ONLINE NEWS POPULARITY:

Elena Gomez

35459621

1. Abstract

In recent years, the number of people that read online news is increasing. Mainly because it is cheaper and easier than reading newspapers. As online news is reaching more and more people, companies take the opportunity to advertise their services or products on these web pages. It is important for media companies to know which articles are interesting for readers and become more popular, and which articles do not. In order to know if advertising their product or services in a specific article is worth it or not. In this project, I trained different models to predict how popular online news is. I used a database of online news articles from the New York Times in order to predict news popularity based on various features using supervised learning. The number of shares of an article determines if it is popular or not. I treated this problem as a regression problem, by predicting the number of shares, and as a classification problem. By creating the classes "popular" (if the article has more shares than the mean) or "unpopular" (if the article has fewer shares or is equal to the mean). In this project, I thoroughly studied the dataset by using some visualization techniques and I did data preprocessing (removing outliers and/or missing values, removing correlated variables, transforming variables, and creating new variables). I applied different classification techniques: Logistic regression, decision trees, and random forest. And regression techniques such as simple linear regression and multiple regression. Then, I compared the performance of these techniques by comparing different metrics of the models. I focused mainly on accuracy. Finally, I selected the best model and discussed if this work could be useful in the real world and if this model could be used for companies for decision-making.

Keywords data science, online news, popularity, classification, regression, decision-making.

2. Overview and Motivation

Advertising is one of the most effective means for companies to expand their reach and promote their products or services to a larger audience. However, selecting the optimal placement for advertisements can be a complex decision due to various factors. Because maybe they advertise products on web pages or articles that are not popular, so they do not reach as many people as expected. And advertisements are usually very expensive, so do not select the best placement for an advertisement could suppose an important waste of money for a company. To avoid this issue, and help the companies with this decision-making process, I trained some models to predict if an article is popular or not. Some of the features fall into various categories such as quantitative information about the article, such things as number of images, number of videos, etc. And qualitative information about the article, such as which day it was published and which topic the article falls under. The most important variable is the number of shares because it is the variable that determines the popularity of the news, it is the target, the variable that I predicted. I also treated this problem as a classification problem by distinguishing between “popular” and “unpopular” articles.

3. Literature Review

3.1 Relevant Work

There are some papers that talk about online news popularity and in which some techniques have been used to make predictions about the number of shares and the popularity of online news. There are especially 4 papers that I found interesting (they were published between 2010 and 2021). The models that are trained in each paper are the following ones:

In [1], A model based on the behaviour of the user to predict the popularity of online news is proposed. In this paper logistic regression model is used to predict the number of votes that a news article would receive based on the number of votes received by similar news articles in the past.

In [2], They proposed a model based on the content and sentiment of news articles to predict their popularity. The suport vector regression model is trained to predict the popularity of news articles based on the number of views, comments, and shares received by similar news articles in the past.

In [3], The authors proposed a model based on the news source and topic of news articles to predict their popularity. They used a decission tree algorithm to predict online news popularity. Taheylso analyzed the news source and topic of the news article and found that these factors also played an important role in the popularity of news articles.

In [4], In this paper they train random forest regresion model to predict the popularity of news articles based on the number of shares, likes, and comments received by similar news articles on social media platforms. Additionally, they analyzed various other features such as the time of publication, the sentiment of the news article, and the engagement of the news website on social media, and found that these features also played an important role in the popularity of news articles.

3.2 Contributions of The Project

In my project, I predicted online news popularity by using different techniques. Some of them are also developed in the papers I have included in the literature review. Although I trained different models, the performance of some of them is not as satisfactory as expected. But I will evaluate and compare the performance of the model trained in this project later. In addition to the data understanding, data visualization, and data preprocessing, my contributions to the project are the following:

- CLASSIFICATION TECHNIQUES:
    - ➢ Logistic regression.
    - ➢ Decision trees. (We will study with which hyperparameter the performance of the model is better.)
    - ➢ Random forest. (We will study with which hyperparameter the performance of the model is better.)
- REGRESSION TECHNIQUES:
    - ➢ Simple linear regression
    - ➢ Multiple regression
    - ➢ I planned to apply also forward regression, backward regression, and stepwise regression, but after evaluating the performance of simple linear regression and multiple regression, I decided not to train these ones. (We will see this later)

4. Technology Details

To begin with, I did an exploratory analysis and preprocessing of the dataset. In this project, I worked with a database of online news articles from the New York Times database of online news articles from the New York Times.

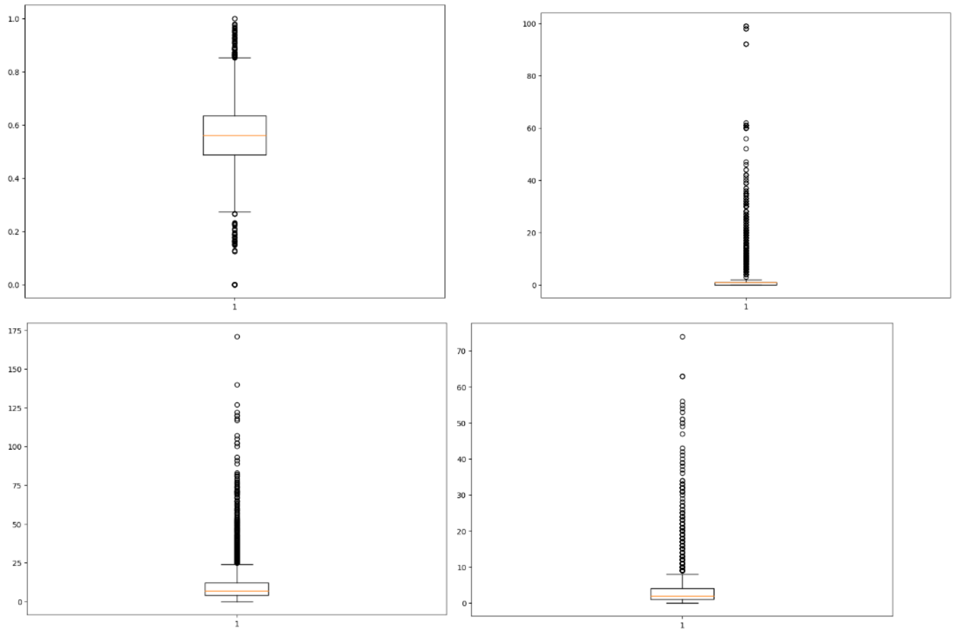
DATA PREPROCESSING

The dataset has 40000 observations, it is too big. So, I took a sample of 10000 observations, because if not the computations would be very expensive.

- Missing values: There are no missing values, so I did not have to remove any variable or observation.
- Irrelevant variables: In the dataset, there are some irrelevant variables that are not necessary to compute the models. So, I removed them.
  Firstly, I removed “url” and “timedelta” because they are non-predictive. I also removed LDA variables. Finally, I removed the variable weekend (“is_weekend”) because there isone variable that determines if the news was published on Saturday(“weekday_is_saturday”) and another one that determines if the news was published on Sunday (“weekday_is_sunday”). So I do not need the variable “is_weekend”.

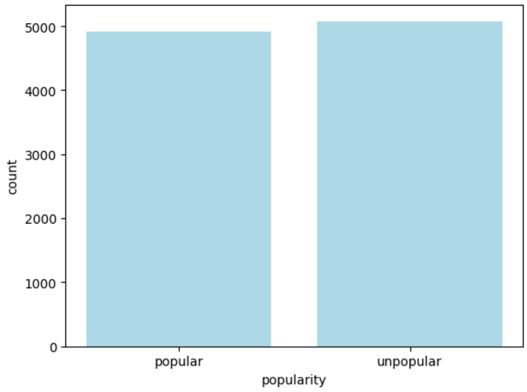
- Converting variables: there are some variables that are factors, but they are considered as "float64".So I had to convert them into factors. These variables are : ' data_channel_is_bus', ' data_channel_is_entertainment’, ' data_channel_is_lifestyle', ' data_channel_is_socmed', ' data_channel_is_tech', ' data_channel_is_world', ' weekday_is_monday', ' weekday_is_tuesday', ' weekday_is_wednesday', ' weekday_is_thursday', ' weekday_is_friday', ' weekday_is_saturday', and ' weekday_is_sunday'.

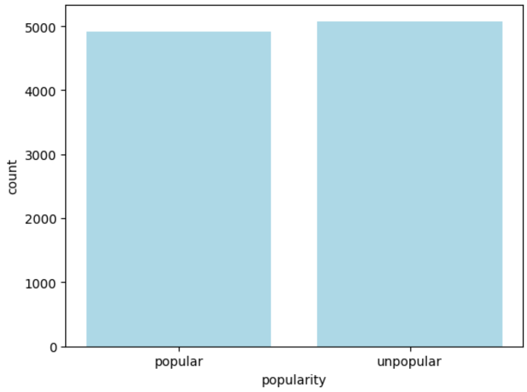
- Creating new variables: I created new variables by using the ones that I already had. The new variables I created consist of transforming the target, which is a numerical variable, into a categorical one. Because I need this variable to be categorical to train the classification models.
  I determined if the popularity of news is "popular" or "unpopular" depending on the number of shares they have.
  I considered that an article is popular if it has equal or fewer shares than the mean of shares and it is unpopular if it has more shares than the mean of shares.
- Outliers: I represented all the numerical and integers variables in different boxplots in order to know if our database has outliers. I scale the data before plotting it. Some of the boxplots are the following ones (plots for the variables ' n_unique_tokens', ' num_imgs', ' num_hrefs', and ' num_self_hrefs'):



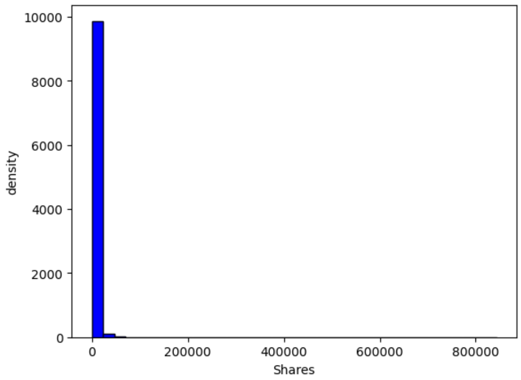
VISUALIZATION TOOLS

I plotted the target variables in order to understand them better. In the image below, we can see a histogram of the popularity. As we can see, more less 1/2 of the articles are popular. And around 1/2 of the articles are unpopular.

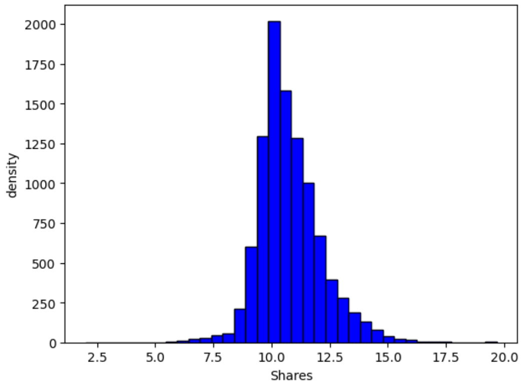




The distribution of the variables ‘shares’ is very skewed as can be seen in the plot below.



I applied a transformation in order to obtain a closer to a normal distribution. In this case, we use the log.



CLASSIFICATION TECHNIQUES

Firstly, I predicted the popularity class, using some classification tools. Classification models are a type of machine learning algorithm that is used to predict the class or category that a data point belongs to. The goal of a classification model is to learn a function that maps input features of a data point to a predicted output class. The class that is predicted is called the target.

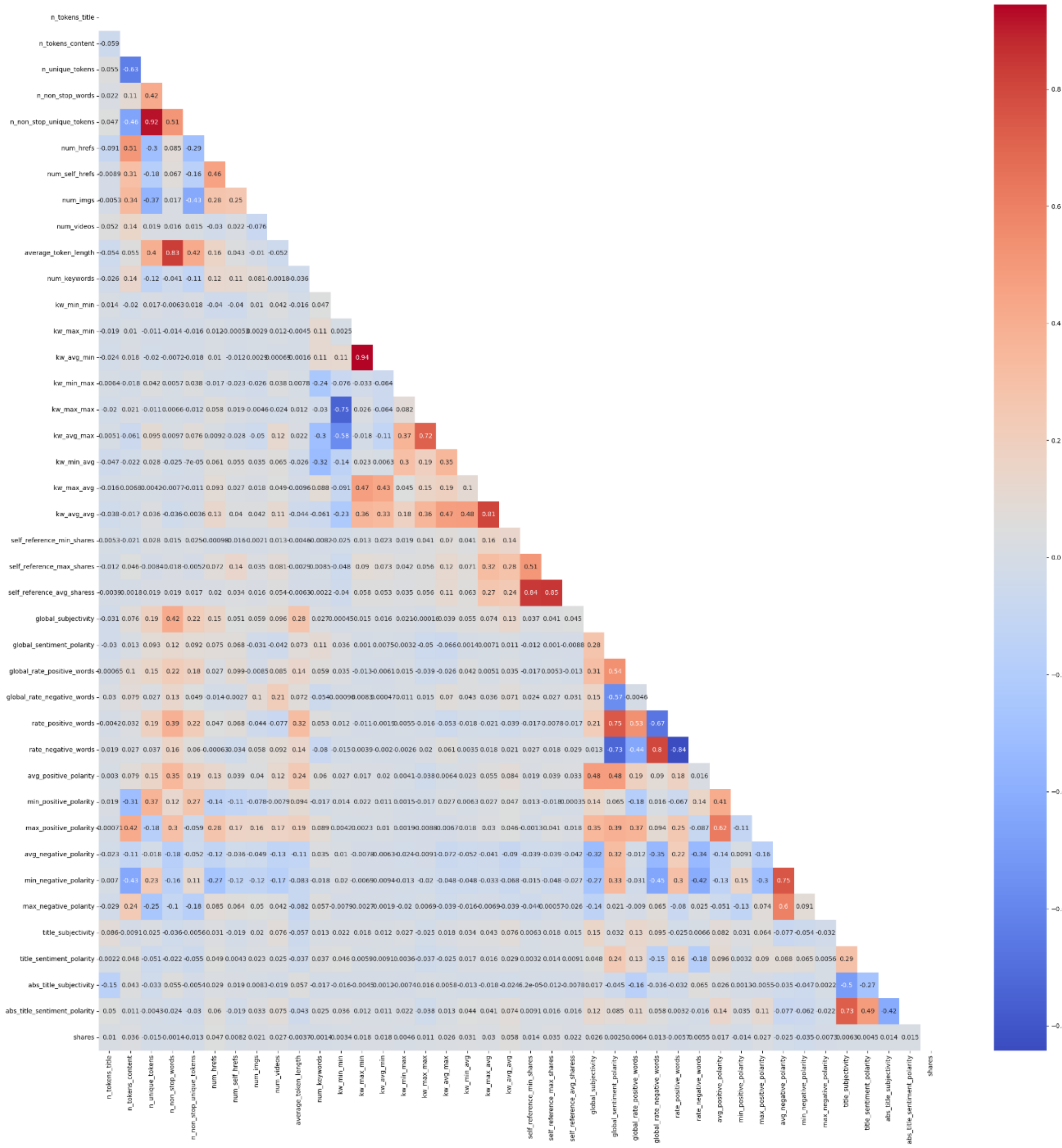
To begin with, I saw the correlation between all the numerical variables. And I removed variables with a strong correlation, because if not I would obtain collinearity errors.

Correlation is a measure of the linear relationship between two variables. When two variables are strongly correlated, it means that they tend to vary together in a predictable way. In the context of machine learning, this can be problematic because it can lead to overfitting, which occurs when a model is too complex and captures noise or patterns that are specific to the training data but not generalizable to new data. When two variables are strongly correlated, it means that they contain redundant information. Including both variables in our model can lead to overfitting because the model will give undue importance to both variables, even though they are measuring the same underlying phenomenon. By removing one of the correlated variables, I simplify the models and reduce the risk of overfitting. This way, I obtain a better generalization performance on new, unseen data.

If I do not remove correlated variables, the consequences can be that the models may suffer from overfitting, resulting in poor generalization performance on new data. This means that the models may perform well on the training data but not on new data, which will train a useless model.

In the plot below, we can see the correlation between the variables. Dark red and dark blue squares mean that the correlation is high. Dark blue squares mean that the correlation is strong and negative and dark blue squares mean that the correlation is strong and positive.

If the correlation between two variables is stronger than -0.8 or stronger than 0.8, I removed one of these two in order to reduce collinearity.



After removing correlated variables I splitted the data into train and test. I used 80% of the dataset to train the model and 20% of the dataset to test the model. This means that I used a dataset consisting of 8000 rows for training the model, and a separate dataset of 2000 rows for testing and evaluating its performance.

After pre-processing the data by removing correlated variables and splitting it into separate training and testing sets, I proceeded to train the following models:

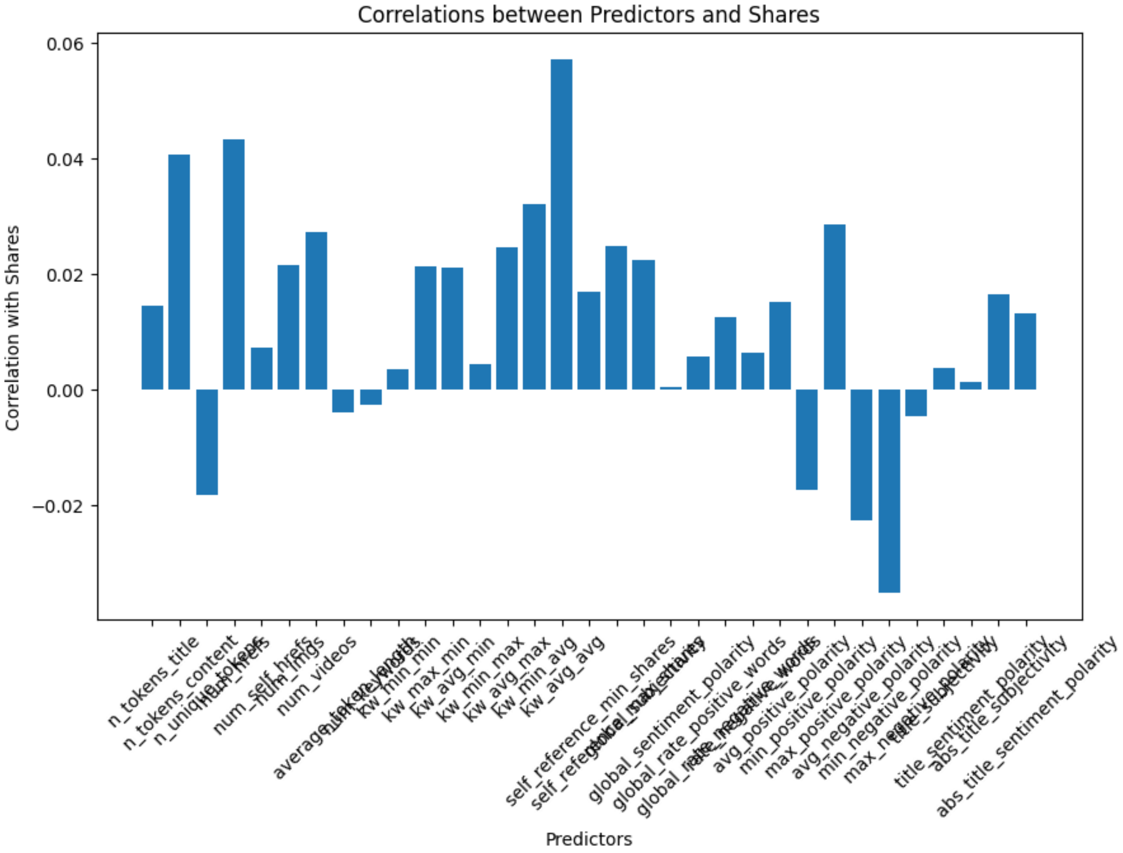
- **Logistic regression**: Logistic regression is a popular statistical model used for binary classification, utilizing the sigmoid function. Logistic regression can be extended to multi-class classification too. It helps us to understand the relationship between the dependent variable and independent variables by estimating probabilities using a logistic regression equation. I will use the division that I have done before in which popularity is divided between popular and unpopular.

- **Decision trees and random forests:** These models divide the data into more homogeneous subsets and then repeatedly into even smaller subsets. Until the subsets are sufficiently homogeneous. I computed these models firstly using the parameters by default, and then, I studied with which parameters I obtain better results. And I trained the models with these parameters that improve their accuracy. I optimized the parameters using a grid search. The hyperparameters that are optimized are the maximum depth and the function used to measure the quality of a split.
    - The maximum depth of any node of the final tree is a limit to stop the further splitting of nodes when the specified tree depth has been reached during the building of the initial decision tree. I trained the model with max depths 2,3,5,10, and 20. A large depth of the tree can produce overfitting while a small depth can produce underfitting.
    - Functions are used to measure the quality of a split. I trained the model with Gini and entropy. The Gini impurity measures the frequency at which any element of the dataset will be mislabelled when it is randomly labeled. Thus, the optimum split is chosen by the features with less Gini Index. And entropy is a measure of information that indicates the disorder of the features with the target. Similar to the Gini Index, the optimum split is chosen by the feature with less entropy. It gets its maximum value when the probability of the two classes is the same and a node is pure when the entropy has its minimum value, which is 0. One of the differences between these two formulas is that entropy is more complex since it makes use of logarithms. So, the calculation of the Gini index will be faster.

Decision trees capture non-linearities and they provide ease of interpretation. They are the basis of popular and powerful tools, for instance, random forests. Random Forest is the most-known bagging learning tool, it uses random subspace methods to reduce correlations between the trees. It improves the predictive performance of DTs by averaging them. The advantages of this model are that is more accurate, and reduces variance and overfitting. I follow the same procedure as when I trained decision trees.

REGRESSION TECHNIQUES

At first, the idea was to apply some linear regression techniques, simple linear regression, and multiple linear regression. But, the results obtained were really disappointing. These disappointing results make sense because the correlations between the predictors and the variable ‘shares’ (the target) are extremely low. This can be seen in the plot below.



I trained a simple linear regression model using as an independent variable the one that is most correlated to “shares”. In this case, is “kw_average_avg”.
Then, I trained a multiple linear regression model.

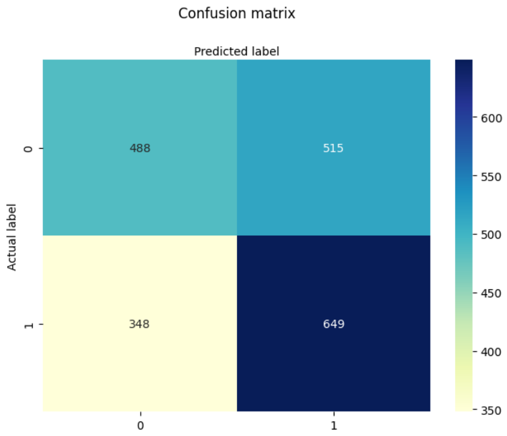
5. Performance Evaluations

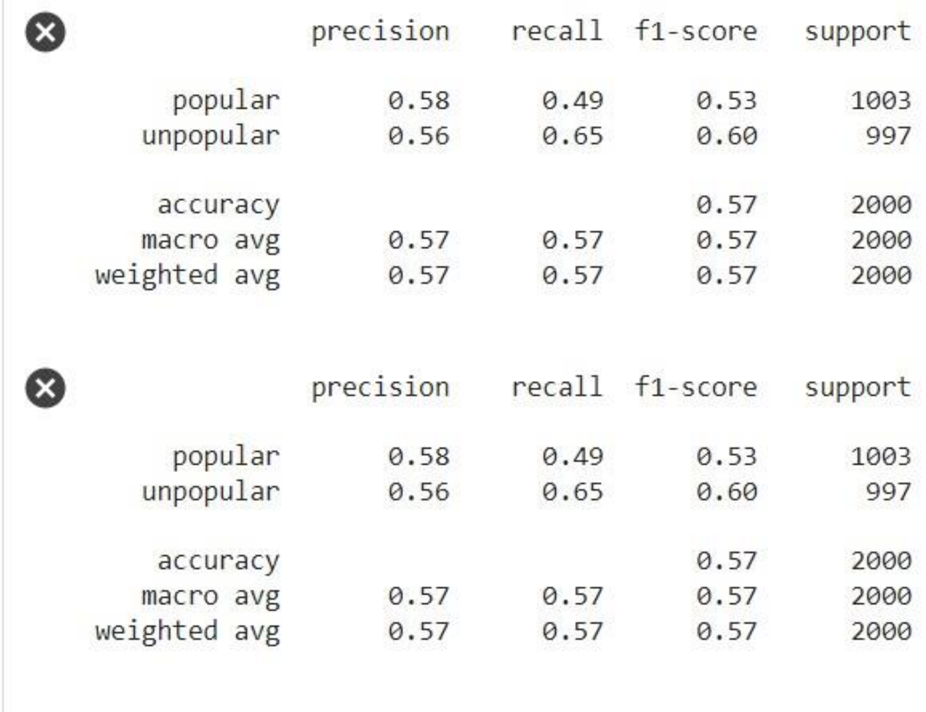
Let’s evaluate the performance of the models.

CLASSIFICATION MODELS

LOGISTIC REGRESSION

After training the logistic regression model I obtained the confusion matrix that can be seen below. This confusion matrix is not satisfactory, because the labels are not corrected classified in plenty of cases. I get an accuracy of 57%.



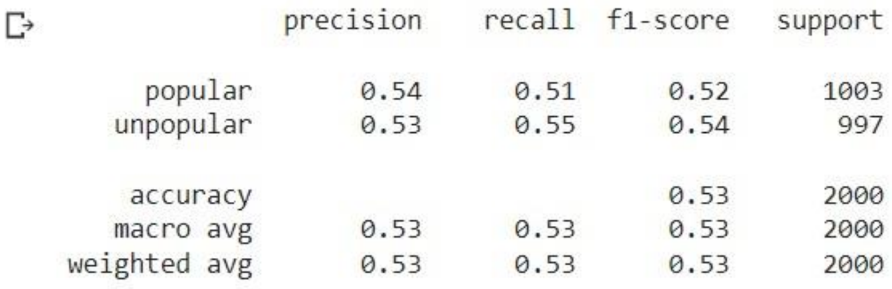


```
              precision    recall  f1-score   support

     popular       0.58      0.49      0.53      1003
   unpopular       0.56      0.65      0.60       997

    accuracy                           0.57      2000
   macro avg       0.57      0.57      0.57      2000
weighted avg       0.57      0.57      0.57      2000
```

```
              precision    recall  f1-score   support

     popular       0.58      0.49      0.53      1003
   unpopular       0.56      0.65      0.60       997

    accuracy                           0.57      2000
   macro avg       0.57      0.57      0.57      2000
weighted avg       0.57      0.57      0.57      2000
```

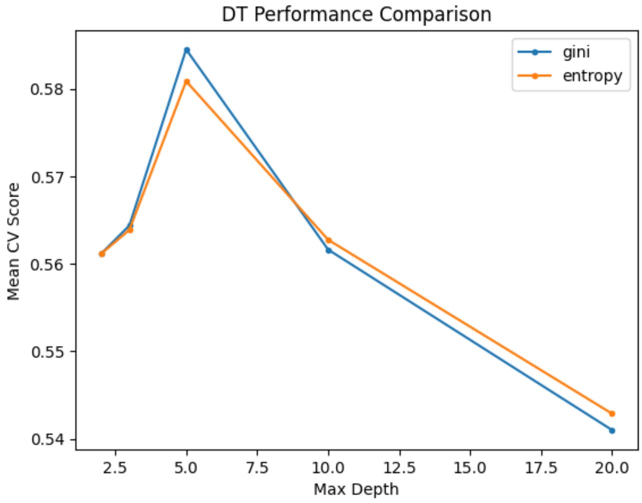
DECISION TREES

Firstly, I computed the models with the parameters by default, and I obtained the following results.

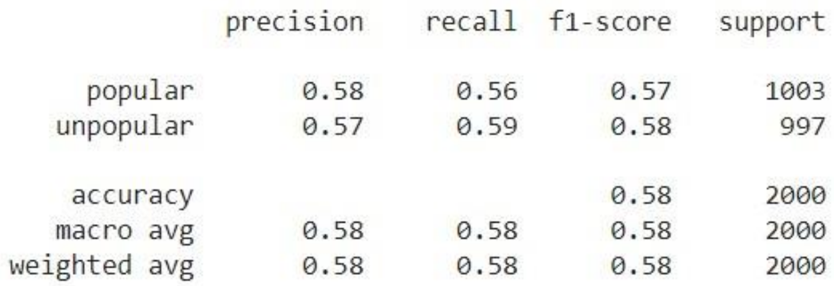


```
              precision    recall  f1-score   support

     popular       0.54      0.51      0.52      1003
   unpopular       0.53      0.55      0.54       997

    accuracy                           0.53      2000
   macro avg       0.53      0.53      0.53      2000
weighted avg       0.53      0.53      0.53      2000
```

The obtained accuracy is 53%, lower than for logistic regression. I studied how the results change when tuning hyperparameters.



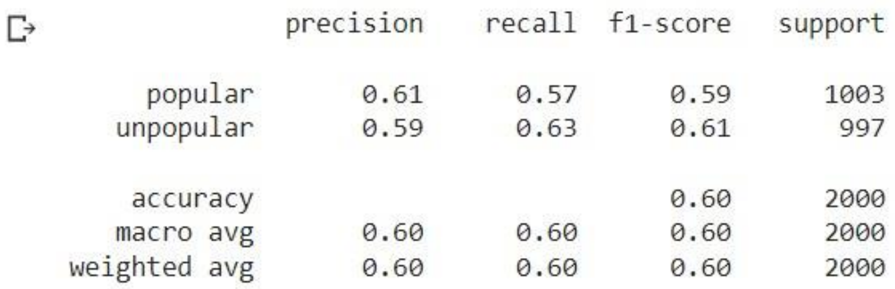
The best result is obtained for maxdepth=5 combined with Gini. For these parameters, the accuracy is increased to 58%.



```
              precision    recall  f1-score   support

     popular       0.58      0.56      0.57      1003
   unpopular       0.57      0.59      0.58       997

    accuracy                           0.58      2000
   macro avg       0.58      0.58      0.58      2000
weighted avg       0.58      0.58      0.58      2000
```

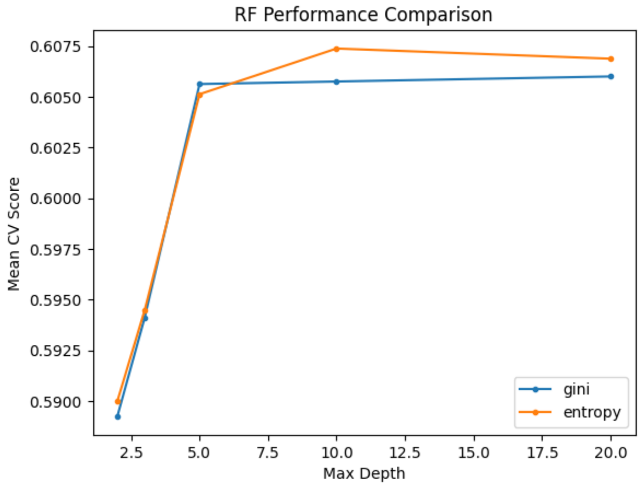
RANDOM FOREST

With the parameters by default, I obtained the following results.

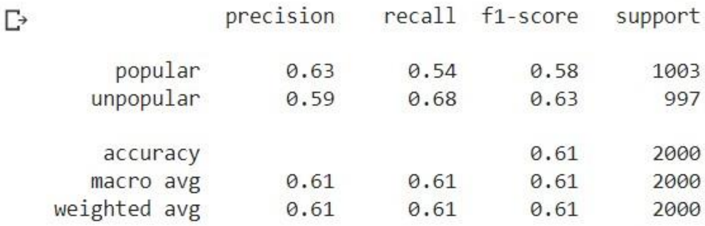


```
              precision    recall  f1-score   support

     popular       0.61      0.57      0.59      1003
   unpopular       0.59      0.63      0.61       997

    accuracy                           0.60      2000
   macro avg       0.60      0.60      0.60      2000
weighted avg       0.60      0.60      0.60      2000
```

When tuning hyperparameters, I got the results that can be seen below.



The best result is obtained for maxdepth=10 combined with Entropy. For these parameters, the accuracy is increased to 61%. Accuracy is improved a little bit (61%).



```
              precision    recall  f1-score   support

     popular       0.63      0.54      0.58      1003
   unpopular       0.59      0.68      0.63       997

    accuracy                           0.61      2000
   macro avg       0.61      0.61      0.61      2000
weighted avg       0.61      0.61      0.61      2000
```

REGRESSION MODELS

SIMPLE LINEAR REGRESSION

As I explained before, I trained a simple linear regression model using as an independent variable the most correlated one to “shares”, which is “kw_avg_avg”, but the results are very unsatisfactory.

MULTIPLE LINEAR REGRESSION

When training multiple linear regression, the results are very unsatisfactory too.

I am not plotting the results for multiple linear regression and simple regression. Because in both cases they are extremely bad. Errors are very high. The high errors in these regression models can be attributed to the correlation between the dependent variables ("shares") and the independent variables. This correlation is very weak. As a result, when I trained the linear regression model, the models, they are not able to accurately capture the relationship between the dependent variable and the independent variables, leading to high errors This correlation can be seen in the plot discussed in the previous section (4. Technology Details. Regression models).

6. Conclusions

The objective of my project was to predict online news popularity. As in recent years, this is the most used way to be informed, companies consider that advertising their services or product in this type of article is a good opportunity to reach out to more people. As it has been explained, in this project, I did deep data preprocessing and data analysis as well as some data modeling by applying regression and classification problems. The results of the models I trained were not as satisfactory as I expected when I decided to do this project, but I knew that it was a challenging project, as I have also read in the papers related to this topic.

The models I trained using linear regression (simple linear regression and multiple linear regression) have been a disaster. But the bad results I got have an explanation. As I have discussed before, linear regression is not a good way to predict the number of shares because the independent variables are not correlated with the dependent variables.

When training classification models the results are much better. The worst model of these ones is logistic regression, and the best ones are decision trees and random forests. Moreover, the results of these 2 models have been improved by applying a grid search and selecting the best hyperparameters to train them.

If I had to choose one of the models to be the final one, I would select Random Forest. Concretely, I would select Random Forest with maxdepth=10 combined with Entropy. Because, with this model, I get the best results of my own project. (Accuracy 61%).

Even though 61% of accuracy is the highest accuracy I got, this is not a really good value for predicting online news popularity. The accuracies that I have obtained are not very high, but I have been reading other papers related to my project. And other people that have also used this database have obtained not very high accuracies, which occurs because data are insufficient to predict the number of shares and popularity.

If I were working in a company and I had to predict this target in order to use the results in a real-world case, I would collect different information from the articles. Maybe information related to the reputation of the author. Because as we have seen the variables of this dataset are not the best ones for predicting online news popularity.

Another solution should be training more models, models different from the ones I trained. Until better results are obtained.

To summarize, we could say that predicting online news popularity is more difficult than I expected and that it is important to check the properties of a dataset before deciding the models that will be trained.

Although, in future work, I would like to continue working on this task, training different models, and studying the properties of the dataset until I obtain better results.

References

[1] G. Szabo and B. A. Huberman, *Predicting the popularity of online content, Communications of the ACM, vol. 53,* no. 8, pp. 80-88, 2010.

[2] R. Bandari, S. Asur, and B. A. Huberman, *The pulse of news in social media: Forecasting popularity*, in Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM), 2012, pp. 26-33.

[3] S. Kumar, F. Morstatter, and H. Liu, *Twitter Data Analytics*. Springer, 2016.

[4] I. Anagnostopoulos, N. Tsapatsoulis, and S. Papadopoulos, *Online news popularity prediction using social media features, Information Processing & Management*, vol. 58, no. 1, pp. 102352, 2021.