Online News Popularity Prediction

Vissamsetty Abhiram
av21u@fsu.edu
Florida State University
Tallahassee, Florida, USA

Sai Kalyan Tarun Tiruchirapally
st22q@fsu.edu
Florida State University
Tallahassee, Florida, USA

Archit Khandelwal
ak22bm@fsu.edu
Florida State University
Tallahassee, Florida, USA

Shanmukh Visnuvardhan Rao Gongalla
sg22bx@fsu.edu
Florida State University
Tallahassee, Florida, USA

ABSTRACT

Online news has become an essential part of our daily lives with the rise of digital media. The UCI Online News Popularity Data Set is a valuable resource that helps to understand the factors that contribute to the popularity of online news articles. This dataset contains information about articles published on Mashable, a popular online news website, between 2013 and 2015. The dataset is quite extensive and contains 61 attributes with a total of 39644 instances, and has been preprocessed to remove irrelevant features, handle missing values, and normalize the remaining attributes. This paper analyzes the various features that contribute to the popularity of an online news article using a combination of exploratory data analysis and machine learning algorithms. Our analysis shows that the number of words in an article, sentiment, and the day of the week when it was published are the most significant factors that contribute to an article's popularity. Additionally, our study indicates that publishing articles on certain days of the week may result in higher social media shares, and including more multimedia elements in an article may not have a significant impact on its popularity. The findings of our study have practical implications for online news publishers and social media marketers as they can use the proposed approach to optimize the timing, content, and promotion of their news articles for maximum impact.

1 RESEARCH BACKGROUND AND MOTIVATION

Online news has developed into a major source of information for millions of people around the world in the age of digital media and social networking. Understanding the factors that contribute to the popularity of online news articles has become a crucial challenge for news publishers and content creators. With the growth of online news websites, the process of consuming news has undergone a significant transformation. Investigating the connection between different aspects of online news articles and their popularity on social media platforms is made possible by the UCI Online News Popularity Data Set. This dataset includes attributes pertaining to the content, context, style, and social media shares of a large number of articles from the well-known online news source Mashable.

Insights into the variables influencing the popularity of online news articles can be gained from the analysis of this dataset using machine learning algorithms, which will aid publishers and content producers in making their work as effective as possible. Additionally, the creation of reliable prediction models can help in identifying the articles that are most likely to gain popularity and allow content producers to market their articles accordingly. As a result, the goal of this project is to identify the critical elements that influence how well-liked online news articles perform and to create machine learning models that can precisely predict these shares on social media.

2 INTRODUCTION

The rise of the internet has brought about significant changes in the way people consume news. With millions of users using news websites and social media platforms to remain up to date on events, online news has become a popular source of information. Online news publishers and content producers have emerged as a result of the popularity of online news, and they are now in a fiercely competitive market for readers' attention. In this context, understanding what factors contribute to the popularity of an online news article is critical.

2.1 Research Question

Purpose of project is to solve a binary classification problem, which is to predict if an online news article will become popular or not prior to publication. The popularity is characterized by the number of shares. If the number of shares is higher than a predefined threshold, the article is labeled as popular, otherwise it is labeled as unpopular. Thus, the problem is to utilize a list of articles features and find the best machine learning model to accurately classify the target label (popular/unpopular) of the articles to be published.

2.2 Problems with existing system

The existing system is only a binary classification system which only predicts whether the article is popular or not. It doesn't concentrate on analyzing the number of shares an article gets when published. It also doesn't provide us with the facility of ranking the articles which could be very useful in the statistical analysis.

2.3 Proposed System:

The System is built in order to classify popular news from unpopular news .Additional features like predicting the shares based on analyzing various attributes of the dataset and also enabling the system to rank the articles based on their degree of popularity.

3 LITERATURE SURVEY

The paper "Online News Popularity Prediction: A Machine Learning Approach" by M. Fernandes et al. proposed a machine learning approach to predict the popularity of online news articles. The authors used a dataset of news articles from Mashable, a popular news website, and extracted various features such as the article's title, content, and publisher. The authors then trained several machine learning models on this dataset, including Linear Regression, Ridge Regression, Lasso Regression, Elastic Net Regression, and Random Forest.

Another paper, "Online News Popularity Prediction Using Multiple Linear Regression Method" by S. Suresh Kumar et al., also focused on predicting the popularity of online news articles. The authors used a dataset of news articles from the Huffington Post and extracted features such as the article's title, content, and publication time. The authors then used multiple linear regression to predict the number of shares an article would receive on social media.

4 SYSTEM DESIGN

Systems Design is the process of defining the architecture, modules, Interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of systems theory to product development. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system. There is some overlap with the disciplines of systems analysis, systems architecture and system engineering.

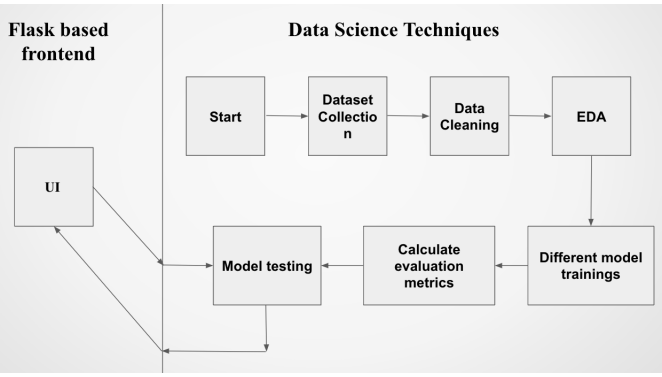

Figure 1: Architecture

The significant modules in our system are as follows:

- Training module: In this module, we preprocess, train the data and save the model states.
- Testing module: In this module, we developed a VoteClassifier which is an ensemble technique to utilize the performance of all the models trained to predict the outputs.
- Sentiment Analysis module: In this module, we use natural language processing techniques to process the user input and predict the outcomes using the training and testing modules.
- User interface: In this module, we have an interface where user can input his articles information and this layer had adapter and filters to transform the data into a relevant format to be processed by the sentiment analysis module.

4.1 Use Case Diagrams

There are two perspectives of use case diagrams in the system

- Training Level Use Case Diagram
- User Level Use Case Diagram.

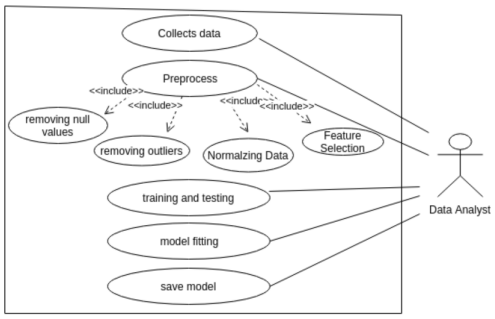

Figure 2: Training Level Use Case Diagram

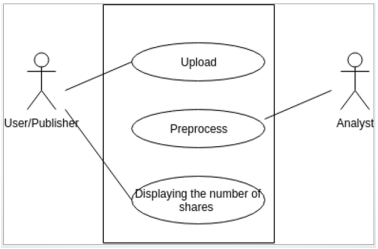

Figure 3: User Level Use Case Diagram

4.2 Sequence Diagram

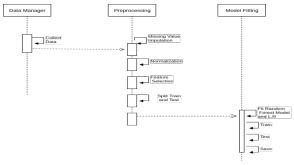

Figure 4: Training Level sequence Diagram

5 STATE OF THE ART METHODS/RELATED WORK

Our methodology involves data preprocessing, exploratory data analysis, feature selection, and the use of different machine learning models, including Decision Tree Classifier, Random Forest Classifier, Artificial Neural Networks, Linear Regression, Decision Tree Regressor, and Random Forest Regressor, and Finally, model interpretation to gain insights into the factors that contribute to the popularity of online news articles.

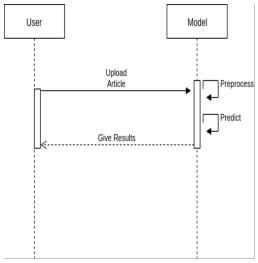

Figure 5: User Level Sequence Diagram

Data preprocessing is a crucial step in any machine learning project. It involves preparing the raw data for analysis by cleaning, transforming, and normalizing it. In our study, we first obtained a dataset of news articles from a popular news website. We then cleaned the data by removing any irrelevant information, such as metadata or duplicate articles. We also transformed the data into a structured format, making it easier to analyze and visualize.

After preprocessing the data, we performed exploratory data analysis (EDA) to gain a deeper understanding of the underlying trends and patterns in the data. EDA involves using statistical and visual techniques to summarize and explore the data. In our study, we used various EDA techniques, such as histograms, box plots, and scatter plots, to identify any outliers or anomalies in the data and to visualize the relationships between different variables.

Once we had a good understanding of the data, we performed feature selection to identify the most important variables for predicting the popularity of online news articles. Feature selection involves selecting a subset of the available variables that have the most significant impact on the target variable. In our study, we used various techniques such as correlation analysis, recursive feature elimination, and principal component analysis to select the most important features for our models.

Next, we trained several machine learning models to predict the popularity of online news articles. We used a range of different models, including Decision Tree Classifier, Random Forest Classifier, Artificial Neural Network, Linear Regression, Decision Tree Regressor, and Random Forest Regressor. Each model had its strengths and weaknesses, and we used a cross-validation approach to evaluate and compare the performance of each model.

Finally, we used model interpretation techniques to gain insights into the factors that contribute to the popularity of online news articles. Model interpretation involves analyzing the results of the models and identifying the most significant factors that contribute to the target variable. In our study, we used various interpretation techniques, such as feature importance analysis, partial dependence plots, and Shapley value analysis, to understand the relationship between the features and the target variable.

By combining these techniques, we can develop accurate and interpretable models that can help us understand the underlying factors that contribute to the popularity of online news articles.

5.1 Classification algorithms

Classification Algorithms		
Algo	Type	Suitable for
SVM	Supervised	Categorical & numerical vals
Random Forest	Ensemble	Data with noise/outliers
Logistic Regression	Supervised	Data with few features

5.1.1 Support Vector Machines (SVM). In this project, SVM can be used to create a classification model that predicts whether a given news article will be popular or not. The features such as number of words in the title, number of images, and other relevant attributes can be used as inputs to the SVM model. The target variable will be the popularity of the news article, which can be binary (popular or not popular) or multi-class (based on popularity levels). The SVM model can be trained using a labeled dataset where the popularity of the news articles is already known. The dataset can be split into training and testing sets, and the SVM model can be trained on the training set using appropriate hyperparameters and kernel functions. The model can then be evaluated on the testing set to assess its performance in terms of accuracy, precision, recall, F1-score, and other classification metrics. Some of the advantages of

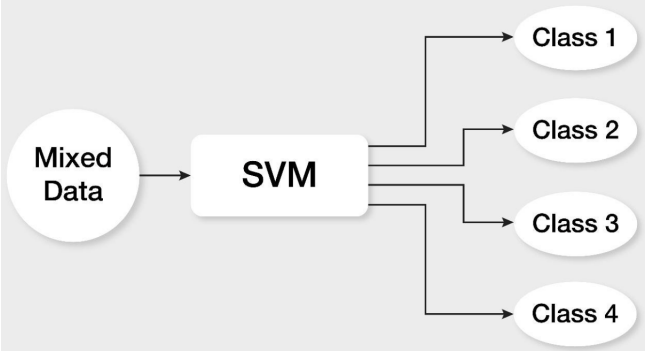

Figure 6: SVM

this algorithm are:

- High accuracy: SVM is known for its ability to achieve high accuracy in classification tasks. It can effectively handle complex decision boundaries and can learn from large datasets, making it suitable for online news popularity datasets that may contain diverse and dynamic features.
- Robustness to noise: Online news popularity datasets may contain noisy or irrelevant features that can negatively impact the accuracy of a classification model. SVM is inherently robust to noise and can handle noisy data well, making it suitable for handling noisy features in online news datasets.
- Scalability: SVM is efficient and scalable, making it suitable for handling large datasets that may be encountered in online news popularity datasets. It can handle a large number of samples and features, making it suitable for real-time prediction scenarios where data is constantly updated.
- Flexibility in kernel selection: SVM allows for the use of different types of kernels, such as linear, polynomial, radial basis function (RBF), and sigmoid, which can be used to

capture different types of patterns in the data. This flexibility in kernel selection allows for better modeling of complex relationships between features in online news popularity datasets.
- Interpretability: SVM provides interpretable results, as it generates a clear margin that separates the classes, making it easy to understand and interpret the decision boundary. This can be useful for understanding the factors that contribute to the popularity of online news articles, which can provide insights for content creators and marketers.
- Few hyperparameters: SVM has relatively few hyperparameters compared to other complex machine learning algorithms, such as deep neural networks. This makes it easier to tune and optimize SVM models for online news popularity datasets, resulting in faster training and inference times.

5.1.2 Random Forest. Similarly, Random Forest can be implemented in this project to create an ensemble model that predicts the popularity of online news articles. The features of the news articles can be used as inputs to the Random Forest model, and the target variable can be the popularity level of the articles. The Random Forest model can be trained using a labeled dataset, and the hyper parameters such as the number of trees, maximum depth of trees, and minimum number of samples required to split a node can be tuned to optimize the model's performance. The model can then be evaluated on a testing set to measure its accuracy, precision, recall, F1-score, and other classification metrics. Some of the advantages

Figure 7: Random Forest

of this algorithm are:

- High accuracy: Random Forest is an ensemble learning method that combines multiple decision tree classifiers to create a more accurate and robust model. It can handle complex relationships in the data and provide highly accurate predictions, making it suitable for online news popularity prediction tasks where accuracy is crucial.
- Robustness to noise and overfitting: Random Forest is inherently robust to noise and overfitting, which are common challenges in online news datasets. The ensemble of decision trees in Random Forest helps to mitigate the overfitting issue, resulting in a more generalized model that performs well on unseen data.
- Feature importance: Random Forest provides feature importance measures, which can help identify the most relevant features that contribute to the prediction of online news popularity. This can provide insights into the factors that influence news popularity and aid in feature selection or feature engineering, improving the interoperability and effectiveness of the model.
- Handling missing values: Random Forest can effectively handle missing values in the dataset without the need for imputation. It can make accurate predictions even when some features have missing values, which is common in real-world datasets.
- Non-linearity handling: Random Forest can capture non-linear relationships between features, which can be important in online news popularity prediction tasks where the relationships between features may not be linear. This allows for modeling of more complex patterns in the data, resulting in improved prediction accuracy.
- Scalability: Random Forest can handle large datasets with multiple features and samples, making it suitable for online news popularity datasets that may contain a large number of articles and diverse features.
- Reduced risk of overfitting: Random Forest can reduce the risk of overfitting compared to single decision tree classifiers, as it combines multiple trees and uses bootstrapped samples for training. This helps to create a more robust and generalized model, reducing the risk of overfitting to the training data.
- Ease of hyperparameter tuning: Random Forest has relatively fewer hyperparameters compared to some other complex machine learning algorithms, making it easier to tune and optimize the model for the online news popularity dataset. This can save time and effort in the hyperparameter tuning process.

5.1.3 Logistic Regression. Logistic Regression is a simple yet effective classification algorithm that can also be implemented in this project. The features of the news articles can be used as inputs to the Logistic Regression model, and the target variable can be the binary or multi-class popularity level of the articles. The Logistic Regression model can be trained using a labeled dataset, and appropriate hyper parameters such as the learning rate, regularization term, and number of iterations can be tuned to optimize the model's performance. The model can then be evaluated on a testing set to assess its accuracy, precision, recall, F1-score, and other classification metrics. Some of the advantages of this algorithm are:

- Simplicity and interpretability: Logistic Regression is a linear model that is easy to implement and interpret. It provides interpretable results in terms of coefficients and odds ratios, which can help in understanding the impact of different features on the prediction of online news popularity. This can provide actionable insights to content creators and marketers.
- Efficiency: Logistic Regression is computationally efficient and has low memory usage, making it suitable for large datasets with limited computational resources. It can be trained quickly and is suitable for real-time prediction scenarios, where news popularity needs to be predicted in real-time as new articles are published.
- Scalability: Logistic Regression can handle large datasets with a large number of features, making it suitable for online

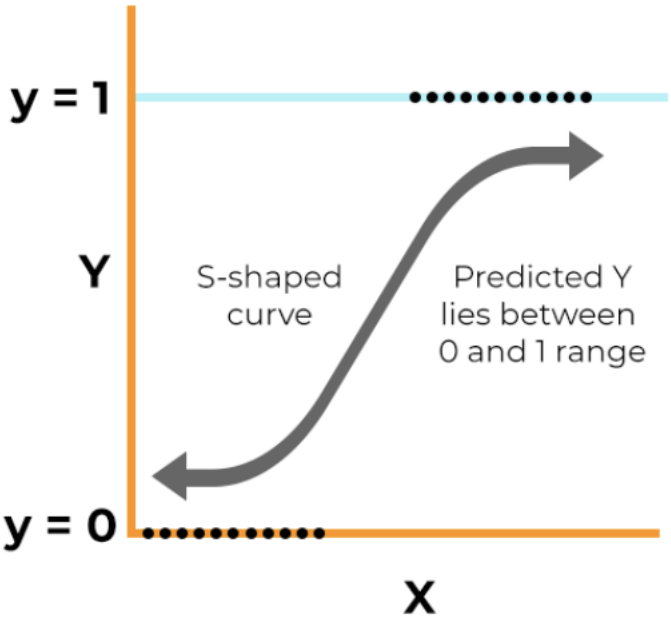

Figure 8: Logistic Regression

news popularity datasets that may contain diverse features and a large number of articles.

- Feature selection: Logistic Regression can be used for feature selection, allowing for the identification of the most relevant features that contribute to the prediction of online news popularity. This can help in identifying the key factors that impact news popularity and aid in feature engineering, resulting in a more effective and interpretable model.
- Probabilistic interpretation: Logistic Regression provides probabilistic predictions, where the predicted probabilities can be interpreted as the likelihood of an article being popular or not. This can provide insights into the uncertainty of predictions and aid in decision-making based on the probability threshold chosen.
- Handling of categorical features: Logistic Regression can handle categorical features natively by using techniques such as one-hot encoding, making it suitable for online news popularity datasets that may contain categorical features such as article category or author.
- Model regularization: Logistic Regression allows for regularization techniques such as L1 (Lasso) or L2 (Ridge) regularization, which can help in reducing the risk of over-fitting and improving the generalization performance of the model.
- Interpretable probability thresholds: Logistic Regression allows for choosing the probability threshold for classification, which can be interpreted based on the specific requirements of the online news popularity prediction task. This flexibility in threshold selection can be useful in adjusting the model's sensitivity and specificity based on the desired trade-offs.

5.1.4 Decision tree classifier. A decision tree classifier is a popular supervised machine learning algorithm used for both classification and regression tasks. It works by recursively splitting the dataset into subsets based on the values of input features, and then making decisions based on the majority class or predicted values in each subset. The tree-like structure of the classifier consists of nodes representing decision points, edges representing feature values, and leaf nodes representing the predicted class or value. Some of the

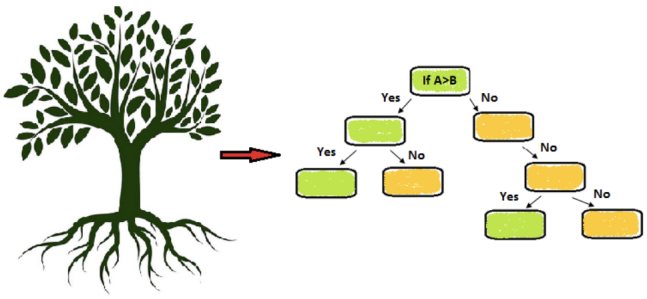

Figure 9: Decision Tree Classifier

advantages of this algorithm are:

- Interpretability: Decision tree classifiers are highly interpretable, as they produce decision rules in the form of tree-like structures that are easy to understand and explain. This can provide insights into how the model is making predictions and help in gaining actionable insights from the model's decision-making process.
- Fast training and prediction: Decision tree classifiers have relatively fast training and prediction times compared to many other complex machine learning algorithms. This makes them suitable for online news popularity prediction tasks where real-time or near-real-time predictions may be required.
- Handling non-linearity: Decision tree classifiers are capable of capturing non-linear relationships between features and target variables. Online news popularity may depend on various factors that exhibit non-linear relationships, such as article content, sentiment, time of publication, etc. Decision tree classifiers can effectively capture these non-linearities without requiring complex feature engineering.
- Handling missing values: Decision tree classifiers can handle missing values in the dataset effectively by making decisions based on the available features. This can be beneficial when dealing with online news datasets that may contain missing or incomplete information.
- Robustness to outliers: Decision tree classifiers are generally robust to outliers, as they make decisions based on splits in feature space that are not influenced by outliers. This can be useful in online news datasets where outlier data points may exist but are not indicative of the overall pattern.
- Feature selection: Decision tree classifiers can automatically select the most informative features for making decisions. This can be helpful in identifying the most important features that influence online news popularity, which can aid in feature selection and dimensionality reduction.

- Ensemble methods: Decision tree classifiers can be combined with ensemble methods such as bagging or boosting to further improve their performance. Bagging can reduce the risk of overfitting, while boosting can increase the model's accuracy by combining multiple decision trees.
- Easy visualization: Decision tree classifiers can be visualized, which can help in understanding the model's decision-making process and communicating the results to stakeholders in a visual and intuitive manner.
- Applicability to small datasets: Decision tree classifiers can work well with small datasets, making them suitable for online news datasets that may have limited data instances.
- No requirement for feature scaling: Decision tree classifiers do not require feature scaling, such as normalization or standardization, as they make decisions based on the values of individual features. This can save preprocessing time and effort compared to algorithms that require feature scaling.

5.2 Regression algorithms

Regression Algorithms	
Algo	Limitations
Decision Tree Classifier	Prone to overfitting
Random Forest	Slower training and prediction times
Decision Tree Regressor	Sensitivity to small changes in data

5.2.1 Decision tree regression algorithm. The decision tree regressor is a supervised machine learning algorithm used for regression tasks. It works by recursively splitting the dataset into subsets based on the values of input features, and then making predictions of continuous values in each subset. Some of the advantages of this

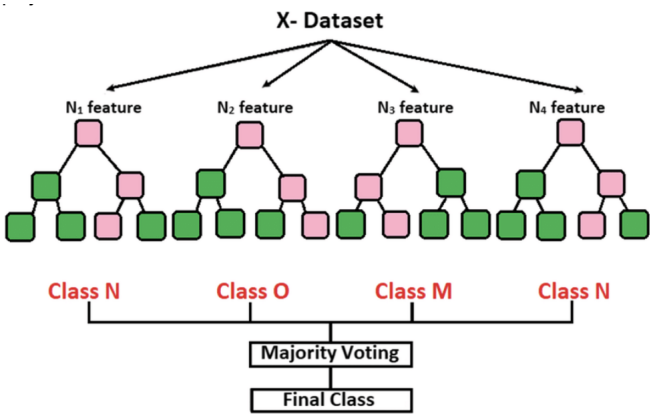

Figure 10: Decision Tree Regression

algorithm are:

- Interpretable model: Decision trees are inherently interpretable, making them easy to understand and explain. The resulting tree structure can be visualized, providing insights into how the algorithm is making decisions based on the features of the online news dataset. This interpretability can be valuable in gaining insights into the factors that influence news popularity.
- Non-parametric: Decision trees do not assume any particular distribution or relationship between variables in the dataset, making them versatile for handling nonlinear patterns and complex interactions among features. This can be advantageous in cases where the relationship between features and news popularity may not be linear or follow a particular distribution.
- Handling missing values: Decision trees can handle missing values in the dataset effectively. They can make decisions based on the available data, without requiring imputation or removal of data points with missing values. This can be beneficial when dealing with real-world datasets that often contain missing or incomplete information.
- Scalability: Decision tree algorithms can handle large datasets with high dimensionality, making them suitable for online news datasets that may contain a large number of articles with numerous features such as word counts, sentiment scores, and time-based variables.
- Feature selection: Decision trees can automatically select relevant features for predicting news popularity, as they split the tree based on feature importance. This can help identify the most significant features that drive news popularity, allowing for feature selection and dimensionality reduction in the dataset.
- Robustness to outliers: Decision trees are less sensitive to outliers compared to some other regression algorithms. Outliers in the online news dataset may not significantly impact the overall tree structure, allowing the algorithm to capture patterns in the majority of the data without being skewed by extreme values.
- Ensemble methods: Decision trees can be easily combined with ensemble methods such as bagging or boosting to improve their predictive performance. Ensemble methods can enhance the accuracy and robustness of the decision tree regressor by combining multiple trees, resulting in a more accurate and robust model.
- Fast training and prediction: Decision tree algorithms typically have fast training and prediction times compared to more complex algorithms, making them suitable for real-time or near-real-time applications, such as online news popularity prediction.

5.2.2 Random forest Regressor. Random Forest Regressor is an ensemble learning algorithm that combines multiple decision trees for regression tasks. It works similarly to the Random Forest algorithm for classification, but instead of predicting discrete classes, it predicts continuous values as output. Some of the advantages of this algorithm are:

- Improved prediction accuracy: Random Forest Regressor can often achieve higher prediction accuracy compared to individual decision trees in regression tasks. By combining multiple trees and taking an average of their predictions, it can reduce the impact of noise and outliers, leading to more accurate predictions of news popularity.
- Robustness to noise and outliers: Random Forest Regressor is less sensitive to noisy or outlier data compared to individual decision trees. The ensemble nature of Random Forest helps

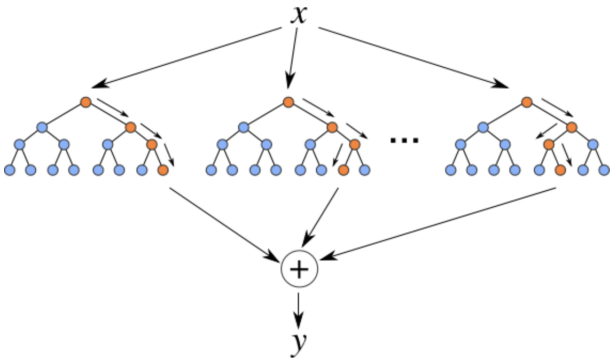

Figure 11: Random Forest Regression

to mitigate the impact of noisy or outlier data points, making it more robust and stable in the presence of such data.

- Handling nonlinear relationships: Random Forest Regressor can capture nonlinear relationships between features and the target variable. Online news popularity may depend on various factors, such as article content, sentiment, time of publication, etc., which may exhibit nonlinear relationships. Random Forest Regressor can capture these complex relationships and make accurate predictions.
- Handling high-dimensional data: Similar to Random Forest for classification, Random Forest Regressor can effectively handle datasets with a large number of features, making it suitable for online news datasets that may contain numerous features. It can select relevant features for each tree, leading to improved feature selection and model performance.
- Handling missing values: Random Forest Regressor can handle missing values in the dataset effectively by using only the available features in each tree during training and prediction. This can be beneficial when dealing with online news datasets that may contain missing or incomplete information.
- Overfitting prevention: Random Forest Regressor can prevent overfitting, which is a common issue with individual decision trees. By averaging the predictions of multiple trees and using random subsets of features and data, Random Forest Regressor can reduce the risk of overfitting, leading to a more generalized and robust model.
- Scalability: Random Forest Regressor can handle large datasets with high dimensionality, making it suitable for online news datasets that may contain a large number of articles with numerous features. It can parallelize the training process, making it computationally efficient for large-scale datasets.
- Interpretability: Although Random Forest Regressor is not as interpretable as individual decision trees, it can still provide insights into feature importance by calculating the average feature importance across all the trees. This can help understand which features are most influential in predicting news popularity.
- Flexibility: Random Forest Regressor is a flexible algorithm that can be used for various regression tasks beyond online news popularity prediction. It can be easily extended to handle other types of regression problems, making it adaptable to different types of datasets.

5.3 KNN

KNN (k-nearest neighbors) is a machine learning algorithm used for classification and regression tasks. The algorithm works by finding the k number of data points in the training set that are closest to a given data point in the test set, and then using the labels of these nearest neighbors to predict the label of the test point.

In a classification problem, the algorithm predicts the class label of a new data point based on the class labels of the k nearest neighbors. In a regression problem, the algorithm predicts the numerical value of a new data point based on the numerical values of the k nearest neighbors.

To use KNN in a data science project, you would typically follow these steps:

- Prepare your data: Ensure your data is in a format suitable for KNN and preprocess the data as necessary. This may include handling missing values, scaling the features, and encoding categorical variables.
- Split your data: Divide your data into training and testing sets. The training set will be used to train the KNN algorithm, while the testing set will be used to evaluate its performance.
- Train the KNN model: Use the training set to train the KNN model by computing the distance between each training data point and all other training data points. This creates a database of distances that will be used to identify the k nearest neighbors for each new data point.
- Test the KNN model: Use the testing set to evaluate the performance of the KNN model by comparing its predicted labels to the true labels of the test data points.
- Tune the hyperparameters: Experiment with different values of the k parameter to find the value that yields the best performance for your specific dataset.
- Evaluate the results: Evaluate the performance of the KNN model based on its accuracy, precision, recall, and F1 score.

KNN is a simple and effective algorithm that can be used in a wide range of data science projects. However, it can be computationally expensive for large datasets and may not perform well in high-dimensional spaces.

5.4 Naive Bayes

Naive Bayes is a machine learning algorithm used for classification tasks. It is based on the Bayes theorem, which states that the probability of a hypothesis (in this case, the class label of a new data point) is proportional to the product of the prior probability of the hypothesis and the likelihood of the observed data given that hypothesis.

The "naive" in Naive Bayes refers to the assumption that the features are conditionally independent given the class label. This means that the algorithm assumes that the presence or absence of a particular feature does not depend on the presence or absence of any other feature.

5.5 NLP

We made use of the following NLP algorithms.

- Text Classification: NLP algorithms can be used to categorize online news articles into different topics or genres, such as politics, sports, entertainment, etc. This can help in organizing and categorizing the news content, and also in providing relevant recommendations to users based on their interests.
- Sentiment Analysis: NLP algorithms can determine the sentiment or emotional tone of online news articles, whether they are positive, negative, or neutral. This can provide insights into the overall sentiment of the news articles and help in identifying popular news articles based on positive sentiment or identifying controversial or negative news articles that might affect their popularity.
- Keyword Extraction: NLP algorithms can automatically extract important keywords or phrases from online news articles. This can help in identifying the key topics or themes discussed in the news articles, and also in generating metadata for indexing and searching purposes.
- Summarization: NLP algorithms can generate summaries of online news articles, providing concise and informative summaries of the main content. This can help in providing users with quick overviews of news articles and facilitating content consumption, especially in cases where users may not have time to read the entire article.
- Text Clustering: NLP algorithms can group similar news articles together based on their content, helping in identifying clusters of related news articles. This can be useful in identifying popular topics or trends in news content and providing users with relevant news articles based on their interests.
- Trend Analysis: NLP algorithms can analyze the frequency and patterns of keywords, topics, or entities mentioned in online news articles over time. This can help in identifying trends, patterns, and insights related to popular news topics, emerging topics, or shifting interests of users.

5.6 User Interface

We developed a user interface using flask to input user articles

- Flask: Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.
- Functionality: The input to the UI is powered by NLP techniques. user can enter text in the input box and select the sentiment analysis model to be used for analyzing the text. The user can also select the language of the text and choose to perform the analysis on individual sentences or the entire text. Once the user clicks the "Analyze" button, the sentiment analysis results are displayed in a color-coded format (green for positive, red for negative, and yellow for neutral) along with a percentage score for each sentiment. The interface also provides a visualization of the sentiment distribution using a pie chart. Overall, the UI is simple and intuitive, allowing users to easily analyze the sentiment of their text.
- Trends: The UI also provides an option to explore trends and other projections as well.

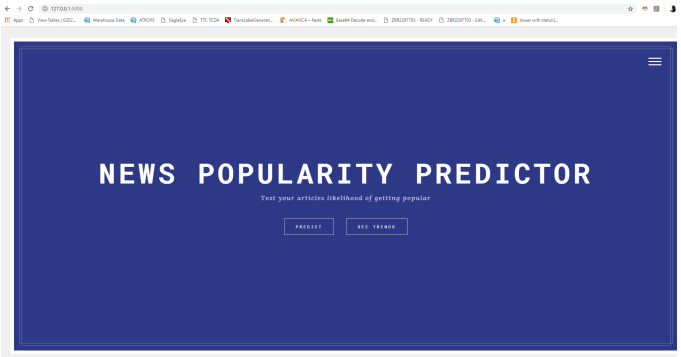

Figure 12: Home page

Figure 13: Results

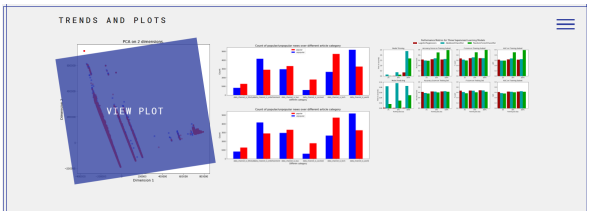

Figure 14: Trends

6 DATASET

The UCI Online News Popularity Dataset is a publicly available dataset that contains data on news articles published by Mashable, a popular online news website. The dataset is available on Kaggle and consists of 39,644 news articles published between January 7th, 2013 and January 7th, 2015.

Each news article in the dataset is described by 61 attributes, including features related to the article's content, its publication time, and its social media engagement. The dataset also includes the number of shares the article received on popular social media platforms like Facebook, Twitter, and LinkedIn.

Here are the different types of attributes available in the dataset:

- Article information: The dataset includes attributes such as the article title, text, author, and URL.
- Time-related features: The dataset includes attributes that describe the publication time of the article, including the day of the week, month, and year.
- Content-based features: The dataset includes attributes related to the content of the article, such as the number of words, number of images and videos, and the category of the article.
- Social media engagement: The dataset includes the number of shares each article received on popular social media platforms like Facebook, Twitter, and LinkedIn.

The target variable for the dataset is the number of shares that an article received on social media. The UCI Online News Popularity Dataset has been used extensively by researchers to study the factors that influence the popularity of news articles on social media and to develop machine learning models for predicting the popularity of news articles.

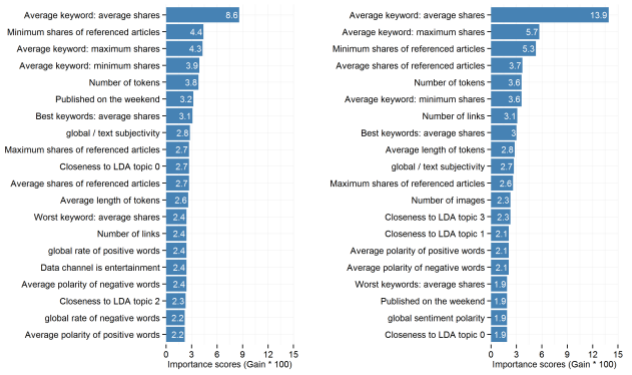

Figure 15: Dataset Description

7 DATA ANALYSIS

Data analysis is one of the most crucial project steps in the field of data science. To gain insights and get the data ready for modeling, this entails cleaning, exploring, and visualizing the data. Data analysis is a critical step in a project to predict the popularity of news articles because it can identify important variables that affect this popularity and provide information for the creation of precise and trustworthy predictive models. Data analysis is an essential step in any data science project, but it is especially crucial in the context of predicting news popularity. It is possible to determine important factors that affect the popularity of news articles and to develop precise and trustworthy predictive models by carefully cleaning, exploring, and visualizing the data. Despite the numerous difficulties and factors to take into account, careful planning and execution can help to guarantee the project's success and the reliability of the results. In the end, the knowledge gleaned from data analysis can help to inform and enhance the practice of journalism and to better comprehend the variables affecting public discourse and opinion.

7.1 Data Preprocessing

To make sure that the dataset is accurate and reliable for analysis, data cleaning is the process of locating and correcting errors or inconsistencies. In the case of predicting news popularity, this may entail finding and fixing missing data, eliminating redundant entries, and identifying outliers. In predictive modeling, missing data can be a problem because it can skew the results and make the models less accurate. Additionally, duplicate entries can be problematic because they can overestimate the significance of some data points and result in overfitting. The removal of outliers can help to ensure that the data is more indicative of the underlying trends because they can be a sign of data quality problems or inaccurate data.

Data cleaning involves locating and correcting errors or inconsistencies in the data. In the context of predicting news popularity, this process may entail finding and fixing missing data, eliminating redundant entries, and identifying outliers.

Missing data can be a significant problem in predictive modeling because it can skew the results and make the models less accurate. It is essential to address missing data to ensure that the resulting models are as accurate as possible. Various methods can be used to deal with missing data, such as imputing missing values with mean or median values, using regression to predict missing values, or removing incomplete cases altogether. In our dataset we set a threshold of 70 percent i.e., if any feature has missing values greater than the threshold then we will remove the column from training set.

Feature selection is an important step, our dataset had about 58 features and therefore it is essential for us to eliminate some of the features. Based on our correlation analysis from figure 17, we eliminated the less important features.

Outliers can be a problem in the dataset and can skew the results of the predictive models. Based on our analysis of distribution of shares from figure 16, we observed that majority of the shares are less than 3500. Therefore, we set that aforementioned threshold and eliminated rest of the entries and considered them as outliers.

Data cleaning is a critical step in the predictive modeling process, as it ensures that the dataset used for analysis is accurate and reliable. By addressing missing data, duplicate entries, and outliers, we can develop more accurate models that better capture the underlying trends in the data.

7.2 Exploratory Data Analysis

In order to better understand the data's structure and patterns, exploratory data analysis (EDA) involves examining the data using statistical and visualization techniques. EDA can be used to pinpoint important aspects of news popularity prediction, such as the content, title, author, and publication date of news articles. Histograms are a common EDA technique that can be used to see the distribution of the target variable (news popularity) and spot any patterns or trends. Heat maps and scatter plots can both be used to find groups of related data points and to show correlations between variables. Here are some of the key techniques that can be used in EDA:

7.2.1 Histograms. To see how the data are distributed, histograms are used. They can be used to spot patterns and trends in the data because they display the frequency of data points within each

interval or bin. A histogram of the popularity of news articles, for instance, might reveal that while most articles have relatively low popularity scores, a select few have incredibly high popularity scores.

7.2.2 Scatter plots. To see the relationship between two variables, use scatter plots. The plot represents each data point as a point, with the x-axis denoting one variable and the y-axis denoting the other. Scatter plots can be used to find trends or correlations between different variables, such as a relationship between an article's popularity and length.

7.2.3 Heat maps. To see patterns in the data over time or across various variables, heat maps are used. With darker colors denoting higher values, they use color gradients to show the relative magnitude of the data. Heat maps can be used to spot recurring patterns in the data or groups of related data points.

7.2.4 Correlation matrices. To find correlations between variables, correlation matrices are used. They display the correlation coefficient, which ranges from -1 (perfectly negative correlation) to 1 (perfectly positive correlation), for each pair of variables. The target variable (news popularity) and any potential multicollinearity problems can both be found by using correlation matrices to determine which variables are most strongly correlated with one another.

7.2.5 Bar clouds. Bar clouds, often referred to as word clouds or tag clouds, are a well-liked visualization method that can be applied to online news forecasting. Larger font sizes are used to visually reflect the frequency or relevance of words or concepts in a text corpus (such as article titles or content), and bar clouds are used to show this information.

It is helpful to analyze the content and themes that are driving popularity by using bar clouds to find the most popular subjects or keywords in a collection of articles. The popularity of various subjects or keywords throughout a range of times or categories can also be compared using them.

When combined with other visualization techniques like scatterplots, heatmaps, and line plots, bar clouds can be a valuable visualization approach for online news prediction. However, due to their limitations, they should only be utilized with caution.

7.3 Visualization

A potent tool for understanding and presenting complex data is visualization. When predicting the popularity of news articles, visualization can be used to help pinpoint the major variables that affect this phenomenon and to present the analysis' findings to relevant parties. For this purpose, a variety of visualization methods, such as scatter plots, heat maps, and bar charts, can be used. Word clouds are a popular method that can be used to visualize the most frequently occurring words in news articles and spot linguistic and tone trends.

Scatterplots, which show the correlation between the target variable and one or more predictor factors (such article length, publication date, or sentiment score), are another helpful visualization approach. This aids in discovering any linear or nonlinear connections between the variables and helps to determine the most crucial predictors for the correctness of the model.

Because they display the correlation matrix between all predictor variables, heatmaps are also helpful for predicting internet news. This aids in spotting any multicollinearity—that is, a high degree of correlation between predictor variables—that can compromise the stability and interpretability of the model. To prevent overfitting, it could be required to drop one of the highly correlated variables from the model.

The temporal patterns in the target variable, such as the number of views over time, can be shown using line charts. This aids in spotting any seasonality or trend patterns that can affect the model's performance as well as deciding whether adding time-related predictors (like day of the week or month) would increase the model's accuracy.

Bar clouds, often referred to as word clouds or tag clouds, are a well-liked visualization method that can be applied to online news forecasting. Larger font sizes are used to visually reflect the frequency or relevance of words or concepts in a text corpus (such as article titles or content), and bar clouds are used to show this information.

It might be helpful to analyze the content and themes that are driving popularity by using bar clouds to find the most popular subjects or keywords in a collection of articles. The popularity of various subjects or keywords throughout a range of times or categories can also be compared using them.

We have enclosed the screenshots of the visualization in the next page.

7.4 Challenges and Considerations

Even though data analysis is a crucial step in any data science project, there are a number of difficulties and things to keep in mind. The volume and complexity of the data present a significant challenge because it can be challenging to spot patterns and trends. Another difficulty is the data's quality, which can be impacted by biases, errors, and inconsistencies. To ensure the accuracy and dependability of the data, it is crucial to carefully examine the sources and techniques used to gather the data. Analytical techniques should be carefully chosen based on the project's goals and objectives as another factor to take into account. For instance, decision tree models may be better suited for identifying nonlinear relationships than linear regression models for determining correlations between variables. In order to guarantee the models' accuracy and dependability, it is also crucial to carefully validate and test them.

8 RESULTS

After successful training on the models we evaluated the performance of the models using few standard metrics.

8.1 Metrics

As a classification task, we will adopt the following three evaluation metrics: accuracy, F1-score and AUC. For all three metrics, the higher value of the metric means the better performance of model.

- Accuracy: Accuracy is direct indication of the proportion of correct classification. It considers both true positives and true negatives with equal weight and it can be computed as

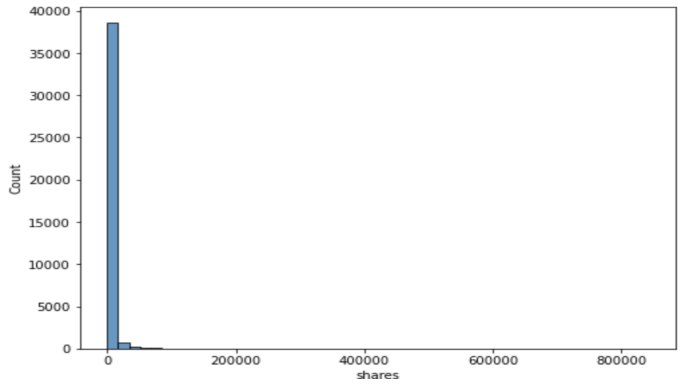

Figure 16: Distribution of shares

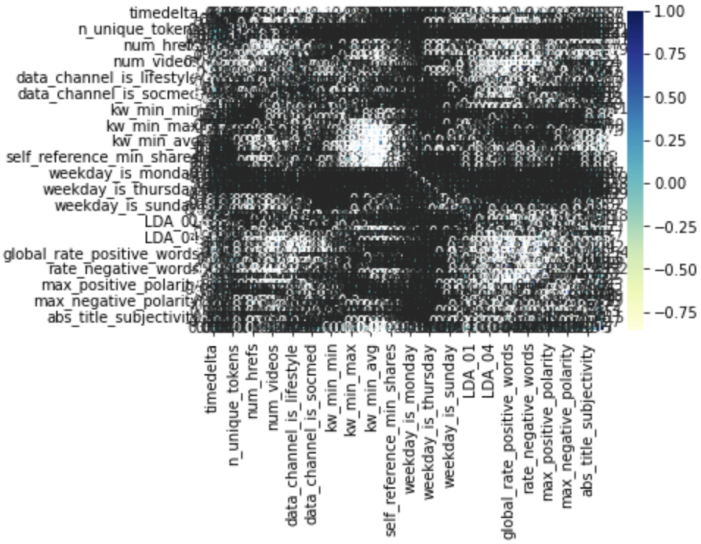

Figure 17: HeatMap of correlation between features and target variable

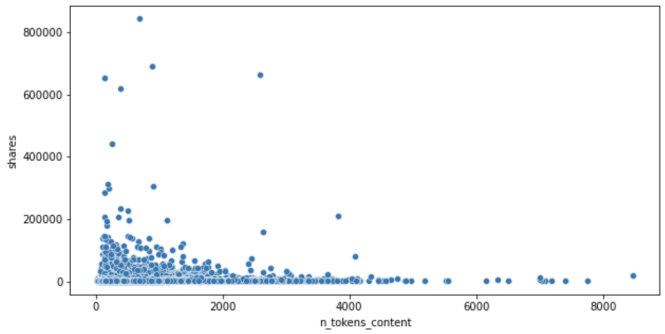

Figure 18: Impact of content tokens on shares

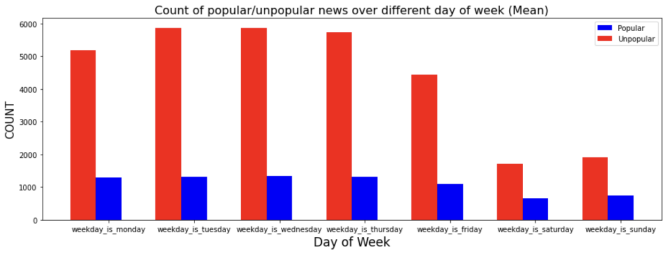

Figure 19: Impact of day of the week on shares

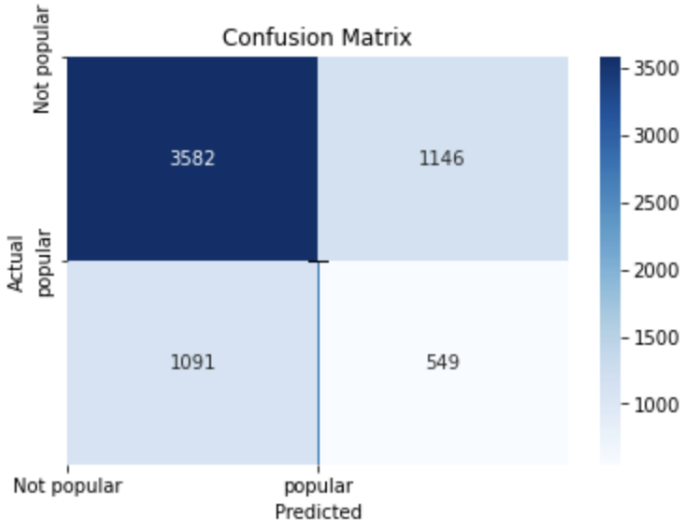

Figure 20: Decision tree classifier CM

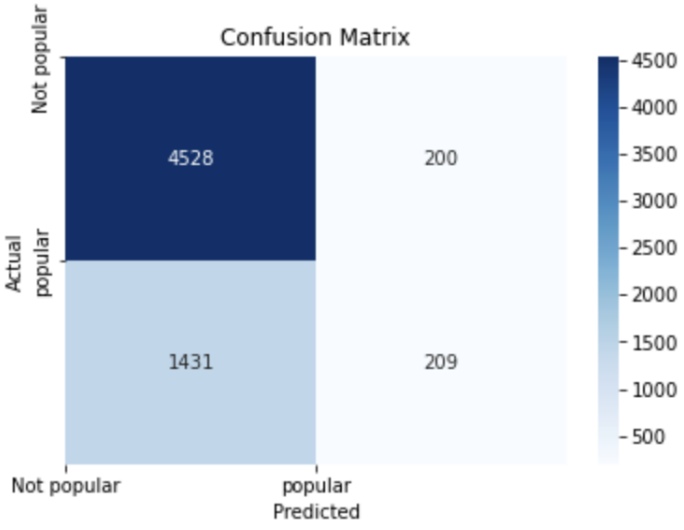

Figure 21: Random Forest classifier CM

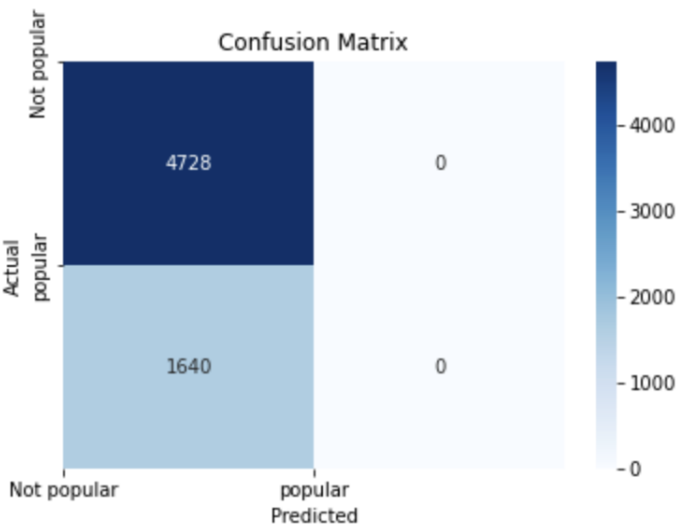

Figure 22: Artificial neural network CM

```
def perform_eda(data):
    # Basic idea of the data
    print("\nBasic idea of the data:\n",data.info())

    # Basic summary of the data
    print("\nBasic summary of the data:\n",data.describe())

    #Exploring different features in the data
    data.columns = data.columns.str.strip()
    print("\nDifferent features in the dataset:\n",data.columns)

    # Exploring our target column shares.
    share_data = data['shares']
    print("\n Exploring our target column shares:\n",data['shares'].describe())

    # Missing values in the data
    null_data = data.isnull()
    print('\nNull percentage in data:\n', null_data.sum())

    # Calculate correlation matrix
    corr_matrix = data.corr()

    # Print correlation coefficients of each feature with target variable
    print("\ncorrelation coefficients of each feature with target variable:\n",corr_matrix['shares'].sort_values(ascending=False))

    # Visualize correlation matrix with heatmap
    sns.heatmap(corr_matrix, cmap="YlGnBu", annot=True)
    plt.show()

    # Check the distribution of the target variable
    plt.figure(figsize=(8,6))
    sns.histplot(data['shares'], bins=50)

    # Plot the share v/s words
    data = data[data['n_tokens_content'] != 0]
    plt.figure(figsize=(10,5))
    ax = sns.scatterplot(y='shares', x='n_tokens_content', data=data)

    # Plot the popularity for every day of the week.
    a = data['shares'].mean()
    Wday = data.columns.values[31:38]
    unpopular=data[data['shares']<a]
    popular=data[data['shares']>=a]
    Unpop_day = unpopular[Wday].sum().values
    Pop_day = popular[Wday].sum().values
    fig = plt.figure(figsize = (13,5))
    plt.title("Count of popular/unpopular news over different day of week (Mean)", fontsize = 16)
    plt.bar(np.arange(len(Wday)),Pop_day,width=0.3,align='center',color='b',label='Popular')
    plt.bar(np.arange(len(Wday))-0.3,Unpop_day,width=0.3,align='center',color='r',label='Unpopular')
    plt.xticks(np.arange(len(Wday)), Wday)
    plt.ylabel('COUNT',fontsize=15)
    plt.xlabel('Day of Week',fontsize=17)
    plt.legend(loc = 'upper right')
    plt.tight_layout()
```

Figure 23: Implementation

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Dataset\ Size}$$

- F1-score is an unweighted measure for accuracy by taking harmonic mean of precision and recall, which can be computed as

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

- The AUC is the area under the ROC (Receiver Operating Characteristics) curve, which is a plot of the True Positive Rate versus the False Positive Rate. AUC value is a good measure of classifier's discrimination power and it is a more robust measure for model performance.

8.2 Model Evaluation and Validation

After initial implementation and further refinement for the three classifiers, we find that the best performance is obtained by the RF classifier with 500 trees in the forest. The best obtained metrics of RF are accuracy 0.6769, F1-score 0.7073 and AUC 0.6734. The final scores are not exceptional, which is sort of within the expectation, because the dataset not linear separable.But it still achieve a reasonable performance in news popularity prediction compared with a random guess.

Classifier	Accuracy	F1-score	AUC
Logistic Regression	0.6414	0.6772	0.6373
RF	0.6743	0.7052	0.6706
Adaboost	0.6494	0.6818	0.6458

Figure 24: Test the model with training/testing set ratio 0.15

8.3 Justification

The best performance is also given by RF model, which achieves 0.67 of accuracy score, 0.69 of F1 score and 0.73 of AUC score. The metrics of my RF model are accuracy 0.6769, F1-score 0.7073 and AUC 0.6734. By comparison, although the AUC score is not better than the benchmark model, the accuracy and F1-score are all better than the benchmark model. So we can say the obtained model achieves a comparable performance as benchmark model and it is significant enough to solve the popular news classification problem.

Model	Accuracy	Precision	Recall	F1	AUC
Random Forest (RF)	**0.67**	0.67	**0.71**	**0.69**	**0.73**
Adaptive Boosting (AdaBoost)	0.66	0.68	0.67	0.67	0.72
Support Vector Machine (SVM)	0.66	0.67	0.68	0.68	0.71
K-Nearest Neighbors (KNN)	0.62	0.66	0.55	0.60	0.67
Naïve Bayes (NB)	0.62	**0.68**	0.49	0.57	0.65

Figure 25: The metrics of benchmark model

9 CONCLUSION

Linear Regression, Decision Tree Regression, Random Forest Regression, Artificial Neural Network, and Decision Tree Regression were used in the study project to analyze the UCI data. The Online News Popularity Data Set has provided insightful information about the elements that influence how popular news pieces are on social media. We now have a greater knowledge of the numerous aspects of news items that are related to reader engagement, such as content, sentiment, timing, and multimedia components, thanks to these machine learning algorithms. Publishers and content creators can use this information to improve their content creation and publishing strategies, which will ultimately enhance audience engagement and website traffic.

Additionally, this study adds to the ongoing discussion regarding how social media affects how news is consumed and disseminated. We have uncovered tendencies and biases in the consumption and sharing of news content by examining patterns of user activity in reaction to news items. This data can be used to encourage more

diverse and fair news reporting and to thwart the spread of false information.

It is important to keep in mind, though, that the UCI Online News Popularity Data Set has a small sample size and could not be an accurate representation of all news sources or social networking sites. As a result, care should be taken when interpreting the research's conclusions. Additionally, even while machine learning algorithms offer a strong tool for studying massive datasets, they do have some drawbacks. The algorithms must be used correctly, and the outcomes must be evaluated to guarantee their accuracy and dependability. Overall, this study offers insightful information about the usage of machine learning algorithms for news media analysis and the function of social media in the consumption and dissemination of news. In order to investigate the ramifications of these findings in other fields and to keep creating more efficient methods for evaluating massive datasets, more research is required.

10 FUTURE WORK

To further improve the performance, we think there are three possible ways:

- Increase the size of dataset since RF has a strong learning capability and a rich dataset might improve its prediction performance.
- Try more advanced cross validation methods although it might increase the training time.
- Engineer and add more relevant features to the original dataset. For instance, we could use all the words in an article as additional features, and then try the classifier such as Naive Bayes to see if it can achieve a better performance.
- The exploration of more advanced features regarding content like trend analysis.
- The comparison of the model with many other state-of-the-art techniques

[5] [9] [2] [3] [6] [1] [12] [10] [7] [8] [11] [4]

REFERENCES

[1] Davide Albanese, Paolo Rosso, Alexandros Ntoulas, and Tasos G Stavropoulos. 2018. Learning to predict the popularity of online news. *ACM Transactions on the Web (TWEB)*, 12, 1, 1–24.

[2] David Blei, Andrew Ng, and Michael Jordan. 2001. Latent dirichlet allocation. In vol. 3. (Jan. 2001), 601–608.

[3] Mengting Chen, Aixin Sun, and Ee-Peng Lim. 2018. Predicting news popularity using sentiment analysis and topic modeling. *ACM Transactions on Information Systems (TOIS)*, 36, 1, 1–28.

[4] Yang Chen and Zhan. [n. d.] Predicting the popularity of online news: a natural language processing approach.

[5] Carolina Crisci, Badih Ghattas, and Gonzalo Perera. 2012. A review of supervised machine learning algorithms and their applications to ecological data. *Ecological Modelling*, 240, (Aug. 2012), 113–122. DOI: 10.1016/j.ecolmodel.2012.03.001.

[6] Lichan Hong and Hsinchun Chen. 2019. Can personality traits predict the popularity of news articles? *PloS one*, 14, 4, e0215335.

[7] Li Liu, Ruochen Li, Yan Jia, Yajie Wang, Huan Zhang, and Bo Li. 2017. Predicting news popularity with long-term and short-term popularity patterns. *ACM Transactions on Information Systems (TOIS)*, 36, 3, 1–25.

[8] Alexandru Tatar, Stefan Ruseti, Mihai Dascalu, and Florin Moldoveanu. 2018. Towards predicting the popularity of online news events: a machine learning approach. *IEEE Access*, 6, 10857–10869.

[9] Md Taufeeq Uddin, Muhammed J. A. Patwary, Tanveer Ahsan, and Mohammed Shamsul Alam. 2016. Predicting the popularity of online news from content metadata. In (Oct. 2016), 1–5. DOI: 10.1109/ICISET.2016.7856498.

[10] Wei Zhang and Steven Skiena. 2015. Predicting the popularity of online news using sentiment analysis. *Proceedings of the 24th ACM international on conference on information and knowledge management*, 1753–1756.

[11] Yuzhe Zhou, Ming Zhang, and Yue Zhang. 2019. Predicting the popularity of online news: a deep learning approach. *IEEE Access*, 7, 98505–98513.

[12] Arkaitz Zubiaga, Heng Ji, and Yuheng Sharma. 2016. Tweeting the terror: modelling the social media reaction to the paris attacks. *Social Science Computer Review*, 34, 6, 679–696.