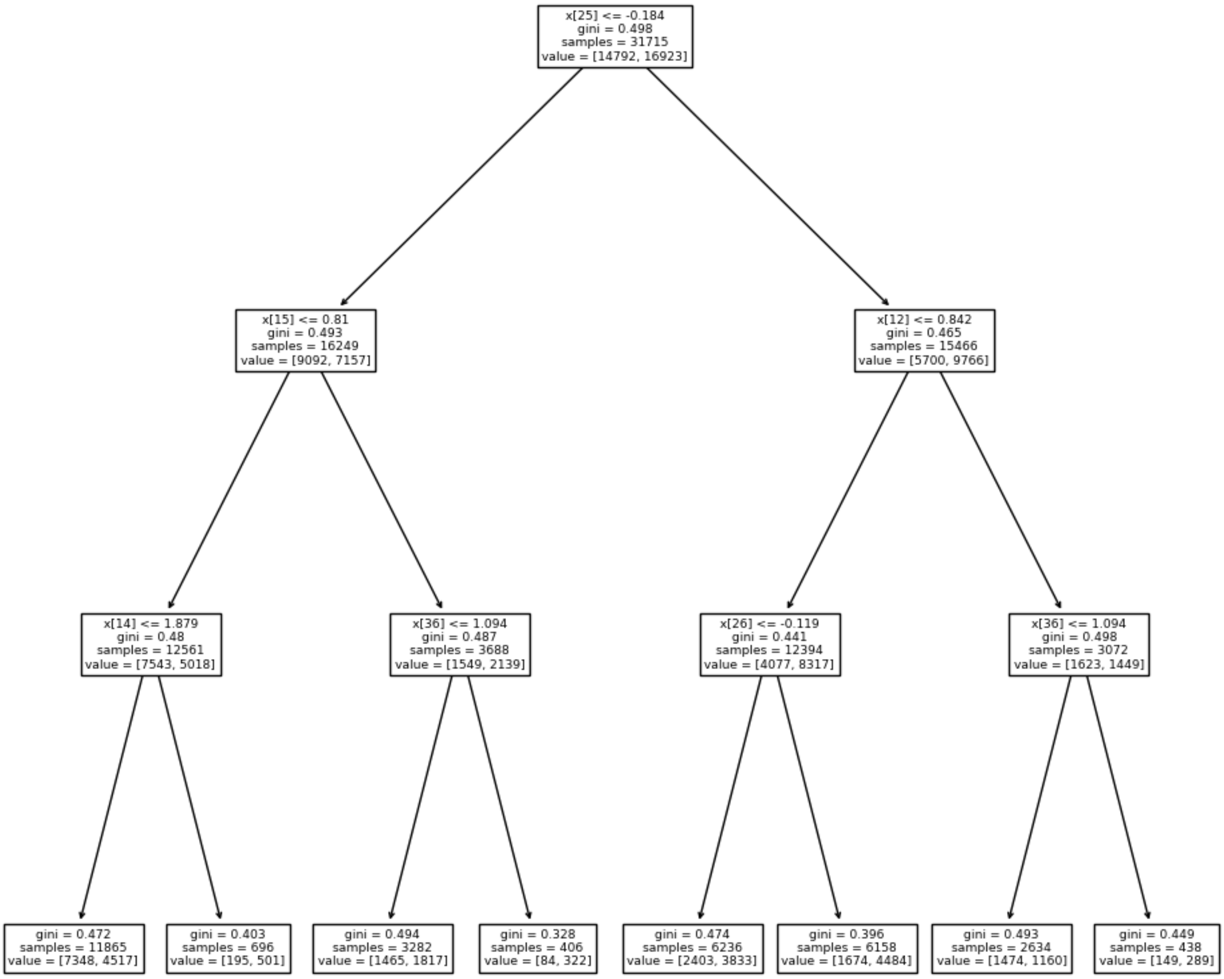
Polynomial:

```
Classification report for classifier SVC(kernel='poly'):
              precision    recall  f1-score   support

           0       0.62      0.60      0.61      3698
           1       0.66      0.69      0.67      4231

    accuracy                           0.64      7929
   macro avg       0.64      0.64      0.64      7929
weighted avg       0.64      0.64      0.64      7929
```


2204
1494
1331
2900
0
1
True label
0
1
Predicted label
2800
2600
2400
2200
2000
1800
1600
1400

## Decision Trees:

The accuracy score of the Decision tree classifier was approximately 0.634.



(Visualisation of the decision tree with max_depth = 3)

## Random Forest Classifier:

The random forest classifier yielded a score value of 0.6593.

### AdaBoost Classification:

The AdaBoost classification yielded a score value of 0.6567.

### GradientBoost Classification:

The GradientBoost classification yielded a score value of 0.6679.

### Hyperparameter Tuning:

- GridSearchCV- This model yielded a score value of 0.6487.
- RandomizedSearchCV- This model yielded a score value of 0.6496.

## 5. References

- Demir, M. (2023, April 7). Machine Learning Algorithm Series Polynomial Kernel SVM: Understanding the Basics and Applications. Medium. https://blog.devgenius.io/machine-learning-algorithm-series-polynomial-kernel-svm-understanding-the-basics-and-applications-89b4b42df137

- The RBF kernel in SVM: A Complete Guide. (2022, December 12). PyCodeMates. https://www.pycodemates.com/2022/10/the-rbf-kernel-in-svm-complete-guide.html

- Y. (2017, July 31). MLND-Online-News-Popularity-Prediction/Project Report.pdf at master · ymdong/MLND-Online-News-Popularity-Prediction. GitHub. https://github.com/ymdong/MLND-Online-News-Popularity-Prediction

- UCI Machine Learning Repository: Online News Popularity Data Set. (n.d.). UCI Machine Learning Repository: Online News Popularity Data Set. https://archive.ics.uci.edu/ml/datasets/online+news+popularity
- Navlani, A. (2018, November 20). AdaBoost Classifier in Python. https://www.datacamp.com/tutorial/adaboost-classifier-python
- Nelson, D. (2023, January 19). Gradient Boosting Classifiers in Python with Scikit-Learn. Stack Abuse. https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/
- GeeksforGeeks. “Hyperparameter Tuning.” *GeeksforGeeks*, Aug. 2022, www.geeksforgeeks.org/hyperparameter-tuning.